Быстрые решения часто встречающихся проблем

# 

Кристофер Шмитт





R PYCCKAA PELAKUNA

Day File and Jones and Change

## **CSS Cookbook**

## THIRD EDITION

*Christopher Schmitt* foreword by Dan Cederholm



Кристофер Шмитт

## CSS рецепты программирования

3-е издание

«Русская редакция»

«БХВ-Петербург»

2011

УДК 681.3.068

ББК 32.973.26-018.1 Ш73

#### Шмитт К.

Ш73 CSS. Рецепты программирования. 3-е изд.: Пер. с англ. — М.: Издательство «Русская редакция»; СПб.: «БХВ-Петербург», 2011. — 672 с.: ил.

ISBN 978-5-7502-0400-7 («Русская редакция») ISBN 978-5-9775-0649-6 («БХВ-Петербург»)

Книга посвящена технологии Cascading Style Sheets (каскадные таблицы стилей). Обсуждаются основы CSS, а также средства, обеспечивающие наилучшие методы практической разработки: Web-типографика, изображения, элементы страницы, списки, ссылки и навигация, формы, таблицы, макеты страниц, печать Web-страниц, программистские трюки, а также проектирование с помощью CSS. Организованное как удобный справочник третье издание станет хорошим подспорьем для всех работающих с CSS. В отличие от предыдущего издания в нем также рассматриваются Firefox 3, IE8 и Chrome.

Для Web-дизайнеров и разработчиков сайтов

УДК 681.3.068 ББК 32.973.26-018.1

Подготовлено к печати по лицензионному договору с O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

Nutshell Handbook и O'Reilly являются товарными знаками или охраняемыми товарными знаками O'Reilly Media, Inc. в США и/или других странах. Все другие товарные знаки являются собственностью соответствующих фирм.

Все названия компаний, организаций и продуктов, а также имена лиц, используемые в примерах, вымышлены и не имеют никакого отношения к реальным компаниям, организациям, продуктам и лицам.

#### Кристофер Шмитт

#### CSS. Рецепты программирования. 3-е изд.

#### Перевод с английского языка Татьяны Коротяевой

Совместный проект издательства «Русская редакция» и издательства «БХВ-Петербург»





Лицензия ИД № 02429 от 24.07.00. Подписано в печать 03.09.10. Формат 70×100<sup>1</sup>/<sub>16</sub>. Печать офсетная. Усл. печ. л. 54,18. Тираж 2000 экз. Заказ №

Санитарно-эпидемиологическое заключение на продукцию № 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

> Отпечатано с готовых диапозитивов в ГУП "Типография "Наука" 199034, Санкт-Петербург, 9 линия, 12

ISBN 978-0-596-15593-3 (англ.) ISBN 978-5-7502-0400-7 («Русская редакция») ISBN 978-5-9775-0649-6 («БХВ-Петербург») © Оригинальное издание на английском языке, O'Reilly Media, Inc., 2010, 2007, 2004

© Перевод на русский язык, издательство «БХВ-Петербург», 2010

<sup>©</sup> Оформление и подготовка к изданию, издательство «БХВ-Петербург», издательство «Русская редакция», 2010

## Оглавление

Пре	дисловие	
Вве	дение	
Для і	кого предназначена эта книга	
Допу	ущения, принятые в книге	
Соде	ержание книги	
Согл	ашения, использованные в книге	
Испо	ользование кода примеров	
Как с	связаться с издательством O'Reilly	
Safar	ri <sup>®</sup> Books Online	
Благ	одарности	
Гла	ва 1. Использование базовых элементов HTML	
1.1.	Выбор текстового редактора	
1.2.	Кодирование простейшей HTML-страницы	
1.3.	О DOCTYPE и его влиянии на макет страниц в Web-обозревателе	
1.4.	Разметка заголовков	
1.5.	Подобающее цитирование	
1.6.	Вставка изображения	
1.7.	Вставка звуковых файлов средствами HTML5	
1.8.	Встраивание видео средствами HTML5	
1.9.	Эффективное применение strong и ет	
1.10.	. Создание списков	
1.11.	. Формирование ссылки на Web-страницу	
1.12.	. Разметка таблиц	
1.13.	. Создание vCard (hCard) средствами HTML	49
1.14.	. Разметка события (hCalendar)	
1.15.	. Проверка правильности HTML-разметки	51
Гла	ва 2. Основы CSS	54
2.1.	Применение правил CSS к Web-странице	55
2.2.	Применение базовых селекторов для стилевого оформления	58
2.3.	Применение селекторов прямого потомка	
2.4.	Применение селекторов соседнего элемента	
2.5.	Применение селекторов атрибута	
2.6.	Применение псевдоклассов	
2.7.	Применение псевдоэлементов	
2.8.	Когда использовать селекторы класса и селекторы ID	75
2.9.	О свойствах CSS	80

0 10		0.1
2.10.	Представление о блочной модели	
2.11.	Связь стилей с Web-страницей	
2.12.	Об источниках CSS	90
2.13.	Порядок сортировки правил в CSS	91
2.14.	Применение !important для переопределения отдельных правил CSS	
2.15.	Определение специфичности	
2.16.	Как применять разные типы таблиц стилей	
2.17	Вставка комментариев в таблицы CSS	101
2.17.	Организация информации в таблице стилей	102
2.10. 2 10	Работа со свойствали с сокращениой формой записи	104
2.19.	Габота со своиствами с сокращенной формой записи	104
2.20.	задание альтернативной таолицы стилей	100
2.21.	Применение плавающеи модели	10/
2.22.	Применение автовыравнивания плавающих элементов	109
2.23.	Применение абсолютного позиционирования	112
2.24.	Применение относительного позиционирования	114
2.25.	Применение сцепленного позиционирования	116
2.26.	Создание стека элементов с помощью <i>z-index</i>	118
2.27.	Проверка правил CSS	118
Гла	ва 3. Web-типографика	121
3.1.	Залание шрифтов	122
32	Применение Web-безопасных шрифтов	125
33	Залание стилизованного амперсанда	128
3.5.	Задание стилизованного ампереанда	120
5.4. 25	Плание фаилов шрифтов	124
3.3.	принудительная разоивка деиствительно длинных слов	134
3.6.	Задание размеров шрифта	135
3.7.	Достижение большего соответствия размеров шрифтов	
	в разных Web-обозревателях	137
3.8.	Установка дефисов, длинных и коротких тире	141
3.9.	Центрирование текста	142
3.10.	Задание выравнивания текста по ширине	142
3.11.	Обозначение избыточного текста многоточием	144
3.12.	Удаление свободного пространства между загодовками и абзацами	145
3 1 3	Залание простой буквищы	146
3 14	Формирование центрированной приполнятой букраны большего размера	147
2 1 5	Формирование центрированной приподнятой буквицы с помощ то изображения.	1/0
2.13.	Декоративное оформление приподнятой буквицы с помощью изображения	151
5.10. 2.17	Создание заголовка со стилизованным текстом	131
3.17.	Создание заголовка со стилизованным текстом и рамкои	153
3.18.	Стилевое оформление заголовка с помощью текста и изображения	155
3.19.	Создание врезки в HTML-тексте	157
3.20.	Расположение врезки сбоку от колонки	159
3.21.	Создание врезки с рамкой	160
3.22.	Создание врезки с изображениями	162
3.23.	Задание отступа в первой строке абзаца	165
3.24	Залание отступа лля всего абзана	166
3 25	Созлание обратного отступа	169
3 26	Стипевое оформление первой строки абзана	171
3.20.	Стилевое оформление первой строки абзана с помощи о изображания	171
5.41.	стилсьое оформление первои строки аозаца с помощью изооражения	1/3

3.28.	Создание эффекта выделенного текста	174
3.29.	Изменение цвета при выделении текста	175
3.30.	Изменение межстрочных интервалов	176
3.31.	Добавление графической обработки HTML-текста	178
3.32.	Вставка тени за текстом	180
3.33.	Корректировка межбуквенных и межсловных пробелов	183
3.34.	Достижение ритмичности базовых линий в Web-типографике	186
3.35.	Стилевое оформление верхних и нижних индексов без нарушения базовой линии	188
3.36.	Отображение нескольких колонок текста	189
Гла	ва 4. Изображения	192
4 1	Преобразование средствами CSS цветных изображений в черно-белые в IE	192
4 2	Вывод рамки вокруг изображения	193
43	Установка вокруг изображения рамки со скругленными углами	195
4.4	Улапение рамок устанавливаемых по умолчанию вокруг изображений	170
т.т.	в некоторых Web-обозревателях	197
45	Установка фонового изоблажения	199
4.5.	Создание полосы фоновых изображений	200
4.0. 4 7	Размещение фоновых изображения	200
ч.7. Д 8	Применение нескольких фонорых изображений в одном HTML -элементе	201
ч.о. 1 0	Применение нескольких фоновых изооражении в одном ПТИП-элементе	204
ч.). Л 10	Гоздание закрепленного фонового изображения	200
4.10.	Изменение размеров изображений при изменении размеров окна	209
7.11.	Изменение размеров изображении при изменении размеров окна	212
1 12	Растажание изображения на все окно Web обозравателя	212
4.12.	Гастяжение изображения на все окно w со-обозревателя	214
4.13.	Создание масштаоирусмых изображении	213
4.14.	Задание способа визуализации изображении в web-обозревателе	217
4.15.	Поворот изображении средствами ССС	210
4.10.	Создание градиентов средствами CSS	220
4.17.	Создание прозрачных PNO-изооражении для про и оолее поздних версии	222
4.18.	применение РМО-фаилов с прозрачностью с помощью заvаSспрі	224
4.19.	Наложение Н і містекста на изооражение	227
4.20.	замещение нтмL-текста изооражением	229
4.21.	Создание панорамного представления изооражения	232
4.22.	Комоинирование разных графических форматов	233
4.23.	Скругленные углы колонок с фиксированнои ширинои	239
4.24.	Скругленные углы (метод раздвижных дверей)	241
4.25.	Скругленные углы (метод горной вершины)	245
4.26.	Скругление углов с помощью JavaScript	248
4.27.	Вставка тени от элемента средствами CSS	252
4.28.	Вставка тени за изображением	253
4.29.	Вставка сглаженной тени за изображением	256
4.30.	Создание выносок с текстом	260
4.31.	Как помешать людям похищать ваши изображения	263
4.32.	Автоматическая вставка отражений в изображения	265
4.33.	Применение изображений-спрайтов	267
4.34.	Обрезка фоновых изображений	269
4.35.	Применение масок к изображениям и рамкам	271

Гла	ва 5. Элементы страницы	274
5.1.	Удаление полей страницы	274
5.2.	Сброс стилей, назначаемых Web-обозревателем по умолчанию	277
5.3.	Раскрашивание полосы прокрутки в Е	281
5.4.	Методы центрирования элементов на Web-странице	284
5.5.	Создание рамки страницы	289
5.6.	Вывод рамки вокруг окна просмотра Web-обозревателя	291
5.7.	Настройка горизонтального разделителя	293
5.8.	Добавление просмотра изображения в режиме Lightbox	295
5.9.	Изменение непрозрачности элементов	300
5.10.	Регулировка непрозрачности цвета фона	302
Гла		206
і ла.		207
0.1.	изменение формата списка	200
0.2.	изменение цвета маркера списка	210
0.3.	задание в списке отступов, одинаково отооражаемых в разных web-ооозревателях.	211
0.4. 6.5	установка разделителей между элементами списка	311 212
0.3.	Создание пользовательских текстовых маркеров в списках	215
0.0. 6 7	Создание пользовательских маркеров-изооражении в списках	313 210
0./.	Вставка обльших пользовательских маркеров-изображении в списки	220
0.8.	улучшение представления списка с помощью графики	320
0.9.	Создание внутристрочных списков	224
0.10.	Создание обратных отступов в списке	227
0.11.	Стидерод сформицация опродоктивности сталини	327
0.12. 6.12	Стилевое оформление списка определении.	329
0.13.	Градиционное отооражение киносценария с помощью элемента языка птист	225
614	Правращание опщака в дараво кото догов	226
6.15	Превращение списка в дерево каталогов	3/0
0.15.	создание системы звездного реитинга	540
Гла	ва 7. Ссылки и навигация	345
7.1.	Легкое создание текстовых меню и подменю	345
7.2.	Удаление подчеркивания ссылок и другие приемы стилевого оформления	347
7.3.	Изменение цвета ссылок	350
7.4.	Удаление пунктирных линий при щелчке мышью по ссылкам в Internet Explorer	351
7.5.	Изменение цвета ссылок в разных секциях страницы	352
7.6.	Вывод пиктограмм в конце ссылок разных типов	353
7.7.	Меняющиеся курсоры	354
7.8.	Изменение внешнего вида ссылки при наведении указателя мыши	
	без применения JavaScript	356
7.9.	Анимация ссылок с помощью свойств CSS3 transition при наведении указателя	
	Мыши	358
7.10.	Создание навигационных текстовых меню и визуальных эффектов при наведении	
	указателя мыши на его пункты	362
7.11.	Вставка подменю в вертикальные меню	366
7.12.	Создание горизонтальных навигационных меню	369
7.13.	Построение горизонтальных навигационных меню с раскрывающимися меню	375
7.14.	Создание навигационного меню с клавишами доступа	377
7.15.	Создание переходов с помощью ссылок-цепочек	379

7.16.	Создание с помощью изображений визуальных эффектов наведения указателя	
	мыши на ссылку	382
7.17.	Создание сворачиваемых меню	387
7.18.	Создание контекстных меню	389
7.19.	Создание всплывающих подсказок с помощью атрибута title	392
7.20.	Разработка динамического меню из закладок	393
7.21.	Изменяющиеся стили ссылок-якорей	395
Гла	ва 8. Формы	400
8.1.	Изменение пробельной зоны вокруг формы	400
8.2.	Удаление пробельной зоны вокруг формы	401
8.3.	Залание стилей лля элементов ввола	402
8.4.	Изменение стилей элементов формы при шелчке по ним кнопкой мыши	405
8.5.	Применение разных стилей к разным элементам ввола в олной форме	406
8.6	Назначение стилей элементам <i>textarea</i>	407
87	Залание стилей элементов select и option	409
8.8	Создание поля поиска в стиле Macintosh	411
89	Стипи кнопок формы	413
8.10.	Создание изображения кнопки Submit	417
8 1 1	Установка кнопки Submit-Once-Only	418
8 12	Созлание кнопки Submit в виле HTML-текста	419
813	Как заставить текстовую HTML-ссылку лействовать как кнопка Submit	421
8 14	Разпаботка Web-формы без таблиц	422
8 1 5	Разработка формы из двух колонок без применения таблиц	424
8 16	Включение в форму информации для пользователей	427
817	Стипевое оформление клавиш доступа в Web-формах	429
8.18.	Группировка общих элементов формы	431
8 1 9	Ввол ланных в форму в виде электронной таблицы	432
8 20	Пример лизайна: простая регистрационная форма	436
8.21.	Пример дизайна: бланк регистрации	442
Гла	ва 9. Таблицы	454
9.1.	Задание границ и отступов в ячейках и таблицах	454
9.2.	Задание объема свободного пространства вокруг ячейки	457
9.3.	Задание стиля заголовка	458
9.4.	Задание стилей в ячейках таблицы	459
9.5.	Задание стилей для элементов шапки таблицы	461
9.6.	Удаление зазоров у изображений, помещенных в ячейки таблицы	463
9.7.	Устранение зазоров между ячейками таблицы	465
9.8.	Создание чередующихся цветов фона в строках таблицы	466
9.9.	Создание эффекта выделения цветом строки таблицы	469
9.10.	Пример проекта: элегантный календарь	470
Гла	ва 10. Проектирование Web-страниц, предназначенных для печати	480
10.1.	Применение к Web-странице таблицы стилей для печати	480
10.2.	Замена цветного логотипа черно-белым при печати Web-страниц	482
10.3.	Создание готовой к печати Web-формы	484
10.4.	Отображение URI после ссылок	489
10.5.	Вставка специальных символов перед ссылками	490

10.6.	Вставка в документ для печати разрывов страниц	492
10.7.	Пример дизайна: версия страницы для печати, созданная средствами CSS	. 493
Г	a 11 Marray a ann anns	502
I ЛАВ	а 11. макеты страниц	. 503
11.1.	Построение одноколоночного макета	503
11.2.	Создание двухколоночного макета	505
11.3.	Построение двухколоночного макета с колонками фиксированной ширины	509
11.4.	Создание "резинового" многоколоночного макета с помощью плавающей модели	512
11.5.	Создание с помощью плавающей модели макета с тремя колонками фиксированной ширины.	. 515
11.6.	Созлание с помощью позиционирования "резинового" многоколоночного макета	518
117	Создание с помощью позиционирования многоколоночного макета с колонками	
11.7.	фиксированной ширины	521
11.8	Чиспользование плавающей молели лля отображения колонок в произвольном	
11.0.	поралие	522
110	Порядае	5/1
11.9.	Проектирование можетор, не зарисящих от разрешения	544
11.10.	проектирование макетов, не зависящих от разрешения	544
Глав	а 12. Программистские трюки, искусственные приемы и выявление	
неис	правностей	. 548
12.1.	Переопределение встроенных стилей	. 549
12.2.	Диагностика ошибок CSS и проблем Web-обозревателей	. 549
12.3.	Применение букмарклетов для поиска ошибок	. 551
12.4.	Применение расширений Web-обозревателя для поиска ошибок	. 552
12.5	Исправление Internet Explorer 6	553
12.6	Исправление Internet Explorer 6 с помощью JavaScript	555
12.0.	Использование условных комментариев лля перелачи стилей в разные версии	000
12.7.	Internet Explorer	556
12.8	Применение фильтров CSS пля передачи правил в практически любой	550
12.0.	Web-обозреватель	558
120	Web-0005реватель	550
12.9.	Установка интеллектуальной системы передачи СОЗ для современных	550
12 10		559
12.10.	тестирование дизаина саита на нескольких платформах с помощью	561
12.11	сдинственного компьютера	301
12.11.	проверка web-саита с помощью текстового web-осозревателя	302
Глав	а 13. Проектирование с помощью CSS	. 564
13.1.	Существенное увеличение размера шрифта	. 564
13.2	Созлание неожиланного сочетания	566
133	Комбинирование разнородных элементов для создания контраста	569
13.4	Привлечение взгляла с помощью контраста	570
13.5	Проверка достаточности цветового контраста	572
13.6	Вылеление питаты с помощью эффектных кавычек	574
13.0.	Залание перемещающейся фоновой сцены при изменении пользователем	574
13.1.	размеров окна	576
13.8	Вставка анимации в элементы на странице	578
13.0	Создание отображения фейерверков при прокрутке страницы пользорателем	583
13.9.	Создание отображения фенерверков при прокрутке страницы пользователем	591
13.10.	Проектирование с помощью сетки (наборы ССС)	585
10.11.	проектирование с помощвю сетки (насоры Соо)	

13.12.	Пример дизайна: связный Web-дизайн	588
13.13.	Пример дизайна: флаг США	605
Глав	а 14. Взаимодействие с JavaScript	. 618
14.1.	Проверка, включен ли JavaScript в Web-обозревателе	618
14.2.	Применение разных таблиц стилей в зависимости от времени суток	619
14.3.	Перенаправление на сайт для мобильных устройств в зависимости от ширины	
	экрана Web-обозревателя	621
14.4.	Вставка в Web-страницу библиотеки JavaScript	622
14.5.	Применение селекторов CSS3 в IE6 и IE7	623
14.6.	Создание с помощью JavaScript HTML-таблицы с окраской зебры	625
14.7.	Выделение цветом строки таблицы при наведении указателя мыши	627
14.8.	Визуальное оформление простого наведения указателя мыши	629
14.9.	Формирование строки одной высоты из элементов с разным объемом контента	630
14.10.	Задание ссылки, открывающей новое окно	633
14.11.	Превращение элемента div целиком в элемент, активируемый щелчком мыши	634
14.12.	Поддержка средствами JavaScript формата PNG в IE6	636
14.13.	Предоставление кода на HTML5 и свойств CSS3 Web-обозревателям,	
	способным их обрабатывать	638
Прил	ожение 1. Источники	. 641
Основ	вые обучающие материалы, относящиеся к языку HTML и технологии CSS	. 641
Лизай	нерские ресурсы.	642
Диску	ссионные группы	643
Справ	очные материалы	644
Прогр	аммные средства	645
Прил	южение 2. Свойства CSS 2.1 и специализированные дополнения	. 647
Прил	южение 3. Селекторы, псевдоклассы и псевдоэлементы CSS 2.1	. 661
Прил	южение 4. Селекторы и псевдоклассы CSS3	. 664
Прил	южение 5. Стилевое оформление элементов форм	. 667
Пред	метный указатель	. 668

## Предисловие

Искусный повар скажет вам, что главное в приготовлении вкусной еды — высококачественные ингредиенты. Автор, Кристофер Шмитт, уже все закупил для вас. Собрав сотни рецептов CSS в одной книге, он предлагает вам универсальный магазин, в котором вы можете выбрать все нужные ингредиенты для создания стильных и гибких Web-страниц.

Когда я начал изучать премудрости CSS, пробы и ошибки были моими основными средствами в постижении творческих возможностей этой технологии: "Мне хотелось бы превратить этот список в горизонтальное навигационное меню" или "Мне нужно предоставить клиенту стилевое оформление элементов формы средствами CSS". Несколько часов, а то и дней тратилось на подключение различных правил, удаление некоторых из них и проверку бесконечного числа комбинаций. Этот подход в расчете на удачу срабатывал (иногда), но определенно съедал массу времени, несмотря на то что я, будучи любопытным человеком, находил его даже увлекательным.

Мне бы хотелось тогда иметь подобную книгу. Вместо того чтобы озадачиваться созданием стилей для каждого элемента страницы, я мог бы пролистать "CSS. Рецепты программирования", найти рецепт и начать приготовление. Работа наугад была бы исключена, и я смог бы заняться тем, что люблю больше всего: творчеством.

Модульная структура книги делает ее незаменимым справочным пособием для дизайнеров и разработчиков любого уровня. Решая разные проблемы, начиная с управления типографикой, ссылками и навигацией и заканчивая макетами целых страниц, Кристофер не только четко поясняет применение стилей для решения поставленной задачи, но и сообщает о проблемах, которые могут быть связаны с определенными Webобозревателями. Предлагая в дополнение полезные приемы и хитрости для решения каждодневных задач, он снабжает вас важными знаниями, столь необходимыми для того, чтобы стать успешным дизайнером CSS.

Например, в недавней статье обсуждалась общая проблема практичности и удобства: если в конце формы отображается кнопка **Submit** (Отправить), некоторые пользователи не могут избавиться от привычки двойного щелчка кнопкой мыши. Кнопка может быть "нажата" дважды, что приведет к дублированию отправки формы. Что делать? В упомянутой ранее статье никакого решения не предлагалось. Именно в этой книге, что неудивительно, нашелся рецепт для решения этой маленькой проблемы средствами CSS с добавлением штриха на JavaScript.

И в этом основная цель данной книги: реальные проблемы и средства, которые принесут реальные результаты. Вы наверняка слышали о том, как технология CSS улучшает жизнь, облегчая сопровождение и поддержку страниц. Пришло время воспользоваться ею, а с этой книгой у вас будет на один предлог меньше для отказа от этого.

Итак мой совет — освободите часть вашего стола, потому что книга "CSS. Рецепты программирования" займет на нем постоянное место. К счастью для вас, при необходимости до нее будет легко дотянуться рукой.

> Дэн Седерхольм (Dan Cederholm), основатель студии Simplebits Сейлем, Массачусетс

> > (http://www.simplebits.com)

## Введение

В каждой книге есть своя история, даже в книгах о Web-дизайне.

Предлагаемая книга посвящена технологии Cascading Style Sheets (каскадные таблицы стилей) или CSS (аббревиатура, обычно используемая для ее обозначения). CSS — простая стандартизованная синтаксическая структура, предоставляющая дизайнерам всесторонний контроль над визуальным представлением их Web-страниц и являющаяся важнейшим компонентом современного Web-дизайна.

По сравнению со средствами разработки эры 1990 гг. сегодня благодаря CSS Webдизайнеры обладают большим набором возможностей для управления дизайном Webсайта и тратят меньше времени на его редактирование и сопровождение. CSS также раздвигает рамки традиционного Web-дизайна, позволяя проектировать внешний вид Web-страницы и управлять им при выводе на печатающее устройство.

Простота технологии каскадных таблиц стилей заключается в отсутствии необходимости специального аппаратного или программного обеспечения для их применения. Основные требования — компьютер, современный Web-обозреватель, такой как Firefox, Safari или Internet Explorer для Windows (приведенные лишь для примера), и ваш любимый редактор Web-страниц. В качестве последнего можно использовать любую программу-редактор, начиная от Notepad в OC Windows или TextEdit для компьютеров Macintosh и заканчивая развитым, с технологией WYSIWYG (What You See Is What You Get, что видишь, то и получаешь), редактором Adobe Dreamweaver в режиме отображения кода страницы.

Теперь, когда вы знаете, о чем эта книга, позвольте мне рассказать ее историю.

Одни считают, что Web-дизайн появился тогда, когда Тим Бернерс-Ли (Tim Berners-Lee), создатель "Всемирной паутины", сформировал первый набор Web-страниц. Другие могут сказать, что он возник, когда появился тег center как элемент собственных расширений языка HTML (HyperText Markup Language, язык разметки гипертекста), разработанных компанией Netscape.

Как бы это ни звучало иронично, я склонен считать, что реально Web-дизайн появился благодаря книгам. Первыми книгами, способствовавшими росту интернет-активности в 1990 гг., были книга Линды Вейнман (Lynda Weinman), первая полноцветная книга о Web-графике, "Designing Web Graphics" (Проектирование Web-графики), изданная в январе 1996 г., и книга Дэвида Зигеля (David Siegel) "Creating Killer Web Sites" (Создание сногсшибательных Web-сайтов), вышедшая в том же году через несколько месяцев. Эти книги положили начало революционным изменениям во "Всемирной паутине" наряду с создателями технологий, сделавшими подобные изменения возможными.

К сожалению, методы, описанные в них, будучи передовыми в свое время, в наши дни уже устарели. Когда я писал эти строки, прошло уже 13 лет с момента выхода названных книг; в том же году, когда вышли в свет первые книги Вейнман и Зигеля о Webдизайне с описанием тегов font, вложенных таблиц и однопиксельных GIFизображений, впервые была представлена технология CSS.

С тех пор CSS пришлось пройти долгий путь. Только сейчас, спустя 13 лет, потраченных на разработку и развитие CSS, с появлением и широким распространением Webобозревателя Internet Explorer 8 для Windows, Web-дизайнеры, разработчики и обычные пользователи Web-обозревателей могут использовать технологию CSS в соответствии с ее потенциалом.

Помимо IE8 о себе заявили и другие Web-обозреватели, часто опережающие Internet Explorer в поддержке новых свойств и функций. Web-обозреватели Firefox, Safari, Chrome и Opera реализуют последние спецификации CSS3 и HTML5 сразу по мере обсуждения их членами рабочих групп консорциума World Wide Web (W3C).

Если вы серьезно относитесь к созданию современных, удобных и передовых Webсайтов, используйте технологию CSS и эту книгу с коллекцией решений распространенных проблем Web-дизайна средствами технологии CSS. Они помогут вам внести свою лепту в историю Web-дизайна.

#### Для кого предназначена эта книга

Это книга для Web-дизайнеров и разработчиков, сталкивающихся с проблемами проектирования средствами CSS. С ее помощью Web-разработчики смогут решать общие задачи, связанные с дизайном Web-страниц посредством каскадных таблиц стилей.

"CSS. Рецепты программирования" идеально подходит тем, кто хотел бы использовать CSS в Web-проектах, но старается избегать изучения новой технологии. Если вы относитесь к этому типу читателей, применяйте каждый раз по одному решению из книги. Используйте ее как руководство и обращайтесь к ней, когда вам нужен новый метод или искусственный прием, и вы готовы к его изучению.

Даже если вы считаете себя специалистом в области CSS, но не владеете основами дизайна, эту книгу полезно держать рядом с компьютером. В ней описаны элементы дизайна, начиная от основ Web-типографики и заканчивая макетами страниц, а также включена глава, описывающая приемы образцового дизайна и способная подсказать вам удачное решение.

#### Допущения, принятые в книге

В книге приняты некоторые допущения, касающиеся вас, дорогой читатель.

Одно из них предполагает наличие у вас некоторого опыта Web-дизайна и разработки в качестве любителя, студента или профессионала.

"CSS. Рецепты программирования" — это не введение в технологию CSS и не подробное описание реализации CSS в разных Web-обозревателях, поэтому люди, начинающие изучение Web-дизайна и методов разработки, могут счесть ее более сложной, чем общее и полное описание теоретических основ CSS.

Если вы ищете книгу, посвященную подробному изучению спецификации CSS, вам следует обратиться к книге Эрика Мейера (Eric A. Meyer) "Cascading Style Sheets: the Definitive Guide" (Каскадные таблицы стилей: подробное руководство)<sup>1</sup>, третье издание O'Reilly Media. Она послужит солидным дополнением к данной книге.

Если вы применяете программы, подобные редактору Adobe Dreamweaver, только в режиме WYSIWYG (What You See Is What You Get, что видишь, то и получаешь) или в режиме дизайна и редко сталкивались с разметкой в режиме просмотра кода, у вас могут возникнуть затруднения, т. к. придется сразу же отказаться от большинства примеров, предложенных в книге. Для знакомства с основами кодирования вручную на языке HTML загляните в книгу Дженнифер Нидерст Роббинс (Jennifer Niederst Robbins) "Learning Web Design" (Изучение Web-дизайна) (изд. O'Reilly Media).

Несмотря на то, что средства WYSIWYG позволяют создавать дизайн-проекты с помощью CSS, некоторые из них не справляются с оригинальными методами, предложенными в данной книге, и могут вызвать проблемы, если вы попытаетесь реализовать эти методы, редактируя страницы только в режиме WYSIWYG.

Для того чтобы полностью использовать предлагаемые в книге решения, вы должны уметь редактировать HTML-код и таблицы CSS вручную. Часть представленного в книге кода можно воссоздать с помощью управляемых диалоговыми окнами программ построения Web-страниц (dialog-box driven Web page building application), но и на этом пути могут возникнуть проблемы.

Другое допущение состоит в том, что Web-дизайнеры и разработчики, обладающие навыками создания страниц с помощью макетов на базе HTML-таблиц, тегов font и однопиксельных GIF-изображений, сочтут эту книгу полезной и разочаровывающей одновременно.

Web-дизайнерам, применяющим на практике методы, подобные этим старым способам разработки, технология CSS покажется сложной. Все еще существует не всегда правильное отображение Web-страниц (так называемый "browser hell — произвол Web-обозревателей"), часто связанное с разработкой Web-обозревателя, в которой его создатели были склонны по-своему интерпретировать спецификацию CSS или не реализовали ее в полном объеме. Чувство неудовлетворенности — естественная составляющая процесса познания. К изучению дизайна средствами CSS следует подходить с терпением и чувством юмора.

К счастью, разработчики основных Web-обозревателей, кажется, решили проблему. В выпусках современных версий обозревателей технология CSS похоже реализована корректно, но попытка поддержки совместимости с более старыми или менее популярными Web-обозревателями все еще может быть серьезным испытанием.

И все же преимущества применения CSS, включая более полный контроль над впечатлением и ощущением, создаваемыми Web-страницей, а также более легкое техниче-

<sup>&</sup>lt;sup>1</sup> Эрик А. Мейер. CSS. Каскадные таблицы стилей. Подробное руководство. — СПб.: Символ-Плюс, 2008 г. — *Прим. ред.* 

ское обслуживание многостраничных Web-сайтов, перевешивают неприятности, связанные с произволом Web-обозревателей.

Поскольку в книге есть несколько решений, использующих язык JavaScript и библиотеку на JavaScript, jQuery, предполагается, что у вас есть базовые знания языка сценариев, а также навыки корректного включения в Web-документ кода на JavaScript.

Если это является препятствием, рекомендуем загрузить код с Web-сайта издательства O'Reilly, чтобы увидеть оригинальный работающий пример. С другой стороны, если вы ищете книгу с решениями, в которых технологии CSS отведена второстепенная роль по сравнению с языком JavaScript, посмотрите книгу Дэнни Гудмана (Denny Goodman) "JavaScript & DHTML Cookbook" (JavaScript и DHTML. Сборник рецептов) (издательство O'Reilly Media)<sup>1</sup>.

И последнее допущение — вам нужен источник, предлагающий быстрые решения для широко распространенных проблем Web-дизайна на базе технологии CSS. Предлагаемые решения, охватывающие все, начиная от Web-типографики и заканчивая многоколоночными макетами страниц, совместимы с современными Web-обозревателями, имеющими номера версий большие или равные 5, за исключением Web-обозревателей Safari и Chrome.

Каждый раз, когда это возможно, я подчеркиваю, что тот или иной метод способен вызвать проблемы в современных Web-обозревателях. Несмотря на то, что в книге есть глава о программистских трюках и искусственных приемах для скрывания таблиц стилей от Web-обозревателей со слабой реализацией полной спецификации CSS, книга не гарантирует создания выверенных до пиксела, совершенных дизайн-проектов во всех Web-обозревателях. Даже с помощью традиционных методов Web-дизайна, начиная с 1990 гг., этого никогда не удавалось добиться (дополнительную информацию см. на http://dowebsitesneedtolookexactlythesameineverybrowser.com/).

## Содержание книги

Для меня лучший вариант применения книги, подобной этой, — освоить ее, раскрывая время от времени, при попытке решить конкретную проблему, что я и делал с первым ее изданием при написании этого, когда нужно было освежить память. Для этой цели она может оказаться полезной, если будет лежать на рабочем столе Web-разработчика или в непосредственной близости от него — всегда в пределах досягаемости для разрешения проблемы, связанной с CSS или Web-дизайном. Тем не менее ничто не мешает вам прочесть эту книгу от первой страницы до последней.

Далее приведен краткий обзор содержания каждой главы.

- В главе 1 "Использование базовых элементов HTML" разбираются решения на основе семантической разметки контента.
- В *главе 2 "Основы CSS"* обсуждаются основы CSS, а также некоторые средства, обеспечивающие наилучшие методы практической разработки.

<sup>&</sup>lt;sup>1</sup> Гудман Д. JavaScript и DHTML. Сборник рецептов. — СПб.: Питер, 2004.

- В *главе 3 "Web-munoграфика"* описываются способы применения CSS для задания шрифтов на Web-странице, заголовков, врезок, абзацных отступов и многое другое.
- В *главе 4 "Изображения"* обсуждаются средства CSS, непосредственно связанные с применением стилей и свойств, относящихся к Web-графике.
- ◆ В *главе 5 "Элементы страницы"* собран набор несвязанных друг с другом тем, которые необязательно уместны в конкретной главе, но все влияют на дизайн страницы в целом. Решения данной главы включают центрирование элементов, установку фонового изображения, размещение рамки страницы и другие средства оформления.
- ◆ В главе 6 "Списки" описаны различные варианты стилевого оформления элементов списка. Решения включают вопросы совместимости разных Web-обозревателей при отображении отступов в тексте, создание обратных отступов, включение пользовательских изображений в качестве маркеров списка и многое другое.
- ◆ В главе 7 "Ссылки и навигация" показано, как использовать CSS для управления представлением ссылки и наборов ссылок. В главу включены как простые, например удаление подчеркивания ссылок, так и более сложные решения, такие как создание динамического визуального меню.
- ◆ В *славе 8 "Формы"* обсуждаются искусственные приемы преодоления недостатков стандартных методов визуализации форм в Web-обозревателях. Решения, предлагаемые в этой главе, включают среди прочего установку стилей для элементов формы, установку кнопки **Submit** с одноразовым нажатием и стилевое оформление регистрационной формы.
- В главе 9 "Таблицы" показано, как задавать стили в HTML-таблицах. Несмотря на то, что CSS помогает исключить необходимость применения дизайн-проектов на базе HTML-таблиц, иногда требуется стилевое оформление табличных данных, таких как календари или статистические сведения. В эту главу включены решения для установки отступов в ячейке таблицы, удаления зазоров в ячейках таблицы при вставке изображений и стилевое оформление календаря.
- ◆ В главе 10 "Проектирование Web-страниц, предназначенных для печати" предоставлена информация о стилях, применяемых при печати Web-страниц. Решения, обсуждаемые в данной главе, включают формирование отдельной таблицы стилей для печати Web-страницы, задание стилей для Web-форм и вставку URL (Uniform Resource Locator, унифицированный указатель информационного ресурса) после ссылок.
- В *главе 11 "Макеты страниц"* рассказывается о том, как можно использовать технологию CSS для создания макетов. В эту главу включены решения для формирования как одноколоночных, так и многоколоночных макетов.
- ◆ В главе 12 "Программистские трюки, искусственные приемы и выявление неисправностей" описаны способы скрывания таблиц стилей, которые не могут быть обработаны определенными Web-обозревателями. Приведены примеры скрывания таблиц стилей от Internet Explorer 5 для Windows и других версий.
- ◆ Глава 13 "Проектирование с помощью CSS" является определяющей. Подчеркивая, что технология CSS лишь средство воплощения дизайна, она описывает роль

текста увеличенного размера, значение контраста и способы создания панорамного отображения.

- ◆ В *главе 14 "Взаимодействие с JavaScript"* показано, как для получения сложных визуальных эффектов применять библиотеку на JavaScript, jQuery.
- ◆ В приложении 1 "Источники" дан перечень ссылок и адресов Web-сайтов, содержащих дополнительную информацию о CSS.
- ◆ В приложении 2 "Свойства CSS 2.1 и специализированные дополнения" приведена таблица свойств CSS, помогающих определить впечатление и ощущение, производимые HTML-элементами, а иногда и звучание их на Web-странице.
- ◆ В приложении 3 "Селекторы, псевдоклассы и псевдоэлементы CSS 2.1" дан список селекторов, псевдоклассов и псевдоэлементов, включенных в спецификацию CSS 2.1.
- ◆ В приложении 4 "Селекторы и псевдоклассы CSS3" дан список селекторов и псевдоклассов, включенных в новую спецификацию CSS3.
- В приложении 5 "Стилевое оформление элементов форм" показано, как различные современные Web-обозреватели обрабатывают и отображают элементы форм. Печатная версия книги содержит введение к этому приложению, а также информацию о том, как получить полную версию. Интерактивная версия (http://oreilly.com/ catalog/9780596155933/) этого приложения содержит справочные таблицы, позволяющие быстро проверить, какие свойства CSS поддерживаются, а также полный обзор элементов форм, содержащий скриншоты всех тестов.

#### Соглашения, использованные в книге

В книге применялись следующие типографические соглашения.

- Элементы интерфейса, а также интернет-адреса выделены полужирным шрифтом.
- Моноширинным шрифтом обозначаются команды, режимы, переключатели, переменные, атрибуты, клавиши, функции, типы, классы, пространства имен, методы, модули, свойства, параметры, значения, объекты, события, обработчики событий, теги языков XML и HTML, макросы, содержимое файлов, вывод команд.

#### Примечание

В разделе с таким заголовком содержатся совет, предложение или общее замечание.

#### Предупреждение

В разделе с таким заголовком содержатся предостережения.

#### Использование кода примеров

Эта книга увидела свет для того, чтобы помочь вам выполнять вашу работу. Как правило, вы можете использовать код, приведенный в этой книге, в ваших Web-страницах и в ваших дизайн-проектах. Не нужно связываться с нами для получения разрешения, если вы не воспроизводите большие объемы кода. Например, для написания программы, использующей несколько порций кода, не требуется разрешения. Для продажи и распространения дисков CD-ROM с примерами из книг издательства O'Reilly, безусловно, необходимо разрешение. Ответ на вопрос с помощью ссылки на эту книгу и цитирования фрагмента кода не требует разрешения. Включение большого объема кода примеров из этой книги в сопроводительную документацию к вашей продукции несомненно нуждается в разрешении.

Мы приветствуем библиографические ссылки, хотя и не настаиваем на них. Такие ссылки обычно включают заголовок, имя автора, издательство и ISBN (International Standard Book Number, стандартный международный номер книги). Например: "*CSS Cookbook*, Third Edition, by Christopher Schmitt. Copyright 2010 O'Reilly Media, Inc., 978-0-596-15593-3"<sup>1</sup>.

Если вам кажется, что использование кода примеров выходит за рамки законного применения или разрешения, данного ранее, вы можете связаться с нами по адресу электронной почты permissions@oreilly.com.

## Как связаться с издательством O'Reilly

Пожалуйста, посылайте замечания и вопросы, относящиеся к этой книге, издателю:

O'Reilly Media Inc., 1005 Gravenstein Highway North Sebastopol, CA 95472 800-998-9938 (в США и Канаде) 707-829-0515 (международный или местный) 707-829-0104 (факс)

У нас есть посвященная этой книге Web-страница, на которой приведен список опечаток, примеры и другая дополнительная информация. К этой странице можно обратиться по адресу:

#### http://www.oreilly.com/catalog/9780596155933

У этой книги есть и еще один Web-сайт:

#### http://csscookbook.com

Для отправки замечаний и вопросов, касающихся этой книги, посылайте письма по адресу:

#### bookquestions@oreilly.com

Для получения дополнительной информации о книгах, конференциях, центрах распространения (Resource Centers) и сервисе O'Reilly Network посетите Web-сайт:

#### http://www.oreilly.com

<sup>&</sup>lt;sup>1</sup> Или ее русский эквивалент, который вы держите в руках: Шмитт К. CSS. Рецепты программирования. — 3-е изд. — СПб.: БХВ-Петербург, 2010. — Прим. ред.

## Safari<sup>®</sup> Books Online

Safari Books Online — это электронная библиотека по требованию, позволяющая для получения быстрых ответов на интересующие вас вопросы легко найти более 7 500 оригинальных справочных пособий и видео о передовых технологиях.

Подписавшись, вы сможете прочесть любую страницу и просмотреть любой видеофильм из нашей интерактивной библиотеки. Читайте книги на вашем мобильном телефоне или других мобильных устройствах. Просматривайте заголовки новых книг до того, как они выйдут из печати, и получите эксклюзивный доступ к рукописям, готовящимся к печати, и связь по почте с их авторами. Копируйте и вставляйте код примеров, формируйте свои списки "Избранного", загружайте отдельные главы, делайте закладки в важных разделах, создавайте замечания, печатайте страницы и пользуйтесь массой функций, экономящих время.

Издательство O'Reilly Media включило эту книгу в библиотеку Safari Books Online. Для получения полного доступа к этой книге и другим книгам сходной тематики издательства O'Reilly и других издательств подключайтесь бесплатно на Web-сайте http://my.safaribooksonline.com.

#### Благодарности

Прежде всего, большое спасибо Дэвиду Сигелу (David Siegel) и Линде Вейнман (Lynda Weinman) за их влияние и поддержку, ощущавшуюся со времени появления Webдизайна.

Я не написал бы ни одной книги, посвященной столь любимой мной сфере деятельности, без поддержки и помощи Молли Хольцшлаг (Molly Holzschlag).

Самой высокой оценки и уважения достойны развивающие и продвигающие Webдизайн на основе технологии CSS собратья Web-разработчики: Дуглас Боуман (Douglas Bowman), Тантек Челик (Tantek Çelik), Ден Седерхольм (Dan Cederholm), Майк Дэвидсон (Mike Davidson), Итан Маркот (Ethan Marcotte), Эрик А. Мейер (Eric A. Meyer), Марк Ньюхаус (Mark Newhouse), Дэйв Ши (Dave Shea), Николь Салливан (Nicole Sullivan), Стефани Салливан (Stephanie Sullivan) и Джеффри Зельдман (Jeffrey Zeldman).

Особая благодарность научным редакторам Web-евангелисту обозревателя Opera Брюсу Лосону (Bruce Lawson), Шелли Пауэрс (Shelley Powers), Эдду Дамбиллу (Edd Dumbill), а также корректору Одри Дойль (Audrey Doyle) за потраченное время, компетентность и терпение.

Я также благодарен Татьяне Диаз (Tatiana Diaz), моему редактору предыдущего издания этой книги.

В этом издании роль Татьяны досталась Симону Сен Лорану (Simon St.Laurent). Его спокойный нрав и умение управлять процессом подготовки книги к изданию сделали свист вращающихся лезвий, оповещающий о наступлении последнего срока сдачи, не столь устрашающим.

Я глубоко признателен моим друзьям, которые знают, что я действительно безраздельно предан Web-технологиям и не обидятся на меня за то, что я не назвал их всех по именам. Спасибо Джессике, кормившей меня на праздновании дня рождения и выхода предыдущего издания книги шоколадным тортом с самодельной шоколадной глазурью и кусочками шоколада. Мне он понравился чрезвычайно, и мой дантист тоже оценил эту превосходную работу. Я не рассчитываю на еще один торт, но включил тебя в раздел с благодарностями.

Благодарю свою семью за любовь и высокую оценку. Ваша поддержка всегда, и в хорошие, и в плохие времена, была непоколебима как скала. И как всегда, я с нетерпением жду нашей новой встречи.

Спасибо Ари Стайлз (Ari Stiles) за доброе отношение ко мне и выкраивание времени для работы над этой книгой. Я люблю Вас.

И снова я посвящаю эту книгу моему отцу. Спасибо за то, что ты лучший отец на свете.

Кристофер Шмитт (Christopher Schmitt) Осень 2009 г. http://christipherschmitt.com

http://twitter.com/teleject

## Использование базовых элементов HTML

## 1.0. Введение

Эффективное использование каскадных таблиц стилей (Cascading Style Sheets, CSS) требует умелого применения языка HTML. В этой главе дается обзор основных принципов корректного использования HTML, чтобы наставить вас на путь истинный. Вы рассмотрите базовые, но очень важные приемы, позволяющие создать HTMLстраницу, проверить корректность ее разметки, чтобы убедиться в отсутствии опечаток и ошибок, и воспользоваться новыми возможностями версии языка HTML5 для вставки видео и звуковой информации.

#### Примечание

Если вы хорошо знаете предлагаемый материал, можете пролистать эту главу. Даже ее беглый просмотр поможет вам сформировать полезные навыки, которые облегчат работу в дальнейшем.

#### Структурирование документов

Для разработки дизайна ваших Web-страниц необходимо, прежде всего, наполнить Web-документ информационным содержимым или контентом, обычно представляющим собой простой текстовый файл. Этот контент в текстовом файле следует пометить тегами, т. е. воспользоваться *языком разметки гипертекста* (HyperText Markup Language) или языком HTML. HTML формирует структуру документов с помощью элементов. Когда во всем контенте вы примените для обозначения этих элементов теги, например, р для абзацев и h2 для заголовков, на Web-странице начнет формироваться присущая ей структура HTML-документа.

Затем Web-обозреватель, используя собственную таблицу стилей, отобразит структуру этого документа. Данный процесс именуют стандартной визуализацией Web-страницы. Это стандартное отображение вряд ли выиграет какие-либо награды на конкурсе дизайна. Оно является всего лишь отправной точкой, позволяющей напрямую связать визуальное представление или дизайн с каскадными таблицами стилей (CSS) и языком JavaScript и обеспечить внешний вид и поведение Web-страницы.

#### Семантическая разметка

Эта глава — пример использования семантического HTML. Семантическая разметка — это "радикальная" система обозначений, в которой мы применяем для любого контента соответствующий ему HTML-элемент.

Например, для обозначения абзаца используется простой тег р в начале и в конце текста абзаца.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

#### Избегайте мешанины из тегов

Семантический HTML не слишком широко распространен в Web-пространстве. Поскольку различные HTML-элементы выглядят по-разному при отображении в обозревателе, Web-дизайнеры для получения нужного визуального представления приготовляют зачастую странные смеси из HTML-элементов, которые обычно называют теговой мешаниной или солянкой из тегов.

Для повышения контроля над внешним видом дизайнеры могут добавить презентационные или оформительские HTML-теги, противоречащие семантически размеченному контенту, как, например, такие:

<font face="Georgia, Times, serif" size="+2">Lorem ipsum dolor</font> <font face="Arial, Helvetica, san-serif" size="1">sit amet, consectetuer adipiscing elit, sed diam <b>nonummy</b> nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</font>

Эти дополнительные HTML-теги присутствуют здесь для управления внешним видом лишь одного абзаца.

Если вы применяете традиционное HTML-кодирование, каждому единичному элементу HTML-страницы сайта может потребоваться кодирование с помощью дополнительных элементов, задающих конкретные цвета, шрифты, выравнивание и расположение, задуманные дизайнером. Этот процесс крайне утомителен и приводит к ошибкам.

Представьте себе, что вы проектируете Web-сайт из 20 страниц и хотели бы вставить в него конкретные элементы дизайна, такие как цвета, шрифты, размеры и выравнивание. Теперь представьте себе сопровождение Web-сайта из 1 000 страниц. А что вы скажете о Web-сайте, содержащем 1 000 000 страниц?

HTML-документы сайтов быстро обрастают дополнительным кодом, что делает и контент, и код почти неуправляемыми.

#### HTML — это структура документа

Итак, очень важно средствами HTML добиться максимально возможной корректности структуры документа. Благодаря применению семантического экономного кодирования Web-разработчики сокращают время, необходимое для технической поддержки, и формируют исходную структуру, к которой могут применяться таблицы стилей.

## 1.1. Выбор текстового редактора

#### Проблема

Вы хотите выбрать текстовый редактор для разметки контента средствами языка HTML.

#### Решение

Для кодирования на языке HTML приспособлены многочисленные программные приложения. Одни свободно распространяются, а другие требуют оплаты.

К базовым текстовым редакторам, устанавливаемым вместе с операционной системой, относятся следующие:

- Notepad (Windows);
- ♦ TextEdit (Mac OS);
- ♦ gedit (Linux).

Далее перечислено несколько свободно распространяемых текстовых редакторов с бо́льшим набором функциональных возможностей:

- Notepad++ (Windows; http://notepad-plus.sourceforgenet/uk/site.htm);
- TextWrangler (Mac OS; http://www.barebones.com/products/TextWrangler/);
- ♦ jEdit (Windows, Mac OS и Linux; http://www.jedit.org/).

В качестве более профессиональных, коммерческих интегрированных сред разработки (IDE), попробуйте одну из следующих:

- Adobe Dreamweaver (Windows и Mac OS; http://www.adobe.com/products/dreamweaver/);
- ♦ Panic Software's Coda (Mac OS; http://www.panic.com/coda/).

#### Обсуждение

Некоторые приложения, подходящие для редактирования HTML, приходят в составе операционных систем, таких как Mac OS и Windows. К ним относятся редакторы TextEdit и Notepad (Блокнот) соответственно.

#### Предупреждение

Для работы с HTML не пользуйтесь текстовыми процессорами. Несмотря на то, что эти программы идеально подходят для создания обычных документов, предназначенных для печати, они вставят в ваш текст нежелательное и ненужное вам постороннее форматирование.

Перед использованием TextEdit выберите вариант форматирования Plain text (неформатированный, свободно читаемый текст) с помощью последовательности команд меню File | Preferences (Файл | Параметры). Иначе текстовый редактор может формировать длинные строки из HTML-элементов.

Если вы применяете Notepad, выберите в меню вариант Format | WordWrap (Формат | Перенос по словам). Он позволит размещать длинные строки в пределах окна приложения, облегчая тем самым их редактирование.

#### Примечание

В обоих редакторах, TextEdit и Notepad, обязательно сохраняйте HTML-файл с расширением html. Не вставляйте в конец названия файла расширение html. Такие названия, как example.txt.html или example.html.txt, приводят к большим затруднениям.

Несмотря на то, что эти редакторы программного кода, свободно распространяемые и заранее установленные в операционной системе, не обладают широким набором параметров, многие Web-дизайнеры полагаются на них при работе с языком HTML.

#### Более надежный и все еще бесплатный

Еще один текстовый редактор, также бесплатный, — TextWrangler компании Bare Bones Software. TextWrangler не так многофункционален, как флагманский продукт компании BBEdit, но вполне может подойти для ваших нужд. Редакторы TextWrangler и BBEdit — приложения, предназначенные только для Mac OS.

Для OC Windows существуют такие варианты, как Notepad++ и TextPad (http://www.textpad.com/).

Для ОС UNIX существуют редакторы vi/vim и Emacs. Другой возможный вариант — редактор jEdit, версии которого есть и для Mac OS, и для Windows.

#### Программные решения IDE

Программные продукты с более широким набором функциональных возможностей часто и стоят больше, но они предлагают полный набор средств для решения практически любых проблем, возникающих при создании Web-сайтов. Популярные продукты в этом сегменте — редакторы Adobe Dreamweaver Panic и Software's Coda.

#### См. также

По адресу http://www.notepad.org/logo.htm графический баннер "Made with Notepad" (Сделан с помощью Notepad), который можно поместить на вашу Web-страницу.

#### 1.2. Кодирование простейшей HTML-страницы

#### Проблема

Вы хотите создать свою первую HTML-страницу.

#### Решение

Начните с исходного контента, например, такого:

My Basic Web Page Epsum factorial non deposit quid pro quo hic escorol.

#### Затем создайте HTML-элемент, включающий весь документ.

<html> My Basic Web Page Epsum factorial non deposit quid pro quo hic escorol. </html>

#### Далее вставьте в документ элементы head и body.

<html> <head>

```
</head>
<body>
My Basic Web Page
Epsum factorial non deposit quid pro quo hic escorol.
</body>
</html>
```

#### В элемент head включите элемент title.

```
<html>
<head>
<title>CSS Cookbook</title>
</head>
<body>
My Basic Web Page
Sed quis custodiet ipsos custodes?
</body>
</html>
```

В элемент body введите элементы heading (h1) и paragraph (p), и страница будет отображаться, как показано на рис. 1.1.

```
<html>
<head>
<title>CSS Cookbook</title>
</head>
<body>
<h1>My Basic Web Page</h1>
Sed quis custodiet ipsos custodes?
</body>
</html>
```



Рис. 1.1. Стандартное отображение простейшей Web-страницы на языке HTML

#### Обсуждение

У каждой Web-страницы должен быть элемент html, включающий в себя весь документ целиком. В каждом элементе html есть два обязательных элемента — head и body.

Элемент head содержит сведения о документе, часто называемые *метаданными*. Элемент head должен включать в себя элемент title. Этот текст обычно располагается в

верхней части окна обозревателя и используется при создании закладок. При формировании тега title важно быть лаконичным и избегать пространных описаний.

Если в элемент title не включен текст, обозреватели будут использовать вместо него либо имя файла, либо первые несколько слов из документа.

#### Примечание

В элементе title может содержаться только текст. Включать в него другие HTMLэлементы запрещено.

Контент Web-документа помещается в элемент body. Если вам необходимо отредактировать или переработать Web-страницу, большую часть времени вы проведете в этом элементе.

В приведенном примере заголовок задан в элементе h1, а стандартный элемент р используется для обозначения абзаца.

#### См. также

Разд. 1.1 о выборе текстового редактора.

## 1.3. О *DOCTYPE* и его влиянии на макет страниц в Web-обозревателе

#### Проблема

Вы хотите сделать корректную Web-страницу, соответствующую стандартам.

#### Решение

В спецификации языка HTML 4.01 есть три типа документа: strict (строгий), Transitional (переходный), Frameset (с фреймами). В спецификациях HTML5 и XHTML 1.1 — только один тип документа, а в спецификации XHTML 1.0, как и в HTML 4.01, — три типа.

В HTML-документе приводится только одно определение типа документа (Document Type Declaration, DTD). Используйте одно из следующих объявлений **DOCTYPE**, наиболее подходящее для вашего проекта:

♦ HTML 4.01 Strict DTD:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

♦ HTML 4.01 Transitional DTD:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd">
```

• HTML 4.01 Frameset DTD:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/1999/REC-html401-19991224/frameset.dtd">
```

#### ♦ HTML5 DTD:

<!DOCTYPE html>

#### • XHTML 1.0 Strict DTD:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

#### XHTML 1.0 Transitional DTD:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

#### • XHTML 1.0 Frameset DTD:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

#### ♦ XHTML 1.1 DTD:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

Далее приведена базовая страница с DTD спецификации HTML5 и обязательными элементами head, body и html.

```
<!DOCTYPE html>
<html>
<head>
    <title>CSS Cookbook</title>
    </head>
    <body>
    <hl>My Basic Web Page</hl>
    Epsum factorial non deposit quid pro quo hic escorol.
    </body>
</html>
```

#### Обсуждение

Объявление **DOCTYPE**, сокращение от Document Type Declaration (Определение типа документа), определяет блоки, формирующие документ на языках HTML и XHTML, и сообщает Web-обозревателям и валидаторам (средство проверки допустимости), какая версия языка HTML или XHTML применяется в вашем документе.

Объявление **DOCTYPE** должно присутствовать в начале каждого документа Webстраницы, перед элементом html для того, чтобы гарантировать соответствие вашей разметки и CSS стандартам и обеспечить обработку страниц Web-обозревателями на основе подходящих DTD.

#### Режим обратной совместимости

Спецификация XHTML требует наличия подходящего **DOCTYPE** в начале документа, в противном случае страницы могут не пройти проверку допустимости и Webобозреватели перейдут в так называемый режим совместимости со старыми версиями Web-обозревателей или *режим обратной совместимости* (quirks mode).

Table with no width a	ttribute in a <div> - Microsoft Internet Explorer</div>	
<u>File Edit View Favorites</u>	s <u>T</u> ools <u>H</u> elp	A.
🕝 Back 🔹 🐑 💌 😫	🚯 🔎 Search 👷 Favorites 🤣 🔗 🤹 🗉 🔻 🖵 🖪 🕄 🏭 🔛 👯 🦓	
Address 🙋 C:\Datafiles\Docs	\Freelance\Other_Gigs\OReilly\CSS_Cookbook\quirks.html	🔽 🄁 Go 🛛 Links 🎽 📆 🔹
RoboForm - Search	🔽 🍰 Logins 👻 🥋 (passcards) 🛛 🖓 Meryl Evans 🛛 🎲 Save 💋 Generate 🖉	
Thi	s is a paragraph before opening the table.	
Т	his is a table with one cell.	
Thi	s is a paragraph after closing the table.	
		~
<		

Рис. 1.2. Ширина таблицы в режиме обратной совместимости Internet Explorer 6 при отсутствии DOCTYPE

🕴 Table with no width attribute in a <div> - Mozilla Firefox</div>		
Elle Edit <u>V</u> iew <u>G</u> o <u>B</u> ookmarks <u>T</u> ools <u>H</u> elp		0
🔶 - 🏟 - 🎯 🔇 🚷 🗋 file:///C:/Datafiles/Docs/Freelance/Other_Gigs/OReilly/CSS_Cookbook/quirks.html 🔍 🖉 Go	A9,	
🗋 Google 🗋 Informit 🗋 LG 🗋 meryl 🗋 Renner 🔅 MTGood 🔅 GoodSpace 🗋 PKK 🗋 IVIZ 🗋 Shevlik 🗋 ISS 🦂 POPFile 🗋 ThingsIV	Vant 🚺 CC 📔 Fur	l It 📄 AW Contests 🛛 »
💢 Disable* 🔒 Cookies* 🛱 CSS* 🔂 Forms* 🔌 Images* 🕖 Information* 🖃 Miscellaneous* 💋 Outline* 🖪 Resize* 🕗 Tools* 🖺 View Sour	rce• 🔝 Options•	S
🗠 🖆 hReview Cr 🛛 🚺 ITworld.com 🛛 📆 TAZR.com v 🗶 🐚 DMNews go 🛛 🗋 QuirksMade 🔀 🗋 Activating t 🗵	Table with 🗵	🔀 Tech Info 区 🗵
This is a paragraph before opening the table.		
This is a table with one cell.		
This is a paragraph after closing the table.		
Done	Open Notebook	PR:n/a Adblock

Рис. 1.3. Ширина таблицы в стандартном режиме Firefox 1.5 с HTML 4.01 Strict DOCTYPE

Режим обратной совместимости устанавливается, если обозреватель трактует Webстраницу как дефектную. В результате такие страницы интерпретируются как некорректно размеченные и, следовательно, вынуждающие современные обозреватели неправильно их визуализировать, даже если у этих страниц безупречный HTML- и CSSкод.

Web-страница без объявления DOCTYPE, с устаревшим или непонятным из-за ошибок DOCTYPE вызывает переход в режим обратной совместимости. Поэтому при кодировании страниц проверяйте наличие и корректность объявления DOCTYPE, чтобы Webобозреватели не отображали страницы в режиме обратной совместимости.

#### Примечание

Если у Web-страницы DOCTYPE, соответствующий спецификации HTML5, современные Web-обозреватели переходят в стандартный режим, даже если в реальной разметке нет элементов HTML5. Internet Explorer для OC Windows версий 6 и 7 игнорирует параметры спецификации HTML5.

На рис. 1.2 и 1.3 показано, как таблица с шириной 100 %, содержащаяся в элементе div, отображается в Internet Explorer 6 в режиме обратной совместимости, и как она должна выглядеть в стандартном режиме.

#### Почему не использовать самый новый вариант DOCTYPE?

Один из возможных вариантов — применение новейших вариантов **DOCTYPE**, например, из спецификации HTML5. Но это не единственная возможность. В отличие от новых версий приложений, более новые варианты объявления типов документов не уменьшают актуальности уже имеющихся.

Например, вы только в крайнем случае согласились бы установить гораздо менее производительный графический редактор Photoshop 4 на современные компьютеры. А синтаксис и объявление DOCTYPE спецификации HTML4 все еще можно применять, не боясь, что Web-обозреватели не смогут визуализировать ваш контент.

#### Самая маленькая Web-страница

Предлагаемое решение содержит пример относительно короткой страницы на HTML5. Но можно привести даже более короткий и все же корректный пример.

```
<!DOCTYPE html>
<title>Small HTML5</title>
Hello world</p
```

Соответствие этих трех HTML-элементов спецификации HTML5 подтверждается проверкой этой страницы валидатором, размещенным по адресу http://validator.w3.org/ check?uri=http%3A%2F%2Fjsbin.com%2Fowata&ss=1.

#### См. также

Спецификацию HTML5 лля DTD на странице http://dev.w3.org/html5/spec/ Overview.html#the-doctype; спецификацию HTML 4.01 для DTD на странице http://www.w3.org/TR/html401/intro/sgmltut.html#h-3.3; консорциума валидаторы

W3C на http://www.w3.org/QA/Tools/#validators; статью о ростуре из журнала "A List Apart" на сайте http://www.alistapart.com/stories/doctype/; статью о режиме обратной совместимости на странице http://www.quirksmode.org/index.html?/css/quirksmode.html; информацию о режиме обратной совместимости Web-обозревателя Mozilla, объясняющую различия режимов визуализации и особенности обработки в режиме обратной совместимости, на http://developer.mozilla.org/en/docs/Mozilla%27s\_Quirks\_Mode; страницу обозревателя Opera, посвященную ростуре, на http://www.opera.com/docs/specs/doctype/.

#### 1.4. Разметка заголовков

#### Проблема

Вы хотите в одном и том же документе различать заголовки по степени важности.

#### Решение

Примените один из шести имеющихся элементов для заголовков, от h1 до h6, как показано на рис. 1.4.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
 <title>CSS Cookbook</title>
</head>
<body>
 <h1>My Basic Web Page</h1>
 Epsum factorial non deposit quid pro quo hic escorol.
 <h2>Secondary Heading</h2>
 Feles mala! cur cista non uteris? stramentum novum in ea posui.
 <h3>Tertiary Heading</h3>
 Por scientie, musica, sport etc., li tot Europa usa li sam
 vocabularium.
 <h4>Quaternary Heading</h4>
 Lex clavatoris designati rescindenda est.
 <h5>Quinary Heading</h5>
 Ire fortiter quo nemo ante iit.
 <h6>Senary Heading</h6>
 Interdum feror cupidine partium magnarum europe vincendarum.
</body>
```

```
</html>
```



Рис. 1.4. Стандартное отображение заголовков шести уровней

#### Обсуждение

Для разметки заголовков документа вы можете выбирать вариант из шести заголовков разных уровней.

При разметке используйте заголовки разных уровней по порядку. Например, если вы применяете элемент h2, заголовок под ним должен быть помещен в элемент h3 (а не в элемент h4 или h5). Заголовок страницы не следует заключать в элемент h2 (используйте для него элемент h1). Если говорить кратко, не пропускайте теги заголовков разных уровней!

При создании документа вовсе не обязательно использовать заголовки *всех* уровней. Но следите за тем, чтобы не переусердствовать с элементом h1, т. к. это может снизить ваш ранг в поисковых машинах. Применяйте элемент h1 один раз для главного заголовка вашей заметки в блоге или на Web-странице; после этого для подзаголовков в документе используйте элементы h2 и h3.

#### Примечание

Если вашему документу понадобятся элементы h4, h5 и h6, разбейте контент на несколько страниц или проанализируйте структуру документа. Документ, нуждающийся в заголовках шести разных уровней, скорее всего, перегружен информацией и не сможет удержать внимание среднестатистического посетителя в течение даже короткого промежутка времени.
Кроме того, если вас беспокоит внешний вид страницы, не волнуйтесь: благодаря мощи CSS дизайн заголовков (как и других частей страницы) можно изменить.

#### Примечание

Грамотное применение заголовков помогает людям, пользующимся программами считывания с экрана (экранными дикторами). Для иллюстрации посмотрите видео по адресу http://www.youtube.com/watch?v=AmUPhEVWu\_E.

#### См. также

Главу 3 о модификации заголовков и других часто используемых приемах верстки.

# 1.5. Подобающее цитирование

## Проблема

Вы хотите сослаться на цитаты средствами HTML, как показано на рис. 1.5.

## Решение

Для цитирования большого фрагмента текста используйте элемент blockquote.

```
<blockquote cite="http://www.example.com/">
Si fallatis officium, quaestor infitias eat se quicquam scire de factis
vestris.
</blockquote>
```



Рис. 1.5. Стандартное отображение цитат

Для цитирования отдельных фраз используйте элемент q.

```
Virgil said <q>Sic itur ad astra</q>.
```

## Обсуждение

blockquote — элемент блочного уровня. Это означает, что текст, помещенный в элемент blockquote, отделяется от остального текста за счет вставки разрыва (переноса) строки перед ним и после него.

Элемент q — *строчный* элемент, не добавляющий разрывов строк. Строчные элементы подходят для цитирования коротких фрагментов текста внутри абзаца.

#### Примечание

Как правило, элемент q визуализируется со знаками кавычек вокруг содержащегося в нем текста. Но Web-обозреватель Internet Explorer в ОС Windows не отображает эти кавычки.

Атрибут cite не обязателен для элементов blockquote и q. Его значение — URI (Universal Resource Identifier, универсальный индикатор ресурсов) источника, откуда взята цитата.

## См. также

Главу 3 с описанием часто используемых приемов верстки.

# 1.6. Вставка изображения

## Проблема

Вы хотите поместить на Web-страницу изображение, как показано на рис. 1.6.



Рис. 1.6. Изображение, включенное в документ

## Решение

Для задания местоположения файла, содержащего изображение, используйте элемент img.

```
<img src="saint-george-island.jpg" />
```

В элемент img вставьте атрибут alt для задания альтернативного текста на случай отключения изображений или применения пользователями специальных возможностей, таких как программы считывания с экрана.

```
<img src="saint-george-island.jpg" alt="St. George Island beach at sunset with an approaching storm" \ />
```

# Обсуждение

Элемент img не обращается к контенту, находящемуся в самом документе. Он только определяет место своего размещения в документе и задает свое местоположение относительно HTML-документа.

#### Дополнительные подсказки

Несмотря на то, что изображение стоит тысячи слов, значение атрибута alt должно быть сравнительно коротким описанием.

Как показано на рис. 1.6, некоторые Web-обозреватели отображают следом за курсором текст, именуемый всплывающей подсказкой и хранящийся в атрибуте изображения title.

<img src="saint-george-island.jpg" alt="St. George Island beach at sunset with an approaching storm" title="St. George Island beach" />

## Файловые форматы

К самым распространенным форматам изображений, поддерживаемых Webобозревателями, относятся GIF и JPEG. При выборе каждого из них для хранения разных типов изображений есть свои "за" и "против".

Если исходить из способа сжатия, метода уменьшения размера файла изображения, формат GIF лучше подходит для областей с ровными цветовыми полями и небольшим количеством цветовых переходов, а формат JPEG — для фотографий и трудноуловимых изменений цвета.

Все Web-обозреватели поддерживают формат PNG, но сейчас альфа-канал прозрачности поддерживается только в Internet Explorer 8 для Windows. *Альфа-канал* допускает непрозрачность или разную степень прозрачности в изображении в отличие от формата GIF, который может лишь один цвет сделать прозрачным. Если PNG-изображение с альфа-прозрачностью визуализируется в старых версиях IE, прозрачные части обычно превращаются в блоки с заливкой сплошным белым цветом.

## Чувствительность к регистру символов в имени файла

При задании файла изображения средствами языка HTML убедитесь в том, что в имени файла нет пробелов и что соблюдено соответствие строчных и заглавных букв. Не-

смотря на то, что ваша ОС возможно нечувствительна к регистру символов, Webсервер, хранящий ваши Web-файлы, может этим свойством не обладать, что помешает визуализации изображений в Web-обозревателе.

## См. также

Главу 4 с описанием верстки Web-страниц, содержащих изображения.

# 1.7. Вставка звуковых файлов средствами HTML5

# Проблема

Вы хотите добавить звуковую информацию на Web-страницу, созданную с помощью стандарта HTML5.

# Решение

Для задания звукового файла используйте элемент audio, как показано на рис. 1.7.



Рис. 1.7. Вставка звуковой информации на Web-страницу

С элементом audio связано пять атрибутов: src, autobuffer, autoplay, loop и controls. Если у вас нет атрибута controls, звуковой проигрыватель не отображается.

#### Звуковая совместимость

Во время написания этой книги не было ни одного типа звуковых файлов, который бы воспроизводился во всех Web-обозревателях, поддерживающих элемент audio спецификации HTML5, что отражено в табл. 1.1.

	Firefox 3.5	Safari 4	Chrome 3 beta	Opera 10
Ogg Vorbis	Да		Да	
MP3		Да	Да	
WAV	Да	Да		Да

Таблица 1.1. Поддержка форматов звуковых файлов стандарта HTML5

Для формирования универсального решения используйте элемент audio наряду с элементом source, который ссылается на файлы форматов и OGG, и MP3. Далее вставьте код для Flash Player с элементами embed и object.

```
<audio controls autobuffer>
<source src="html5audio.ogg" />
<source src="html5audio.mp3" />
<!-- include Adobe Flash player EMBED and OBJECT code here -->
</audio>
```

#### Примечание

Если вы действительно вставляете звуковые файлы, не рекомендуется устанавливать для них атрибут autoplay, т. к. это может помешать пользователям Web, применяющим программы-экранные дикторы.

## См. также

Разд. 1.8 о вставке видео на Web-страницу.

# 1.8. Встраивание видео средствами HTML5

# Проблема

Вы хотите включать в ваши Web-страницы видеоданные.

# Решение

Используйте элемент video из спецификации HTML5, как показано на рис. 1.8.

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
<meta charset="utf-8">
<title>CSS Cookbook</title>
</head>
<body>
<hl>Video Example</hl>
<video src="html5video.ogg" width="320" height="240"
controls poster="html5video.jpg">
<a href="html5video.jpg">
<a href="html5video.ogg">Download movie</a>
</video>
</body>
</html>
```



Рис. 1.8. Видеоданные, включенные в Web-страницу

Ширину и высоту элемента video задавать не обязательно. Если вы не укажете элемент video с соответствующими атрибутами, видеоролик будет воспроизводиться с размерами по умолчанию, заданными в видеофайле.

У видеофайла может быть собственная афиша или *постер* — статическое изображение, представляющее видеоданные в целом и напоминающее по виду пиктограмму.

Его можно переопределить с помощью атрибута poster. Изображение-постер может иметь любой формат, поддерживаемый Web-обозревателями (GIF, JPEG или PNG).

#### Примечание

Несмотря на то что атрибут controls не обязателен, из соображений удобства советую вам применять его, чтобы не вызвать раздражения у посетителей вашего сайта.

Между тегами video вы можете поместить альтернативный текст, включая ссылку на загрузку видеофайла для Web-обозревателей, не распознающих элемент video. Такой подход позволит посетителям Web-сайта просмотреть контент с помощью сторонних программ.

#### См. также

Web-caйт http://www.videolan.org/, содержащий сведения о средствах экспорта в программном приложении VLC, которое можно использовать для преобразования стандартных видеофайлов в формат OGG (поддерживаемый обозревателями Firefox и Opera).

# 1.9. Эффективное применение strong и ет

Вы хотите выделить в абзаце определенные слова или фразы, как показано на рис. 1.9.



Рис. 1.9. Стандартное отображение выделенного текста

## Решение

Для обозначения визуального выделения в документе примените элементы strong и ет.

Ma quande lingues coalesce, li grammatica del <strong>resultant lingue</strong> es plu simplic e regulari quam ti del coalescent lingues. Li nov lingua franca va esser plu simplic e regulari quam li existent Europan lingues. It va esser tam simplic quam Occidental: in fact, it va esser Occidental. A <em>un Angleso</em> it va semblar un simplificat Angles, quam un skeptic Cambridge amico dit me que Occidental es.

# Обсуждение

В стандартном отображении назначение элемента strong — отображение текста жирным шрифтом, а элемент ет отображает текст курсивом. Элемент ет следует применять для привлечения внимания или для создания контрастного отображения одного или нескольких слов по сравнению с остальной частью предложения. Например:

- Darth Vader translates loosely as Dark Father in Dutch.
- There are not 57, but 50 states in the United States of America.
- If you join him, he will *complete* your training.

Элемент strong — альтернативный вариант акцентирования внимания на словах или фразах.

Хотя использование em и strong помогает нарушить монотонность текста, применяйте их экономно и согласованно, чтобы избежать их чрезмерного употребления.

# См. также

Главу 3 с описанием других часто используемых приемов верстки.

# 1.10. Создание списков

Вы хотите сформировать на Web-странице перечень пунктов, как показано на рис. 1.10.



Рис. 1.10. Стандартное отображение маркированного списка

# Решение

Поместите список фраз в элемент ul.

```
    First Item
    Secondary Item
    Tertiary Item
    Quaternary Item
```

Затем примените элемент 11 для обозначения каждого пункта списка.

```
First Item
Secondary Item
```

```
Tertiary ItemQuaternary Item
```

44

В языке HTML есть три вида списков: маркированный, нумерованный и список определений.

Сформировать маркированный и нумерованный списки очень просто. Для разметки пунктов *маркированного* списка используйте элементы ul и li, что в результате приведет к появлению кружка слева от каждого пункта списка.

#### Примечание

Маркированный список обычно применяется для формирования основы навигационного меню.

У нумерованных списков, в которых вместо элемента ul используется элемент оl, есть последовательный номер, добавляемый в начало каждого пункта.

Как показано на рис. 1.11, *списки определений*, которые применяются для определения терминов, действуют несколько иначе, чем маркированные и нумерованные списки. Каждый пункт списка разбивается на две части: термин (dt) и определение (dd).

```
<dl>
<dd>First Term</dt>
<dd>Seriatim</dd>
<dt>Secondary Term</dt>
<dd>Sequentia</dd>
<dt>Tertiary Term<dt>
<dd>Sequens mirabitur aetas</dd>
</dl>
```



Рис. 1.11. Стандартное отображение списка определений

## См. также

Главу 6 о списках и главу 7 о ссылках и навигации.

# 1.11. Формирование ссылки на Web-страницу

# Проблема

Вы хотите сослаться на другую Web-страницу.

# Решение

Примените элемент anchor.

```
This book's <a href="http://www.csscookbook.com/">Web site</a> contains links to download more materials.
```

Для ссылки на другую страницу того же Web-сайта сошлитесь на ее имя файла.

Check out the <a href="about.html">About page</a> for more information.

# Обсуждение

Как и в случае элемента img (см. *разд. 1.6*), некоторые Web-обозреватели отображают, как показано на рис. 1.12, всплывающую подсказку, если в элементе anchor присутствует атрибут title и его значение.

```
This book's <a href="http://www.csscookbook.com/" title="Link to the book
site">Web site</a> contains links to download more material.
```



Рис. 1.12. Всплывающая подсказка, отображаемая поверх ссылки

## Ссылка на другую Web-страницу одного и того же сайта

Когда создаются ссылки в пределах одного Web-сайта, вместо ссылки-якоря (anchor) используйте *относительные ссылки*. Относительные ссылки — это адреса, которые корректны, только если вы посещаете их с определенных Web-страниц.

Предположим, что у вас есть Web-сайт, состоящий из четырех страниц, хранящихся в одной и той же *корневой папке*, главном каталоге, содержащем файлы Web-сайта, как показано на рис. 1.13.

Включение в ссылку всей информации, необходимой для указания Web-обозревателю местоположения, означает, что вы создали *абсолютную ссылку*, которая выглядит следующим образом.

<a href="http://www.csscookbook.com/services.html">Services Page</a>

Если вы хотите создать ссылку с индексной страницы на другую страницу одного и того же Web-сайта, примените *относительную ссылку*. Относительная ссылка немного компактнее абсолютной и, как в данном примере, может ссылаться в атрибуте href только на имя файла.

<a href="services.html">Services Page</a>

Относительные ссылки не содержат ни полного протокола http://, ни имени домена.



Рис. 1.13. Пример структуры каталога

Когда Web-обозреватель переходит по относительной ссылке, для формирования точной ссылки, на которую он должен перейти, используется имя домена отображаемой в данный момент страницы.

#### Перемещение по папкам

Так же как ваш персональный компьютер наполнен многочисленными папками, хранящими множество файлов для проекта, Web-сайты составлены из наборов папок и файлов. Для того чтобы сослаться из одного документа на другой документ в пределах одного Web-сайта, применяйте относительные ссылки.

Скажем, у вас есть главная страница технических спецификаций в папке specs, в свою очередь находящейся в папке widget. Структура файлов на сервере может выглядеть следующим образом:

- products/
  - widget/
    - specs/
      - specs.html

Для формирования ссылки на страницу widget со страницы specs используйте обозначение ../, чтобы сообщить Web-обозревателю о необходимости перехода в родительский каталог или на уровень выше.

<a href="../widget.html">Widget Page</a>

Если вы хотите подняться в структуре папок на два уровня выше и сослаться на главную страницу products со страницы specs, нужно сформировать ссылку следующим образом:

<a href="../../products.html">Product Page</a>

#### Применение ссылок относительно корня сайта

Использование относительных ссылок для переходов между папками большого Webсайта иногда может оказаться сложным и даже запутанным процессом. В этом случае можно применить другой тип ссылок — *ссылку относительно корня сайта*.

Вот как следует использовать ссылку относительно корня сайта для записи ссылки со страницы specs на главную страницу products из предыдущего примера:

<a href="/products/products.html">Product Page</a>

Прямой слэш обозначает протокол и доменное имя URI, своего рода сокращенная форма записи для ссылок.

#### Ссылки на конкретные элементы на Web-странице

Вы можете ссылаться на конкретные элементы в HTML-документе, создавая якори или привязки. Якорь можно создать, присвоив HTML-элементу атрибут id.

h2 id="hireme">Hire Me</h2>

Затем сошлитесь на этот якорь, указав перед именем id хэш-символ (#).

<a href="#hireme">Hire Me</a>

Если ссылку щелкнуть кнопкой мыши, Web-обозреватель выполнит переход к той части документа, у которой соответствующее имя id.

#### Примечание

Если весь документ помещается в поле вывода или окне Web-обозревателя, не будет никаких видимых изменений, указывающих на то, что Web-обозреватель перешел к ссылкеякорю.

Дизайнеры применяют якори для создания в верхней части страницы ее оглавления, что позволяет быстро переходить к разным частям документа. Такой подход особенно полезен на Web-страницах с большим объемом контента, т. к. избавляет пользователей от избыточной прокрутки.

## См. также

Главу 7 о ссылках и навигации.

# 1.12. Разметка таблиц

# Проблема

Вы хотите создать простую HTML-таблицу, как показано на рис. 1.14.

## Решение

Используйте специальные элементы, относящиеся к разметке табличных данных.

```
<caption>
```

```
Know Your IE6 Adoption Rate
</caption>
 
  2002
  2003
  2004
  2005
  2006
  2007
  2008
  <t.h>2009</t.h>
\langle t.r \rangle
  %
  45
  62
  82
  81
  78
  50
  45
  36
</t.r>
CSS Cookbook
                                     -
```



Рис. 1.14. Стандартное отображение простой HTML-таблицы

Во-первых, вставьте в начало и конец табличных данных тег . Тег определяет таблицу в целом.

Необязательный элемент <caption> предназначен для краткого описания табличных данных и вставляется сразу после открывающего элемента .

Далее, если у вашей таблицы есть заголовок, добавьте элемент thead в одну или несколько строк заголовка таблицы. Для обозначения тела таблицы, чтобы отделить его от заголовка таблицы, используйте тег tbody.

Затем вставьте теги табличных строк tr для обозначения каждой строки таблицы. Этот элемент охватывает группы отдельных ячеек таблицы. Сначала вы определяете строку, затем добавляете входящие в нее ячейки.

#### Примечание

Для столбца таблицы нет специального тега. Столбцы формируются только благодаря построению следующих друг за другом строк таблицы.

После этого вы применяете тег th для обозначения каждой ячейки, которую хотите пометить как ячейку заголовка таблицы, включающего в нашем примере годы и проценты. В тег следует вставить конкретное содержимое ячейки. По умолчанию Webобозреватели отображают текст в ячейках заголовка жирным шрифтом.

Для разметки отдельных ячеек таблицы используйте тег td. Как и в тег th, в тег td вставляется конкретное содержимое ячейки.

#### Примечание

Для того чтобы избежать разметки многочисленных табличных ячеек, вручную опробуйте генератор простых HTML-таблиц по адресу http://www.askthecssguy.com/kotatsu/ index.html.

## См. также

Главу 9 о таблицах.

# 1.13. Создание vCard (hCard) средствами HTML

# Проблема

Вы хотите поместить на Web-страницу контактную информацию, такую как на визитной карточке (рис. 1.15).



Рис. 1.15. Стандартное отображение контактной информации в формате hCard

## Решение

Используйте атрибуты class со специальными атрибутами, перечисленными в спецификации микроформата hCard (http://microformats.org/wiki/hcard).

```
<div class="vcard">
  <span class="fn n">Josiah Bartlet</span>
  <div class="org">White House</div>
  <div class="adr">
    <div class="adr">
    <div class="street-address">1600 Pennsylvania Avenue NW</div>
    <span class="locality">Washington</span>,
    <span class="region">DC</span>,
    <span class="postal-code">20500</span>
  </div>
</div>
```

Микроформат hCard предоставляет способ отображения контактной информации, включая людей, организации и местонахождения с помощью атрибутов class спецификации XHTML. Это один из многих форматов, подробно описанных в проекте Microformats Project (http://microformats.org/), цель которого предоставить стандарты для кодирования на Web-страницах средствами семантического HTML информации, пригодной для машинного считывания. Подобно дизайнерским шаблонам hCard стандартизует способ представления информации, позволяющий программному обеспечению сторонних фирм собирать информацию и использовать ее во всевозможных благих целях.

Для экономии времени и исключения возможных опечаток воспользуйтесь генератором hCard Creator (http://microformats.org/code/hcard/creator), чтобы сгенерировать разметку контактной информации на языке HTML.

## Расширение hCard

Web-сервис H2VX (http://h2vx.com/vcf/), пригодный для использования на сайте в виде букмарклета,<sup>1</sup> просматривает разметку Web-страницы, выискивая в Web-адресе данные формата hCard. Если он находит данные hCard, то предлагает посетителю загрузить данные как электронную визитку стандарта vCard.

Затем посетитель сайта может импортировать данные vCard в адресную книгу своего любимого приложения, например Outlook (Windows) или Address Book (Mac OS X).

Исполняющая программа (https://addons.mozilla.org/en-US/firefox/addon/4106) — расширение Firefox, обнаруживающее микроформатированный текст на Web-странице, а затем предоставляющее параметры для различных вариантов представления данных в зависимости от типа применяемого микроформата.

Аналогичное встраиваемое расширение есть и для Safari http://zappatic.net/safarimicroformats/.

# См. также

Валидатор hCard по адресу http://en.hcard.geekhood.net/; *разд. 1.14* об использовании HTML для разметки события.

<sup>&</sup>lt;sup>1</sup> Кстати, русский вариант названия "закладурка" ничуть не хуже (http://ru.wikipedia.org/wiki/%D0%91% D1%83%D0%BA%D0%BC%D0%B0%D1%80%D0%BA%D0%BB%D0%B5%D1%82). — Прим. пер.

# 1.14. Разметка события (hCalendar)

# Проблема

Вы хотите применить HTML для разметки информации о событии.

# Решение

Используйте атрибуты class и title со специальными атрибутами, перечисленными в спецификации микроформата hCalendar (http://microformats.org/wiki/hcalendar).

```
<div class="vevent" id="hcalendar-The-CSS-Summit">
  <a class="url" href="http://csssummit.com/">
    <abbr class="dtstart" title="2009-07-18T09:00-04:0000">July 18,
2009 9</abbr>
    - <abbr class="dtend" title="2009-07-18T18:00-04:00">6pm</abbr>
    : <span class="dtend" title="2009-07-18T18:00-04:00">6pm</abbr>
    : <span class="summary">The CSS Summit</span>
    at <span class="location">Online Conference</span></a>
<//div>
```

# Обсуждение

Микроформат hCalendar, основанный на файловом формате iCalendar, применяемом для обмена данными о событии, использует стандартизованный язык HTML для кодирования в Web-документе информации о времени и месте события.

Каждое отдельное событие определяется с помощью класса vevent. Он задает контент как входной элемент формата hCalendar.

Время начала события, dtstart, и summary — свойства, обязательные для каждого события hCalendar, в то время как dtend и location — необязательные свойства.

Памятка со списком необязательных свойств микроформата hCalendar доступна по адpecy http://microformats.org/wiki/hcalendar-cheatsheet.

## См. также

Генераторы форматов hCalendar (http://microformats.org/wiki/hcalendar/creator) и Conference Schedule (http://dmitry.baranovskiy.com/work/csc/) для упрощения представления собственного события hCalendar; *paзд. 1.13* о включении в Web-страницу контактной информации.

# 1.15. Проверка правильности HTML-разметки

# Проблема

Вы хотите убедиться в том, что HTML-разметка вашей Web-страницы синтаксически корректна.

Используйте валидатор W3C (http://validator.w3.org/) для ввода URI Web-документа и тестирования его HTML-корректности, как показано на рис. 1.16.

Вы также можете протестировать код, загрузив файл CSS или введя правила CSS.

	w C consolute sen	mitte: Desi W	C.P. Lecunology	BOOKS, TECH C	2
11.701					
W3C Mile		ITML,) of Web	e documents		
Jump	fo: Congratulatio	ns · Icons	Source Lis	ting	
This documen	t was successfu	lly checked	as XHTM	L 1.0 Strict!	
Result:	Passed				
Address :	http://www.csscookbook.com/				
Modified:	(undefined)				
Server:	Apache/2.0.48 (Fedora)				
Size:	(undefined)				
Content-Type:	text/html				
Encoding :	iso-8859-1	(detect autom	natically)		
Doctype :	XHTML 1.0 Strict	(detect autom	natically)	:	
Root Element:	html				
Root Namespace:	http://www.w3.org/1999/xhtml				
	The W3C validators	s rely on comm	unity support	for hosting and	d
	Donate and he	lp us build bett	er tools for a	better web.	

Рис. 1.16. Проверка корректности Web-страницы

## Обсуждение

Консорциум W3C разместил на своем Web-сайте надежную программу проверки кода HTML. Но ее вывод порой трудно понять. При выполнении проверки убедитесь в том, что выбраны команды More Options | Verbose Output (Дополнительные параметры | Словесный вывод).

Этот вариант отклика программы предоставляет больше справочной информации, касающейся ошибок в вашем коде, давая больше возможностей для поиска и устранения проблем.

## Создание букмарклета для HTML-валидатора

С помощью букмарклета проверяйте в HTML-валидаторе любую страницу, которую вы посетили в Web-пространстве. Букмарклет — это крошечная программка на языке JavaScript, спрятанная в элементе закладки Address (адрес).

Создайте новую закладку (строку в списке **Избранное**), назовите ее "HTML-валидатор" и затем замените любое содержимое адресного поля следующей строкой:

```
javascript:void(document.location='http://validator.w3.org/check?
charset=%28detect+automatically%29&doctype=Inline&ss=1&group=0&
verbose=1&uri='+escape(document.location))
```

При посещении другого Web-сайта щелчок кнопкой мыши по букмарклету пропустит через HTML-валидатор страницу, загруженную в Web-обозреватель в данный момент.

## См. также

Разд. 2.7 о проверке допустимости правил CSS.

# глава 2 Основы CSS

# 2.0. Введение

*Каскадные таблицы стилей* (Cascading Style Sheets, CSS) предоставляют простой способ отображения информационного содержимого или контента на ваших Webстраницах. Технология CSS на первый взгляд может показаться сложной, но в этой главе вы увидите, как просто пользоваться CSS.

Далее приведено упражнение с использованием стандартного приветствия "Hello, world!". Сначала откройте текстовый редактор или предпочитаемую вами программу для редактирования Web-страниц и введите следующие строки:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title>CSS Cookbook</title>
<head>
<body>
Hello, world!
</body>
</html>
```

Сохраните файл и просмотрите его в вашем Web-обозревателе. Как видно на рис. 2.1, эта строка отображается без каких-либо особенностей.



Рис. 2.1. Стандартное отображение HTML-текста без применения CSS

Для изменения стиля HTML-текста с помощью шрифта семейства sans serif (без засечек) добавьте следующую строку CSS (рис. 2.2):

Hello, world!

Или, сохранив шрифт по умолчанию, измените его размер, увеличив параметр fontsize до 150 % с помощью следующей строки кода (рис. 2.3):

```
Hello, world!
```



Рис. 2.2. Шрифт изменен на sans serif с помощью технологии CSS

00	CSS Cookbook	$\Box$
Back Forward R	C X A IP eload Stop Home Print 1Password	
CSS Cookbook	🛞 🖉 Technology Books, T 🛞	🚺 Christopher Schmitt : 🛞
Hello, world!		

Рис. 2.3. Увеличение размера текста

В этой главе вы узнаете о селекторах и свойствах, организации таблиц стилей и позиционировании. Приведенные здесь простые примеры подготовят вас к пониманию более сложных примеров из последующих глав.

# 2.1. Применение правил CSS к Web-странице

# Проблема

Вы хотите применить правила CSS для задания дизайна вашей Web-страницы.

## Решение

Начните с чистой страницы в редакторе Notepad (Блокнот), вашем любимом текстовом редакторе или другой программе для разработки Web-страниц, такой как Adobe Dreamweaver или Microsoft Expression.

#### Примечание

Если вы используете простой текстовый редактор, убедитесь в том, что выбраны параметры сохранения файла в виде неформатированного текста (Plain Text), а не в формате RTF.

Вставьте следующий HTML-код между тегами body и сохраните файл с именем cookbook.html:

<html> <head> <title>CSS Cookbook</title>

```
</head>
<body>
<h1> Title of page</h1>
This is a sample paragraph with a
<a href="http://csscookbook.com">link</a>.
</body>
</html>
```

Внесите в код следующие изменения (помеченные жирным шрифтом), чтобы переопределить стиль ссылок, маркированных списков и заголовков. Результат показан на рис. 2.4.

```
html>
<head>
<title>CSS Cookbook</title>
<style type="text/css">
<!...
body {
 font-family: verdana, arial, sans-serif;
}
h1 {
 font-size: 120%;
3
a {
 text-decoration: none;
}
p {
 font-size: 90%;
}
-->
</style>
 </head>
 <body>
  <h1> Title of page</h1>
  This is a sample paragraph with a
 <a href="http://csscookbook.com">link</a>.
 </body>
```

</html>



Рис. 2.4. Страница отображается иначе после вставки CSS

CSS содержит правила, состоящие из двух элементов: селекторов и свойств.

*Селектор* определяет, какой части вашей Web-страницы назначается стилевое оформление. В состав селектора входит одно или несколько свойств и их значения.

*Свойство* сообщает Web-обозревателю о том, что изменить, а *значение* позволяет ему узнать, каким должно быть это изменение.

В приведенном далее примере блока объявления селектор сообщает обозревателю об увеличении на Web-странице размера содержимого элемента h1 до 120 % по сравнению с размером по умолчанию:

```
h1 {
  font-size: 120%
}
```

В табл. 2.1 приведены селекторы, свойства и значения, использованные в реализации данного примера. В столбце "Результат" объясняется, что произойдет, когда к селектору будет применено свойство с соответствующим значением.

Селектор	Свойство	Значение	Результат
h1	font-size	120%	Размер текста больше стандартного размера
p	font-size	90%	Размер текста меньше стандартного размера

Таблица 2.1. Частичный перечень селекторов, свойств и значений, примененных в реализации данного примера

Стандартная синтаксическая запись CSS включает селектор (обычно тег, который вы хотите выделить), за которым следуют свойства и значения, заключенные в фигурные скобки:

селектор { свойство: значение; }

Для большей наглядности многие разработчики используют следующий формат записи:

```
селектор {
свойство: значение;
}
```

Добавление пробельной зоны и переносов строк облегчают чтение CSS-кода. Оба способа записи CSS верны. Применяйте тот, который более удобен вам.

Технология CSS также позволяет селекторам использовать несколько свойств одновременно для создания более сложных стилевых оформлений. При назначении одному селектору нескольких свойств используйте точку с запятой для отделения одного свойства от другого, как показано в следующем примере кода. Обратите внимание на точку с запятой после последнего свойства, за которым нет других свойств. Такой синтаксис позволит нам быстро добавлять новые свойства и снизит вероятность дополнительной ошибки из-за забытого разделителя.

```
селектор {

свойство: значение;

свойство: значение, значение, значение;

свойство: значение, значение, значение;

}

селектор, селектор {

свойство: значение;

}
```

## Размещение правил CSS

В случае внутренних таблиц стилей (см. *разд. 2.11*) правила CSS помещаются в HTMLэлемент style.

```
<style type="text/css">
<!--
-->
</style>
```

Элемент style информирует Web-обозреватель о том, что он содержит правила CSS, предназначенные для форматирования, и что обозреватель должен быть готов обработать контент. HTML-комментарий приведен здесь для того, чтобы защитить более старые версии Web-обозревателей, не знающие о том, как правильно отображать правила CSS. В новейших версиях обозревателей HTML-комментарий не понадобится.

## См. также

*Разд. 2.2* с дополнительной информацией о селекторах; *приложения 3* и 4 с перечнем селекторов.

# 2.2. Применение базовых селекторов для стилевого оформления

# Проблема

Вы хотите использовать базовые селекторы для назначения уникальных стилей разным частям Web-страницы.

# Решение

Используйте разные типы селекторов для задания стилей различным фрагментам Webстраницы (рис. 2.5):

```
<html>
<head>
<title>CSS Cookbook</title>
<style type="text/css">
<!--
* {
font-family: verdana, arial, sans-serif;
}
```

h1 {

```
font-size: 120%;
 #navigation {
  border: 1px solid black;
  padding: 40px;
 }
 li a {
  text-decoration: none;
 }
 p {
  font-size: 90%;
 -->
 </style>
</head>
<body>
 <h1>Title of Page</h1>
 This is a sample paragraph with a
<a href="http://csscookbook.com" class="warning">link</a>. Lorem ipsum dolor sit
amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut
laoreet dolore magna <em class="warning">aliquam erat volutpat</em>. Ut wisi enim
ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut
aliquip ex ea commodo consequat.
 <a href="http://csscookbook.com">Apples</a>
  <a href="http://csscookbook.com">Bananas</a>
  <a href="http://csscookbook.com">Cherries</a>
 </body>
```

```
</html>
```



Texнология CSS позволяет разными, иногда очень оригинальными способами точно указать, к каким частям Web-страницы применять стилевое оформление.

Для того чтобы лучше понять, как выбирать фрагменты Web-страницы для применения селекторов, разработчик должен знать, что информационное содержимое, размеченное с помощью HTML-кода, формирует структуру.

Несмотря на то, что элементы, используемые в HTML-коде реализации данного примера, могут выглядеть как беспорядочное нагромождение, показанное на рис. 2.6, они образуют определенную структуру.



Рис. 2.6. Элементы, использованные в реализации примера

Эта структура может быть не видна посетителю Web-страницы, но это ключевой компонент в процессе визуализации, выполняемой Web-обозревателем.

Когда обозреватель извлекает Web-страницу с сервера и начинает ее отображать, элементы страницы размещаются в структуре, которую компонует программное обеспечение Web-обозревателя.

Несмотря на то, что процесс размещения элементов в организационной структуре скорее программно ориентирован, хорошим визуальным представлением для нее может служить организационная диаграмма компании.

Организационная диаграмма для HTML-кода, использованного в данном примере, соответствует показанной на рис. 2.7.

## Селекторы типа

Селекторы типа — это селекторы, именующие элемент или HTML-тег, к которому применяется стиль. Приведенные далее правила применяют стилевое оформление шрифта к элементам Web-страницы h1 и p (puc. 2.8):

```
h1 {
  font-size: 120%;
}
```

```
p {
  color: blue;
}
```



Рис. 2.7. Элементы, использованные на Web-странице, организованы в структуру



Рис. 2.8. Элементы, выбранные с помощью правил CSS

#### Селекторы класса

Если вы хотите применять много раз одно и то же правило CSS к разным элементам, используйте *селектор класса*.

Например, селекторы класса можно применять для выделения жирным шрифтом, как предупреждений в абзаце, так и элемента списка.

Сначала создайте селектор класса warning для предупреждающих сообщений, который начинается с точки (.):

```
<html>
 <head>
 <title>CSS Cookbook</title>
 <style type="text/css">
 <!--
  * {
   font-family: verdana, arial, sans-serif;
  }
 body {
h1 {
   font-size: 120%;
  ι
  #navigation {
  border: 1px solid black;
  padding: 40px;
  }
 li a {
   text-decoration: none;
  }
 p {
   font-size: 90%;
  }
  .warning {
   font-weight: bold;
  }
  -->
 </style>
 </head>
 <bodv>
 <h1>Title of Page</h1>
 This is a sample paragraph with a
<a href="http://csscookbook.com>link</a>. Lorem ipsum dolor sit amet, consectetuer
adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna
<em class="warning">aliquam erat volutpat</em>. Ut wisi enim ad minim veniam, quis
nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo
consequat.
 <a href="http://csscookbook.com">Apples</a>
   <a href="http://csscookbook.com">Bananas</a>
   <a href="http://csscookbook.com">Cherries</a>
 </body>
```

```
</html>
```

Затем добавьте атрибут class к ссылке и элементу списка для применения к ним стилевого оформления, показанного на рис. 2.9.

62



Рис. 2.9. Изменение внешнего вида Web-страницы с помощью селекторов класса CSS

```
<html>
 <head>
  <title>CSS Cookbook</title>
  <style type="text/css">
  <!--
  * {
   font-family: verdana, arial, sans-serif;
  }
h1 {
   font-size: 120%;
  }
  #navigation {
   border: 1px solid black;
   padding: 40px;
  }
  li a {
   text-decoration: none;
  }
  p {
   font-size: 90%;
  1
  .warning {
   font-weight: bold;
  }
  -->
  </style>
 </head>
 <body>
  <h1>Title of Page</h1>
```

```
This is a sample paragraph with a
<a href="http://csscookbook.com" class="warning">link</a>. Lorem ipsum dolor sit
amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut
laoreet dolore magna <em class="warning">aliquam erat volutpat</em>. Ut wisi enim
ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut
aliquip ex ea commodo consequat.

<lass="warning">aliquam erat

<lass="warning">aliquam erat

<</li>

<</li>

<</li>
<</li>
<</li>

<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>

<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>
<</li>

<</li>
```

На рис. 2.10 показано, какие части документа выбраны заданным селектором класса.



Рис. 2.10. Структура страницы с элементами, к которым применен стиль

## Селекторы ID

*Селекторы ID* (селекторы идентификатора) похожи на селекторы класса, за исключением того, что они появляются в документе только один раз. Селектор ID может появляться много раз в документе CSS, но элемент, на который ссылается селектор ID, встречается в HTML-документе единожды.

Часто селекторы ID включаются в тег div для обозначения основных разделов документа, но могут применяться в любой части документа.

Для создания селектора ID используйте хэш-символ или знак "решетка" (#), а следом за ним вводите метку или имя:

```
#navigation {
   border:1px solid black;
   padding: 40px;
}
```

Затем вставьте атрибут id со значением navigation (рис. 2.11):

```
    <a href="http://csscookbook.com">Apples</a>
    <a href="http://csscookbook.com">Bananas</a>
    <a href="http://csscookbook.com">Cherries</a>
```



Рис. 2.11. Применение стиля к маркированному списку

#### Селекторы потомка

Селекторы потомка (descendant selector) позволяют более тонко управлять выбором фрагментов Web-страницы по сравнению с селекторами типа и класса. Обычно они состоят из двух элементов, причем второй элемент является потомком первого:

```
li a {
  text-decoration: none;
}
```

Вставьте HTML-код, в котором элемент а появляется внутри 1i, как показано на рис. 2.12:

```
    class="warning"><a href="http://csscookbook.com">Apples</a>
    <a href="http://csscookbook.com">Bananas</a>
    <a href="http://csscookbook.com">Cherries</a>
```

В этом примере всякий раз, когда ссылка или элемент а появляется в пункте списка или элементе li, применяется это правило CSS.



Рис. 2.12. Выбраны ссылки внутри элементов списка

#### Универсальные селекторы

*Универсальные селекторы* обозначаются "звездочкой" (\*) и применяются ко всем элементам (рис. 2.13).

В приведенном далее коде каждый элемент, содержащий HTML-текст, будет отображаться шрифтом Verdana, Arial или каким-либо шрифтом семейства sans-serif (без засечек):

```
* {
  font-family: verdana, arial, sans-serif;
}
```



Рис. 2.13. Каждый элемент получает стилевое оформление, заданное универсальным селектором

## См. также

Селекторы в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/ selector.html; Selectutorial, учебное пособие по применению селекторов CSS (http://css.maxdesign.com.ua/selectutorial/); руководство по поддержке селекторов Web-обозревателями на странице Web-сайта Westciv http://westciv.com/style\_ master/academy/browser\_support/selectors.html; *главу 3* с дополнительными сведениями о Web-типографике; *приложение 3* с перечнем селекторов.

# 2.3. Применение селекторов прямого потомка

# Проблема

Вы хотите задать стилевое оформление для селекторов потомка, но только для дочерних элементов, отстоящих от родительского элемента лишь на один уровень.

# Решение

Используйте *селектор прямого потомка* или дочерний селектор (child selector), который обозначается правой угловой скобкой (>), часто стоящей между двумя селекторами типа, как показано далее:

```
strong {
  text-decoration: underline;
}
div > strong {
  text-decoration: none;
}
```

# Обсуждение

В случае селектора прямого потомка стиль будет применен только к тому элементу, который является *прямым* потомком (дочерним элементом) родительского элемента.

Подчеркивается только элемент strong, не входящий в другой элемент, в данном случае элемент div (рис. 2.14):

```
Nothing happens to this part of the sentence because this
<strong>strong</strong> isn't the direct child of div.
<div>
However, this <strong>strong</strong> is the child of div.
Therefore, it receives the style dictated in the CSS rule.
</div>
```

На организационной диаграмме (рис. 2.15) показано, на какие элементы влияет данное правило CSS.

Как показано на рис. 2.14 и 2.15, первый элемент strong подчеркнут, т. к. он размещен вне элемента div.

Если в документе не представлено отношение родитель-ребенок, стиль не поддерживается. Это небольшая, но важная разница между селектором прямого потомка и селектором потомка.





Рис. 2.14. Действие правила с селектором прямого потомка

Рис. 2.15. Структура разметки документа с выделенным селектором прямого потомка

Примечание

Селекторы прямого потомка не поддерживаются в Internet Explorer 6 и более ранних версиях.

## См. также

Селекторы прямого потомка в спецификации CSS 2.1 на странице http://www.w3.org/TR/ CSS2/selector.html#child-selectors.

# 2.4. Применение селекторов соседнего элемента

# Проблема

Вы хотите назначить стиль элементу, следующему за другим конкретным элементом.

# Решение

Примените *селектор соседнего или смежного элемента* (adjacent sibling), который обозначается знаком плюс (+), стоящим между двумя селекторами, как показано в следующем примере:

```
li +li {
  font-size: 200%;
}
```

# Обсуждение

Термин "соседние элементы" описывает отношение между двумя элементами, размещенными рядом в информационном потоке разметки Web-страницы.

Эффект от применения этого правила показан на рис. 2.16. Обратите внимание на то, что стиль применяется только ко второму и третьему элементам списка, т. к. они рас-

положены следом за таким же элементом. К первому элементу списка стиль не применяется, т. к. перед ним нет элемента того же уровня.

Увидеть, на какие элементы воздействует это правило CSS, демонстрирующее селекторы соседних элементов, можно на организационной диаграмме рис. 2.17.

#### Примечание

Селекторы соседних элементов не поддерживаются в Internet Explorer 6 и более ранних версиях.



Рис. 2.16. Селекторы соседних элементов влияют только на два последних элемента списка



Рис. 2.17. Выделены элементы, к которым применяется стиль

# См. также

Селекторы соседних элементов в спецификации CSS 2.1 на странице http:// www.w3.org/TR/CSS2/selector.html#adjacent-selectors.

# 2.5. Применение селекторов атрибута

# Проблема

Вы хотите применить стилевые элементы, основываясь на существующих атрибутах HTML-элементов, а не вставлять дополнительный атрибут class.

# Решение

У селектора атрибута спецификации CSS2 есть четыре способа поиска элемента.

[attribute]

Ищется совпадение с заданным атрибутом. Например:

```
a[href] {
  text-decoration: none;
}
```

В результате, когда бы ни встретился в элементе HTML-кода атрибут href, ссылка не будет подчеркиваться.

[attribute=val]

Ищется совпадение с заданным значением. Например:

```
a[href="csscookbook.com"] {
  text-decoration: none;
}
```

В результате, когда бы ни появилась в HTML-коде ссылка, указывающая на csscookbook.com, она не будет подчеркиваться.

```
[attribute~=val]
```

Ищется значение, в котором содержится указанный атрибут, отделенный пробелом. Например:

```
a[title~="tv hd digital"] {
  text-decoration: none;
}
```

В результате, где бы ни появилось слово "digital" в атрибуте title элемента-якоря, ссылка не будет подчеркиваться.

```
[attribute|=val]
```

Ищется значение, содержащее заданное val с дефисом. Например:

```
a[title|="anti"] {
  color: red;
}
```

В результате, где бы ни появилось слово "anti-" в атрибуте title элемента-якоря, ссылка будет окрашена в красный цвет.

## Обсуждение

Несмотря на то, что селекторы CSS2 пользуются поддержкой основных Webобозревателей (за исключением Internet Explorer 6 и более ранних версий), как раз сейчас начинают внедряться следующие новые дополнения к селекторам атрибутов из спецификации CSS3, называемые селекторами атрибутов с соответствием подстрок.

[attribute^=val]

Ищутся атрибуты, значения которых начинаются с val. Например:

```
a[href^="mailto:"] {
  padding-right: 15px;
  background: url(icon-email.png) no-repeat right;
}
```

В результате, если какая-либо ссылка содержит mailto:, в ее конец добавляется пиктограмма электронной почты.

[attribute\$=val]

Ищутся атрибуты, значения которых заканчиваются val. Например:

```
a[href$='.rss'], a[href$='.atom'] {
  padding-right: 15px;
  background: url(icon-rss.png) no-repeat right;
}
```

В результате, если какая-либо ссылка содержит указание на объединение материалов для рассылки, в конец такой ссылки вставляется пиктограмма RSS.

```
[attribute*=val]
```

Ищутся атрибуты, в значениях которых содержится подстрока val. Например:

```
a[href *="username"] {
   padding-right: 15px;
   background: url(icons-star.png) no-repeat right;
}
```

В результате, как только в ссылке на сайт социальной сети появляется конкретное имя пользователя, справа от ссылки вставляется пиктограмма звездочки.

## См. также

Селекторы атрибута в спецификации CSS2 на странице http://www.w3.org/ TR/CSS2/selector.html#attribute-selectors; селекторы атрибута в спецификации CSS3 на странице http://www.w3.org/TR/css3-selectors/#attribute-selectors; статью сообщества разработчиков Opera (Opera Developer Community) о селекторах CSS3 на Webстранице http://dev.opera.com/articles/view/css3-attribute-selectors/.
# 2.6. Применение псевдоклассов

## Проблема

Вы хотите добавить стили для элементов, которые (обычно) не заданы именами, атрибутами или контентом.

## Решение

Создайте псевдокласс. В следующем примере используется *псевдокласс*, который создает общий эффект при наведении указателя мыши на HTML-ссылки:

```
a:link {
  color: blue;
a:visited {
  color: purple;
}
a:hover {
  color: red;
}
a:active {
  color: gray;
}
```

## Обсуждение

В приведенном наборе основная (еще не посещавшаяся) ссылка окрашена в синий цвет. Как только на ссылку наводят указатель мыши, она меняет цвет на красный. При щелчке ссылки кнопкой мыши ссылка становится серой. Когда же после посещения ссылки вы возвращаетесь на страницу, ссылка окрашивается фиолетовым цветом.

К псевдоклассам также относятся следующие три: :first-child (выбирающий первый дочерний элемент), :focus (см. *разд. 7.4*) и :lang (n).

### Псевдоклассы CSS

В спецификации CSS3 введен список новых псевдоклассов. Хотя Internet Explorer и не поддерживает эти новые селекторы, их поддержка другими Web-обозревателями растет с каждым днем, что отражено в табл. 2.2.

Селектор	Firefox 2	Firefox 3.5	Opera 9	Opera 10	Safari 3.1	Safari 4	Chrome
:target	Да	Да	Да	Да	Да	Да	Да
:enabled	Да	Да	Да	Да	Да	Да	Да
:disabled	Да	Да	Да	Да	Да	Да	Да
:checked	Да	Да	Да	Да	Да	Да	Да
:default		Да	Да	Да			

Таблица 2.2. Поддержка Web-обозревателями псевдоклассов CSS3

Таблица	2.2	(окончание)
ruomuqu		

Селектор	Firefox 2	Firefox 3.5	Opera 9	Opera 10	Safari 3.1	Safari 4	Chrome
:valid	Да	Да	Да	Да			
:invalid	Да	Да	Да	Да			
:in-range	Да	Да	Да	Да			
:out-of-range	Да	Да	Да	Да			
:required			Да	Да			
:root	Да	Да	Да	Да	Да	Да	Да
:not()	Да	Да		Да	Да	Да	Да
:nth-child()				Да	Да	Да	Да
:nth-last-child()				Да	Да	Да	Да
:nth-of-type()				Да	Да	Да	Да
:nth-last-of-type()				Да	Да	Да	Да
:last-child		Да		Да	Да	Да	Да
:first-of-type				Да	Да	Да	Да
:last-of-type				Да	Да	Да	Да
:only-child		Да		Да	Да	Да	Да
:only-of-type				Да	Да	Да	Да
:empty		Да		Да	Да	Да	Да

### См. также

Псевдоклассы в спецификации CSS2 на странице http://www.w3.org/TR/CSS2/ selector.html#pseudo-class-selectors; псевдоклассы в спецификации CSS3 на странице http://www.w3.org/TR/css3-selectors/#pseudo-classes.

## 2.7. Применение псевдоэлементов

### Проблема

Вы хотите задать стиль для определенных характеристик элемента без введения новой разметки, например с помощью элемента span.

## Решение

Примените псевдоэлемент. На рис. 2.18 показан пример включения псевдоэлемента ::first-letter:

```
p::first-letter {
  font-size: 200%;
  font-weight: bold;
}
```



Рис. 2.18. Стилевое оформление первой буквы (буквицы)

### Примечание

В спецификации CSS3 был добавлен идентификатор, состоящий из двойного двоеточия, но для того чтобы добиться его поддержки в Internet Explorer 8, необходимо скопировать правила CSS в исходном варианте с одинарным двоеточием, обеспечивающим кроссплатформенную поддержку Web-обозревателей.

На рис. 2.19 показано применение псевдоэлемента ::first-line для стилевого оформления всей первой строки. Если первая строка не является законченным предложением



Рис. 2.19. Стилевое оформление первой строки

или содержит начало второго предложения, псевдоэлемент ::first-line все равно окажет влияние только на первую строку.

```
p::first-line {
  font-size: 200%;
  font-weight: bold;
}
```

## Обсуждение

Для стилевого оформления документа с помощью большинства селекторов разработчик может использовать элементы и их размещение внутри Web-документа.

Но иногда разработчики с помощью псевдоэлементов могут задать стиль фрагмента, который не помечен как элемент. К псевдоэлементам CSS2 относятся ::first-letter, ::first-line, ::before и ::after.

### См. также

Псевдоэлементы в спецификации CSS 2.1 на странице http://www.w3.org/TR/ CSS2/selector.html#pseudo-element-selectors; псевдоэлементы в спецификации CSS3 на странице http://www.w3.org/TR/css3-selectors/#pseudo-elements.

# 2.8. Когда использовать селекторы класса и селекторы ID

### Проблема

Вы хотите определить наилучший способ применения селекторов класса и селекторов ID.

## Решение

Используйте селекторы класса, когда стиль в документе нужно применять много раз, и селекторы ID для однократного применения стиля в документе.

В приведенной далее таблице стилей #banner и #content — селекторы ID, а .title и .content — селекторы класса.

```
body {
  margin: 0;
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: .75em;
  padding: 0;
  }
  #banner {
  margin-top: 0;
  margin-bottom: 0;
  background-color: #900;
```

```
border-bottom: solid 1px #000;
padding: 5px 5px 5px 10px;
line-height: 75%;
color: #fff;
}
#sub banner {
background-color: #ccc;
border-bottom: solid 1px #999;
font-size: .8em;
font-style: italic;
padding: 3px 0 3px 10px;
}
#content {
position: absolute;
margin-left: 18%;
width: 40%;
top: 100px;
padding: 5px;
}
#nav1 {
position: absolute;
width: 30%;
left: 60%;
top: 100px;
padding: 5px;
}
#nav2 {
position: absolute;
padding: 5px 5px 5px 10px;
top: 100px;
width: 15%;
}
#footer {
text-align: center;
padding-top: 7em;
}
.warning {
font-weight: bold;
color: red;
}
.title {
font-size: 120%;
}
.content {
font-family: Verdana, Arial, sans-serif;
margin-left: 20px;
margin-right: 20px;
}
.footer {
font-size: 75%;
}
```

76

### Далее приведены селекторы ID и селекторы класса в HTML-коде:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
 <title>CSS Cookbook</title>
 k href="1-2.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div id="header">
 <h1>CSS Collection</h1>
 <h2>Showcase of CSS Web Sites</h2>
</div>
<div id="content">
 <h3>Content Page Title</h3>
 Content Item Title
 Content goes here.
</div>
<div id="navigation">
 <h3>List Stuff</h3>
 <a href="http://csscookbook.com/">Submit a site</a><br />
 <a href="http://csscookbook.com/">CSS resources</a><br />
 <a href="http://csscookbook.com/">RSS</a><br />
 <h3>CSS Cookbook Stuff</h3>
 <a href="http://csscookbook.com/">Home</a><br />
 <a href="http://csscookbook.com/">About</a><br />
 <a href="http://csscookbook.com/">Blog</a><br />
 <a href="http://csscookbook.com/">Services</a><br />
</div>
<div id="blipverts">
 <h3>Ads go here.</h3>
</div>
<div id="siteinfo">
  Copyright 2006
 </div>
</body>
</html>
```

## Обсуждение

Селекторы ID обозначают уникальные атрибуты, представленные одним экземпляром в структурном дереве документа, тогда как селекторы класса могут применяться на Web-странице многократно. Запомните, что в селекторах ID используется знак решетки (#), а селекторы класса начинаются с точки (.).

Как правило, разработчики будут пользоваться селекторами ID для обозначения уникальных секций Web-страницы. В только что приведенном примере страница разделена на следующие секции:

- ♦ header
- ♦ content

- navigation
- ♦ blipverts
- ♦ siteinfo

Назначив этим секциям их собственный селектор ID, дизайнеры могут применить специальные стили к этим областям страницы, не используя эти стили для оформления других секций. Реализуется это с помощью сочетания селекторов потомка с селекторами ID.

В следующем примере к разным элементам h3 применяются разные правила CSS:

```
#content h3 {
  font-size: 2em;
  font-weight: bold;
}
#navigation h3 {
  font-size: 0.8em;
  font-weight: normal;
  text-decoration: underline;
}
```

### Элементы группировки в HTML5

Несмотря на то, что спецификация HTML5 на современном этапе все еще остается проектом в стадии разработки, ко времени написания этой книги в ней созданы новые элементы, заменяющие распространенные способы группировки HTML-документа с помощью элемента div. Далее перечислены некоторые из этих новых элементов спецификации HTML5.

- ♦ hearder
- ♦ nav
- section
- ♦ article
- ♦ aside
- ♦ footer

Вместо группировки содержимого средствами HTML, приведенной далее:

```
<div id="header">
...
</div>
<div id="content">
...
</div>
<div id="navigation">
...
</div>
<div id="blipverts">
...
</div>
<div id="siteinfo">
...
</div>
```

вы выполняете группировку в документе на HTML5 следующим образом, в результате получая более четкую разметку:

<header>

```
//header>
</section>
...
</section>
</nav>
...
</nav>
<aside>
...
</aside>
<footer>
...
</footer>
...
```

Вы можете начать применять элементы HTML5, но есть несколько предупреждений.

Во-первых, вы должны задать новый **DOCTYPE** для HTML5, который легче запомнить, чем DOCTYPE для XHTML:

<!DOCTYPE html>

Затем необходимо использовать JavaScript, чтобы заставить Internet Explorer интерпретировать новые элементы как элементы блочного уровня:

```
<script type="text/javascript">
document.createElement("header");
document.createElement("section");
document.createElement("nav");
document.createElement("aside");
document.createElement("footer");
</script>
```

### Примечание

Несмотря на то, что вы можете положиться на JavaScript при внедрении элементов блочного уровня в Internet Explorer, некоторые Web-дизайнеры решили сделать шаг назад: они все еще применяют элементы div, но задают значения атрибутов id, соответствующие новым элементам HTML5.

Благодаря этому приему они готовятся к тому моменту, когда спецификация HTML5 получит более широкую поддержку в Web-обозревателях. И тогда разработчики смогут с помощью простого поиска и замены в своем коде преобразовать разметку Web-страницы в соответствии с HTML5.

Кроме того, при задании стилевого оформления элементов не забудьте описать их как элементы блочного уровня:

```
header, section, nav, aside, footer {
  display: block;
}
```

```
79
```

## См. также

Активизируемый щелчком кнопки мыши список элементов HTML5 на странице http://simon.html5.org/html5-elements; селекторы ID в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/selector.html#id-selectors; селекторы класса в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/selector.html#class-html.

# 2.9. О свойствах CSS

## Проблема

Вы хотите узнать больше о свойствах CSS.

## Решение

В примерах этой главы используются популярные свойства, такие как color, fontfamily, font-size и text-decoration. Свойства заключаются в фигурные скобки и сразу следом за свойствами располагаются их значения, как показано в следующей обобщенной синтаксической записи:

```
selector {
  property: value;
}
```

Реальный пример может выглядеть так:

```
li {
   list-style-type: circle;
}
```

Каждый раз, когда в документе появляется элемент li, стандартный знак маркера в нем заменяется кружком.

## Обсуждение

Селекторы определяют, к каким фрагментам Web-документа применять стиль, в то время как свойства указывают, как селекторы должны быть изменены.

Например, свойство color означает изменение цвета элемента, но не указывает конкретного цвета для изменения. Это задача значения свойства. В табл. 2.3 перечислены дополнительные свойства и их значения, а также результат их воздействия.

Свойство	Значение	Результат
font-weight	bold	Добавляет жирное начертание текста
border-color	Имя цвета или шестнадцатеричное HTML-значение (например, #000000 для черного и #ffffff для белого)	Добавляет цвет рамки

### Таблица 2.3. Краткий перечень свойств CSS

### Таблица 2.3 (окончание)

Свойство	Значение	Результат
border-style	solid	Добавляет сплошную линию
	dotted	Добавляет пунктирную линию
	dashed	Добавляет штриховую линию
	double	Добавляет двойную линию
text-align	left	Выравнивает текст по левому краю
	center	Выравнивает текст по центру
	right	Выравнивает текст по правому краю
	justify	Выравнивает текст по ширине

Изучение нового языка, даже не такого сложного, как CSS, может испугать, если твердо не усвоить, какими средствами или свойствами он обладает. Если вы новичок в CSS, потратьте некоторое время и напишите код с использованием максимально возможного количества свойств, перечисленных в *приложении 2*. Чем лучше вы познакомитесь со свойствами CSS, тем легче вам будет кодировать Web-страницы.

### См. также

Таблицу с полным перечнем свойств на сайте консорциума W3C на странице http://www.w3.org/TR/CSS21/propidx.html; описание CSS Properties на сайте HTML Dog (http://www.htmldog.com/reference/cssproperties/); подробное описание применения свойства border в *pa30. 4.4*; полный перечень свойств CSS в *приложении 2*.

# 2.10. Представление о блочной модели

## Проблема

Вы хотите лучше понять блочную или контейнерную модель и поведение окружающих информационный блок полей (margin), рамки (border) и отступов (padding).

### Решение

У каждого элемента блочного уровня, такого как р или div, есть четыре края: верхний, правый, нижний и левый. Все эти стороны блочных элементов формируются из трех зон, окружающих информационное наполнение блока (рис. 2.20).

Таким образом, каждый блочный элемент состоит из четырех секций.

- Content (контент) реальное информационное наполнение, такое как текст, изображения, апплеты на языке Java и другие объекты. Область контента находится в середине блока.
- Padding (отступ) пробельная зона, которая окружает область контента.



Рис. 2.20. Вид блочной модели спереди и сбоку

- Вогder (рамка) следующая наружная зона, окружающая зону отступа и формирующая рамку вокруг блока.
- ◆ Margin (поле) зона, начинающаяся от края рамки и располагающаяся снаружи от нее.

По умолчанию величина поля (margin) равна 0, т. е. поле сливается с краем рамки. Рамка (border) со значением 0 совпадает с краем пробельной зоны или отступа.

Очевидно, что отступ (padding) со значением 0 выравнивается по контенту. Положительные значения расширяют блоки.

### Обсуждение

Для того чтобы представить себе блочную модель, мысленно нарисуйте картонную коробку, лежащую на полу.

Посмотрите на нее сверху и вы увидите ее четыре края: верхний, правый, нижний и левый. Коробка может быть большой или маленькой, т. к. вы можете изменять свойства height (высота) и width (ширина).

```
div {
    height: 150px;
    width: 150px;
}
```

Кладите в коробку книги до тех пор, пока всю ее не заполните книгами.

```
<div>
Moby Dick
The red Badge of Courage
The Catcher in the Rye
</div>
```

Для того чтобы легче было увидеть края коробки, обведите ее рамкой, как показано на рис. 2.21:

div {
 border: thin solid #000000;

```
height: 150px;
width: 150px;
}
```



Рис. 2.21. Блок, обведенный рамкой

Книги лежат друг на друге или тесно прижаты друг к другу, что плохо для них, особенно в данном примере, т. к. это коллекционные книги.

Поэтому добавим отступы (padding) между книгами и блоком со свойством padding, обеспечивающим дополнительное воздушное пространство и защиту. Чем больше вы зададите отступы, тем меньше книг поместится в коробку (блок). В пример, показанный на рис. 2.22, добавлен некоторый отступ:

```
div {
  border: thin solid #000000;
  height: 150px;
  width: 150px;
  padding: 10px;
}
```

### Примечание

Вставка отступа изменяет общий размер блока, несмотря на установку ширины и высоты, равных 150 пикселам. После добавления отступа со всех сторон блока новая ширина равна 170 пикселам (отступ, равный 10 пикселам, устанавливается слева и справа). Высота также теперь равна 170 пикселам.

Вам понадобится еще одна коробка для хранения книг, которые не поместились в первую. Создайте рядом с первым еще один блок и введите в него оставшиеся книги. Поместите новый блок под первым блоком, как показано на рис. 2.23:

```
<div>
Moby Dick
The red Badge of Courage
The Catcher in the Rye
</div>
```

```
<div>
The Red Queen
Awakening
The Scarlet Letter
</div>
```



Рис. 2.22. Добавленный отступ



Рис. 2.23. Дополнительный перечень добавленных книг

Но вы хотите раздвинуть блоки (коробки), чтобы они не лежали один на другом. Для этого измените расстояние между блоками с помощью свойства margin (рис. 2.24):

```
div {
  border: thin solid #000000;
  height: 150px;
  width: 150px;
  padding: 10px;
  margin: 25px;
}
```



Рис. 2.24. Вставка полей в элементы блочного уровня

Для того чтобы лучше выделить оба блока, измените свойство border. Как поле или отступ, рамка может быть настолько толстой или тонкой, насколько захотите. На рис. 2.25 рамка увеличена до 5 пикселов:

```
div {
  border: 5px double #000000;
  height: 150px;
  width: 150px;
  padding: 10px;
  margin: 25px;
}
```

К этому моменту блочная модель обоих элементов изменена единообразно. Вы отрегулировали величину полей, отступов и толщину рамок со всех сторон. Но можно также изменять конкретные края блочной модели.

Например, если нужно откорректировать правую сторону элемента div, оставив настройку для других сторон без изменения, код может быть похож на приведенный далее (рис. 2.26).



Рис. 2.25. Рамка, увеличенная до 5 пикселов



Рис. 2.26. Настройка свойств правой стороны блока

```
div {
  border: 5px double #000000;
  height: 150px;
  width: 150px;
  padding: 10px;
  margin: 0px;
  border-right: 1px solid #000000;
  padding-right: 1px;
  margin-right: 1px;
}
```

Вы с таким же успехом можете изменить другие края. Например, при использовании свойства margin код может выглядеть следующим образом:

```
div {
  margin-top: lpx;
  margin-right: lpx;
  margin-bottom: lpx;
  margin-left: lpx;
}
```

Корректируя параметры сторон и различные свойства блочной модели, разработчики могут улучшить формат отображения своих Web-страниц.

### См. также

Блочную модель CSS 2.1 на странице http://www.w3.org/TR/CSS21/box.html; блочную модель информационного и обучающего Web-caйта Brain Jar на странице http://www.brainjar.com/css/positioning/default.asp; и интерактивную CSS Box Model на сайте http://www.redmelon.net/tstme/box\_model/.

# 2.11. Связь стилей с Web-страницей

## Проблема

Вы хотите знать разные способы добавления стилей к Web-странице.

## Решение

Стили можно применить тремя способами: внутренним, внешним и встроенным (inline). Внутренняя таблица стилей вставляется в начальную часть HTML-документа, в тег head.

```
<style type="text/css">
<!--
#header {
width: 100%;
height: 100px;
font-size: 150%
}</pre>
```

```
#content {
  font-family: Verdana, Arial, sans-serif;
  margin-left: 20px;
  margin-right: 20px
}
.title {
  font-size: 120%
}
-->
</style>
```

### Примечание

Обратите внимание на использование комментариев сразу после элемента <style>. Комментарии вставляются для того, чтобы скрыть CSS-описание и уберечь его от отображения на макете Web-страницы или визуализации его обозревателем каким-либо нежелательным образом.

Внешние таблицы стилей хранятся в отдельном файле, который объединяется с HTMLфайлом с помощью механизма связывания (linking). Приведенный далее код сохраняется в собственном файле:

```
/* CSS Document */
h1 {
  font-size: 150%;
}
h2 {
  font-size: 120%;
}
p {
  font-family: Verdana, Arial, Helvetica, sans-serif;
}
```

#### Предупреждение

Обратите внимание на то, что во внешней таблице стилей нет элемента style.

Для связывания внешней таблицы стилей, содержащей описанные paнee стили, добавьте в код Web-страницы между тегами head следующую строку:

<link href="screen.css" rel="stylesheet" type="text/css" media="screen" />

Встроенные стили действуют аналогично элементу font, потому что они вставляются в разметку, на которую воздействуют.

```
<h1 style="font-family: Verdana, Arial, sans-serif;
font-size: 150%; color: blue;">Page Title</h1>
```

Hello, world!

### Обсуждение

Есть три типа таблиц стилей.

- ♦ Внешняя. Все Web-страницы связываются с внешней таблицей стилей, которая не содержит ничего, кроме стилей CSS. Если вы хотите изменить цвет шрифта на всех страницах, связанных с этой таблицей стилей, просто измените внешнюю таблицу. Свяжите страницу с таблицей стилей с помощью тега link.
- Внутренняя. У отдельной Web-страницы есть своя таблица стилей, поэтому стили применяются только к этой странице, а не ко всем Web-страницам. Определите внутренние стили в тегах style.
- Встроенная. Встроенные стили действуют аналогично тегу font, применяя стилевое оформление к конкретному тегу на Web-странице. Разработчики редко используют встроенные стили.

Внешние и внутренние таблицы стилей Web-сайтов требуют меньше времени в процессе технического сопровождения по сравнению со встроенными стилями. Отсутствие необходимости применять тег font для каждого элемента, нуждающегося в стилевом оформлении, сохраняет файл компактным и логически стройным.

Например, вы получили в наследство Web-страницу, на которой весь текст синий и для управления его размером использован элемент font. К вам приходят заявки на изменение цвета текста с синего на черный, поэтому вы ищите все вхождения элемента для замены цвета с blue на black, подобные приведенной далее строке:

```
<font size="2" color="blue">Text goes here</font>
```

Для замены во всех элементах p blue на black во внешней таблице стилей потребуется два шага. Открыть файл CSS и изменить цвет.

```
p {
  color: black;
}
```

Во внутренней таблице стилей вы можете изменить цвет текста с синего на черный поиском тега style в верхней части страницы и заменой blue на black.

```
<style>
<!--
p {
color: black;
}
-->
</style>
```

### Когда применять встроенные стили

При использовании встроенных стилей замена цвета потребует столько же времени, сколько нужно для исправления исходного файла с тегом font:

Test goes here.

Почему же может возникнуть потребность в применении встроенных стилей, учитывая затраты времени на внесение изменений? Такое случается редко, но у вас может быть

контент, появляющийся один раз на всем Web-сайте и нуждающийся в особом стилевом оформлении. Вместо того чтобы загромождать внешнюю таблицу стилей стилем для одного фрагмента, вы используете встроенные стили.

### Когда применять внутренние таблицы стилей

Что касается внутренних и внешних таблиц стилей, то в большинстве сайтов применяются внешние таблицы. Тем не менее, начиная писать CSS-код для верстки Webстраницы, лучше всего отталкиваться от внутренней таблицы стилей. Когда вы достигнете точки, в которой проект завершается или становится слегка громоздким, перенесите таблицу стилей в отдельный файл. Затем внесите изменения, необходимые для перехода к внешней таблице стилей.

У вас также может быть специальная страница, не связанная с Web-сайтом или использующая особые стили. В этом случае легче применять внутреннюю таблицу стилей, чем загромождать внешнюю таблицу.

### См. также

Секцию Style Sheets в спецификации языка HTML 4.01 на странице http:// www.w3.org/TR/html401/present/styles.html; "CSS How to Insert a Style Sheet" (CSS, как вставить таблицу стилей) на Web-сайте W3Schools http://www.w3schools.com/ css/css\_howto.asp.

# 2.12. Об источниках CSS

## Проблема

Вы хотите знать все варианты связывания правила CSS с документом.

### Решение

Вы можете применять стили в документе следующими способами:

- с помощью собственной внутренней таблицы стилей Web-обозревателя или пользовательского агента;
- с помощью таблицы стилей пользователя (если он ее создал);
- с помощью вашей (авторской) таблицы стилей, которая может быть:
  - встроенной таблицей стилей;
  - внедренной или внутренней таблицей стилей;
  - импортированной таблицей стилей;
  - связанной или внешней таблицей стилей.

## Обсуждение

Чем выше в приведенном списке находится источник происхождения правил CSS, тем шире область действия у этих правил по сравнению с другими, порожденными други-

ми источниками. Знание этого списка полезно при поиске и устранении возможных проблем, возникающих в ваших Web-проектах.

## См. также

*Разд. 2.13* с информацией о порядке сортировки правил в CSS; *главу 11* о программистских трюках, искусственных приемах и выявлении неисправностей.

# 2.13. Порядок сортировки правил в CSS

## Проблема

Вы хотите знать, в каком порядке Web-обозреватель применяет правила CSS.

### Решение

Основное практическое правило гласит: "Ближайшее к контенту правило CSS побеждает" любое другое правило CSS.

## Обсуждение

При наличии множества способов связывания CSS с Web-документом (см. *разд. 2.12*) у обозревателя должна быть возможность разрешения потенциальных конфликтов в случае появления одних и тех же или похожих правил из разных источников.

Для разрешения конфликтов в ваших правилах CSS следуйте приведенным далее рекомендациям:

- собственные пользовательские стили обладают более высоким приоритетом по сравнению со стилями Web-обозревателя;
- у авторских (ваших) стилей более высокий приоритет, чем у пользовательских стилей;
- внутренние стили имеют более высокий приоритет по сравнению с внешними или импортированными;
- встроенные стили обладают более высоким приоритетом, чем внутренние, внешние или импортированные.

Например, у нас есть набор абзацев, для которых задан шрифт sans-serif, как показано на рис. 2.27.

```
p {
  font-family: "Gill Sans", Trebuchet, Calibri, sans-serif;
}
```

Но если мы введем еще одно правило для стилевого оформления абзацев шрифтом serif и поместим это новое правило перед предыдущим, как показано в приведенном далее коде, абзацы останутся неизменными:

```
p {
font-family: Garamond, "Hoefler Text", "Times New Roman", Times, serif;
}
p {
font-family: "Gill Sans", Trebuchet, Calibri, sans-serif;
}
```



Рис. 2.27. В качестве шрифта абзацев задан шрифт семейства sans-serif

Только когда мы вставим правило со шрифтом serif после правила со шрифтом sansserif, в Web-обозревателе произойдет изменение, показанное на рис. 2.28:

```
p {
  font-family: "Gill Sans", Trebuchet, Calibri, sans-serif;
}
p {
  font-family: Garamond, "Hoefler Text", "Times New Roman", Times, serif;
}
```

И снова побеждает правило CSS, ближайшее к контенту.

Но из этого правила есть исключение — и тут в игру вступает специфичность (см. *разд. 2.15*).

### См. также

*Разд. 2.12* с информацией о способах связывания правила CSS с документом; *разд. 2.15* о том, как определить специфичность.



Рис. 2.28. Абзацы отображены шрифтом семейства serif

# 2.14. Применение *!important* для переопределения отдельных правил CSS

## Проблема

Вы хотите сделать отдельные правила CSS важнее остальных.

## Решение

Для переопределения отдельного правила CSS примените объявление !important:

```
p {
  font-size: 12px !important;
}
```

# Обсуждение

В некоторых Web-обозревателях у пользователя может быть таблица стилей, применяемая для просмотра Web и позволяющая задавать размеры шрифта и другие свойства CSS в соответствии со вкусами пользователя.

Но как разработчик Web-документа, вы, возможно, хотите быть уверены в том, что ваши проекты отображаются так, как вы планировали. Правило !important дает вам (очень) слабую уверенность в том, что ваши проекты останутся нетронутыми.

### Пользователь управляет своим интерфейсом

Природа Web-пространства такова, что проекты никогда не воспроизводятся одинаково или с пиксельной точностью на разных дисплеях. Следовательно, объявление !important не может гарантировать того, что ваши стили будут отображаться в Webобозревателе пользователя так, как вы рассчитывали. При отображении Web-страницы в обозревателе "последнее слово" за пользователем.

Несмотря на то, что вы как разработчик пишите правила !important, пользователь тоже может задать это объявление в своей собственной таблице стилей.

В спецификации CSS2 правила !important, которые пользователь может написать, переопределяют любые правила !important, написанные дизайнером.

### См. также

Правила !important в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/cascade.html#important-rules.

# 2.15. Определение специфичности

## Проблема

Вы хотите понять, как разрешаются конфликтные ситуации в таблицах CSS, если источник и порядок сортировки у правил CSS одни и те же.

## Решение

В каждом правиле CSS есть информация, которая позволяет Web-обозревателю (и нам) узнать вес или специфичность правила.

Рассмотрим следующие три правила CSS:

```
#header p.big {
  font-family: Impact, Haettenschweiler, "Arial Narrow Bold", sans-serif;
}
p.big {
  font-family: Futura, "Century Gothic", AppleGothic, sans-serif;
}
p {
  font-family: "Gill Sans", Trebuchet, Calibri, sans-serif;
}
```

Чем большей специфичностью обладает правило CSS, тем больше у него шанс победить в конфликте с другим правилом. Как показано на рис. 2.29, при просмотре в Webобозревателе первое правило CSS (со шрифтом Impact) побеждает.

Для того чтобы понять, почему первое правило выигрывает, определим специфичность правила CSS. Таблица 2.4 поможет в этом.

Пример селектора	Встроенный стиль	Количество селекторов ID	Количество селекторов класса	Количество элементов
р	0	0	0	1
p.big	0	0	1	1
#header p.big	0	1	1	1

Таблица 2.4. Справка для определения специфичности

В соответствии с табл. 2.4:

- ♦ значение специфичности селектора р 0,0,0,1;
- у селектора р.big благодаря наличию селектора класса значение специфичности 0,0,1,1;
- ◆ у селектора #header p.big значение специфичности равно 0,1,1,1 благодаря наличию селекторов класса и идентификатора.

В этих примерах у последнего селектора наибольшая специфичность, и следовательно, он выигрывает в соревновании.



#### Рис. 2.29. Победившее правило CSS

## Обсуждение

Источник правил CSS и порядок их сортировки помогают Web-обозревателю определить правила, побеждающие другие правила (и объявление !important позволяет определенным правилам переопределять другие правила). Когда эти методы определения правил-победителей терпят неудачу, возникает конфликт. В спецификации CSS есть наготове способ обработки таких конфликтов — специфичность самого правила CSS.

Чем больше специфичность правила CSS, тем выше вероятность того, что правило выиграет.

### Примечание

Специфичность при наличии универсального селектора равна 0,0,0,0. Унаследованные значения не обладают специфичностью.

Есть несколько интерактивных калькуляторов специфичности CSS, которые помогут вам определить специфичность правил. Один такой калькулятор можно найти на странице http://www.suzyit.com/tools/specificity.php.

### См. также

Статью Эрика Мейера (Eric Meyer) о специфичности на странице http://meyerweb.com/ eric/css/link-specificity.html; статью Молли Хольцшлаг (Molly Holzschlag) о специфичности в CSS2 и CSS 2.1 на сайте http://www.molly.com/2005/10/06/css2-and-css21specificity-clarified/.

# 2.16. Как применять разные типы таблиц стилей

### Проблема

Вам нужно предоставить таблицы стилей для разных типов устройств вывода информации (media type), таких как акустические (aural), печатные или переносные (handheld).

### Решение

Создайте отдельные внешние таблицы стилей для разных устройств вывода и назовите их в соответствии с типом устройства, например print.css, screen.css, handheld.css. Затем для связывания с этими стилями используйте на Web-странице элемент link с типом устройства. Другой способ — применение правила @media.

Далее приведен текст файла print.css:

```
body {
  font: 10pt Times, Georgia, serif;
  line-height: 120%;
}
```

Новый файл, названный screen.css:

```
body {
  font: 12px verdana, arial, sans-serif;
  line-height: 120%;
}
```

Затем еще один файл, названный projection.css:

```
body {
  font: 14px;
  line-height: 120%;
}
```

Далее на Web-странице сошлитесь на эти три файла, используя следующие строки в разделе head. У каждой ссылки свой тип устройства вывода:

```
<link rel="stylesheet" type="text/css" href="/css/print.css"
media="print" />
<link rel="stylesheet" type="text/css" href="/css/screen.css"
media="screen" />
<link rel="stylesheet" type="text/css" href="/css/projection.css"
media="projection" />
```

Для задания в одной таблице стилей разных правил для различных типов устройств можно воспользоваться правилом @media:

```
<style type="text/css">
<!--
Qmedia print {
body {
  font: 10pt Times, Georgia, serif;
 }
}
Qmedia screen {
body {
  font: 12pt Verdana, Arial, sans-serif;
 }
}
Qmedia projection {
body {
  font-size: 14pt;
 }
}
@media screen, print, projection {
body {
  line-height: 120%;
 }
}
-->
</style>
```

## Обсуждение

Разработав стили для печати, добавьте их в файл print.css, и только эти стили будут применяться во время печати. Это гарантирует печать страницы без потери места на

листе и чернил на печатание изображений. Только устройства, поддерживающие конкретный тип вывода, будут видеть соответствующие этому типу стили CSS. Таблицы стилей с учетом типа устройства вывода не влияют на устройства других типов и на саму Web-страницу.

Правило @media позволяет вам поместить все типы устройств в одну таблицу стилей.

На рис. 2.30 показано, как выглядит Web-страница в оригинальном экранном формате. Пользователям не нужно печатать боковые компоненты, поэтому скопируйте файл таблицы стилей screen.css и сохраните его как новый с именем print.css. Вместо того, чтобы начинать с нуля, измените содержимое файла screen.css, оптимизируя вывод Web-страницы на печать. В файле print.css были изменены следующие компоненты файла screen.css.

```
#sub banner {
background-color: #ccc;
border-bottom: solid 1px #999;
 font-size:.8em;
 font-style: italic;
padding: 3px 0 3px 5px;
}
#nav1 {
position: absolute;
width: 30%;
 left: 60%;
 top: 100px;
padding: 5px 5px px 5px 0;
}
#nav2 {
position: absolute;
 width: 15%;
 left: 1%;
 top: 100px;
padding: 5px 5px px 5px 0;
}
h1 {
 text-align: left;
 color: #fff;
 font-size: 1.2em;
 text-align: left;
margin-bottom: 5px;
margin-top: 5px;
}
.entry {
padding-bottom: 20px;
padding: 5px;
border: solid 1px #999;
background-color: #fcfcfc;
margin-bottom: 25px;
}
```



Рис. 2.30. Так будет выглядеть напечатанная Web-страница при отсутствии файла print.css

На рис. 2.31 показано, как страница выглядит при печати благодаря файлу print.css.

```
#sub_banner {
  display: none;
}
#nav1 {
  display: none;
}
#nav2 {
  display: none;
}
h1 {
  display: none;
}
.entry {
  padding: 5px;
}
```

css collection web sites without tables	file///C/Datafiles/Docs/Freelance/Other_Gigs/OReilly/CSS_Cookbook/css
C55 Co	Usetion
The List - 1879	) entries
artViper	
http://www.artvip	er.net/
	permalink
DMNews.com	
http://www.dmne	ws.com/
	permalink
Spyro Interactive	
http://www.spyroi	interactive.com/
	permalink
Arrow Loans	
http://www.arrow	loans.co.uk/
	permalink
CSS Remix	
CSS Remix: CSS-B	ased Website Gallery
	permalink
Lebensklang	
http://www.leben	sklang.at/
	permalink
Film and Music En	ntertainment Ltd
http://www.fame.	uk.com/
	permalink
nom blog.v2.oran	ige
http://nom.obbik.c	
	permalink
Gross Design Stu	dio
http://www.aross	desian.it/
	permalink
Arcsin	
http://arcsin.se/	
	permalink
Previous <u>Next</u>	
1 of 1	6/5/2006 9:33 AM

Рис. 2.31. После создания файла print.css и связывания с ним Web-страницы выводится версия, более удобная для печати

На странице удален sub\_banner с подзаголовком и скрыты две навигационные колонки. Нет необходимости выводить элемент h1 и его удаление сэкономит место в верхней части листа. У выводимых строк светло-серые поля вывода требуют больших затрат чернил (порошка), поэтому их вывод упрощен: сохранены только отступы между ними.

Не забудьте вставить элемент link в HTML-код страницы:

```
<link rel="stylesheet" type="text/css" href="/css/print.css" media="print" />
<link rel="stylesheet" type="text/css" href="/css/screen.css" media="screen" />
```

Вот и все, что нужно. Технология CSS упрощает многие вещи, включая подготовку страницы для вывода на различные устройства. В табл. 2.5 перечислены современные типы устройств вывода информации, или описатели носителей (media types), которые включены в спецификацию CSS 2.1.

Тип носителя	Устройства
all	Применяется для любых устройств
aural	Применяется для звуковых синтезаторов и синтезаторов речи
braille	Применяется для устройств чтения азбуки Брайля (устройств для слепых) с тактильной обратной связью
embossed	Применяется для устройств печати азбуки Брайля
handneld	Применяется для переносных или портативных устройств, таких как карманные компьютеры или смартфоны
print	Применяется для принтеров и предварительного просмотра
projection	Применяется для вывода презентаций с помощью проекторов
screen	Применяется для цветных мониторов
speech	Применяется для синтезаторов речи
tty	Применяется для устройств с моноширинными шрифтами и сеткой символов фиксированной высоты, таких как телетайпы, терминалы и портативные устройства с ограниченными возможностями символьного вывода
tv	Применяется для телевизоров и WebTV

### Таблица 2.5. Перечень типов носителей информации

### См. также

Главу 10 о настройке стилей при печати; раздел типов устройств вывода в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/media.html; статью "ALA's New Print Styles" (Новые стили для печати от ALA) на http://www.alistapart.com/ articles/alaprintstyles и статью "Pocket-Size Design: Taking Your Website to the Small Svreen" (Проектирование карманного размера: перенос вашего Web-сайта на маленький экран) на http://www.alistapart.com/articles/pocket.

# 2.17. Вставка комментариев в таблицы CSS

## Проблема

Вы хотите добавить в таблицу CSS комментарии для отслеживания вносимой информации.

## Решение

Вставляйте знаки /\* и \*/ в любое место описания стилей для обозначения начала и конца комментариев.

```
/* Это комментарий */
a {
text-decoration: none;
}
```

```
/* Это тоже комментарий */
hl, h2 {
font-size: 100%; /* И это комментарий */
color: #666666;
}
```

## Обсуждение

Вы можете посмотреть на старый код и не вспомнить, почему вы предприняли определенные шаги при его разработке. Комментарии могут пояснить и организовать код, помогая лучше понять его при просмотре спустя какое-то время. Комментарии также помогают понять назначение кода тому, кто не является его автором. Webобозреватели игнорируют содержимое, заключенное между знаками /\* и \*/.

Если вы делите код на секции, комментарии полезны для присвоения имен секциям, например, заголовок, примечание, основная навигация, дополнительная навигация и т. д. Комментарии очень помогают тестировать ваши Web-страницы. Если вы не уверены в правиле стилевого оформления или не знаете, как оно повлияет на страницу, поместите стиль в комментарий, чтобы отключить его.

```
/*
a {
text-decoration: none;
}
*/
```

Стилевое правило для text-decoration не будет действовать, т. к. комментарий выводит его из обращения. Если нет других стилей для элемента а, ссылки будут подчеркиваться до тех пор, пока вы не удалите знаки комментария.

### См. также

Комментарии в спецификации CSS 2.1 на странице http://www.w3.org/TR/ CSS21/syndata.html#comments.

## 2.18. Организация информации в таблице стилей

### Проблема

Вы хотите узнать, как организовать информацию в таблице стилей, чтобы легче было управлять ею.

## Решение

Можно управлять таблицей CSS, группируя вместе общие визуальные элементы страницы. В приведенном далее перечне показан вариант упорядочивания элементов, сгруппированных в таблице стилей.

- 1. Элементы (h1—h6, p, a, list, links, images).
- 2. Типографика.

- 3. Макет страницы (заголовок, информационное содержимое, навигация, глобальная навигация, дополнительная навигация, боковая полоса, подвал (footer)).
- 4. Теги формы (form, fieldset, label, legend).
- 5. Контент (статьи, события, новости).

Далее показаны комментарии из трех таблиц стилей, в которых стили сгруппированы по-разному:

```
/* Типографика и цвета
-----*/
[css code ]
/* Структура
-----* /
[css code ]
/* Заголовки
----*/
[css code ]
/* Изображения
-----*/
[css code ]
/* Списки
-----*/
[css code ]
/* Элементы формы
-----*/
[css code ]
/* Комментарии
-----*/
[css code ]
/* Боковая полоса
-----*/
[css code ]
/* Общие элементы
-----*/
[css code ]
```

## Обсуждение

То, что хорошо для одного человека, может быть плохо для другого. Предложенный в решении набор — это рекомендация, основанная на сочетании практического опыта и лучших методов, больше всего подходящая для Web-сайтов малого и среднего размеров.

Для разных проектов вы можете подобрать способ упорядочивания, который вам больше всего подходит и учитывает ваши личные предпочтения. Посетите любимые Web-сайты и просмотрите их таблицы стилей для того, чтобы изучить способы их организации.

## См. также

Статью "CSS organization Tip 1: flags" (Организация CSS, совет 1: флаги) Дуга Боумана (Doug Bowman) о методе отыскания правил в ваших файлах CSS на странице http://www.stopdesign.com/log/2005/05/03/css-tip-flags.html.

# 2.19. Работа со свойствами с сокращенной формой записи

### Проблема

Вы хотите использовать в таблицах стилей свойства с сокращенной формой записи.

### Решение

Начните с корректно размеченного фрагмента.

```
<h3>Свойство с сокращенной формой записи</h3>
<бъединяйте свойства с помощью сокращенной формы записи и экономьте время,</p>
затраты на набор и несколько байтов в размере файла. Ваши таблицы стилей также
будет легче читать.
```

Затем используйте один экземпляр свойства font вместо трех: font-style, font-size и font-family:

```
h3 {
 font: italic 18pt verdana, arial, sans-serif;
}
p {
 border: 2pt solid black;
}
```

## Обсуждение

От нескольких свойств CSS можно отказаться в пользу свойств с сокращенной формой записи.

Свойство border — свойство с сокращенной формой записи, которое объединяет три свойства:

- ♦ border-color
- ♦ border-width
- ♦ border-style

Свойство font — свойство с сокращенной формой записи, объединяющее пять свойств. Свойство font включает в себя значения из следующих свойств:

- ♦ font-style
- font-size/line-height
- ♦ font-family
- ♦ font-weight
- ♦ font-variant

Вводите значения так же, как вы вводили бы их для любого другого свойства за исключением font-family и font-size/line-height. Для свойства font-family вводите названия шрифтов в порядке желательного приоритета, отделяя одно от другого запятой. Если вы используете оба свойства, и font-size, и line-height, отделите их значения знаком прямого слэша (/):

```
h3 {
  font: italic 18pt/20pt verdana, arial, sans-serif
}
```

В табл. 2.6 приведена сводка свойств с сокращенной формой записи, доступных Webразработчикам.

Свойство	Значения	Пример		
background	background-color	<pre>background: url(book.gif) #999 no-repeat top;</pre>		
	background-image			
	background-repeat			
	background-attachment			
	background-position			
border	border-width	border: thin solid #000		
border-left	border-style			
border-right	border-color			
border-top				
border-bottom				
font	font-style	font: 14px italic Verdana, Arial,		
	font-variant	sans-serif;		
	font-weight			
	font-size/line-height			
	font-family			
	caption			
	icon			
	menu			
	message-box			
	small-caption			
	Status-bar			
list-style	list-style-type	list-style: circle inside;		
	list-style-position			
	list-style-image			
margin	margin-top	margin: 5px 0px 5px 10px;		
	margin-right	margin: 15px 0;		
	margin-bottom	margin: 5px;		
	margin-left			
padding	padding-top	padding: 5px 10% 15px 5%;		
	padding-right	padding: 7px 13px;		
	padding-bottom	padding: 6px;		
	padding-left			

Таблица 2.6. Свойства с сокращенной формой записи

## См. также

Свойство border с сокращенной формой записи в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/box.html#border-shorthand-properties и свойство font с сокращенной формой записи на http://www.w3.org/TR/CSS21/about.html#shorthand, а также полный перечень свойств CSS в *приложении 2*.

# 2.20. Задание альтернативной таблицы стилей

## Проблема

Вы хотите предоставить другие варианты стилей пользователям, которым может потребоваться более крупный текст или другая цветовая схема.

## Решение

Используйте элемент link с атрибутом title и свяжите его с альтернативными таблицами стилей. Атрибут title позволяет пользователю при просмотре списка доступных стилей видеть, какие варианты имеются в наличии. Для отображения списка в Webобозревателе Firefox выберите кнопкой мыши последовательность команд View | Page Styles (Вид | Стили страницы).

```
<link href="default.css" rel="stylesheet" title="default styles"
type="text/css" media="screen" />
<link href="green.css" rel="stylesheet" title="green style"
type="text/css" media="screen" />
<link href="blue.css" rel="stylesheet" title="blue style"
type="text/css" media="screen" />
```

К сожалению, этот способ не работает в Web-обозревателях Internet Explorer 6.0 или Safari.

## Обсуждение

Альтернативные таблицы стилей действуют аналогично таблицам стилей для разных устройств вывода, описанным в *разд. 2.16*. Вместо создания стилей для отдельных устройств вывода вы предоставляете пользователям несколько вариантов стилевого оформления экрана. Более того, этот метод не требует применения языка JavaScript. У некоторых пользователей JavaScript отключен, что, безусловно, повлияет на динамическое изменение таблиц стилей (style sheet switcher).

Все, что от вас требуется — сделать копию стандартной таблицы стилей и переименовать ее. Внесите изменения в таблицу стилей и вставьте элемент link с атрибутом title (рис. 2.32).



Рис. 2.32. Смена таблиц стилей в меню Web-обозревателя

### См. также

Статью "Invasion of Body Switchers" (Внедрение переключателей стилей) Энди Кларке (Andy Clarke) и Джеймса Эдвардса (James Edwards) на сайте http:// www.alistapart.com/articles/bodyswitchers, в которой показаны способы создания на языке JavaScript переключателя таблиц стилей; CSS Style Switcher tutorial (Пособие по переключателям стилей CSS) Амита Гасте (Amit Ghaste) на его странице http://www.ghaste.com/pubs/styleswitcher.html.

# 2.21. Применение плавающей модели

### Проблема

Вы хотите разместить на странице слева или справа изображения так, чтобы текст обтекал их, а не располагался над или под изображением (рис. 2.33).

### Решение

Сначала создайте для изображений селекторы класса:

```
.leftFloat {
  float: left
}
.rightFloat {
  float: right
}
```

### Предупреждение

Использовать для классов имена, описывающие способ отображения, как я сделал в данном примере, не рекомендуется. Это сделано только для большей наглядности.

Затем добавьте селектор класса в разметку страницы (рис. 2.34):

```
<img src="csscookbook.gif" class="leftFloat" alt="cover" />
This is the book cover for the <em>CSS Cookbook</em>.
<img src="csscookbook.gif" class="rightFloat" alt="cover" />
This is the book cover for the <em>CSS Cookbook</em>.
```


Рис. 2.33. По умолчанию текст не обтекает изображения

Рис. 2.34. Благодаря применению свойства float текст обтекает изображения

### Обсуждение

До того как соблюдение стандартов стало настоятельной рекомендацией, дизайнеры пользовались атрибутом align элемента img для расположения изображения сбоку и обеспечения обтекания его текстом. Консорциум W3C выступил против применения атрибута align и теперь рекомендует использовать вместо него свойство float.

Для сдвига объекта влево или вправо от его исходного положения свойство float может применяться не только к изображениям.

Второе изображение может наложиться на абзац, связанный с первым изображением. Это сбивает с толку и требует корректировки. Для исправления данного недостатка используйте свойство clear:

```
p {
  clear: left;
}
```

Свойство clear заставляет абзац выводиться после вывода потока с изображением. Во втором элементе img свойство clear вместо выравнивания со вторым элементом р сдвигает изображение на одну строку ниже предыдущего изображения (рис. 2.34).

### См. также

Свойства float в спецификации CSS 2.1 консорциума W3C на странице http://www.w3.org/TR/CSS21/visuren.html#floats; *главу* 8, содержащую примеры использования float с колонками страницы; посвященный плавающим моделям раздел CSS/edge Web-сайта Эрика Мейера (Eric Meyer) по адресу http://meyerweb.com/ eric/css/edge/.

# 2.22. Применение автовыравнивания плавающих элементов

### Проблема

Вы хотите, чтобы плавающий элемент не накладывался на остальной контент независимо от наличия других HTML-элементов.

### Решение

Сначала рассмотрим вариант, в котором элемент float накладывается на часть макета, как показано на рис. 2.35.

```
<div>
<img src="schmitt-csscookbook.jpg" alt="cover" />
Lorem ipsum dolor sit amet, consectetuer adipiscing elit,
sed diam nonummy nibh euismod tincidunt ut laoreet dolore
magna aliquam erat volutpat...
</div>
```

Затем зададим правила CSS, например, такие:

```
div {
  border: 1px solid black;
  padding: 25px;
}
img {
  border-right: 1px solid #999;
  border-bottom: 1px solid #999;
  float: left;
  padding: 1px;
}
p {
  float: right;
  width: 87%;
}
```



Рис. 2.35. Изображение и абзац накладываются на границу элемента

Для того чтобы заставить границу элемента div охватывать плавающие объекты, используйте метод вертикального автовыравнивания плавающих объектов (self-clearing float).

Сначала задайте правила CSS:

```
.clearfix:after {
  content: ".";
  display: block;
  height: 0;
  clear: both;
  visibility: hidden;
}
/* CSS rule for IE6 */
* html .clearfix {
  height: 1%;
  }
/* CSS rule for IE7 */
*:first-child+html .clearfix {
  min-height: 1px;
}
```

110

Затем для родительского элемента div добавьте селектор класса со значением clearfix (рис. 2.36):

```
<div class="clearfix">
<img src="schmitt-csscookbook.jpg" alt="cover" />
Lorem ipsum dolor sit amet, consectetuer adipiscing elit,
sed diam nonummy nibh euismod tincidunt ut laoreet dolore
magna aliquam erat volutpat...
</div>
```



Рис. 2.36. Выровненные плавающие элементы

### Обсуждение

Метод вертикального выравнивания, обсуждавшийся в *разд. 2.21*, учитывает присутствие дополнительного элемента, следующего сразу после плавающего объекта.

Другой метод, применяемый Web-разработчиками, состоит во вставке в разметку после плавающего объекта элемента div или br и последующем задании свойства clear этого элемента:

```
<div>
<img src="schmitt-csscookbook.jpg" alt="cover" />
Lorem ipsum dolor sit amet, consectetuer adipiscing elit,
sed diam nonummy nibh euismod tincidunt ut laoreet dolore
magna aliquam erat volutpat...
<div style="clear: both;"></div>
</div>
```

Если к Web-документу или документам прикладывают руки многие, практически невозможно быть уверенным в том, что такие подпорки будут вставляться всеми единообразно и согласованно.

#### Автовыравнивающиеся плавающие объекты

Метод вертикального автовыравнивания плавающих объектов, впервые опубликованный на Web-caйте Position is Everything (http://positioniseverything.net/easyclearing.html), предлагал способ выравнивания плавающих элементов без дополнительной разметки. Ho Internet Explorer 7 и более ранние версии не могут формировать автогенерируемый контент с помощью псевдоэлементов :after.

Для того чтобы обойти ограничения Web-обозревателей и обманным путем заставить их выравнивать плавающие элементы, понадобятся два правила CSS — одно для IE7 и другое для IE6.

#### Примечание

Вы можете запрятать эти правила CSS с помощью условных комментариев, так что их будут видеть только обозреватели IE.

#### Применение overflow

Другой метод вертикального выравнивания плавающих объектов — применение редко используемого свойства CSS overflow:

```
div {
  border: 1px solid black;
  padding: 25px;
  overflow: hidden;
  zoom: 1
}
```

Благодаря свойству overflow элемент выравнивает все плавающие объекты, находящиеся внутри него. (Свойство zoom предназначено для IE, если потребуется. Если нет, вы можете исключить его.)

### См. также

*Разд. 2.21* с информацией об использовании плавающих объектов; другие способы выравнивания по вертикали плавающих элементов на сайте http://www.sitepoint.com/blogs/2005/02/26/simple-clearing-of-floats/.

# 2.23. Применение абсолютного позиционирования

### Проблема

Вы хотите задать положение элемента относительно окна, а не с помощью его позиции по умолчанию.

### Решение

В таблице стилей используйте свойство position со значением absolute. Для указания позиции элемента также воспользуйтесь свойством bottom, left или обоими:

```
.absolute {
  position: absolute;
  bottom: 100px;
  left: 50px;
}
```

### Обсуждение

Проектирование с помощью абсолютного позиционирования выводит контент из *нор-мального потока макета страницы* и помещает его в пределах текущего блока или окна точно туда, куда указывают свойства CSS. Пример кода, используемого в реализации примера, предписывает Web-обозревателю расположить элемент с классом absolute точно на 100 пикселов вниз от верхнего края окна просмотра и на 50 пикселов вправо от его левого края.

Посмотрите на стандартный поток вывода изображения и абзаца, показанный на рис. 2.37.



Рис. 2.37. Стандартное отображение контента

Теперь примените абсолютное позиционирование к элементу div, охватывающему контент, вставив атрибут class и значение absolute (рис. 2.38):

```
<div class="absolute">
<img src="csscookbook.gif" alt="cover" />
Lorem ipsum dolor sit amet, consectetuer adipiscing elit,
sed diam nonummy nibh euismod tincidunt ut laoreet dolore
magna aliquam erat volutpat...
</div>
```

Для изменения абсолютной позиции также можно использовать свойства right и bottom. Свойство bottom означает нижний край окна просмотра независимо от того, какого размера окно вы сделали.

#### Предупреждение

В данном случае абсолютное позиционирование элементов использовалось для сдвига блока контента во всех направлениях только для демонстрации его принципа действия. На

практике его применение требует осторожности, поскольку элементы с абсолютной позицией будут оставаться на месте, даже когда "резиновая" верстка (flexible layout) Webстраниц изменится в результате возможности гибкой перенастройки Web-обозревателей и/или изменения размера шрифта.



Рис. 2.38. Абсолютное позиционирование помещает элемент в соответствии с его местоположением в окне

### См. также

Абсолютное позиционирование в спецификации CSS 2.1 консорциума W3C на странице http://www.w3.org/TR/CSS21/visuren.html#absolute-positioning и учебное пособие по позиционированию W3Schools по адресу http://www.w3schools.com/css/css\_ positioning.asp.

# 2.24. Применение относительного позиционирования

### Проблема

Вы хотите располагать контент, основываясь на его местоположении в документе. Другими словами, при визуализации Web-обозревателем потока макета страницы положение элемента изменяется относительно его стандартной позиции.

### Решение

В таблице стилей используйте свойство position со значением relative. Добавьте также свойство top, left или оба для указания нового положения элемента.

}

С помощью следующего правила CSS для изображения картинка может перемещаться по тексту абзаца, как показано на рис. 2.39:

```
.relative {
position: relative;
top: 100px;
left: 20px;
          000
                                                     CSS Cookbook
                                                                                                          0
           (>
                           C
                                  (X)
                                        ( 1
                                               (1P)
                                                                 ☐ (file:///User ☆▼) = (G * petter font sQ
           Back
                 Forward
                           Reload
                                  Stop
                                        Home
                                               Print
                                                      1Password
                    CSS Cookbook
                                              Technology Books, Tech Con.
                                                                               Christopher Schmitt : Design
          Lorem incum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet
          dol 🐗
                    na aliquam erat volutpat ...
```

Рис. 2.39. Относительное позиционирование располагает элемент, основываясь на его позиции в нормальном потоке документа

### Обсуждение

В отличие от абсолютного позиционирования отображение примера начинается не с верхнего левого края окна, а с того места, где располагался бы элемент р, если бы он был единственным. Код предписывает Web-обозревателю сместить абзац на 100 пикселов вниз от верхней границы и на 20 пикселов вправо от левой границы исходной позиции абзаца, а не от границы окна.

В случае абсолютного позиционирования контент смещается от границ текущего блока точно в то место, которое задают свойства.

### См. также

Относительное позиционирование в спецификации CSS 2.1 консорциума W3C на странице http://www.w3.org/TR/CSS21/visuren.html#relative-positioning и учебное пособие по позиционированию от W3Schools на странице http://www.w3schools.com/ css/css\_positionin.asp.

# 2.25. Применение сцепленного позиционирования

### Проблема

Вы хотите смещать элемент с учетом ограничений, налагаемых размерами другого элемента. Например, вам нужно поместить изображение обложки книги в темное поле, а не в левый верхний угол окна просмотра Web-обозревателя, как показано на рис. 2.40.



Рис. 2.40. Изображение с абсолютным позиционированием, помещенное в левый верхний угол окна просмотра Web-обозревателя

### Решение

Сначала в родительском элементе задайте значение relative в свойстве position:

```
#content {
  position: relative;
  width: 200px;
  height: 200px;
  margin: 10% auto;
  background: #2942c4;
}
```

Затем задайте дочерний элемент с абсолютным позиционированием, используя свойства смещения top, right, bottom и left для сдвига этого элемента в границах родительского элемента (рис. 2.41):

```
#positioned {
  position: absolute;
  top: 20px;
  left: 20px;
}
```



Рис. 2.41. Изображение, связанное с размерами непосредственного родительского элемента

### Обсуждение

Если у элемента абсолютное позиционирование, он изымается из нормального потока и его размещение зависит от элемента, который его содержит (родителя). В большинстве случаев оказывается, что это основной или корневой элемент Web-документа, как правило, элемент html.

Но контекст этого элемента-контейнера может изменяться.

Если родительский элемент тоже позиционирован, элемент с абсолютным позиционированием оказывается отсоединенным от корневого элемента (обычно левого верхнего угла окна просмотра, если не заданы свойства смещения). Описанный эффект называется изменением контекста родительского элемента. Я называю это сцеплением, потому что так короче.

#### См. также

Статью Дуга Боумана (Doug Bowman) "Making the absolute, relative" (Превращение абсолютного в относительное) на Web-странице http://stopdesign.com/archive/ 2003/09/03/absolute.html.

# 2.26. Создание стека элементов с помощью *z-index*

### Проблема

На вашей странице есть элемент, накладывающийся на другой элемент и мешающий его просмотру.

### Решение

Используйте свойство z-index в сочетании со свойством position, имеющим одно из значений: absolute, relative или fixed:

```
div.image {
  position: relative;
  z-index: 20;
  width: 13px;
  height: 14px;
  background-image: url(star.gif);
  background-repeat: no-repeat;
}
```

### Обсуждение

Цифровые изображения формируются из слоев. Верхний слой закрывает все, что находится на нижних слоях. Эта аналогия подходит для свойства z-index. Элемент с бо́льшим значением z-index закрывает элемент с меньшим значением.

#### Примечание

Свойство z-index действует, если элемент позиционирован со значением absolute, relative или fixed. Без подходящего свойства position z-index не используется.

Если вы применяете несколько элементов со свойством z-index, старайтесь использовать значения кратные 10 (т. е. 10, 20, 30), а не 1, 2, 3 и т. д. Такой подход позволит включать другие, незапланированные элементы, не нарушая порядка наложения элементов, и не приведет к необходимости переопределять их значения.

### См. также

Описание свойства z-index в спецификации CSS2 на странице http://www.w3.org/ TR/CSS2/visuren.html#z-index.

# 2.27. Проверка правил CSS

### Проблема

Вы хотите убедиться в том, что ваши правила CSS не испорчены ошибками и опечатками.

### Решение

Перейдите на Web-страницу http://jigsaw.w3.org/css-validator/, показанную на рис. 2.42, и введите URI страницы, которую хотите проверить.

Кроме того, вы можете ввести код для тестирования двумя способами: загрузив файл CSS или введя правила CSS с клавиатуры.



Рис. 2.42. Ввод Web-адреса для проверки корректности правил CSS

### Обсуждение

Проверка CSS отличается от проверки HTML-кода тем, что вам не нужно объявлять применяемый ростуре документа.

Несмотря на то, что у многих коммерческих программных продуктов есть встроенные валидаторы (например, у Adobe Dreamweaver), W3C CSS Validator поддерживается в актуальном состоянии и лучше комментирует результаты, особенно если используется спецификация CSS3.

#### Примечание

Если в таблице стилей применяются правила CSS3, обязательно задайте в раскрывающемся списке выбора профиля **CSS level 3**. В момент написания книги по умолчанию правила CSS проверялись только на соответствие спецификации CSS 2.1.

#### Создание букмарклета для валидатора CSS

С помощью букмарклета отправляйте любую страницу, которую вы посетите в Webпространстве, непосредственно в W3C CSS Validator. Букмарклет — это небольшая программка на языке JavaScript, спрятанная в поле закладки Address.

Создайте новую закладку, назовите ее CSS Validator и затем замените любое встретившееся в адресном поле содержимое следующей строкой:

```
javascript:void(document.location='http://jigsaw.w3.org/
css-valitor/validator?profile=css21&usermedium=all
&warning=1&lang=en&uri='+escape(document.location))
```

При посещении вами какого-либо Web-сайта щелчок кнопкой мыши по букмарклету выполнит проверку загруженной в Web-обозреватель страницы валидатором CSS.

### См. также

Pacширение Firefox (https://addons.mozilla.org/en-US/firefox/addon/2289), предназначенное для передачи просматриваемой в данный момент страницы в W3C CSS Validator, открывающийся на новой закладке Web-обозревателя.

# глава з Web-типографика

# 3.0. Введение

До появления технологии CSS Web-разработчики применяли теги font для установки цвета, размера и стиля текста в разных частях Web-страницы:

<font face="Verdana, Arial, sans-serif" size="+1" color="blue">
Hello, World!
</font>

Несмотря на то, что для изменения отображения шрифта этот метод был эффективен, средства были ограничены.

Использование многочисленных тегов font на многих и многих страницах в результате приводило к длительным исправлениям, увеличению размера файла Web-документа и росту вероятности появления ошибок в разметке. Технология CSS помогает устранить эти проблемы проектирования и сопровождения.

Сначала введите в элемент р контент:

```
Hello, World!
```

Затем задайте стили в теге документа head для определения внешнего вида абзаца:

```
<style type="text/css" media="all">
p {
    color: blue;
    font-size: small;
    font-family: Verdana, Arial, sans-serif;
    }
</style>
```

Теперь благодаря этой технологии структура абзаца и его визуальное представление разделены. А это разделение чрезвычайно упрощает процесс редактирования и сопровождения дизайна Web-сайта, включая его типографику. Изменения стилей могут выполняться в таблице стилей без необходимости внесения изменений на уровне Web-содержимого.

По сравнению с предшествующими технологиями, Web-разработчикам стало не только легче редактировать типографику страниц, но и управлять ею. Помимо обсуждения способов задания цвета, стиля и размера шрифтов в этой главе также пойдет речь о способах создания буквиц, визуально привлекательных врезок (pull quote), изменении интерлиньяжа (leading) и о многом другом.

# 3.1. Задание шрифтов

### Проблема

Вы хотите задать гарнитуру шрифта на Web-странице.

### Решение

```
Воспользуйтесь свойством font-family:
body {
font-family: Georgia, Times, "Times New Roman", serif;
}
```

### Обсуждение

Задать шрифты, с помощью которых Web-обозреватель будет визуализировать Webстраницу, можно написав разделенный запятыми список значений для свойства fontfamily. Если Web-обозреватель не найдет шрифт, указанный в списке первым, он попытается найти следующий шрифт и т. д. до тех пор, пока не найдет шрифт.

Если название шрифта содержит пробелы, заключите его в апострофы или двойные кавычки.

В конец списка вариантов шрифтов для выбора следует включить обобщенное название семейства шрифтов (generic font family). В технологии CSS предлагается пять приведенных в табл. 3.1 значений для семейств шрифтов.

Обобщенные названия семейств шрифтов	Примеры шрифтов		
serif	Georgia, Times, "Times New Roman", Garamond, "Century Schoolbook"		
sans-serif	Verdana, Arial, Helvetica, Trebuchet, Tahoma		
monospace	Courier, "MS Courier New", Prestige		
cursive	"Lucida Handwriting", "Zapf-Chancery"		
fantasy	Comic Sans, Whimsy, Critter и Cottonwood		

Таблица 3.1. Семейства шрифтов в технологии CSS

У всех Web-обозревателей есть список шрифтов, которые принадлежат пяти семействам, перечисленным в табл. 3.1. Если шрифт не выбран правилом CSS или недоступен на компьютере пользователя, Web-обозреватель применяет шрифт одного из этих семейств.

#### Проблема поиска шрифтов

Больше всего проблем возникает у семейства fantasy, т. к. в него попадает любой шрифт, не принадлежащий четырем другим семействам. Разработчики редко пользуются этим шрифтом, поскольку они не представляют, какие символы будут выводиться на экран!

Другое создающее проблемы значение — cursive, потому что некоторые системы не могут отображать курсивный (рукописный) шрифт. Если Web-обозреватель не может применить курсивный шрифт, он использует для его замены другой стандартный шрифт. Поскольку текст, помеченный как cursive, в действительности может выводиться не курсивным шрифтом, разработчики часто также избегают применения этого обобщенного названия.

Если вы хотите использовать необычный шрифт, который может быть не установлен на большинстве пользовательских компьютеров, эмпирическое правило рекомендует выбрать в качестве последнего значения в свойстве font-family: serif, sans-serif или monospace. Это обеспечит пользователю, по крайней мере, возможность прочесть Webдокумент.

#### Наследование шрифтов на всей Web-странице

Не нужно устанавливать те же самые свойства для каждого используемого вами тега. Дочерний элемент *наследует* или имеет те же значения свойств, что и его родительский элемент, если спецификация CSS, определяющая данное свойство, может быть унаследована. Например, если вы задаете свойство font-family для отображения шрифтом serif абзаца, который содержит элемент ет как дочерний, для текста элемента ет также устанавливается шрифт serif:

```
 The water fountain
with the broken sign on it<em>indeed</em> broken.
```

Наследования не происходит в двух случаях.

Первый — неотъемлемая часть спецификации CSS и относится к элементам, способным формировать блок. Такие элементы, как h2 и p, называются элементами блочного уровня (block-level element) и могут иметь другие свойства, например, поля, рамки, отступы и фон, как видно на рис. 3.1.



Рис. 3.1. Блочная модель элемента блочного уровня

Поскольку эти свойства не передаются дочернему элементу, вы не должны писать дополнительные правила для того, чтобы скомпенсировать визуальные эффекты, которые возникли бы при их наследовании. Например, если бы вы применили поле margin 15 % к элементу body, это правило пришлось бы применять к каждому элементу h2 и p, являющихся дочерними для данного элемента body. Если бы эти свойства наследовались, страница выглядела бы так, как показано на рис. 3.2.



Рис. 3.2. Гипотетический макет при наследовании свойств margin и border

Поскольку одни свойства определены как наследуемые, а другие — нет, в действительности страница выглядит в CSS-совместимом Web-обозревателе так, как показано на рис. 3.3.



Рис. 3.3. Вид страницы при отсутствии наследования определенных свойств блочного элемента

Второй случай, при котором не происходит наследования, — это, конечно же, отсутствие в Web-обозревателе поддержки спецификации CSS. К счастью, такого не происходит в последних версиях любых современных Web-обозревателей, а самым ярким примером мог бы быть старый Netscape Navigator 4.

### См. также

Спецификацию CSS 2.1 о наследовании на странице http://www.w3.org/TR/CSS21/ cascade.html#inheritance; спецификацию CSS 2.1 о значениях свойства font-family на http://www.w3.org/TR/CSS21/fonts.html#propdef-font-family; публикации о технологии CSS и проблемах Netscape 4 на странице http://www.mako4css.com/cssfont.htm.

# 3.2. Применение Web-безопасных шрифтов

### Проблема

Вы хотите задать шрифты, которые есть на машинах большинства посетителей вашего Web-сайта.

### Решение

Используйте так называемые Web-безопасные шрифты, представляющие собой файлы шрифтов, входящие в состав операционных систем Macintosh и Windows.

#### Примечание

Если вы используете Linux, можно инсталлировать шрифты Microsoft TrueType, установив пакет msttcorefonts. Дополнительную информацию ищите по адресу http://embraceubuntu.com/ 2005/09/09/installing-microsoft-fonts/.

Далее приведены примеры списков Web-безопасных шрифтов семейства sans-serif:

```
font-family: Verdana, Geneva, sans-serif;
font-family: Arial, Helvetica, sans-serif;
font-family: Tahoma, Geneva, sans-serif;
font-family: "Trebuchet MS", Area, Helvetica, sans-serif;
font-family: "Lucida Sans Unicode", "Lucida Grande", sans-serif;
```

Далее приведены примеры списков Web-безопасных шрифтов семейства serif:

```
font-family: Georgia, "Times New Roman", Times, serif;
font-family: "Palatino Linotype", "Book Antigua", Palatino, serif;
font-family: "MS Serif", New York, serif;
```

#### К моноширинным Web-безопасным шрифтам относятся следующие списки:

font-family: "Courier New", Courier, monospace; font-family: "Lucida Console", Monaco, monospace;

#### К курсивным Web-безопасным шрифтам относится следующий список:

```
font-family: "Comic Sans MS", cursive;
```

### Обсуждение

В ОС Windows и Macintosh вы можете найти примерно 13 шрифтов, перечисленных в табл. 3.2.

Arial Sans-serif ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789 ABCDEFGHIJKLMNOPQRSTUVWXYZ
Atial Black Constant
Anai Black Sans-seni abcoergnijkimnopqrstuvwxyz
0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ
Comic Sans MS Cursive abcdefghijklmnopqrstuvwxyz
0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ
Courier New Monospace abcdefghijklmnopqrstuvwxyz
Georgia Serif abcdefshijklmppgrstuwwyz
0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ
Helvetica Sans-serif abcdefghijklmnopqrstuvwxyz
0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ
Impact Sans-serif abcdefghijkimnopgrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
ABCDEEGHIKLMNOPORSTUVWXYZ
Times Serif abcdefghijklmnopqrstuvwxyz
0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ
Times new Roman Serif abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQKSTUVWXTZ
0123456789
ABCDEFGHIJKLMNOPORSTUVWXYZ
Verdana Sans-serif abcdefghijklmnopqrstuvwxyz
0123456789
₫ঽ᠔᠙☜☞d₽७©©⊗€ <sup>≈</sup> ≳Ю础→♠ቈ₽₽♥∞≠€
Wingdings (symbol) ତିଣ୍ଲୁ≏ଲୁନ୍ନୁର୍କ୍କୁର୍କୁର

Таблица 3.2. Кроссплатформенные шрифты

#### Примечание

Шрифты Courier, Helvetica и Times устанавливаются на большинстве систем X11 UNIX/Linux. Остальные шрифты, представленные в табл. 3.2 как Web-безопасные для OC Windows и Mac OS X, обычно не устанавливаются.

#### Расширение списка Web-безопасных шрифтов

Популярные высокопроизводительные программные приложения Microsoft Office и Apple iWork устанавливают дополнительные файлы шрифтов. Если предположить, что у большого числа компьютерных пользователей есть одно из них (в зависимости от установленной операционной системы), список Web-безопасных шрифтов можно расширить.

Web-дизайнер Джейсон Кренфорд Тиге (Jason Cranford Teague) именно это и сделал. Исследовав перечни шрифтов для программных приложений, он составил каталог с расширенным списком Web-безопасных шрифтов, который можно отсортировать по их именам, насыщенности, ОС или рангу (вероятности их установки на компьютере пользователя) (рис. 3.4). Просмотреть список можно по адресу http://tr.im/xCGi.

O O Web-safe Fonts   Web Typography   Speaking In Styles						
Image: State Forward     Cd     X     Image: State Forward     IP       Back Forward     Reload     State Forward     Print     IPassword						
🛇 Web-safe Fonts   Web Typogra 🥹 👔 Christopher Schmitt : Designer, 🥸 🕼 Technology Books, Tech Confer 😔 🕇						
Web-safe Fonst-2						
Font Name	Weight and Style	OS Rank	Sample			
1 Academy Engraved LET		3-Likely	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789			
2 Agency FB	bold	4-Less Likely	ABCDEFGHIJKLMNDPORSTUVWXYZ abcdafghijkhmopgrstuvwxyz D123456789			
3 Algerian		4-Less Likely	ABCDEFGHIJKLMNOPQRSTUVWXYZ ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789			
4 American Typewriter	bold	2-Almost Certain	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789			
5 Andale Mono		1-Certain	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz 0123456789			
			ABCDEFGHIJKLMNOPQRSTUVWXYZ			
< > (	*********	*********				

Рис. 3.4. Каталог с расширенным списком Web-безопасных шрифтов

#### Более надежное комплектование

Несмотря на то, что просто указание шрифтов, имеющихся на пользовательских машинах, — хорошее решение для межплатформенной разработки, свойство font-family позволяет дизайнерам выбирать шрифты, не входящие в базовый комплект. Не ограничивайте Web-проекты несколькими семействами шрифтов.

Например, шрифт Gill Sans — отличный шрифт семейства sans-serif, но он обычно не устанавливается на компьютерах. Для того чтобы задать список шрифтов, учитывающий пожелание использовать в Web-проектах шрифт Gill Sans, но при этом указать альтернативы, примените следующий код CSS:

```
p {
  font-family: "Gill Sans", Trebuchet, Calibri, sans-serif;
}
```

Идеолог дизайна Натан Форд (Nathan Ford) изучает этот подход и предлагает дополнительные возможные списки шрифтов в статье "Better CSS Font Stacks" (Улучшенные списки шрифтов в CSS) в своем блоге (http://unitinteractive.com/blog/2008/06/ 26/better-css-font-stacks/).

#### См. также

Перечень Web-безопасных шрифтов и примеры их отображения на Web-странице http://www.fonttester.com/web\_safe\_fonts.html.

### 3.3. Задание стилизованного амперсанда

### Проблема

Вы хотите применить в заголовке стилизованный амперсанд вместо амперсанда, предлагаемого стандартным Web-безопасным шрифтом.

#### Решение

Сначала амперсанд в заголовке заключите в элемент span:

```
<hl>The Lorem Ipsum <span class="amp">&amp;</span> Dolor</hl>
```

Затем задайте для селектора класса список шрифтов, чтобы включить шрифты со стилизованным символом амперсанда, например, как показанный на рис. 3.5:

```
span.amp {
  font-family: "Goudy Old Style", "Palatino", "Book Antiqua", serif;
  font-style: italic;
  font-weight: normal;
}
```

### Обсуждение

Для набора амперсанда в тексте Web-страницы используйте специальный HTML-код астр, так в языке HTML кодируются специальные символы, например, знаки "меньше" (<) или "больше" (>), для того чтобы Web-обозреватель не воспринимал их как часть разметки.



Рис. 3.5. Изменение стиля амперсанда на более привлекательный

#### Примечание

Для ввода в текст Web-документа знаков "меньше" и "больше" используйте обозначения alt и agt соответственно.

#### Стилизованные амперсанды

Специалист по типографике Роберт Брингхерст (Robert Bringhurst) в своей книге "The Elements of Typographic Style" (Основы стиля в типографике) (изд. Hartley and Marks)



Рис. 3.6. Каталог шрифтов со стилизованными амперсандами

предлагает применять наилучший амперсанд, имеющийся во время работы с текстом. Он также утверждает, что часто курсивные варианты семейств шрифтов содержат амперсанды лучшей формы, чем шрифты с обычным начертанием или прямые шрифты.

Web-дизайнер Дэн Седерхольм (Dan Cederholm) применил эти рекомендации к Webдизайну и даже исследовал амперсанды (рис. 3.6) в различных семействах шрифтов, найденных в OC Windows и Macintosh (http://simplebits.com/notebook/2008/08/ 14/ampersands.html).

### См. также

Список HTML-кодов для специальных символов на странице http:// htmlhelp.com/reference/html40/entities/latin1.html; учебник Ричарда Раттера (Richard Rutter) "The Elements of Typographic Style Applied to the Web" (Элементы типографики, применяемые в Web) http://webtypography.net.

# 3.4. Внедрение файлов шрифтов

### Проблема

Вы хотите использовать на вашей Web-странице (рис. 3.7) файл шрифта с гарнитурой Museo.



### Решение

Для назначения имени в свойстве font-family используйте правило @font-face:

```
@font-face {
```

```
font-family: "Museo 300";
}
```

Далее свяжите файл шрифта с форматом шрифта:

```
@font-face {
  font-family: "Museo 300";
  font-style: normal;
  font-weight: normal;
  src: url("fonts/Museo300-Regular.otf") format("opentype");
}
```

Затем вставьте значение font-family внедренного шрифта в начало задаваемого списка шрифтов:

```
h2 {
  font-family: "Museo 300", Verdana, Geneva, sans-serif;
  font-weight: normal;
}
```

### Обсуждение

Описание внедрения шрифта стало частью спецификации CSS2 с 1998 г. Internet Explorer для Windows стал поддерживать правило @font-face, начиная с версии 4, но Web-обозреватель IE поддерживает только формат шрифта Embedded OpenType Font (eot), содержащий код DRM (Digital Rights Management, управление цифровыми правами).

Другие открытые форматы файлов для внедрения шрифтов OpenType Face (otf) и TrueType Format (ttf) поддерживаются в Safari 3.1, Opera 10, Firefox 3.5 и более поздних версиях этих Web-обозревателей, как показано в табл. 3.3.

	ttf	otf	eot
Safari 3.1 и более поздние версии	Да	Да	
Opera 10 и более поздние версии	Да	Да	
Firefox 3.5 и более поздние версии	Да	Да	
IE4 и более поздние версии			Да

Таблица 3.3. Поддержка Web-обозревателями форматов файлов шрифтов

#### Примечание

Появился новый формат шрифта, Web Open Font Format (WOFF), что дает некоторую надежду. Поддержка включена в Firefox 3.6. Дополнительную информацию см. на странице http://hacks.mozilla.org/2009/10/woff/.

#### Внедрение с поддержкой разными Web-обозревателями

Для преобразования файла шрифта в еоt-файл и обеспечения поддержки разными Web-обозревателями корпорация Microsoft предлагает приложение, названное Web Embedding Fonts Tool или WEFT (http://www.microsoft.com/typography/WEFT.mspx). Несмотря на то что это средство работает, оно в течение некоторого времени не обновлялось. Обязательно прочтите внимательно руководство.

Для кодирования внедрения шрифта с поддержкой разными Web-обозревателями правило @font-face позволяет ссылаться на несколько файлов:

```
@font-face {
  font-family: "Fontin Sans";
  src: url("fonts/font-file.otf")format("opentype"),
   url("fonts/font-file.eot") format("embedded-opentype");
}
```

Этот метод также позволяет устанавливать связь с альтернативным местоположением файла в случае неработоспособности одного из Web-серверов:

```
@font-face {
  font-family: "Museo 300";
  font-style: normal;
  font-weight: normal;
  src: url("http://example.com/fonts/font-file.otf")format("opentype"),
    url("http://example.com/fonts/font-file.eot") format("embedded-opentype"),
    url("http://csscookbook.com/fonts/font-file.otf")format("opentype"),
    url("http://csscookbook.com/fonts/font-file.eot")
format("embedded-opentype");
}
```

#### Проблема, связанная с внедрением шрифтов

Во время написания книги ряд поставщиков, продававших шрифты, не лицензировали их для внедрения в Web-страницы. Если же они все-таки продавали лицензию на шрифт, цена ее была непомерно высока. (Внедрение шрифтов отличается от создания изображения с набором символов и включения этого изображения в Web-страницу. Это вполне законное действие, если вы заранее купили шрифты, используемые для создания изображений.)

Несмотря на то, что формат eot был разработан, чтобы помочь специалистам по типографике управлять цифровыми правами, связанными с их работой, внедрение этого формата все еще не завоевало популярности.

Специалисты по созданию шрифтов опасаются того, что внедренный шрифт можно довольно легко скопировать, и он перестанет быть их источником дохода, что вполне справедливо, особенно потому, что именно с таким типом поведения сражались фотографы с тех пор, как Web-обозреватель Mosaic ввел элемент img, и Американская ассоциация компаний звукозаписи (Recording Industry Association of America, RIAA) со времени появления файлообменной сети Napster.

Некоторые специалисты по типографике находят способы продавать шрифты и в то же время оставлять их доступными для внедрения. Например, Йос Бейвенга (Jos Buivenga), чей шрифт использован в реализации данного примера, выпускает несколько шрифтов из семейства для бесплатного распространения (http://www.josbuivenga. demon.nl/). Для получения дополнительных прав, позволяющих пополнить список шрифтов, вы вносите небольшую плату. Некоторые другие специалисты по типографике, например компания Fonthead Design (http://fonthead.com), при покупке их шрифтов включают внедрение в типовую лицензию.

#### Примечание

Список свободно распространяемых шрифтов, пригодных для внедрения см. по адресу http://www.fotsquirrel.com/.

#### Сторонние обходные пути

Ряд решений, предложенных сторонними компаниями или специалистами, позволяет применять внедрение шрифта без кражи файлов шрифтов. Web-дизайнер Ричард Раттер (Richard Rutter) в июле 2008 г. предложил подобное решение (http:// clagnut.com/blog/2166/):

"Дизайнеры вовсе необязательно должны загружать файл шрифта с собственного Web-сервера. Они могут сослаться на файл шрифта, находящийся на другом сервере. И в этом кроется реальная возможность.

Когда вы встраиваете географическую карту Google в свою Web-страницу, вы не загружаете пакет изображений карт с сайта Google и не оставляете их на своем сервере, вы ссылаетесь на сайт Google, который затем предоставляет их имеющим лицензию доменам. Точно такой же подход можно применить и к шрифтам. Создатели шрифтов могли бы выдавать лицензию на внедрение шрифтов и предоставлять их с помощью своей системы хранения или через доверенные сторонние системы только зарегистрированным Web-сайтам".

#### Другие методы

К альтернативным возможностям включения в проекты Web-страниц шрифтов разных гарнитур относятся Flash и изображения.

sIFR 3 — название типичного обходного пути, применяющего Flash и JavaScript и позволяющего вставлять шрифты без внедрения. Дополнительную информацию см. на странице http://wiki.novemberborn.net/sifr3/How+to+use.

Другое решение — установить собственные шрифты на странице, заменить HTMLтекст изображениями. Дополнительную информацию см. в *разд. 4.20*.

### См. также

Правило @font-face в спецификации CSS на странице http://www.w3.org/TR/2008/ REC-CSS2-20080411/fonts.html#font-descriptions; статью "Bulletproof @font-face syntax" (Надежный синтаксис @font-face) в блоге Пола Айриша (Paul Irish) по адресу http://tr.im/Gxhf.

# 3.5. Принудительная разбивка действительно длинных слов

### Проблема

Вы хотите добиться разбивки длинного слова (или длинной строки символов).

### Решение

Примените свойство word-wrap со значением break-word, результат показан на рис. 3.8:

```
p {
  border: 1px solid black;
  width: 150px;
  padding: 12px;
}
p.break {
  word-wrap: break-word;
}
```

```
CSS Cookbook
CC
4
       X
                              ( 1)
                                     (1P)
                                                               57
Back
      Forward
                Reload
                               Home
                                      Print
                                            1Password
     CSS Cookbook
                    0
                            Christopher Schm... 💮
                                                 A Technology Books
The doctor says I have
Pneumonoultramicrosc
opicsilicovolcanoconio
sis and I should buy a
dictionary.
The doctor says I have
Pneumonoultramicroscopicsilicovolcanoconiosis
and I should buy a
dictionary.
```

Рис. 3.8. Самое длинное слово в словаре разделено и помещено внутри рамки

### Обсуждение

Появившись в спецификации CSS3, свойство word-wrap первым было применено в Internet Explorer. После того Safari и Firefox 3.5 тоже приняли его на вооружение.

Стандартное значение свойства word-wrap — default, обеспечивающее обычное отображение длинного слова, т. е. выход его за пределы поля.

### См. также

Свойство word-wrap в спецификации CSS3 на странице http://www.w3.org/TR/css3text/#word-wrap; *pa3d. 3.11* о создании вырезок из больших фрагментов текста.

# 3.6. Задание размеров шрифта

### Проблема

Вы хотите задать размер шрифта, применяемого на Web-странице.

### Решение

Установите значения размеров шрифта с помощью свойства font-size:

```
p {
font-size: 0.9em;
}
```

### Обсуждение

Свойство font-size может принимать разные значения и выражаться в разных единицах измерения. В приведенном решении использовалась единица измерения ет (кегельная или круглая шпация). Есть и другие единицы измерения, например проценты.

Задание размера шрифта в процентах заставляет Web-обозреватель вычислять размер шрифта, основываясь на размере шрифта родительского элемента. Например, если для элемента body задан размер шрифта 12 пикселов и размер шрифта элемента р определен как 125 %, в тексте абзацев размер шрифта будет 15 пикселов.

Вы можете использовать для задания размера шрифта проценты, единицы длины и ключевые слова свойства font-size, определяющие размер шрифта.

### Единицы длины

Единицы длины бывают двух категорий: абсолютные и относительные. К абсолютным относятся следующие:

- ♦ дюймы (in);
- сантиметры (сm);
- миллиметры (mm);
- ♦ пункты (pt);
- пики (pc).

В терминах спецификации CSS пункт равен 1/72 дюйма, а пика равна 12 пунктам.

В свойстве font-size задание отрицательного значения в единицах длины, например -25 см, недопустимо.

#### Относительные единицы

Относительные единицы задают значение свойства в зависимости от аналогичного значения другого свойства. Относительные единицы длины включают следующие:

- em (кегельная);
- ▲ x-height (ex) (х-высота);
- ♦ pixels (px) (пикселы).

Единицы ет ссылаются на размер стандартного шрифта, установленного в глобальных параметрах Web-обозревателя пользователя, в то время как единицы x-height (ex) ссылаются на высоту строчного символа x в шрифте.

Пиксел — это мельчайшая точка, которую можно установить на экране компьютера.

#### Задание нулевого и отрицательных значений размера шрифта

В спецификации CSS не говорится о том, как разработчики Web-обозревателя должны трактовать текст с нулевым значением размера шрифта. Следовательно, разные Web-обозреватели интерпретируют это значение непредсказуемо.

Например, в Web-обозревателях Firefox или Mozilla такой текст не виден. В Internet Explorer для Macintosh и в Safari текст не скрыт, а напротив отображается со стандартным значением размера шрифта. Web-обозреватель Opera отображает такой текст уменьшенным, но все еще разборчивым. Safari 4 для Macintosh превращает текст в мелкий, неразборчивый, но все же видимый (с размером шрифта, равным 0.1 em), как показано на рис. 3.9.



Рис. 3.9. Safari 4 для Macintosh отображает неразборчивый текст, если размер шрифта задан равным нулю

Если вы хотите сделать текст невидимым, вместо задания нулевого размера шрифта используйте свойства CSS visibility или display.

```
p {
  display: none;
}
```

### См. также

Свойство font-size в спецификации CSS 2.1 на странице http://www.w3.org/TR/ CSS21/fonts.html#font-size-props.

# 3.7. Достижение большего соответствия размеров шрифтов в разных Web-обозревателях

### Проблема

Вы хотите, чтобы размер шрифта был постоянным в разных Web-обозревателях и операционных системах.

### Решение

Задайте значение для свойства font-size в элементе body равным 62.5 %:

```
body {
  font-size: 62.5%;
}
```

Затем в Internet Explorer для Windows задайте font-size для элементов-потомков form и table, равным 1 em:

```
input, select, th, td {
  font-size: lem;
}
```

Теперь размер шрифтов в вашем документе будет эквивалентен 10 пикселам для каждой единицы 1 ет. Например, если вы добавите объявление элемента body из первой части предложенного решения, то следующее правило задаст размер шрифта в абзаце, равным 19 пикселам:

```
p {
  font-size: 1.9em // отображает текст размером 19 пикселов
}
```

## Обсуждение

Поскольку отображение в Web-обозревателях меняется в разных операционных системах и при разных настройках видеопараметров, задание размера шрифта в фиксированных (или абсолютных) единицах имеет мало смысла. На самом деле лучше всего избегать абсолютных размеров в Web-документах, если вы не разрабатываете стилевое оформление документов для определенного типа вывода. Например, если создается таблица стилей для печати Web-документа, абсолютные единицы длины предпочтительней. Дополнительную информацию о создании таблицы стилей для печати смотрите в *главе 11*.

### Применение пикселов

Несмотря на то, что пикселы применяются для согласованного управления размерами шрифтов Web-документа в большинстве платформ и Web-обозревателей, не стоит использовать пикселы при разработке Web-типографики.

Основная проблема при задании размера шрифта в пикселах касается не размерной точности, а специальных возможностей. Людям с ослабленным зрением может потребоваться изменить размер шрифта для облегчения чтения документа.

Но если вы на вашей Web-странице применяете пикселы для задания шрифта, пользователи Internet Explorer 7 не смогут изменить размер шрифта. Поскольку Internet Explorer для Windows — самый популярный Web-обозреватель на планете, использование пикселов для задания размера шрифта становится проблемой для подавляющего большинства пользователей, нуждающихся в изменении размера шрифта в их Webобозревателях.

#### Примечание

В Internet Explorer 8 и всех остальных Web-обозревателях есть свойства масштабирования, которые позволяют увеличивать шрифты (даже если их размер задан в пикселах) и изображения.

Если вам все же нужны абсолютные единицы для задания размера, следует применять пикселы, а не пункты, несмотря на то, что специалисты по подготовке к печати больше привыкли к измерениям в пунктах. Причина в том, что операционные системы Macintosh и Windows отображают размеры в пунктах по-разному, а размер в пикселах, как правило, остается одним и тем же.

#### Примечание

Несмотря на то, что формально пикселы — это относительные единицы измерения, дизайнеры воспринимают их как абсолютные. Пиксел — относительная единица с точки зрения его физического размера, но его можно отнести к абсолютным единицам с точки зрения соотношения размеров (size ratio) на Web-странице, которое особенно важно для дизайнера.

Если важна именно легкость просмотра, перейдите на единицы измерения ет. В приведенном решении мы задали размер текста в абзаце 0.9 ст. Это значение эквивалентно установке размера шрифта, равного 90 % от размера стандартного шрифта, заданного в глобальных параметрах Web-обозревателя.

Однако применение единиц измерения ет порождает новую проблему. Теперь она связана с удобством использования или пригодностью. Несмотря на то, что вы можете изменять размер шрифта на Web-странице, если вы зададите размер меньше стандартного размера шрифта в Web-обозревателе (например, 0.7 em), Internet Explorer для Windows выведет на экран мелкий, почти неразборчивый текст (рис. 3.10). Вывод таков: будьте осторожны с относительными размерами, т. к. очень легко сделать текст нечитаемым.



Рис. 3.10. Почти нечитаемый шрифт, заданный с помощью единиц измерения ет

#### Применение ключевых слов шрифта

Еще одно возможное решение — использование ключевых слов в свойстве font-size. В спецификации CSS 2.1 есть семь ключевых слов, определяющих абсолютные размеры шрифта, которые можно применять для задания размера (рис. 3.11): xx-small, x-small, small, medium, large, x-large.

Есть еще два ключевых слова для задания относительных размеров в свойстве fontsize: larger и smaller. Если в дочернем элементе задан размер larger, Webобозреватель может интерпретировать значение родительского элемента font-size как small и увеличить текст в дочернем элементе до medium.



У ключевых слов свойства font-size есть два преимущества: они облегчают увеличение или уменьшение размера текста в большинстве Web-обозревателей, и размеры шрифта в обозревателях никогда не становятся меньше 9 пикселов, гарантируя разборчивость текста на экране. Если вы задаете маленький размер текста, используйте шрифт семейства sans-serif, такой как Verdana, для улучшения разборчивости текста.

#### Использование единиц ет для управления шрифтами

Несмотря на то, что ключевые слова, обозначающие размер шрифта, предоставляют общее управление типографическими размерами, дизайнеры обычно хотят иметь больше вариантов для выбора, чем несколько обеспечиваемых ключевыми словами. Решение, предложенное в этом примере и разработанное Ричардом Раттером (Richard Rutter) (http://www.clagnut.com/), предлагает такой вариант управления.

По умолчанию для Web-типографики обозреватели устанавливают значение 16 пикселов, равное ключевому слову medium. При задании размера шрифта в элементе body равным 62.5 % стандартное значение 16 пикселов уменьшается до 10 пикселов:

(16 pixels) 62.5% = 10 pixels

Как упоминалось ранее, единица измерения ет (кегельная) — это размер по умолчанию в пользовательском Web-обозревателе. При манипуляции с размером шрифта по умолчанию в элементе body одна единица ет теперь равна 10 пикселам.

1 em = 10 px

Такое решение позволяет Web-разработчику, желающему управлять размерами шрифтов, заданными в пикселах, сделать это, минуя ограничения, проявляющиеся при использовании пикселов в качестве значений.

Например, если Web-разработчик хочет задать размер заголовка равным 24 пикселам, в то время как размер текста абзаца равен 15 пикселам, значения, установленные в правиле, базирующемся на этом решении, будут выглядеть следующим образом:

```
body {
  font-size: 62.5%;
}
input, select, th, td {
  font-size: lem;
}
h2 {
  font-size: 2.4em;
}
p {
  font-size: 1.5em;
}
```

#### См. также

Оригинальную статью Ричарда Раттера (Richard Rutter) с подробным описанием предложенного решения на сайте http://www.clagnut.com/blog/348/; статью "CSS Design: Size Matters" (Дизайн CSS: размер имеет значение), написанную Тоддом Фарнером

(Todd Fahrner) (приглашенным членом рабочей группы (Working Group) по CSS консорциума W3C) на странице http://www.alistapart.com/articles/sizematters/; спецификацию CSS 2.1 для получения более подробной информации об определении значений Web-обозревателями на странице http://www.w3.org/TR/CSS21/cascade.html#q1; спецификацию CSS 2 о единицах длины на странице http://www.w3.org/TR/REC-CSS2/syndata.html#length-units; *pa30. "Font Size"* (размер шрифта) в *главе 5* книги Эрика Мейера (Eric A. Meyer) "Cascading Style Sheets: the Definitive Guide" (Каскадные таблицы стилей: подробное руководство) (издательство O'Reilly Media).

# 3.8. Установка дефисов, длинных и коротких тире

### Проблема

Вы хотите вместо дефиса применять длинные (Em) и короткие (En) тире, как показано на рис. 3.12.



Рис. 3.12. Применение длинных и коротких тире

### Решение

Задайте длинное тире его десятичным кодом —:

```
Look I don't care if IE6 can't render the page
correctl&#8212;what? we're having a baby?
```

Для короткого тире используйте десятичный код –:

I took the Myers–Brigg test and all I got was this "I'm hard to talk to" t-shirt at work

### Обсуждение

Распространенный способ представления длинных и коротких тире — применение специальных HTML-кодов: &em; и &en; соответственно. Но для лучшей межплатформенной и межобозревательской поддержки лучше применять их десятичные обозначения.

### См. также

Статью о разделении коротких и длинных тире на странице http://www.alistapart.com/ articles/emen.

# 3.9. Центрирование текста

### Проблема

Вы хотите центрировать текст абзаца или заголовка.

### Решение

```
Воспользуйтесь свойством text-align со значением center:
```

```
h3 {
  text-align: center;
}
p {
  text-align: center;
}
```

### Обсуждение

Значение center свойства text-align предназначено для управления выравниванием внутристрочного содержимого в блочном элементе.

### См. также

Свойство text-align в спецификации CSS 2.1 на странице http://www.w3.org/TR/ CSS21/text.html#alignment-prop; *paзd. 5.4* о центрировании различных элементов на Web-странице.

# 3.10. Задание выравнивания текста по ширине

## Проблема

Вы хотите выровнять левый и правый края текста, как показано на рис. 3.13.

### Решение

Воспользуйтесь свойством text-align:

```
p {
  width: 600px;
  text-align: justify;
}
```



Рис. 3.13. Абзац, выровненный с обеих сторон

### Обсуждение

Хорошо ли выравнивается текст по ширине Web-обозревателем? Что касается спецификации CSS 2.1, это зависит от алгоритмов, разработанных проектировщиками, сделавшими Web-обозреватель средством просмотра Web-страниц. Поскольку нет общепринятого алгоритма выравнивания текста по ширине, в разных обозревателях текст выглядит по-разному, даже если поставщик Web-обозревателя обеспечивает техническую поддержку выравнивания по ширине или выключки строк.

Хорошо выравнивание по ширине поддерживается в Internet Explorer, Safari, Firefox, Chrome и Opera. В этих Web-обозревателях у выровненного по ширине текста привлекательный внешний вид. В других обозревателях выровненный по ширине текст может выглядеть плохо; например, у него могут быть слишком большие расстояния между словами.

#### Примечание

Выровненный по ширине текст трудно читать людям, страдающим дислексией. Дополнительную информацию о дизайне для людей с затруднением чтения см. по адресу http://www.thepickards.co.uk/index.php/200512/designing-for-dyslexia/.

#### См. также

Свойство text-align в спецификации CSS 2.1 на странице http://www.w3.org/ TR/CSS21/text.html#alignment-prop.
# 3.11. Обозначение избыточного текста многоточием

# Проблема

Вы хотите избежать выхода за заданные границы родительского элемента, как показано на рис. 3.14.

O       CSS Cookbook         Image: CSS Cookbook       Image: CSS Cookbook						
The doctor says I have Pneumonoultramicro and I should buy a dictionary.						
The doctor says I ha						
			1			

Рис. 3.14. Дополнительный текст обозначается многоточием

# Решение

Примените свойство text-overflow (наряду с собственным свойством -o-text-overflow Web-обозревателя Opera):

```
p {
  border: 1px solid black;
  width: 150px;
  height: 100px;
  padding: 12px;
  border: 1px solid black;
  overflow: hidden;
  padding: 1em;
  text-overflow: ellipsis;
  -o-text-overflow: ellipsis;
}
p.nowrap {
  white-space: nowrap;
  height: auto;
}
```

## Обсуждение

В настоящее время Web-обозреватели Safari и Орега поддерживают свойство text-overflow для сокращения текста и подстановки многоточия (...).

## См. также

Свойство text-overflow в спецификации CSS3 на странице http://www.w3.org/TR/ 2003/CR-css3-text-20030514/#text-overflow.

# 3.12. Удаление свободного пространства между заголовками и абзацами

# Проблема

Вы хотите уменьшить расстояние между заголовком и абзацем.

# Решение

Задайте нулевые значения свойств margin и padding для заголовка и абзаца:

```
h2 + p {
  margin-top: 0;
  padding-top: 0;
}
h2 {
  margin-bottom: 0;
  padding-bottom: 0;
}
p {
  margin: 1em 0 0 0;
  padding: 0;
}
```

# Обсуждение

Применив селектор атрибута, вы задаете поле и отступ между абзацем и заголовком, равными 0.

У Web-обозревателей есть собственные внутренние таблицы стилей, предписывающие значения по умолчанию для HTML-элементов. У этих стилей есть предопределенные значения для свойств margin и padding у заголовков и абзацев.

Упомянутые значения по умолчанию облегчают чтение документов без стилевого оформления, но очень часто мешают Web-разработчикам.

# См. также

Стандартную таблицу стилей для HTML 4 в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/sample.html.

# 3.13. Задание простой буквицы

# Проблема

Вы хотите начать абзац с увеличенной первой буквы.

# Решение

Разметьте абзац текста с помощью элемента р:

Online, activity of exchanging ideas is sped up. The distribution of messages from the selling of propaganda to the giving away of disinformation takes place at a blindingly fast pace thanks to the state of technology …

Воспользуйтесь псевдоэлементом :first-letter для стилизации первой буквы абзаца (рис. 3.15):

```
p:first-letter {
  font-size: 1.2em;
  background-color: black;
  color: white;
}
```



Рис. 3.15. Простая буквица

# Обсуждение

В спецификации CSS предлагается легкий способ стилизации первой буквы абзаца как традиционной приподнятой буквицы или выступающего инициала (initial cap) и буквицы (drop cap): используйте для этого псевдоэлемент :first-letter.

Псевдоэлемент : first-letter поддерживается современными Web-обозревателями, но для поддержки более старых версий Internet Explorer требуется другое решение.

Включите первую букву первого предложения первого абзаца в элемент span c атрибутом class:

<span class="initcap">O</span>nline, activity of exchanging ideas is sped up. The distribution of messages from the selling of propaganda to the giving away of disinformation takes place at a blindingly fast pace thanks to the state of technology …

Затем задайте стиль для приподнятой буквицы:

```
p .initcap {
  font-size: 1.2em;
  background-color: black;
  color: white;
}
```

Выступающие инициалы и буквицы, также называемые версалами (versals), традиционно увеличиваются при печати на величину от нескольких пунктов до трех строк текста.

#### См. также

Псевдоэлемент : first-letter в спецификации CSS 2.1 на странице http://www.w3.org/ TR/CSS21/selector.html#x52.

# 3.14. Формирование центрированной приподнятой буквицы большего размера

## Проблема

Вы хотите поместить большую приподнятую буквицу в центре абзаца.

# Решение

Создайте декоративное оформление, задающее красную строку<sup>1</sup> абзаца (рис. 3.16):

```
p {
  text-indent: 37%;
  line-height: lem;
}
p:first-letter {
  font-size: 6em;
  line-height: 0.6em;
  font-weight: bold;
}
```

# Обсуждение

Это решение работает благодаря влиянию свойства text-indent, которое сдвигает начало первой строки к середине абзаца.

<sup>&</sup>lt;sup>1</sup> Термин "красная строка" используется для обозначения первой строки абзаца с отступом. — Прим. пер.



Рис. 3.16. Центрированная приподнятая буквица увеличенного размера

Его значение равно 37 %, что немного больше одной трети расстояния от левого края абзаца, как показано на рис. 3.17, но недостаточно для центрирования буквицы.



Рис. 3.17. Текст с заданной красной строкой

Обратите внимание на то, что в этом примере приподнятая буквица центрируется, если ширина символа равна 26 % ширины абзаца. Другими словами, если буква для приподнятой буквицы или ширина абзаца у вас другие, необходимо откорректировать значения в правилах CSS, для того чтобы добиться центрирования буквицы.

#### См. также

*Разд. 3.30*, в котором регулируется межстрочный интервал или интерлиньяж с помощью высоты строки; свойство text-indent в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/text.html#propdef-text-indent.

# 3.15. Декоративное оформление приподнятой буквицы с помощью изображения

# Проблема

Вы хотите использовать изображение для оформления приподнятой буквицы.

# Решение

Включите первую букву первого предложения первого абзаца в элемент span:

```
<span class="initcap">O</span>nline, activity of exchanging
ideas is sped up. The distribution of messages from the selling of
propaganda to the giving away of disinformation takes place at a
blindingly fast pace thanks to the state of technology…
```

Скройте содержимое элемента span:

```
span.initcap {
  display: none;
}
```

Задайте изображение, которое будет использоваться в фоне абзаца как приподнятая буквица (рис. 3.18):

```
p {
    line-height: 1em;
    background-image: url(initcap-o.gif);
    background-repeat: no-repeat;
    text-indent: 35px;
    padding-top: 45px;
}
```



## Обсуждение

Первый шаг в данном решении — создание изображения, которое будет использоваться в качестве приподнятой буквицы. После создания изображения оцените его ширину и высоту. В этом примере изображение буквы равно 55×58 пикселов (рис. 3.19).



Рис. 3.19. Изображение приподнятой буквицы

Далее скройте первую букву HTML-текста, задав для свойства display значение none. Затем поместите изображение в фон абзаца, убедитесь в том, что изображение не повторяется, задав значение no-repeat в свойстве background-repeat:

```
background-image: url(initcap-o.gif);
background-repeat: no-repeat;
```

Уже зная размеры изображения, задайте его ширину как значение для красной строки абзаца, а высоту изображения как величину отступа перед абзацем или над абзацем (рис. 3.20):

```
text-indent: 55px;
padding-top: 58px;
```



Рис. 3.20. Высвобождение пространства для буквицы

Затем откорректируйте значения свойств text-indent и padding-top для того, чтобы расположить приподнятую буквицу на уровне базовой линии шрифта (см. рис. 3.18).

#### Облегчение просмотра

Учтите, что пользователи, отключившие изображения, не увидят первой буквы, т. к. в решении не применяется атрибут alt для изображения. Если вы хотите применить изображение, но сохранить атрибут alt для вывода при отключенных изображениях, используйте HTML-символ, замещающий изображение:

```
<img src="initcap-o.gif" alt="O" />nline, activity of exchanging
ideas is sped up. The distribution of messages from the selling
of propaganda to the giving away of disinformation takes place at
a blindingly fast pace thanks to the state of technology…
```

Учтите также, что, когда в данном решении отображается элемент alt, возможность корректировки свободного пространства между начальной буквой и остальным HTMLтекстом теряется. HTML-текст начинается точно от правого края изображения и его нельзя сместить ближе к букве, которая отображается на месте изображения.

#### См. также

Разд. 3.13 о задании простой приподнятой буквицы.

# 3.16. Создание заголовка

#### со стилизованным текстом

#### Проблема

Вы хотите с помощью свойств CSS сформировать заголовок, отличающийся от стандартного. Например, вывести заголовок, показанный на рис. 3.21, курсивным шрифтом (рис. 3.22).





Рис. 3.22. Заголовок со стилевым оформлением

#### Решение

Сначала корректно разметьте заголовок:

```
<h2>Designing Instant Gratification</h2>
Online, activity of exchanging ideas is sped up. The
distribution of messages from the selling of propaganda to the
giving away of disinformation takes place at a blindingly fast
pace thanks to the state of technology…
```

Затем воспользуйтесь свойством font с сокращенной формой записи, позволяющим легко изменить стиль заголовка:

```
h2 {
  font: bold italic 2em Georgia, Times, "Times New Roman", serif;
  margin: 0;
  padding: 0;
  }
  p {
  margin: 0;
  padding: 0;
  }
```

## Обсуждение

Свойство с сокращенной формой записи объединяет несколько свойств в одном. Свойство font — одно из таких экономящих время средств. В свойстве font могут быть представлены следующие значения:

- ♦ font-style
- ♦ font-variant
- font-weight
- font size/line-height
- ♦ font-family

Первые три можно приводить в любом порядке, а остальные должны задаваться в указанном порядке.

Если вы хотите включить значение line-height, поместите слэш (/) между значениями font size и line-height:

```
p {
  font: lem/1.5em Verdana, Arial, sans-serif;
}
```

Задавая стилевое оформление заголовков, помните о том, что у Web-обозревателей есть собственные значения по умолчанию для отступов и полей вокруг тегов абзацев и заголовков. Эти значения основываются скорее на математических расчетах, а не на эстетических представлениях, поэтому без колебаний корректируйте их для улучшения визуального представления вашего Web-документа.

## См. также

Свойство font с сокращенной формой записи в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/fonts.html#propdef-font.

# 3.17. Создание заголовка со стилизованным текстом и рамкой

# Проблема

Вы хотите задать стилевое оформление нижней и верхней сторон рамки, окружающей заголовок (рис. 3.23).



#### Решение

Воспользуйтесь свойствами border-top и border-bottom при задании стиля заголовка:

```
h2 {
  font: bold italic 2em Georgia, Times, "Times New Roman", serif;
  border-bottom: 2px dashed black;
  border-top: 10px solid black;
  margin: 0;
  padding: 0.5em 0 0.5em 0;
  font-size: 1em;
  }
  p {
  margin: 0;
  padding: 10px 0 0 0;
  }
```

#### Обсуждение

Помимо верхней и нижней сторон рамки у блочного элемента может задаваться стиль левой и правой сторон рамки с помощью свойств border-left и border-right соответственно. Свойства border-top, border-bottom, border-left и border-right — это свойства с сокращенной формой записи, позволяющие разработчикам задавать ширину, стиль линии и цвет для каждой стороны рамки.

Вместо стандартных объявлений свойств border с сокращенной формой записи, приведенных в решении, правило CSS увеличилось бы на четыре дополнительных объявления:

```
h2 {
  font: bold italic 2em Georgia, Times, "Times New Roman", serif;
  border-bottom-width: 2px;
  border-bottom-style: dashed;
  border-bottom-color: black;
  border-top-width: 10px;
  border-top-style: solid;
  border-top-color: black;
  margin: 0;
  padding: 0.5em 0 0.5em 0;
  font-size: 1em;
}
```

Есть также свойство border с сокращенной формой записи для свойств border-top, border-bottom, border-left и border-right всех вместе. Свойство border задает одина-ковые значения для ширины, стиля и цвета всех четырех сторон рамки, окружающей элемент:

```
h2 {
    border: 3px dotted #333333;
}
```

При задании рамки проверьте значение отступа (padding) вокруг блока для того, чтобы было достаточно свободного пространства между рамкой и текстом заголовка. Это

улучшит читаемость. При недостатке свободного пространства у заголовка его текст может выглядеть зажатым.

# См. также

Подробную информацию о стилях рамок и свойстве border с сокращенной формой записи в *разд. 5.5*.

# 3.18. Стилевое оформление заголовка с помощью текста и изображения

# Проблема

Вы хотите поместить под заголовком повторяющееся изображение, например траву, показанную на рис. 3.24.



Рис. 3.24. Фоновое изображение, используемое вместе с заголовком

# Решение

Bocпoльзyйтecь свойствами background-image, background-repeat и background-position:

```
h2 {
  font: bold italic 2em Georgia, Times, "Times New Roman", serif;
  background-image: url(tall_grass.jpg);
  background-repeat: repeat-x;
  background-position: bottom;
```

```
border-bottom: 10px solid #666;
margin: 10px 0 0 0;
padding: 0.5em 0 60px 0;
}
```

# Обсуждение

Обратите внимание на высоту изображения, используемого как фон. В данном примере она равна 100 пикселам (рис. 3.25).



Рис. 3.25. Изображение высокой травы

Задайте значение свойства background-repeat равным repeat-x, что вызовет повторение вывода изображения по горизонтали:

```
background-image: url(tall_grass.jpg);
background-repeat: repeat-x;
```

#### Примечание

Местонахождение изображения, задаваемое как url(), касается его местоположения относительно таблицы стилей, а не HTML-документа.

Затем установите значение свойства background-position равным bottom:

background-position: bottom;

В свойстве background-position можно задавать два значения, относящиеся к горизонтальной и вертикальной осям. Значения этого свойства могут указываться в единицах длины (таких как пикселы), процентах или ключевым словом. Для размещения элемен-



Рис. 3.26. Изображение выровнено по нижней границе оси У и по середине оси Х

та по оси X используйте одно из следующих ключевых слов: left, center или right. Для оси Y применяйте ключевые слова: top, center или bottom.

Если для одной из осей не указана позиция, изображение располагается в центре по этой оси, как показано на рис. 3.26.

background-position: bottom;

Таким образом, в данном решении изображение расположено внизу по оси *Y* и повторяется вдоль оси *X*.

## См. также

Задание фонового изображения для целой Web-страницы в разд. 4.5.

# 3.19. Создание врезки в HTML-тексте

## Проблема

Вы хотите задать стилевое оформление текста врезки (pull quote), отличающееся от стандартного (рис. 3.27). Невыделенные врезки, в отличие от стилизованных, сливаются с остальным текстом.

0	0		CSS	Cookbook		$\bigcirc$
Back	Forward Reload Sto	op Home Print	1P 1Password	file:///Us	ers/christophe 🏠 🔻 - 🚷 🕻 Google	Q
	CSS Cookbook	🕲 🚺 Christ	opher Schmi	tt : Designer, 🕲	🖉 🖧 Technology Books, Tech Confe 🔇	+ =
						4
	Lorem ipsum dolor faucibus tellus eget metus dictum place eget pede ut risus f <i>Ma quande lii</i> <i>regulari quan</i> Nunc pharetra, libe elit, dapibus in, por lobortis ut, placerai posuere sed, iaculis parturient montes, tincidunt ut, tortor. Etiam orci. Phasell	r sit amet, consecte t elit. Lorem ipsum rat. Quisque quan aucibus ultricies. I ngues coalesce, li n ti del coalescen ero non dictum mo tritior aliquet, posu t vitae, tempus et, i s quis, consectetue nascetur ridiculus Vestibulum placer us imperdiet dapib	tuer adipis a dolor sit a a tellus, lob n sit amet l i gramma t lingues. Illis, nisi od arcu. Vestil r vel, erat. mus. Done at. Nunc p us urna. Se	cing elit. Proin di imet, consectetue ortis quis, consec ibero eget justo t <i>tica del resultar</i> io porta metus, ir I. Phasellus pede. Julum molestie ti Cum sociis natoq c magna magna, haretra pede eget ed euismod ultrici	ictum adipiscing sapien. Nam r adipiscing elit. Nam id mi vel uat id, congue eget, mi. Proin incidunt ornare. <i>Int lingue es plu simplic e</i> John Smith at the movies a ornare nulla erat a velit. Ut leo Cras posuere. Nunc magna enim, ncidunt metus. Mauris nisl orci, ue penatibus et magnis dis posuere sit amet, sollicitudin eget, turpis. Suspendisse vitae tellus. ies augue.	

Рис. 3.27. Стилизованная врезка

# Решение

Для смыслового выделения врезки в коде разметки используйте элемент блочной цитаты blockquote:

#### <blockquote>

```
Ma quande lingues coalesce, li grammatica del resultant
lingue es plu simplic e regulari quam ti del coalescent
lingues.
<div class="source">John Smith at the movies</div>
</blockguote>
```

С помощью технологии CSS задайте значения свойств margin, padding и color для элемента blockquote:

```
blockquote {
  margin: 0;
  padding: 0;
  color: #555;
}
```

Затем задайте стили для элементов р и div, вложенных в элемент blockquote:

```
blockquote p {
  font: italic 1em Georgia, Times, "Times New Roman", serif;
  font-size: 1em;
  margin: 1.5em 2em 0 1.5em;
  padding: 0;
  }
  blockquote .source {
   text-align: right;
   font-style: normal;
   margin-right: 2em;
  }
```

# Обсуждение

Врезка применяется в дизайне для того, чтобы завладеть вниманием читателя и заставить его продолжить чтение. Легко создать врезку, изменив цвет фрагмента основного текста.

Можно усилить выделение, добавив контраст: изменить гарнитуру шрифта для врезки, чтобы она отличалась от основного текста. Например, если основной текст Webдокумента набран шрифтом sans-serif (без засечек), задать для врезки шрифт семейства serif (с засечками).

#### См. также

Подробную информацию о проектировании врезок с помощью технологии CSS в *разд. 3.21* и *3.22*.

# 3.20. Расположение врезки сбоку от колонки

# Проблема

Вы хотите поместить врезку сбоку от основного текста.

# Решение

Задайте отступ для левого края текста:

```
#content {
   padding-left: 200px;
}
```

Затем примените свойство float, чтобы заставить основной текст обтекать врезку:

```
blockquote {
  padding: 0;
  margin: 0;
  float: left;
  width: 180px;
  text-align: right;
  color: #666;
}
```

Далее задайте отрицательное значение поля, чтобы сместить врезку в область отступа слева от текста, как показано на рис. 3.28:

```
blockquote {
  padding: 0;
  margin: 0;
  float: left;
  width: 180px;
  margin-left: -200px;
  text-align: right;
  color: #666;
}
```

# Обсуждение

Расположение врезки слева от текста — двухшаговый процесс.

Сначала выделяется место для врезки с помощью отступа элемента, содержащего весь контент. Затем для блочной цитаты blockquote в плавающей врезке задается отрицательное значение, чтобы полностью извлечь ее из основного текста.

Этот прием не ограничивается врезками, он также полезен для размещения слева от текста фотографий, подкрепляющих контент.

## См. также

Главу 10 о дополнительных способах обтекания текста на Web-странице.



Рис. 3.28. Врезка слева от основного текста

# 3.21. Создание врезки с рамкой

#### Проблема

Вы хотите создать стилевое оформление врезки с помощью обрамления ее сверху и снизу, как показано на рис. 3.29.

#### Решение

Для задания обрамления слева и справа от врезки вместо задания верхней и нижней стороны рамки используйте свойства border-left и border-right:

```
border-left: 1em solid #999;
border-right: 1em solid #999;
```

Для разметки текста врезки используйте элемент blockquote:

#### <blockquote>

```
«Ma quande lingues coalesce, li
grammatica del.»
</blockquote>
```

Далее задайте правила CSS для рамки и текста, включенного во врезку:

```
blockquote {
  float: left;
  width: 200px;
  margin: 0 0.7em 0 0;
```

```
padding: 0.7em;
color: #666;
background-color: black;
font-family: Georgia, Times, "Times New Roman", serif;
font-size: 1.5em;
font-style: italic;
border-top: lem solid #999;
border-bottom: lem solid #999;
}
blockquote p {
margin: 0;
padding: 0;
text-align: left;
line-height: 1.3em;
}
```



Рис. 3.29. Стилевое оформление врезки с помощью обрамления сверху и снизу

#### Обсуждение

Для элемента blockquote задайте свойства float и width. Эти два свойства CSS позволят основному контенту обтекать врезку со всех сторон:

```
float: left;
width: 200px;
```

Выделите врезку из окружающего текста, изменив в ней цвета фона и текста:

```
color: #666;
background-color: black;
```

Используйте свойства border-top и border-bottom, чтобы добиться соответствия цвету текста во врезке:

border-top: lem solid #999; border-bottom: lem solid #999;

### См. также

В *славе 11* несколько приемов создания макета страницы с использованием свойства float; в *разд. 3.17* стилизацию заголовков с помощью рамок; в *разд. 13.3* и *13.4* дополнительную информацию о применении контраста при разработке страницы.

# 3.22. Создание врезки с изображениями

## Проблема

Вы хотите создать стилевое оформление врезки с помощью изображений в виде фигурных скобок, расположенных с двух сторон от нее, как показано на рис. 3.30.



Рис. 3.30. Врезка с изображениями

#### Решение

Используйте элемент blockquote для разметки текста врезки:

#### <blockquote>

Ma quande lingues coalesce, li grammatica del resultant lingue es plu simplic e regulari quam ti.

#### </blockquote>

Затем задайте стиль для врезки, поместив одно изображение в фон элемента blockquote, а другое — в фон элемента p: blockquote {

```
background-image: url(bracket left.gif);
background-repeat: no-repeat;
 float: left;
 width: 175px;
margin: 0 0.7em 0 0;
padding: 10px 0 0 27px;
 font-family: Georgia, Times, "Times New Roman", serif;
 font-size: 1.2em;
 font-style: italic;
 color: black;
}
blockquote p {
margin: 0;
padding: 0 22px 10px 0;
width:150px;
 text-align: justify;
 line-height: 1.3em;
background-image: url(bracket right.gif);
background-repeat: no-repeat;
background-position: bottom right;
}
```

# Обсуждение

В данном решении изображения фигурных скобок выводятся парой: одна в левом верхнем углу врезки, а другая в правом нижнем углу. В спецификации CSS разрешается назначать элементу блочного уровня только одно фоновое изображение.

Для того чтобы расположить изображения в нужных местах страницы, поместите одно изображение в фон элемента blockquote, а другое в фон элемента p, дочернего элемента тега blockquote:

```
blockquote {
  background-image: url(bracket_left.gif);
  background-repeat: no-repeat;
  float: left;
  width: 175px;
}
blockquote p {
  background-image: url(bracket_right.gif);
  background-repeat: no-repeat;
  background-position: bottom right;
}
```

Затем откорректируйте значения отступа, поля и ширины элементов blockquote и р для того, чтобы добиться полной видимости изображений:

```
blockquote {
  background-image: url(bracket_left.gif);
  background-repeat: no-repeat;
  float: left;
  width: 175px;
  margin: 0 0.7em 0 0;
  padding: 10px 0 0 27px;
  }
  blockquote p {
  margin: 0;
  padding: 0 22px 10px 0;
  width:150px;
  background-image: url(bracket_right.gif);
  background-repeat: no-repeat;
  background-position: bottom right;
  }
```

164

Достоинство предложенного решения заключается в том, что при изменении размера шрифта текста (рис. 3.31) изображения меняют позицию самостоятельно.



Рис. 3.31. Фоновые изображения остаются в углах врезки при изменении размера шрифта

#### См. также

Разд. 7.20.

# 3.23. Задание отступа в первой строке абзаца

# Проблема

Вы хотите создать отступ в первой строке (красную строку) каждого абзаца, как показано на рис. 3.32.



Рис. 3.32. Абзацы с красной строкой

# Решение

Для создания красной строки воспользуйтесь свойством text-indent:

```
p {
  text-indent: 2.5em;
  margin: 0 0 0.5em 0;
  padding: 0;
}
```

# Обсуждение

Значения в свойстве text-indent можно задавать в абсолютных и относительных единицах длины, а также в процентах. Если вы указываете проценты, то они базируются на ширине элемента, а не на ширине страницы в целом. Другими словами, если задан абзацный отступ 35 % для абзаца с шириной 200 пикселов, величина отступа будет равна 70 пикселам.

#### См. также

Более подробную информацию о свойстве text-indent в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/text.html#propdef-text-indent.

# 3.24. Задание отступа для всего абзаца

### Проблема

Вы хотите задать отступы для абзаца целиком, как показано на рис. 3.33.



Рис. 3.33. Абзацы с заданными отступами

## Решение

Для достижения желаемого эффекта воспользуйтесь селекторами класса:

```
p.normal {
   padding: 0;
   margin-left: 0;
   margin-right: 0;
}
p.large {
   margin-left: 33%;
   margin-right: 5%;
}
```

```
p.medium {
  margin-left: 15%;
  margin-right: 33%;
}
```

Затем вставьте соответствующий атрибут в разметку текста.

Tifum factorial non deposit quid pro quo hic escorol. Olypian
quarrels et gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum e
pluribus unum. Defacto lingo est igpay atinlay. Marquee selectus non provisio
incongruous feline nolo contendre.

Gratuitous octopus niacin, sodium glutimate. Quote meon an
estimate et non interruptus stadium. Sic tempus fugit esperanto hiccup estrogen.
Glorious baklava ex librus hup hey ad infinitum. Non sequitur condominium facile
et geranium incognito.

Li Europan lingues es membres del sam familie. Lor separat
existentie es un myth. Por scientie, musica, sport etc., li tot Europa usa li sam
vocabularium. Li lingues differe solmen in li grammatica, li pronunciation e li
plu commun vocabules.

#### Обсуждение

Селекторы класса позволяют выбрать все HTML-элементы, использующие атрибут class. Разница между селекторами класса и типа состоит в том, что селекторы типа позволяют отобрать все экземпляры определенного HTML-элемента. Далее показаны два правила CSS, в первом приведен селектор типа, задающий вывод красным цветом всего контента, размеченного как элемент h2, во втором правиле указан селектор класса, задающий отступ элемента, равный 33 %:

```
h2 {
  color: red;
}
.largeIndent {
  padding-left: 33%;
}
```

Сочетание селекторов типа и класса в одном элементе позволяет добиться большей специфичности в стилевом оформлении элементов. В приведенной далее разметке третий элемент выводится красным цветом и с отступом слева равным 33 %:

```
<h2>Этот абзац красный.</h2>
<h3 class="largeIndent">У этого абзаца значительно больше отступ.</h3>
<h2 class="largeIndent">Этот абзац красный и с большим отступом.</h2>
```

Другое решение, в котором можно использовать вместо селекторов класса создание отступа с помощью полей и применение для стилевого оформления абзацев селекторов соседних элементов (adjacent sibling selector):

```
p, p+p+p+p {
  padding: 0;
  margin-left: 0;
  margin-right: 0;
}
```

```
p+p, p+p+p+p {
  margin-left: 33%;
  margin-right: 5%;
  }
  p+p+p, p+p+p+p+p+p {
  margin-left: 15%;
  margin-right: 33%;
  }
```

Этот метод использует селекторы соседних элементов, представленные двумя или несколькими однотипными селекторами, разделенными знаком (знаками) плюс. Например, селектор h2+p задает стиль для абзаца, следующего сразу за заголовком h2.

В данном примере мы хотим задать стили определенных абзацев в соответствии с очередностью их появления на экране. Например, p+p выбирает абзац, следующий за другим абзацем. Если абзацев больше двух, третий абзац (как и все последующие) отображается с тем же стилем, что и второй абзац. Это происходит потому, что третий абзац следует непосредственно за таким же абзацем.

Для разделения стилей второго и третьего абзацев задайте еще одно правило CSS для третьего абзаца, которое выбирает три абзаца, следующие друг за другом:

```
p+p+p, {
  margin-left: 15%;
  margin-right: 33%;
}
```

Затем создайте видоизмененные правила CSS, сгруппировав селекторы. Вместо написания двух правил для стилевого оформления третьего и шестого абзацев, отделите селекторы запятой и пробелом в одном правиле:

```
p+p+p, p+p+p+p+p+p {
  margin-left: 15%;
  margin-right: 33%;
}
```

Главная проблема применения селекторов соседних элементов состоит в том, что они поддерживаются не во всех версиях Web-обозревателя Internet Explorer для Windows. Следовательно, пользователи этой программы не увидят абзацев с отступами. Селекторы соседних элементов поддерживаются в Web-обозревателях Safari, Firefox, Chrome и Opera. В Internet Explorer 8 они поддерживаются почти полностью.

#### Примечание

Вместо применения селекторов атрибутов, для реализации данного решения можно использовать селектор :nth-child(), позволяющий выделить абзацы, к которым применяется стиль. Однако во время написания книги селекторы атрибутов пользовались поддержкой большего числа Web-обозревателей, чем селектор :nth-child().

#### См. также

Селекторы класса в спецификации CSS 2.1 на странице http://www.w3.org /TR/CSS21/selector.html#class-html; селекторы соседних элементов в спецификации CSS 2.1 на http://www.w3.org/TR/CSS21/selector.html#adjacent-selectors.

# 3.25. Создание обратного отступа

# Проблема

Вы хотите создать обратный отступ, или выступ (hanging indent), в первой строке абзаца.

# Решение

Используйте отрицательное значение для свойства text-indent:

```
p.hanging {
  text-indent: -5em;
}
```

# Обсуждение

Типографическая обработка обратного отступа — обычное явление в списках определений (definition list) большинства Web-обозревателей. Последовательность обратных отступов создается с помощью простого кода (рис. 3.34) без изнурительных усилий.

<dl>

```
<dt>Hanging Indent</dt>
```

<dd>A common typographic effect where the first line of a paragraph is aligned with left margin while the proceeding lines are indented. The technique creates the visual effect where the first line is left hanging over lines of text.</dd></dl>



Рис. 3.34. Список определений с отображением обратного отступа по умолчанию

Если вы хотите создать обратный отступ для простого абзаца (не для списка), разметки с помощью списка определений недостаточно. Простой подход, показанный в решении, включает использование свойства text-indent технологии CSS.

#### При создании обратного отступа соблюдайте осторожность

Прежде чем включать в таблицу стилей свойство text-indent, убедитесь в корректной реализации кода. Например, просто вставка в правило CSS наряду с основными свойствами стилевого оформления шрифта свойства text-indent может вызвать проблемы удобочитаемости текста.

На рис. 3.35 обратный отступ уходит за левую границу окна просмотра Web-обозревателя. Читатель может быть сумеет определить обрезанные слова, исходя из содержания оставшегося абзаца, но для него это просто лишняя нагрузка.

Чтобы избежать подобной ситуации, используйте для обратного отступа в первой строке значение, равное левому полю (margin) абзаца. Обратный отступ займет область, которая уже добавлена как поле, гарантируя, что текст в обратном отступе останется видимым.

```
p.hanging {
 text-indent: -5em;
 margin-left: 5em;
}
        CSS Cookbook
                         C
                                            -
                               X
                                     (1P)
               🗋 (file:///Users/ 🏠 🔻 ) - 🚷 🖌 Google
                                                                                                     Q
                                                   1Password
                         Reload
                               Ston
                                             Print
        Back
              Forward
                                      Home
                 CSS Cookbook
                                         Christopher Schmitt : Desi...
                                                                😧 🖉 Technology Books, Tech C.
       nore time with feeling
        Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet
        dolore magna aliquam erat volutpat ...
```

Рис. 3.35. Обратный отступ, уходящий влево за пределы окна просмотра

#### Спаренный обратный отступ

Обратный отступ первой строки и сдвиг влево заголовка в результате создают спаренный обратный отступ:

```
#content p.hanging {
  text-indent: -60px;
  margin: 0 0 0 60px;
  padding: 0;
  }
#content h3 {
  text-indent: -60px;
  margin: 0 0 0 60px;
  padding: 0;
}
```

HTML-разметка для получения описанного эффекта такова:

```
<div id="content">
<h3>Once more time with feeling</h3>
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh
euismod tincidunt ut laoreet dolore magna aliquam erat volutpat…
</div>
```

Или после небольших изменений можно сохранить обратный отступ только у заголовка:

```
#content p {
  margin: 0;
  padding: 0 0 0 60px;
}
#content h3 {
  text-indent: -60px;
  margin: 0 0 0 60px;
  padding: 0;
}
```

Переопределенная HTML-разметка выглядит следующим образом:

```
<div id="content">
<h3>One more time with feeling</h3>
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy
nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat…
</div>
```

#### См. также

Свойство text-indent в спецификации CSS 2.1 на странице http://www.w3.org/TR/ CSS21/text.html#propdef-text-indent.

# 3.26. Стилевое оформление первой строки абзаца

## Проблема

Вы хотите выделить первую строку абзаца жирным шрифтом, как показано на рис. 3.36.

#### Решение

Для задания стиля первой строки воспользуйтесь псевдоэлементом : first-line:

```
p:first-line {
  font-weight: bold;
}
```



Рис. 3.36. Первая строка с заданным жирным начертанием шрифта

## Обсуждение

Как и селектор класса, псевдоэлемент позволяет манипулировать стилем фрагментов Web-документа. В отличие от селектора класса область, помеченная как псевдоэлемент, может измениться из-за изменения размера окна Web-обозревателя или размера шрифта. В данном решении объем текста в первой строке может измениться, если размер окна Web-обозревателя станет другим (рис. 3.37).



Рис. 3.37. Объем текста в первой строке меняется при изменении размера окна Web-обозревателя

## См. также

Псевдоэлемент :first-line в спецификации CSS 2.1 на странице http://www.w3.org/ TR/CSS21/selector.html#first-line-pseudo.

# 3.27. Стилевое оформление первой строки абзаца с помощью изображения

# Проблема

Вы хотите создать стилевое оформление первой строки абзаца и включить в нее изображение, например, как показано на рис. 3.38.



Рис. 3.38. Первая строка с фоновым изображением

# Решение

Воспользуйтесь свойством background-image в псевдоэлементе : first-line :

```
p:first-line {
  font-size: 2em;
  background-image: url(background.gif);
}
```

# Обсуждение

С помощью псевдоэлементов :first-line можно применять стили к первой строке текста элемента, но не к ширине самого элемента.

Помимо свойства background-image псевдоэлемент :first-line предоставляет следующие свойства, позволяющие с большей эффективностью управлять дизайном:

Глава	3
-------	---

- ♦ font
- ♦ color
- background
- ♦ world-spacing
- letter-spacing
- text-decoration

#### См. также

Псевдоэлемент : first-line в спецификации CSS 2.1 на странице http://www.w3.org/ TR/CSS21/selector.html#first-line-pseudo.

# 3.28. Создание эффекта выделенного текста

#### Проблема

Вы хотите выделить цветом порцию текста в абзаце, как показано на рис. 3.39.



Рис. 3.39. Выделенный цветом текст

## Решение

Воспользуйтесь элементом strong, чтобы разметить фрагменты текста, которые вы хотите выделить:

```
The distribution of messages from the selling of propaganda
to the giving away of disinformation takes place at a blindingly
fast pace thanks to the state of technology… <strong>This
change in how fast information flows revolutionizes the
culture.</strong>
```

- ♦ vertical-align
- text-transform
- text-shadow
- line-height
- clear

Затем задайте правило CSS для указания выделения цветом:

```
strong {
  font-weight: normal;
  background-color: yellow;
}
```

# Обсуждение

Несмотря на то, что в данном решении для выделения текста применяется элемент strong, вы также можете для этой цели использовать элемент еm. В спецификации HTML 4.01 заявлено, что элемент еm следует применять для разметки логического выделения текста, в то время как элемент strong "обозначает более сильное выделение".

После разметки текста задайте цвет выделения с помощью свойства background-color. Поскольку некоторые Web-обозреватели выводят жирным шрифтом текст, размеченный с помощью элемента strong, задайте значение normal для свойства font-weight. При использовании элемента ет для того, чтобы помешать Web-обозревателям выводить курсивный текст, обязательно задайте значение normal для свойства font-style, как показано в следующих строках кода:

```
em {
  font-style: normal;
  background-color: #ff00ff;
}
```

#### См. также

Элементы strong и ет в спецификации языка HTML на странице http://www.w3.org/ TR/html401/struct/text.html#edef-STRONG.

# 3.29. Изменение цвета при выделении текста

#### Проблема

Вы хотите изменить цвет фрагмента при выделении его мышью, как показано на рис. 3.40.

## Решение

Примените псевдоэлемент :: selection для задания цвета текста и цвета фона:

```
::selection {
  color: #90c;
  background: #cf0;
}
```

	-
	$\odot$
Back Forward Reload Stop Home Print 1Password	Q)
CSS Cookbook 😵 🧊 Christopher Schmitt : Designer, 😵 🚑 Technology Books, Tech Confer 🕸	+ =
The Lorem Ipsum & Dolor	*
Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisi ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.	
E Pluribus Unum	U
Epsum factorial non deposit quid pro quo hic escorol. Olypian quarrels et gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum e pluribus unum. Defacto lingo est igpay atinlay. Marquee selectus non provisio incongruous feline nolo contendre. Gratuitous octopus niacin, sodium glutimate. Quote meon an estimate et non interruptus stadium. Sic tempus fugit esperanto hiccup estrogen. Glorious baklava ex librus hup hey ad infinitum. Non sequitur condominium facile et geranium incognito. Epsum factorial non deposit quid pro quo hic escorol. Marquee selectus non provisio incongruous feline nolo contendre Olypian quarrels et gorilla congolium sic ad nauseum.	

Рис. 3.40. Задание цвета при выделении мышью фрагмента текста

## Обсуждение

Во время написания книги псевдоэлемент ::selection поддерживался только Webобозревателем Safari. Однако у Web-обозревателя Firefox есть свой собственный селектор CSS.

Для поддержки выделения в обоих Web-обозревателях продублируйте правило CSS, добавив псевдоэлемент ::moz-selection:

```
::selection {
  color: #90c;
  background: #cf0;
}
::-moz-selection {
  color: #90c;
  background: #cf0;
}
```

## См. также

Фрагменты с элементами UI (пользовательского интерфейса) в спецификации CSS3 на странице http://www.w3.org/TR/2001/Cr-css3-selectors-20011113/#UIfragments.

# 3.30. Изменение межстрочных интервалов

# Проблема

Вы хотите оставить больше или меньше места между строками текста. На рис. 3.41 показана стандартная визуализация в Web-обозревателе, а на рис. 3.42 отображены абзацы с бо́льшим межстрочным интервалом.



Рис. 3.41. Стандартный межстрочный интервал абзаца

O CSS Cookbook	$\bigcirc$
Image: Stop     Reload     Stop     Home     Print     1Password       Image: Stop     Reload     Stop     Home     Print     1Password	9
🗋 CSS Cookbook 🛛 👔 Christopher Schmitt : Designer, 🕄 🚑 Technology Books, Tech Confer 😒 🕇	. =
Ea nam mutat cotidieque, facilisi reprimique an mea. Ea est choro saperet detraxit, eos ullum facilis delectus et. Mutat dicunt pertinax mea ea, mea audire ponderum principes	4 •
no, ad aeque civibus cum. Cu nam accumsan apeirian reprehendunt, nam te fastidii voluptatum, iudico maiorum reformidans ut mei. Ea mucius sensibus vis, et laudem deleniti percipit qui. Eius tractatos efficiendi has eu. Maiorum accusam mei no.	
Ut salutatus mnesarchum sententiae vix, mel ex legere scripserit suscipiantur. Ut vis vocent argumentum, no vim takimata intellegam. Delectus evertitur id usu, placerat suscipiantur ea mei. Mel semper animal no, prima concludaturque in eos, mazim dolorem eum ea. Et pri summo falli, ne sit quis platonem neglegentur. Sea te timeam incorrupte, nisl scripserit in sed. No usu amet suavitate.	
Sea cu nullam regione scriptorem, pro oratio suscipit ut, at eos ocurreret philosophia. Mea movet commodo discere et, at sed placerat referrentur. Aliquip feugiat ea pri, vel eu recusabo dissentias quaerendum, his cu mucius mnesarchum. Et iisque eligendi quo. Ex sea autem eruditi, te reque tamquam ceteros his, ne legere explicari est. Vim quot insolens no.	

Рис. 3.42. Увеличенное межстрочное расстояние в тексте

## Решение

Воспользуйтесь значением свойства line-height:

```
p {
   line-height: 1.5em;
}
```

# Обсуждение

Когда значение свойства line-height увеличивается, межстрочный интервал растет. Когда это значение уменьшается, расстояние между строками текста уменьшается и, в конце концов, строки накладываются друг на друга. Дизайнеры отмечают соответствие свойств высоты строки и межстрочного интервала.

Значение line-height может быть числом с указанием единиц измерения, таких как пункты, просто числом или числом с последующим знаком процента. Если значение свойства line-height — просто число, то оно означает процент или масштабный коэффициент размера самого элемента и его дочерних элементов. Отрицательные значения в свойстве line-height запрещены.

В следующем примере устанавливается размер шрифта 12 пикселов и высота строки — 14.4 пиксела [(10 px \* 1.2) \* 1.2 px = 14.4 px)]:

```
body {
  font-size: 10px;
}
p {
  font-size: 1.2em;
  line-height: 1.2;
}
```

Вы также можете задать свойство line-height с помощью свойства font с сокращенной формой записи, содержащего значение line-height, спаренное с font-size. Приведенная далее строка преобразует любой текст в элементе p, задавая font-size — 1 em, line-height — 1.5 em и отображение семейством шрифтов sans-serif:

```
p {
  font:lem/1.5em sans-serif;
}
```

# См. также

Свойство line-height в спецификации CSS 2.1 на странице http://www.w3.org/ TR/CSS21/visudet.html#propdef-line-height; в *разд. 3.16* более подробную информацию о свойстве font.

# 3.31. Добавление графической обработки HTML-текста

# Проблема

Вы хотите применить повторяющуюся графическую обработку текста, например размытые края или полосы, показанные на рис. 3.43.



для создания эффекта прозрачности

### Решение

Поместите элемент span после тега, открывающего элемент заголовка, но перед HTMLсодержимым заголовка:

<h2><span></span>Designing Instant Gratification</h2>

Далее воспользуйтесь версией метода замещения изображением Гилдера — Левина (http://www.mezzoblue.com/tests/revised-image-replacement/#gilderlevin) для вставки PNG-файла с бесшовным рисунком (seamless pattern) поверх HTML-текста:

```
h2 {
 font:3em/lem Times, serif;
 font-weight: bold;
 margin:0;
 position: relative;
 overflow: hidden;
 float: left;
 text-shadow: 0 1px 0 rgba(153,153,153,.8);
}
h2 span {
 position: absolute;
width: 100%;
height: 5em;
background: url(title-glass.png);
}
} q
 clear: left;
}
```
### Обсуждение

Текст в заголовке задан как плавающий элемент с выравниванием по левому краю. Этот прием позволяет фоновому изображению, помещенному в элемент span, располагаться поверх HTML-текста благодаря абсолютному позиционированию.

Стандартно плавающий элемент, заголовок, должен был бы сместиться влево с обтекающим его контекстом с правой стороны. Но свойство clear, помещенное в абзац, препятствует этому.

Свойству height присвоено значение 5em, а свойству overflow — значение hidden, для того чтобы не дать фоновому изображению выйти за пределы заголовка и распространиться на другие фрагменты Web-документа.

### См. также

Дополнительную информацию о методе замещения изображением Гилдера — Левина на http://www.mezzoblue.com/tests/revised-image-replacement/#gilderlevin.

# 3.32. Вставка тени за текстом

### Проблема

Вы хотите поместить тень за текстом заголовка, как показано на рис. 3.44.



### Решение

Используйте свойство text-shadow для задания цвета и расположения тени:

```
h1 {
  font-size: 2.5em;
  font-family: Myriad, Helvetica, Arial, sans-serif;
  width: 66.6%;
  text-shadow: yellow .15em .15em .15em;
  margin: 0 0 0.1em 0;
}
```

### Обсуждение

Первое значение в свойстве text-shadow задает цвет. Первое значение, заданное в единицах длины, .15ем, сдвигает тень по оси X к позиции HTML-текста. Следующее значение сдвигает тень по оси Y. Последняя величина — радиус рассеивания (blur radius) тени. Чем больше радиус, тем разреженнее тень.

### Задание прозрачности тени

Задавая цвет тени в формате RGBA, вы можете указать степень прозрачности. Это позволит цвету тени лучше сочетаться с цветом фона:

```
body {
  Background-color: #000;
}
h1 {
  font-size: 2.5em;
  font-family: Myriad, Helvetica, Arial, sans-serif;
  width: 66.6%;
  text-shadow: rgba(205, 205, 0, .7) .15em .15em .15em;
  margin: 0 0 0.1em 0;
}
```

### Создание эффекта снятия фаски

Установив расстояние для тени на один пиксел ниже и левее, а также задав степень прозрачности 60 %, вы можете получить с помощью свойства text-shadow простой эффект снятия фаски (bevel effect), показанный на рис. 3.45:

```
body {
    background-color: #999;
}
h1 {
    text-shadow: 0 1px 0 rgba(255,255,255,.6);
}
```

### Вставка отсвета огня над текстом

Свойство text-shadow может принимать несколько значений (каждое из которых отделяется запятой). Это позволяет создавать интересные (с вашей точки зрения) эффекты, например отсветы огня в верхней части заголовка, как показано на рис. 3.46:

```
h1 {
 color: red;
 text-shadow: rgba(0, 0, 0, .9) 0px 0px 1px,
  rgba(255, 255, 51, .9) 0px -5px 5px,
  rgba(255, 204, 51, .7) 2px -10px 7px,
  rgba(255, 153, 0, .6) -2px -15px 10px;
```



# The Lorem Ipsum & Dolor

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

#### E Pluribus Unum

Epsum factorial non deposit quid pro quo hic escorol. Olypian quarrels et gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum e pluribus unum. Defacto lingo est igpay atinlay. Marquee selectus non provisio incongruous feline nolo contendre. Gratuitous octopus niacin, sodium glutimate. Quote meon an estimate et non interruptus stadium. Sic tempus fugit esperanto hiccup estrogen. Glorious baklava ex librus hup hey ad infinitum. Non sequitur condominium facile et geranium incognito. Epsum factorial non deposit quid pro quo hic escorol. Marquee selectus non provisio incongruous feline nolo contendre Olypian quarrels et gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum e pluribus unum.





}

#### Известная поддержка

Поддержка свойства text-shadow известна только в Web-обозревателях Firefox 3.5, Орега 9.5 и их более поздних версиях и Safari.

### Текст с тенью в Internet Explorer

Для задания тени за текстом в Internet Explorer 6 и более поздних версиях используйте его собственное свойство filter:

```
h2 {
   filter:shadow(color=#999999,direction=270, strength=1);
}
```

В приведенном коде свойство color задано в виде шестнадцатеричного значения, direction принимает значения от 0 до 360, а свойство strength — длина тени в пикселах.

#### Примечание

Корпорация Microsoft перенесла в IE8 filter и другие свойства, чтобы использовать расширения спецификации CSS, созданные разработчиками IE. Дополнительную информацию см. на странице http://blogs.msdn.com/ie/archive/2008/09/08/microsoft-css-vendorextentions.aspx.

### См. также

Более подробную информацию о свойстве text-shadow в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/text.html#text-shadow-props.

# 3.33. Корректировка межбуквенных и межсловных пробелов

### Проблема

Вы хотите откорректировать в HTML-тексте расстояния между буквами и словами.

### Решение

Для настройки величины пробелов между буквами используйте свойство letterspacing (рис. 3.47):

```
h2 {
  font: bold italic 2em "Helvetica Nue", serif;
  margin: 0;
  padding: 0;
  letter-spacing: -0.1em;
}
```

Для настройки величины пробелов между словами используйте свойство word-spacing (рис. 3.48):

```
h2 {
  font: bold italic 2em "Helvetica Nue", serif;
  margin: 0;
  padding: 0;
  word-spacing: 0.33em;
}
```

0	O CSS Cookbook	$\bigcirc$
Back	Forward Reload Stop Home Print IPassword	Q
	CSS Cookbook 🛛 🕄 Christopher Schmitt : Designer, 🕄 🖉 Technology Books, Tech Confer 😒	+ =
	Derivering Instant Constitution	
	Designing Instant Gratification	
	I had to wake up early for the dentist appointment this morning. Why is it that on the day I have to	
	wake up really early it interferes with a good sleep? I had the blankets wright where they were most comfortable, too.	
	If that ever happens, our society should allow you to tell the other person you were late because you	
	realized you were in the middle of "a good sleepy". It's the positive version of "I was in a car accident."	
	Try the phrase out, see how it works and report back to me.	
		1

Рис. 3.47. Откорректированные межбуквенные пробелы в тексте заголовка

CSS Cookbook	$\bigcirc$
▲ P , C × ↑ ⊕ P , C  Back Forward Reload Stop Home Print 1Password IPassword C file:///Users/christophe ☆▼ • S • Google	Q
CSS Cookbook 😵 👔 Christopher Schmitt : Designer, 😵 🎘 Technology Books, Tech Confer 😵 🕇	=
Designing Instant Gratification	
I had to wake up early for the dentist appointment this marning. Why is it that on the day I have to	
wake up carly it interferes with a good sleep? I had the blankets wright where they were most comfortable, too.	
If that ever happens, our society should allow you to tell the other person you were late because you	
realized you were in the middle of "a good sleepy". It's the positive version of "I was in a car accident."	
Try the phrase out, see how it works and report back to me.	

### Обсуждение

Одна из сильных сторон технологии CSS — ее способ управления Web-типографикой. Web-дизайнеры и разработчики больше не должны применять напоминающие головоломку наборы вложенных элементов font, b и однопиксельные GIF-фокусы для привлекательного оформления текста. Технология CSS может справиться с легкостью с таким действием, как настройка расстояний между двумя буквами или разделение слов в абзаце.

### Кернинг и трекинг

Корректировка межбуквенных расстояний для улучшения эстетического впечатления — старая традиция в графическом дизайне. Есть два термина, описывающих разные влияния величины межбуквенных пробелов на текст: кернинг (kerning) и трекинг (tracking).

Кернинг — это дизайнерский термин, используемый для описания изменения расстояния между парой букв для улучшения визуального впечатления. Примером кернинга может служить изменение расстояния между заглавной буквой Т и строчной буквой і. *Трекинг* определяет настройку расстояний между буквами в большом фрагменте текста, а не между парой букв.

Свойство word-spacing поддерживается в Web-обозревателях Firefox, Internet Explorer 6 для Windows и более поздних версиях, Opera 3.5 и более поздних версиях и Safari.

### Проверенные практические рекомендации

Для повышения эффективности свойства letter-spacing и word-spacing следует задавать в относительных, а не в абсолютных единицах длины. Поскольку пользователи могут изменять размеры шрифтов в своих Web-обозревателях, фиксированное значение расстояния — 5 пунктов, первоначально предназначенное для размера шрифта 12 пикселов, не изменится, даже если пользователь увеличит шрифт. Другими словами, расстояние между буквами 5 пунктов едва ли будет заметно, если будет задан размер шрифта 72 пиксела или больший. В случае относительных единиц, таких как ет, значение 1.5 ет для свойства letter-spacing будет масштабироваться одновременно с изменением размера шрифта.

Кроме того, лучший вариант визуальных эффектов, применяемых к тексту, — сохранение удобочитаемости текста вне зависимости от использованного стилевого оформления. Если вам и вашему клиенту важно общение, слабое стилевое оформление лучше, чем создание текстовых элементов, понятных только посвященным. Если текст становится неразборчивым, вы можете утомить всех, с кем пытаетесь связаться.

### См. также

Свойство letter-spacing в спецификации CSS 2.1 на странице http://www.w3.org/TR/ CSS21/text.html#propdef-letter-spacing; свойство word-spacing на http://www.w3.org/ TR/CSS21/text.html#propdef-word-spacing; дополнительную информацию о кернинге и трекинге на странице http://desktoppub.about.com/cs/typespacing/a/kerningtracking.htm.

# 3.34. Достижение ритмичности базовых линий в Web-типографике

### Проблема

Вы хотите разместить две колонки текста на одной и той же базовой линии, как показано на рис. 3.49.

00	CSS Cookbook
Back Forward Reload St	C The print 1Password Coogle Q
CSS Cookbook	[ Christopher Schmitt : 😵 🖉 Technology Books, T 😒 🕂 🗮
	The Lorem Ipsum & Dolor
H3 heading	Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut
Ma quande lingues coalesce, li	laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation
grammatica del resultant lingue es	ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in
plu simplic e regulari quam ti del	hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros
coalescent lingues.	et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait
John Smith at the movies	nulla facilisi.
	Epsum factorial non deposit quid pro quo hic escorol. Olypian quarrels et gorilla congolium sic ad
	nauseum. Souvlaki ignitus carborundum e pluribus unum. Defacto lingo est igpay atinlay. Marquee
	selectus non provisio incongruous feline nolo contendre. Gratuitous octopus niacin, sodium glutimate.
	Quote meon an estimate et non interruptus stadium. Sic tempus fugit esperanto hiccup estrogen. Glorious
	baklava ex librus hup hey ad infinitum. Non sequitur condominium facile et geranium incognito. Epsum
	factorial non deposit quid pro quo hic escorol. Marquee selectus non provisio incongruous feline nolo
	contendre Olypian quarrels et gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum e pluribus
	unum.
	Plu Sommun Paroles
	Li Europan lingues es membres del sam familie. Lor separat existentie es un myth. Por scientie, musica,
	sport etc., li tot Europa usa li sam vocabularium. Li lingues differe solmen in li grammatica, li
	pronunciation e li plu commun vocabules. Omnicos directe al desirabilita de un nov lingua franca: on
	refusa continuar payar custosi traductores. It solmen va esser necessi far uniform grammatica,
	pronunciation e plu sommun paroles.

Рис. 3.49. Текст в колонке, выровненный по той же базовой линии, что и строки во врезке

### Решение

В селекторе body задайте свойство font-size, равное 62.5 %, как рекомендовалось в *разд. 3.6*.

```
body {
  font-size: 62.5%
}
```

Затем задайте свойство line-height (или межстрочный интервал) в соответствии с рекомендациями разд 3.30:

```
body {
  font-size: 62.5%;
  line-height: 1.83em;
}
```

Рассчитайте значения line-height для других текстовых HTML-элементов по следующей формуле:

(line-height элемента body / font-size HTML-элемента) = line-height HTML-элемента в единицах em

Для элемента h2 с font-size, равным 1.5 em, частное равно 1.2 em:

(1.83 em / 1.5 em) = 1.2 em

Откорректируйте правила CSS, включив новое значение свойства line-height для элемента h2:

```
body {
  font-size: 62.5%;
  line-height: 1.83em;
}
h2 {
  margin: 0;
  font-size: 1.5em;
  line-height: 1.2em;
}
```

Для того чтобы поля элемента h2 остались в соответствии со свойством line-height, задайте для них то же значение:

```
body {
  font-size: 62.5%;
  line-height: 1.83em;
}
h2 {
  margin: 0;
  font-size: 1.5em;
  line-height: 1.2em;
  margin-bottom: 1.2em;
}
```

Выполните аналогичные расчеты и установки для остальных текстовых HTMLэлементов.

### Обсуждение

Несмотря на то, что эффект выравнивания текста в двух или нескольких колонках по одной базовой линии делает ставку на мастерство, которого не хватает на большинстве Web-страниц, необходимость вычислений, зависящих от содержимого страницы, может сделать тщетными усилия даже самых терпеливых Web-дизайнеров, особенно если им или их клиентам потребуются изменения значения font-size. Это кажущееся простым требование в результате приведет к новой серии вычислений.

Для помощи в подобной ситуации Web-дизайнер Джеффри Грозенбах (Geoffrey Grosenbach) создал программу вычисления нужного межстрочного интервала Baseline Rhythm Calculator (http://topfunky.com/baseline-rhythm-calculator/), облегчающую людские страдания.

### См. также

Статью Ричарда Раттера (Richard Rutter) о вертикальном ритме на странице http://24ways.org/2006/compose-to-a-vertical-rhythm.

# 3.35. Стилевое оформление верхних и нижних индексов без нарушения базовой линии

### Проблема

Вы хотите вставить верхние и нижние индексы без корректировки базовой линии текста, как показано на рис. 3.50.



### Решение

Примените HTML-элементы sup и sub для вставки верхних и нижних индексов соответственно:

Lorem ipsum dolor sit amet, consectetuer adipiscing elit

```
<a href="#footnotel">1</a></sup>, sed diam nonummy nibh euismod tincidunt ut
laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam<sup><a
href="#footnote2">2</a></sup>, quis nostrud exerci tation ullamcorper suscipit
lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor
in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu
feugiat nulla facilisis at vero eros et accumsan et H<sub>2</sub>0 iusto odio
dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait
nulla facilisi.
```

Затем отрегулируйте выравнивание текста в элементах sup и sub:

```
sup, sub {
  vertical-align: baseline;
  position: relative;
  top: -0.4em;
}
sub {
  top: 0.4em;
}
```

### Обсуждение

Предложенное решение работает благодаря привязке к базовой линии текста внутри элементов sup и sub, так же, как и в остальном тексте. В дальнейшем для получения нужного вам расположения верхнего и нижнего индекса вы сможете смещать текст относительно базовой линии с помощью относительного позиционирования (см. *разд. 2.24*).

### См. также

Статью об этом методе в блоге Web-дизайнера Пола Армстронга (Paul Armstrong) на странице http://paularmstrongdesigns.com/weblog/stop-superscripts-from-breaking-line-heights-once-and-for-all.

# 3.36. Отображение нескольких колонок текста

### Проблема

Вы хотите разместить длинный фрагмент текста в нескольких колонках, как показано на рис. 3.51.

### Решение

Для размещения текста в нескольких колонках поместите его в элемент div:

```
div id="column">
```

```
<h2>...</h2>
...<h2>...
<h2>...</h2>
...
</div>
```

### Примените специализированные свойства column-gap и column-width:

```
#column {
    -moz-column-gap: 3em;
    -moz-column-width: 11em;
    -webkit-column-gap: 3em;
    -webkit-column-width: 11em;
    padding: 10px;
}
```

Затем задайте разделительные линии с помощью свойства column-rule:

```
#column {
    -moz-column-gap: 3em;
    -moz-column-width: 11em;
    -moz-column-rule: 1px solid #ccc;
    -webkit-column-gap: 3em;
    -webkit-column-width: 11em;
    -webkit-column-rule: 1px solid #ccc;
    padding: 10px;
}
```



amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulnutate velit esse molestie consequat, vel illum

dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

#### E Pluribus Unum

Epsum factorial non deposit quid pro quo hic escorol. Olypian quarrels et gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum e lingo est igpay atinlay. Marquee selectus non provisio incongruous feline nolo contendre. Gratuitous octopus niacin, sodium glutimate. Quote meon an estimate et non interruptus stadium. Sic tempus fugit esperanto hiccup estrogen. Glorious baklava ex librus hup hey ad infinitum. Non sequitur condominium facile et geranium incognito.

Epsum factoriai non deposit quid pro quo hic escorol. Marquee selectus non provisio incongruous feline nolo contendre Olypian quarrels et gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum e pluribus unum.

#### Plu Sommun Paroles

Li Europan lingues es membres del sam familie. Lor separat scientie, musica, sport etc., li tot Europa usa li sam vocabularium. Li lingues differe solmen in li grammatica, li pronunciation e li plu commun vocabules. Omnicos directe al desirabilita de un nov lingua franca: on refusa continuar payar custosi traductores. It solmen va esser necessi far uniform grammatica. pronunciation e plu sommun paroles.

### Обсуждение

Применение свойств колонок экономит время Web-дизайнеров, потому что формирование колонок текста — это очень трудоемкий процесс.

Для получения колонок в проекте Web-дизайнерам пришлось бы сосчитать количество слов для каждой колонки, чтобы обеспечить равное количество слов в каждой из них; включить количество слов для каждой колонки в отдельный элемент div и поместить каждый из этих элементов на свое место с помощью позиционирования или плавающей модели.

### Известные проблемы

Свойства колонок из спецификации CSS3 делают процесс создания колонок текста легким и автоматизированным. Главная проблема заключается в том, что они поддерживаются только собственными расширениями CSS в Web-обозревателях Firefox и Safari.

Решение на языке JavaScript с помощью модуля библиотеки jQuery предлагает альтернативу применению специализированных свойств CSS (http://welcome.totheinter.net/ 2008/07/22/multi-column-layout-with-css-and-jquery/).

Примечание

Приемы создания многоколоночных макетов см. в главе 11.

### См. также

Тест свойств колонок Петера-Пауля Коха (Peter-Paul Koch) на странице http://www.quirksmode.org/css/multicolumn.html.

# глава 4 Изображения

# 4.0. Введение

Когда в начале 90-х годов прошлого века первый Web-обозреватель Марка Андриссена (Marc Andreessen) сделал возможным вывод встроенной в текст фоновой графики, появилась возможность внести визуальную привлекательность во Всемирную паутину. Совместно используемые документы больше не выглядели как наполненные текстом научные статьи, а стали для разработчиков основой развития новой сферы деятельности, Web-дизайна.

С тех давних времен дизайнеры помимо размещения одного-двух изображений на Web-странице начали применять для улучшения Web-сайтов GIF-, JPEG- и PNGфайлы.

В этой главе обсуждается много примеров, касающихся обработки изображений средствами технологии CSS. Примеры включают создание рамок, использование фоновых изображений, скругление углов блоков, замещение HTML-текста изображениями и многое другое.

# 4.1. Преобразование средствами CSS цветных изображений в черно-белые в IE

### Проблема

Вы хотите в Internet Explorer преобразовать цветные изображения на Web-странице в версии в оттенках серого.

### Решение

Используйте специализированное свойство CSS filter, предназначенное для автоматического преобразования цветных изображений в изображения в оттенках серого:

```
img {
  filter: gray;
}
```

#### Примечание

В IE8 корпорация Microsoft передает filter и другие свойства, чтобы сохранить возможность применения расширений CSS от производителей. Дополнительную информацию см. на странице http://blogs.msdn.com/ie/archive/2008/09/08/microsoft-css-vendorextentions.aspx.

### Обсуждение

Свойство filter не самое полезное свойство CSS, тем не менее находит применение.

Примером может быть формирование изображений в оттенках серого в таблицах стилей для печати (см. *главу* 10). Этот подход экономит деньги ваших пользователей, т. к. цветные чернила дороже черных.

Еще один пример — создание для более старых версий Internet Explorer пользовательских таблиц стилей с условными комментариями (см. *разд. 12.7*), в которых вся графика задается как черно-белая. Именно этот подход использовал дизайнер Энди Кларк (Andy Clarke) при переработке своего сайта (http://stuffandnonsense.co.uk/blog/ about/hello/).

### См. также

Свойство filter для получения оттенков серого в документации MSDN на странице http://msdn.microsoft.com/en-us/library/ms533003(VS.85).aspx.

# 4.2. Вывод рамки вокруг изображения

### Проблема

Вы хотите заключить изображение в рамку.

### Решение

Используйте свойство border в элементе img (рис. 4.1):

```
img {
  width: 300px;
  border: 2px solid #666;
  background: #fff;
  padding: 2px;
}
```

### Обсуждение

Если превратить изображение в ссылку, можно создать более сложное оформление с помощью свойства border.

Псевдокласс :hover позволяет менять стиль рамки, когда пользователь укажет мышью на изображение, как показано справа на рис. 4.1:

```
img {
  width: 300px;
```

```
border: 2px solid #666;
background: #fff;
padding: 2px;
}
a:hover img {
border-style: solid;
background: #999;
}
```

Рамка служит обрамлением изображения, и в то же время ее цвет и стиль можно менять, когда пользователь наводит указатель мыши на изображение. Отступ 2 пиксела, заданный в блоке объявления элемента img, позволяет изменять цвет также внутри рамки. Таким образом, простое наведение указателя мыши на изображение создает интересный визуальный эффект с помощью всего двух блоков объявлений.



Рис. 4.1. Рамка вокруг изображения

### См. также

Разд. 4.4 об удалении рамок вокруг изображений.

# 4.3. Установка вокруг изображения рамки со скругленными углами

# Проблема

Вы хотите скруглить прямые углы рамки вокруг изображения.

# Решение

Задайте рамку, а затем примените свойство спецификации CSS3 border-radius вместе со специфичными для Web-обозревателей свойствами border-radius (рис. 4.2, справа):

```
div{
  background-image: url(beach.jpg);
  width: 375px;
  height: 500px;
  border: 8px solid #666;
  border-radius: 40px;
  -moz-border-radius: 40px;
  -webkit-border-radius: 40px;
}
```



Рис. 4.2. Скругленные углы рамки вокруг изображения справа

### Обсуждение

Радиус, половина диаметра окружности, задает величину скругления угла. Чем больше радиус, тем больше будет скругление угла.

Во время написания книги свойство border-radius не поддерживалось само по себе, но специализированные свойства Web-обозревателей Firefox и Safari обеспечивали тот же эффект. Главный недостаток (помимо отсутствия универсальной поддержки в разных Web-обозревателях) заключался в несогласованности названий свойств рамки, что отражено в табл. 4.1.

CSS3	Firefox	Webkit
border-radius	-moz-border-radius	-webkit-border-radius
border-top-left-radius	-moz-border-radius-topleft	-webkit-border-top-left- radius
border-top-right-radius	-moz-border-radius-topright	-webkit-border-top-right- radius
border-bottom-right-radius	-moz-border-radius-bottomright	-webkit-border-bottom- right-radius
border-bottom-left-radius	-moz-border-radius-bottomleft	-webkit-border-bottom- left-radius

Таблица 4.1. Эквиваленты свойства скругления углов

### Задание углов

Скругленными могут также отображаться отдельные углы, а не все четыре угла рамки. Для задания скругления одного или нескольких углов укажите в правиле CSS каждый скругляемый угол отдельно.

Например, в следующем правиле CSS задается скругление всех углов, кроме правого верхнего:

```
div#roundbkgd {
background-image: url(beach.jpg);
width: 375px;
height: 500px;
border: 8px solid #666;
/* левый верхний угол*/
border-top-left-radius: 40px;
-moz-border-radius-topleft: 40px;
-webkit-border-top-left-radius: 40px;
```

/\* правый нижний угол \*/ border-bottom-right-radius: 40px; -moz-border-radius-bottomright: 40px; -webkit-border-bottom-right-radius: 40px;

```
/* левый нижний угол */
border-bottom-left-radius: 40px;
-moz-border-radius-bottomleft: 40px;
-webkit-border-bottom-left-radius: 40px;
}
```

### Известные проблемы

Если изображение встроено в текст или помещено в HTML не как фоновое изображение, скругленные углы отображаются под изображением, а не обрезают углы изображения, как показано на рис. 4.2, слева:

```
img {
  width: 375px;
  height: 500px;
  border: 8px solid #666;
  background: #fff;
  display:block;
  border-radius: 40px;
  -moz-border-radius: 40px;
  -webkit-border-radius: 40px;
}
```

Для решения этой проблемы сохраните значение свойства border-radius относительно малым (не более четырех или пяти пикселов) или задайте изображение как часть фона элемента (см. *разд. 4.5*).

#### Примечание

В Web-обозревателе Орега поддержка свойства border-radius запланирована для основного выпуска, следующего за Opera 10.

### См. также

Свойство border-radius в спецификации CSS3 на странице http://www.w3.org/ TR/2005/WD-css3-background-20050216/#the-border-radius.

# 4.4. Удаление рамок, устанавливаемых по умолчанию вокруг изображений в некоторых Web-обозревателях

### Проблема

Вы хотите удалить рамки вокруг изображений, активизируемых щелчком мыши, таких как изображение, показанное на рис. 4.3.



Рис. 4.3. Изображения с рамкой и без нее

### Решение

Задайте нулевую толщину рамки для изображений:

```
a img {
border: 0;
}
```

### Обсуждение

До появления технологии CSS разработчики задавали рамку вокруг изображения с помощью атрибута border элемента img:

```
<a href="http://csscookbook.com">
<img src="beach.jpg" border="0" alt="beach" />
</a>
```

### См. также

Разд. 4.2 о создании рамки вокруг изображения.

# 4.5. Установка фонового изображения

# Проблема

Вы хотите применить неповторяющееся фоновое изображение.

# Решение

Используйте свойства background-image и background-repeat для управления выводом изображения (рис. 4.4).

```
body {
    background-image: url(bkgd.jpg);
    background-repeat: no-repeat;
}
```



Рис. 4.4. Фоновое изображение выводится один раз в левом верхнем углу

# Обсуждение

Вы можете помещать текст и встроенные в него изображения поверх фонового для придания глубины Web-странице. Можно также вставить в рамку Web-страницу с помощью мозаичного расположения фонового изображения вдоль краев окна Webобозревателя.

### См. также

*Разд. 4.6* о формировании горизонтальной или вертикальной полосы повторяющихся фоновых изображений.

# 4.6. Создание полосы фоновых изображений

### Проблема

Вы хотите вывести ряд повторяющихся фоновых изображений, расположенный вертикально или горизонтально.

### Решение

Для создания горизонтальной полосы из фоновых изображений (рис. 4.5) или полосы, расположенной вдоль оси *X*, воспользуйтесь следующим правилом CSS:

```
body {
  background-image: url(bkgd.jpg);
  background-repeat: repeat-x;
}
```

### Обсуждение

Для того чтобы фоновые изображения повторялись вдоль вертикальной оси, задайте значение repeat-у в свойстве background-repeat.



#### 200

### См. также

Разд. 4.7 о выводе фонового изображения в конкретном месте Web-страницы.

# 4.7. Размещение фонового изображения

### Проблема

Вы хотите задать положение фонового изображения на Web-странице.



Рис. 4.6. Фоновое изображение, смещенное на 75 пикселов вправо и 150 пикселов вниз от левого верхнего угла окна просмотра Web-обозревателя

### Решение

Для указания местоположения фонового изображения используйте свойство background-position. Следующее правило CSS помещает начальную точку изображения на 75 пикселов правее и на 150 пикселов ниже верхнего левого угла окна просмотра Web-обозревателя (рис. 4.6):

```
html {
    height: 100%;
}
```

```
body {
  background-image: url(bkgd.jpg);
  background-repeat: no-repeat;
  background-position: 75px 150px;
}
```

### Обсуждение

Свойство background-position содержит два значения, отделенные пробелом. Первая величина задает положение начальной точки по оси X, а вторая — по оси Y. Если задано только одно значение, оно используется для определения горизонтальной позиции, а вертикальная устанавливается равной 50 %.

В предложенном решении в качестве единиц измерения для определения положения фонового изображения использованы пикселы; однако вы можете также использовать и проценты. При значении 50 % для свойства background-position Web-обозреватель помещает изображение в неподвижном центре (dead center) окна просмотра, как показано на рис. 4.7, в то время как значения 0 и 100 % помещают изображение в левый верхний и правый нижний углы окна соответственно.



Рис. 4.7. Фоновое изображение в центре окна просмотра Web-обозревателя

#### 202

Вместе с процентами можно применять в качестве значений ключевые слова top, center и bottom для оси Y и left, center и right для оси X. Комбинируя эти значения, можно поместить фоновое изображение в восьми точках вдоль краев окна просмотра Web-обозревателя (в углах и между ними), а также в центре окна. Например, для переопределения значения 50 % с рис. 4.7 можно использовать следующее правило CSS:

```
body {
  background-image: url(bkgd.jpg);
  background-repeat: no-repeat;
  background-position: center;
}
```

Для того чтобы поместить фоновое изображение в правый нижний угол (рис. 4.8), можно использовать приведенное далее правило CSS:

```
body {
  background-image: url(bkgd.jpg);
  background-repeat: no-repeat;
  background-position: bottom right;
}
```



Рис. 4.8. Фоновое изображение, помещенное в правый нижний угол

Вы также можете применить свойства background-position и background-repeat для вывода фоновых изображений, расположенных рядом, но не привязанных к контуру окна Web-обозревателя.

### См. также

*Paзд. 4.10* о задании неперемещаемого при прокрутке изображения; свойство background-position в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/colors.html#propdef-background-position.

# 4.8. Применение нескольких фоновых изображений в одном HTML-элементе

### Проблема

Вы хотите поместить несколько фоновых изображений в один HTML-элемент.

### Решение

В спецификации CSS3 свойство background с сокращенной формой записи может принимать множественные наборы данных о фоновых изображениях, разделенные запятыми (рис. 4.9):

```
h2 {
border: 1px solid #666;
border-radius: 20px;
 -moz-border-radius: 20px;
 -webkit-border-radius: 20px;
background: white;
 padding-top: 72px;
 text-align: center;
background: url(mail.gif) top center no-repeat,
  url(printer.gif) 40% 24px no-repeat,
  url(gift.gif) 60% 24px no-repeat,
  url(content-bkgd.png) 50% 50% repeat-x,
  url(heading-sub-bkgd.png) 3em 3em repeat-x,
  url(plane.gif) center no-repeat;
 font-family: "Gill Sans", Trebuchet, Calibri, sans-serif;
 color: #666;
}
```

#### Примечание

Обсуждение метода помещения изображений в фон HTML-элементов см. в разд. 4.7.

# Обсуждение

Во время написания книги входящее в спецификацию CSS3 многослойное расположение множественных фоновых изображений в одном элементе было реализовано в Webобозревателе Safari для Macintosh.



Рис. 4.9. В заголовке фоновые изображения в виде отдельных пиктограмм

### Свойства с сокращенной формой записи

Как и большинство свойств с сокращенной формой записи, код свойства CSS background для множественных фоновых изображений можно разделить на отдельные блоки объявления:

```
h2 {
border: 1px solid #666;
border-radius: 20px;
 -moz-border-radius: 20px;
 -webkit-border-radius: 20px;
 background: white;
 padding-top: 72px;
 text-align: center;
background-image: url(mail.gif),
  url(printer.gif),
  url(gift.gif),
  url(content-bkgd.png),
  url(heading-sub-bkgd.png),
  url(plane.gif);
 background-position: top center,
  40% 24px,
  60% 24px,
  50% 50%,
  3em 3em,
  center;
 background-repeat: no-repeat,
  no-repeat,
  no-repeat,
  repeat-x,
  repeat-x,
```

```
no-repeat;
font-family: "Gill Sans", Trebuchet, Calibri, sans-serif;
}
```

Если все фоновые изображения в правиле CSS повторяются, в код можно поместить одно значение no-repeat и применить его ко всем фоновым изображениям:

```
h2 {
padding-top: 72px; /* отступ, достаточный для вывода изображений */
text-align: center;
background: url(plane.gif), url(mail.gif), url(printer.gif), url(gift.gif);
background-position: center, top center, 40% 24px, 60% 24px;
background-repeat: no-repeat;
}
```

Такое сокращение для одинаковых значений может применяться ко всем свойствам CSS, связанным со свойством background, и свидетельствовать о том, что желательно использование всеми фоновыми изображениями одного и того же значения.

#### Не готово для повседневного использования

В настоящее время ввод новых элементов и вставка фоновых изображений в них единственный способ реализовать приемы включения множественных изображений во всех современных Web-обозревателях. Дополнительную информацию и примеры использования этих приемов см. в *разд. 4.14* и 4.15.

### См. также

Дополнительную информацию о применении слоев для вставки множественных изображений в спецификации CSS3 на странице www.w3.org/TR/2005/WD-css3background-20050216/#layering.

# 4.9. Расположение изображений на рамке блока

### Проблема

Вы хотите поместить изображение на границах HTML-элемента.

### Решение

Сначала создайте изображение, содержащее рамку. Поместите контент в элементы div с уникальным значением атрибута id:

#### <div id="section">

```
<h2>Images on Borders</h2>
Epsum factorial non deposit quid pro quo hic escorol. Olypian quarrels et
gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum
e pluribus unum..
</div><!-- /#section -->
```

```
206
```

Затем для расположения изображения вдоль рамки элемента, как показано на рис. 4.10, примените новое свойство CSS3 border-image:

```
#section {
  margin-right: 40px;
  color: #000;
  font-family: Verdana, Geneva, Tahoma, sans-serif;
  width: 100px;
  text-align: center;
  border-style: solid;
  border-color: #930;
  border-width: 26px 39px 37px 43px;
  border-image: url(frame.png) 26 39 37 43 stretch stretch;
  -webkit-border-image: url(frame.png) 26 39 37 43 stretch round;
}
```



Рис. 4.10. Для обрамления контента применяется одно изображение

# Обсуждение

Новое свойство border-image включено в спецификацию CSS3 и во время написания книги поддерживалось Web-обозревателями Firefox 3.1 и Safari 4 и их более поздними версиями.

#### Примечание

Если в предложенном решении изменяется размер текста, изображение, имитирующее рамку, масштабируется и включает в себя весь текст.

Свойство border-image не только позволяет обрамлять контент с помощью одного изображения, способного масштабироваться, но и предоставляет способ создания графически выразительных кнопок для Web-форм, показанных на рис. 4.11.

Сначала создадим простую кнопку средствами языка HTML:

```
<form action="/" method="get">
<button>Submit</button>
</form>
```

Затем применим свойство border-image для задания показанной на рис. 4.11 визуально привлекательной кнопки, которая гораздо интереснее стандартной:

```
button {
background: none;
width: 250px;
padding: 10px 0 10px 0;
border-style: solid;
border-color: #666;
border-width: 0 17px 0 17px;
border-image: url(bkgd-button.png) 0 17 0 17 stretch stretch;
 -webkit-border-image: url(bkgd-button.png) 0 17 0 17 stretch stretch;
 -moz-border-image: url(bkgd-button.png) 0 17 0 15 stretch stretch;
 color: white;
 font-family: "Gill Sans", Trebuchet, Calibri, sans-serif;
 font-weight: bold;
 text-transform: uppercase;
 text-shadow: 0px 0px 5px rgba(0,0,0,.8);
}
```

При размещении изображения на месте рамки сначала задаются ширины сторон рамки:

border-width: 0 17px 0 17px;

Далее с помощью функции url() в свойство border-image вставляется изображение. Следующие четыре значения в свойстве должны соответствовать значениям в свойстве border-width для верхней, правой, нижней и левой сторон рамки HTML-элемента. Значения 0 в свойстве border-image заставляют Web-обозреватель покрывать изображением верхнюю и нижнюю стороны рамки целиком. Значения 17 означают, что для левой и правой сторон рамки должно использоваться 17 пикселов изображения. Следующие два значения в свойстве равны stretch для того, чтобы изображение покрыло расстояние между сторонами рамки и создало гладкое бесшовное заполнение:

border-width: 0 17px 0 17px; border-image: url(bkgd-button.png) 0 17 0 17 stretch stretch;

Помимо значения stretch можно применять repeat (для мозаичного расположения) и round (тоже для мозаичного расположения, но с целым числом повторов между краями рамки).

00	CSS Cookbook		$\bigcirc$
	file:///Users/christ	opher/Desktop 🦙 🔹 🕄	Google <b>Q</b>
Contacts - Events	💽 Locations 🔹 🧮 Tagspaces 🔹 🧮 Bookn	narks 🔹 🧮 Resources 🔹	t Options
CSS Cookbook	🙁 [ []] Christopher Schmitt : D 🙁	Contraction Technology Books, Tech	⊗ + =
in the second second	SUBMIT		

Рис. 4.11. Одно изображение, применяемое для отображения кнопки

### См. также

Свойство border-image в спецификации CSS3 на странице http://www.w3.org/TR/css3background/#the-border-image.

# 4.10. Создание закрепленного фонового изображения

### Проблема

Вы хотите, чтобы фоновое изображение оставалось в окне просмотра Web-обозревателя, даже если пользователь прокручивает Web-страницу в окне.

# Решение

Используйте свойство background-attachment со значением fixed, как показано далее:

```
body {
  background-image: url(bkgd.jpg);
  background-repeat: no-repeat;
  background-attachment: fixed;
}
```

# Обсуждение

Используя этот метод, вы закрепляете фоновое изображение. Даже если посетитель перемещает страницу в окне, изображение остается на прежнем месте. Другое допус-

тимое значение для свойства background-attachment — scroll, являющееся значением по умолчанию. Поэтому, если вы даже не укажете значение scroll, фоновое изображение будет перемещаться вверх вместе с остальным содержимым, когда посетитель прокручивает страницу от начала к концу.

Например, представьте себе, что вы хотите опубликовать на вашей Web-странице фотографию о недавней поездке и поместить ее слева, а текст — справа. Когда читатель перемещает страницу вниз для продолжения чтения, фотография, сделанная в поездке, остается на месте (рис. 4.12).



Рис. 4.12. Фотография остается на месте, когда посетитель прокручивает страницу в окне

#### Далее приведен код:

```
body {
  background- image: url(bkgd2.jpg);
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: -125px 75px;
 margin: 75px 75px 0 375px;
}
h1, h2, h3 {
  padding-top: 0;
  margin-top: 0;
  text-transform: uppercase;
}
p{
  text-align: justify;
}
```

Более того, вы можете закрепить изображение на уровне блочных элементов, отличных от элемента body. Например, попробуйте использовать заголовок, когда проектируете вывод рецензии на фильм или концерт. Приведенный далее код CSS создает интересный вариант навигации.



Рис. 4.13. Фотография проглядывает сквозь заголовки, а не сквозь элемент body

```
h1, h2, h3 {
  font-size: 200%;
  background-image: url(bkgd2.jpg);
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-position: center;
  padding: 1.5em;
  text-align: center;
  color: white;
}
```

Поскольку для заголовков заданы отступы и светлый цвет, на странице достаточно места для того, чтобы видеть фоновое изображение "сквозь" элементы и читать заголовки. Когда посетитель, читая рецензию, переместит Web-страницу, он увидит оставшуюся часть изображения (рис. 4.13).

### См. также

*Paзд. 4.5* о позиционировании фонового изображения; свойство background-attachment в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/colors.html# propdef-background-attachment.

# 4.11. Изменение размеров изображений при изменении размеров окна Web-обозревателя

### Проблема

Вы хотите, чтобы по мере изменения размеров окна Web-обозревателя фоновое изображение растягивалось или сжималось.

### Решение

Примените свойство background-size вместе со специализированными свойствами, предлагаемыми разработчиками Web-обозревателей (рис. 4.14):

```
body {
  background-image: url(button_redstar.gif);
  background-size: 25% auto;
  -o-background-size: 25% auto;
  -webkit-background-size: 25% auto;
  background-repeat: repeat-x;
  margin-top: 30%;
}
```

### Обсуждение

Если задано свойство background-size, Web-обозреватель изменяет изображение в соответствии со значениями свойства. Первое из них задает ширину изображения, а второе — высоту.



Рис. 4.14. Четыре изображения отображаются одинаково в верхней части окна просмотра даже при изменении его размеров

В предложенном решении указание значения, равного 25 %, заставляет Webобозреватель выложить изображение четыре раза вдоль ширины окна просмотра. Поскольку в свойстве background-repeat задано повторение только по оси X, фон формируется всего из четырех изображений. Значение auto по высоте означает сохранение соотношения высоты и ширины изображения.

### Примечание

Свойство background-size поддерживает Web-обозреватель Firefox 3.6.

### См. также

Свойство background-size в спецификации CSS3 на странице http://www.w3.org/ TR/css3-background/#the-background-size.

# 4.12. Растяжение изображения на все окно Web-обозревателя

# Проблема

Вы хотите, чтобы изображение занимало все окно просмотра Web-обозревателя.

### Решение

Для разработки решения, действующего в разных Web-обозревателях, примените HTML-фреймы.

Сначала создайте файл full-bleed.html и поместите изображение в элемент body:

```
<img id="stretch" src="green_car.jpg" alt="photo of green car" />
```

Примените правила CSS для удаления полей и отступов в элементе body и увеличения ширины и высоты изображения:

```
body {
  margin: 0;
  padding: 0;
}
#stretch {
  position: absolute;
  width: 100%;
  height: 100%;
}
```

### Обсуждение

Это решение лучше всего работает в большинстве Web-обозревателей, тестировавшихся с выполнением удаления изображения из нормального потока документа и последующего повторного задания его ширины и высоты, равными 100 %.

Для помещения контента поверх изображения примените абсолютное позиционирование и задайте для нового контента большее (по крайней мере больше, чем у растянутого изображения) значение в свойстве z-index.

### Применение свойства background-size

Наилучший и самый прямой метод создания эффекта заполнения окна просмотра изображением — применение свойства background-size (см. *разд. 4.11*) для растяжения изображения до размеров окна просмотра в Web-обозревателе Safari:

```
body {
  background-image: url(green_car.jpg);
  -webkit-background-size: 100% 100%;
  -o-background-size: 100% 100%;
}
```

Но если окно Web-обозревателя становится меньше, изображение начинает сжиматься, чтобы сохранить соотношение ширины и высоты, и располагает копии изображения под собой.

### Применение элемента iframe

Другой метод — использование HTML-элемента iframe для некоторой имитации фреймовой структуры HTML:

<iframe width="100%" height="100%" src="full-bleed.html" border="0" noborder="noborder" frameborder="0" padding="0" spacing="0" scrolling="no"></iframe>

Однако в некоторых Web-обозревателях, таких как Opera и Safari, растяжение изображения из файла full-bleed.html выполняется не полностью.

#### Примечание

Другой вариант — применение HTML-элемента frameset. Однако из соображений облегчения просмотра лучше не применять этот способ там, где это возможно.

### См. также

*Разд. 2.23* с дополнительной информацией об абсолютном позиционировании элементов.

# 4.13. Создание масштабируемых изображений

### Проблема

Вы хотите, чтобы размеры изображения менялись в соответствии с изменением размеров окна Web-обозревателя.

### Решение

Задайте ширину изображений в процентах (рис. 4.15):

```
img {
  border: 1px solid #cecece;
  width: 60%;
  float: left;
  margin-right: .7em;
  margin-bottom: .5em;
}
```

Современные Web-обозреватели масштабируют высоту и ширину изображений пропорционально, поэтому задавать и ширину, и высоту нет необходимости.

### Обсуждение

При создании эластичной, или "резиновой", верстки HTML-текст, заданный в колонках, способен растягиваться и втягиваться в соответствии с изменениями размеров окна Web-обозревателя. А изображения с размерами, обычно заданными в пикселах, сохраняют свою величину.


Рис. 4.15. Уменьшенное изображение и изображение большего размера из-за увеличения окна Web-обозревателя

Для того чтобы обеспечить пропорциональное изменение всех элементов страницы при "резиновой" верстке, разработчики могут задавать ширину и высоту в процентах.

## Предотвращение излишнего растяжения изображений

Если размеры изображений задаются в процентах, обозреватели могут растянуть их так сильно, что нарушится целостность изображения. Например, при растяжении изображений в форматах JPEG или GIF начинают проявляться искажения, почти невидимые при нормальном размере.

Для того чтобы уберечь изображения от растяжения до ширины больше определенной, используйте свойство max-width, заданное в единицах длины:

```
img {
  border: 1px solid #cecece;
  width: 60%;
  max-width: 300px;
  float: left;
  margin-right: .7em;
  margin-bottom: .5em;
}
```

## См. также

Главу 11 с дополнительными примерами резиновой верстки.

# 4.14. Задание способа визуализации изображений в Web-обозревателе

# Проблема

Вы хотите задать в Web-обозревателе способ отображения изображений (рис. 4.16).



Рис. 4.16. Демонстрация способов визуализации изображений, предлагаемая центром разработчиков Mozilla Developer Center

# Решение

Примените свойство image-rendering и собственное свойство корпорации Microsoft -ms-interpolation-mode:

```
#content img[src$=".gif"] {
    image-rendering: -moz-crisp-edges;
    -ms-interpolation-mode: nearest-neighbor;
}
```

# Обсуждение

Первоначально свойство image-rendering было свойством формата Scalable Vector Graphics (SVG, масштабируемая векторная графика), но оно было преобразовано для работы с HTML-элементами. Свойство инструктирует Web-обозреватель о том, как визуализировать изображения с измененными размерами.

Правило CSS в предложенном решении отображает графику с резкими границами или высококонтрастную. Для сглаживания отображения применяйте следующие правила CSS:

```
#content img[src$=".gif"] {
    image-rendering: optimizeQuality;
    -ms-interpolation-mode: bicubic;
}
```

#### Примечание

В Web-обозревателях Firefox свойство image-rendering применяется к встраиваемым в текст и фоновым изображениям, а также к элементам video и canvas языка HTML5.

#### Сглаживание изображений в Internet Explorer

Web-дизайнер Итан Маркотт (Ethan Marcotte) разработал для визуализации в Internet Explorer изображений, изменяющих размеры, другое решение, в котором для использования свойства filter, созданного Microsoft, применяется JavaScript (http://unstoppablerobotninja.com/entry/fluid-images/). Простой в использовании модуль библиотеки jQuery, расширяющий упомянутое решение, можно найти на странице http://thinkdrastic.net/journal/wp-content/uploads/2009/05imgresizer.htm.

#### См. также

Подробное описание атрибута SVG свойства image-rendering на странице http://www.zvon.org/xxl/svgReference/Output/attr\_image-rendering.html; статью MSDN о свойстве -ms-interpolation-mode на странице http://msdn.microsoft.com/en-us/library/ms530822(VS.85,loband).aspx; описание свойства image-rendering центра разработчиков Mozilla Developer Center по адресу https://developer.mozilla.org/en/CSS/image-rendering.

# 4.15. Поворот изображений средствами CSS

## Проблема

Вы хотите поворачивать изображения.

#### Решение

Сначала задайте элемент img для отображения в виде блочного элемента:

```
img {
    display: block;
    float: left;
    margin: 20px;
}
```

Затем для поворота изображений на 270°, как показано на рис. 4.17, примените набор специализированных свойств CSS, действующих в обозревателях Safari, Firefox и Internet Explorer:

```
img+img {
  -webkit-transform: rotate(270deg);
```

```
-moz-transform: rotate(270deg);
filter: progid:DXImageTransform.Microsoft.BasicImage(rotation=3);
}
```



Рис. 4.17. Первое изображение визуализировано стандартно, а остальные повернуты

## Обсуждение

Web-дизайнеры поворачивают блочные элементы с помощью специализированных свойств, но только на углы, кратные 90°.

Специализированные свойства transform Web-обозревателей Safari и Firefox позволяют поворачивать элементы на произвольный угол (например, 78°), а свойство корпорации Microsoft BasicImage может повернуть только на четыре фиксированных угла, перечисленных в табл. 4.2.

Таблица 4.2. Таблица простого соответствия угла поворота в разных Web-обозревателях

Угол поворота	Значение фильтра BasicImage
0	0
90	1
180	2
270	3

#### Примечание

Свойства transform Web-обозревателей Safari и Firefox также допускают наклоны при отображении блочных элементов. Во время написания книги в Internet Explorer 8 такого эффекта не было. Дополнительную информацию о нем см. в ссылках *разд. "См. также"* предыдущего примера.

#### См. также

Статью MSDN о фильтре rotation на странице http://msdn.microsoft.com/enus/library/ms532918(VS.85,loband).aspx; статью Mozilla Developer Center о свойстве -moz-transform на странице https://developer.mozilla.org/en/CSS/-moz-transform; сообщение о свойстве -webkit-transform в блоге на сайте Surfin' Safari по адресу http://webkit.org/blog/130/css-transforms/.

# 4.16. Задание градиентов средствами CSS

## Проблема

Вы хотите средствами CSS создать фоновые градиенты (плавные переходы цвета).

## Решение

Для задания градиентов в фоновых элементах, как показано на рис. 4.18, примените специализированные свойства CSS:

```
div.building {
  border: lpx solid #666;
  float: left;
  width: 300px;
  height: 300px;
  margin: 20px;
  background-image: -webkit-gradient(radial,center center,900,center
  bottom,0,from(#0cf),to(white));
  background-image: -moz-radial-gradient(center,900px,center
  bottom,0,from(#0cf),to(white));
  background-repeat: no-repeat;
}
```

## Обсуждение

Во время написания книги градиенты, создаваемые средствами CSS, поддерживались в Safari 4 и Firefox 3.6 и более поздних версиях этих Web-обозревателей благодаря применению их собственных соответствующих свойств.

#### Задание градиентов в Safari

В Web-обозревателе Safari вызовите функцию -webkit-gradient() в свойстве background или background-image:

```
background-image: -webkit-gradient();
```

#### Примечание

При разработке дизайна Web-страницы с использованием градиента CSS я рекомендую применять свойство background-image вместо свойства с сокращенной формой записи background, чтобы не блокировать другие свойства, касающиеся фона, например, задающие значение background-color элемента.



Рис. 4.18. Радиальные градиенты, заданные в фоне элемента

Далее задайте вид градиента, который хотите установить: радиальный (radial) или линейный (linear):

background-image: -webkit-gradient(linear);

Затем с помощью значений, аналогичных значениям свойства background-position, задайте начальную точку градиента и его радиус:

background-image: -webkit-gradient(radial,center center,900);

#### Примечание

Обычно сокращенная форма записи значения свойства background-position в данном случае — просто значение center. Но при задании градиента Web-обозреватель Safari не понимает сокращенной формы записи, поэтому не применяйте ее при указании местоположения градиента.

Радиус воспринимайте как конечную точку радиального градиента. Когда Webобозреватель при визуализации градиента достигает конкретной точки, заданной в радиусе, цветовые переходы прекращаются и цвет остается ровным.

#### Примечание

Во время написания книги не принимались значения радиуса, заданные в разных единицах длины, и любое числовое значение интерпретировалось как значение в пикселах.

После того как начальная точка определена, задайте конечную точку:

Указав начальную и конечную точки, переходите к заданию соответствующих цветов:

```
background-image: -webkit-gradient(radial,center
center,900,center bottom,0,from(#0cf),to(white));
```

#### Примечание

Вы можете применять градиенты CSS в Safari не только к фоновым изображениям блочных элементов, но и к маркерам списка, генерируемому содержимому и к изображениям, формирующим границы.

#### Задание градиентов в Firefox

В то время как в Web-обозревателе Safari тип градиента задается в его собственном специализированном свойстве, в Firefox есть отдельное свойство для каждого типа градиента: -moz-radial-gradient() и -moz-linear-gradient().

В отличие от градиента в Safari вы можете при задании начальной и конечной точек градиентов использовать значения с сокращенной формой записи из свойства background-position и единицы длины.

#### Прозрачность с градиентами

Другой интересный аспект реализации градиентов CSS в Firefox касается прозрачности. Если вы не зададите цвет фона для элемента, вы можете указать фоновый цвет (или цвета) с помощью функции RGBA и сформировать прозрачность градиента:

```
background-image: -moz-linear-gradient(left top, left bottom,
from(rgba(153,51,0,.3)), to(#6b3703), color-stop(0.5, #903000));
```

#### Примечание

Поскольку в решении применяются свойства CSS, поддерживаемые только в некоторых современных Web-обозревателях, лучше создать градиенты в графической программе и затем задать их как фоновые изображения (см. *разд. 4.5*).

#### См. также

Сообщение "Introducing CSS Gradients" (Знакомство с градиентами CSS) в блоге на сайте Surfin' Safari по адресу http://webkit.org/blog/175/introducing-css-gradients/; статью о градиентах в Firefox центра разработчиков Mozilla Developer Center на странице https://developer.mozilla.org/en/CSS/-moz-linear-gradient.

# 4.17. Создание прозрачных PNG-изображений для IE6 и более поздних версий

#### Проблема

Вы хотите создать для Internet Explorer 6 обратно совместимые PNG-изображения с прозрачностью без применения JavaScript.

## Решение

Используйте графический редактор Adobe Fireworks для создания смешанных изображений формата PNG8.

Для изображений с прозрачностью, таких как падающая тень, задайте формат изображения PNG8, размытие 100 % и прозрачность, равную альфа-прозрачности (рис. 4.19).



Рис. 4.19. Сохраненная целостность изображения несмотря на то, что альфа-прозрачность градиента урезана

## Обсуждение

Альфа-прозрачность формата PNG24 не работает в IE6, прозрачность часто отображается в виде блока белого цвета, как показано на рис. 4.20. Adobe Fireworks экспортирует неповрежденный и видимый формат PNG8 с его полупрозрачностью в Internet Explorer 7 и более поздние версии.

В настоящее время Adobe Fireworks — единственное коммерческое приложение, обеспечивающее подобный экспорт смешанного формата PNG8.

#### Примечание

К другим приложениям, выполняющим этот особый тип экспорта, относятся pngquant (http://www.libpng.org/pub/png/apps/pngquant.html) и pngnq (http://pngnq.sourceforge.net/index.html).

#### См. также

Статью "PNG8 — The Clear Winner" (PNG8 — явный победитель) на сайте SitePoint по адресу http://www.sitepoint.com/blogs/2007/09/18/png8-the-clear-winner/.



Рис. 4.20. Отсутствие части градиента с альфа-прозрачностью

# 4.18. Применение PNG-файлов с прозрачностью с помощью JavaScript

## Проблема

Вы хотите использовать множественные PNG-файлы с альфа-прозрачностью (alpha-trasparency).

## Решение

Воспользуйтесь обновленным скриптом Sleight Дрю Маклеллана (Drew McLellan) для включения альфа-прозрачности в Internet Explorer 5.5—6.

Вы можете написать код на JavaScript в отдельном файле или загрузить его с сайта Маклеллана http://allinthehead.com/code/samples/bgsleight.js:

```
if (navigator.platform == "Win32" &&
  navigator.appName == "Microsoft Internet Explorer" &&
  window.attachEvent) {
   window.attachEvent("onload", fnLoadPngs);
}
```

```
function fnLoadPngs() {
var rslt = navigator.appVersion.match(/MSIE (\d+\.\d+)/, '');
var itsAllGood = (rslt != null && Number(rslt[1]) >= 5.5);
for (var i = document.all.length - 1, obj = null;
  (obj = document.all[i]); i--) {
   if (itsAllGood &&
obj.currentStyle.backgroundImage.match(/\.png/i) != null) {
  this.fnFixPng(obj);
 obj.attachEvent("onpropertychange",
this.fnPropertyChanged);
 }
 }
}
function fnPropertyChanged() {
if (window.event.propertyName == "style.backgroundImage") {
 var el = window.event.srcElement;
 if (!el.currentStyle.backgroundImage.match(/x\.gif/i)) {
   var bg = el.currentStyle.backgroundImage;
   var src = bg.substring(5,bg.length-2);
    el.filters.item(0).src = src;
    el.style.backgroundImage = "url(x.gif)";
 }
 }
}
function fnFixPng(obj) {
var bg = obj.currentStyle.backgroundImage;
var src = bg.substring(5,bg.length-2);
obj.style.filter =
"progid:DXImageTransform.Microsoft.AlphaImageLoader(src='"
+ src + "', sizingMethod='scale')";
obj.style.backgroundImage = "url(x.gif)";
}
```

Присоедините файл на JavaScript к Web-странице, поместив следующий код между элементами head:

<script src="/\_assets/js/bgsleight.js" type="text/javascript"></script></script></script></script>

Проверьте, загружено ли однопиксельное прозрачное GIF-изображение (в скрипте именуемое x.gif) на Web-сервер, и исправьте ссылку в скрипте на его местоположение в соответствии с вашими установками.

## Обсуждение

Поддержка альфа-прозрачности в современных Web-обозревателях — почти обычное явление. К Web-обозревателям, включающим собственную поддержку PNG-файлов, относятся Chrome, Opera, Safari и Internet Explorer 7 для Windows. К сожалению, в этот перечень не входит Internet Explorer 6 для Windows.

Для того чтобы устранить этот недостаток, Аарон Будман (Aaron Boodman) создал фрагмент на JavaScript, в котором использовано разработанное корпорацией Microsoft свойство filter для активизации поддержки в Internet Explorer 5.5—6 для Windows встроенных PNG-файлов с альфа-прозрачностью, при этом скрипт не влияет на другие Web-обозреватели, имеющие собственную поддержку формата PNG.

Дрю Маклеллан развил работу Будмана и модифицировал скрипт на JavaScript, примененный в решении, заставив его работать не только со встраиваемыми изображениями, но и с фоновыми (см. http://allinthehead.com/retro/69/sleight-of-hand).

#### Примечание

Если вы используете библиотеку jQuery, модуль, базирующийся на данном решении, можно найти по адресу http://jquery.andreaseberhard.de/pngFix/.

#### Как работает скрипт

Скрипт Маклеллана выполняется при загрузке страницы. Он исследует HTMLразметку и ищет элементы img, указывающие на файлы изображений с расширением png.

Когда код с таким элементом img найден, скрипт переписывает HTML-код динамически, на лету. Сначала он заменяет PNG-изображение однопиксельным прозрачным GIF-изображением.

Затем в свойстве filter Internet Explorer указывается PNG-файл для того, чтобы включить альфа-прозрачность в этом Web-обозревателе. Таким единственно возможным образом PNG-изображение устанавливается в фон.

#### Примечание

Для применения специализированных свойств CSS только в Internet Explorer используйте условные комментарии (см. *разд. 12.7*).

Итак, PNG-изображение отображается как фон за прозрачным GIF-изображением.

#### Примечание

Изображения в формате PNG24 с полной альфа-прозрачностью, как правило, формируют файлы довольно большого размера. Для решения этой проблемы разработан оптимизатор Pngcrush (http://pmt.sourceforge.net/pngcrush/), который можно выполнять в окне MS-DOS или в командной строке UNIX. PNGThing, маленькое и легкое в использовании приложение Mac, основанное на этом коде, можно найти по адресу http://mac.softpedia.com/ Graphics/PNGThing.shtml.

#### См. также

Оригинальную публикацию скрипта Sleight на странице http://www.youngpup.net/ 2001/sleight; подробную информацию о свойстве filter корпорации Microsoft на странице http://msdn.microsoft.com/en-us/library/ms532967(VS.85,classic).aspx.

# 4.19. Наложение HTML-текста на изображение

# Проблема

Вы хотите поместить HTML-текст поверх изображения.

# Решение

Поместите изображение в фон и затем позиционируйте HTML-текст и задайте ему соответствующее стилевое оформление.

Сначала включите текст в элемент div с атрибутом id:

```
<div id="frame">
<div id="banner">
<h1>White House Confidential <br /><span>
Classified Lawn Care Secrets</span></h1>
</div><!-- end #banner -->
...
</div>
```

Вставьте изображение с помощью свойства background-image с обязательным указанием ширины и высоты:

```
#banner {
  width: 550px;
  height: 561px;
  overflow: hidden;
  background-image: url(whitehouse.jpg);
  background-position: 0;
  background-repeat: no-repeat;
  position: relative
}
```

Затем задайте желаемый стиль текста (рис. 4.21):

```
h1 {
  margin: 0;
  padding: 0;
  font-family: Verdana, Arial, sans-serif;
  margin-top: 325px;
  margin-left: 25px;
  position: absolute;
  bottom: 0;
  color: white;
  text-shadow: 0 1px 0 #666;
  text-align: center;
  border-left: 2px solid #666;
  border-right: 2px solid #666;
  border-top: 2px solid #666;
  /* свободное пространство вокруг текста */
```

```
padding-left: 25px;
```

```
/* вставьте полупрозрачное фоновое изображение */
background-image: url(white-banner.png);
background-position: bottom;
background-repeat: no-repeat;
}
h1 span {
font-size: .8em;
}
```



Рис. 4.21. Наложение текста на фотографию

## Обсуждение

Вместо того чтобы встраивать в текст изображение, у которого сугубо декоративное назначение, или делать его частью информационного контента Web-страницы, используйте для вывода изображения свойство background-image. Такой подход сделает страницу более доступной, но при этом сохранит задуманный визуальный ряд.

## См. также

Разд. 4.20 о замещении HTML-текста изображением.

# 4.20. Замещение HTML-текста изображением

# Проблема

Вы хотите заменить HTML-текст (например, заголовок) изображением с более интересным визуальным и типографическим представлением.

# Решение

Воспользуйтесь методом Гилдера — Левина (Gilder/Levin) для замещения текста изображением.

Сначала вставьте элемент span перед HTML-текстом:

```
<h1>
<span></span>
Replacement Text Is Here
</h1>
```

Задайте ширину и высоту замещающего изображения в селекторе h1, а также относительное позиционирование элемента:

```
h1 {
  width: 216px;
  height: 72px;
  position: relative;
}
```

Далее, благодаря установке абсолютного позиционирования элемента span и настройке ширины и высоты элемента span в элементе h1, span накладывается на HTML-текст. Последний шаг — вставка замещающего изображения с помощью свойства background (рис. 4.22):

```
h1 span {
   background: url(replacementimage.jpg) no-repeat;
   position: absolute;
   width: 100%;
   height: 100%;
}
```

# Обсуждение

В Web-разработке существует несколько методов замещения изображением, и у них всех есть собственные достоинства и недостатки.

#### Метод Фарнера замещения текста изображением

Тодд Фарнер (Todd Fahrner) — один из тех, кому приписывают разработку первоначальной концепции метода замещения текста изображением.

В разметку для метода Фарнера (Farhner Image Replacement, FIR) включается формальный элемент span, предназначенный только для вставки в него контента:

#### <h1>

<span>

Replacement Text Is Here

#### </span>

</h1>



Рис. 4.22. HTML-текст, замещенный графикой

Затем правила CSS вставляют замещающее изображение через селектор элемента h1, скрывая при этом текст:

```
h1 {
   background: url(replacementimage.jpg) no-repeat;
   width: 216px;
   height: 72px;
}
h1 span {
   display: none;
}
```

**Проблемы FIR-метода.** Легкая реализация FIR-метода сделала его очень популярным в Web-разработке. Однако экранные считыватели (screen reader), применяемые людьми с ограниченными физическими возможностями, часто будут пропускать считывание HTML-текста, поскольку элемент span отменяет отображение текста на экране. Таким образом, важный текст будет потерян для этой части аудитории сайта.

#### Phark-метод замещения текста изображением

И FIR-метод, и метод замещения текста изображением Гилдера — Левина для достижения результатов используют искусственный тег span. Другой метод замещения текста изображением, созданный Майком Рандлом (Mike Rundle) из Phark.net, не нуждается в теге span.

Сначала откорректируйте HTML-код, удалив тег span:

```
<hl>
Replacement Text Is Here
</hl>
```

В правилах CSS задайте отрицательное значение свойства text-indent вместо применения свойства display для скрытия текста:

```
h1 {
   text-indent: -9000em;
   background: url(replacementimage.jpg) no-repeat;
   width: 216px;
   height: 72px;
}
```

**Проблемы Phark-метода.** Как и другие методы, Phark-метод очень хорошо работает. Его главный недостаток — отсутствие HTML-текста на экране при отключении пользователем вывода изображений в окне своего Web-обозревателя.

#### Подход CSS3 к замещению текста изображением

В спецификации CSS3 при условии ее технической поддержки в Web-обозревателях есть легкий способ замещения текста изображением. Например, для замены текста в элементе h1 таблица CSS будет состоять из одного блока объявления:

```
h1 {
  content: url(logo.gif);
}
```

Спецификация также не накладывает ограничений на тип мультимедийной информации, который может поддерживаться свойством content. Теоретически Web-разработчик мог бы на место анимационного GIF-изображения поместить фильм в формате Quick Time:

```
h1 {
  content: url(logo.mov);
}
```

#### Примечание

Во время написания книги в современных Web-обозревателях не было поддержки этой части спецификации CSS3.

## См. также

Сведения о вставке контента в спецификации CSS3 на страницах http:// www.w3.org/TR/css3-content/#inserting3 и http://my.opera.com/ODIN/blog/css-3image-replacement.

# 4.21. Создание панорамного представления изображения

# Проблема

Вы хотите создать представление изображения, ширина которого меняется при изменении размеров окна Web-обозревателя, как показано на рис. 4.23.



Рис. 4.23. Отображение более широкой панорамы при увеличении окна Web-обозревателя

## Решение

Поместите элемент, ссылающийся на панорамное изображение, в фон элемента блочного уровня:

```
<hl>Visit France City!</hl>
<div><img src="frenchtown.jpg" alt=" " /></div>
<h2>The quaint and charming little destination in France</h2>
```

Поместите элемент с изображением в правый верхний угол блочного элемента и затем скройте изображение с помощью свойства display со значением none:

```
div {
  background-image: url(frenchtown.jpg);
  background-repeat: no-repeat;
  background-position: top right;
  height: 300px;
  border: 1px solid black;
  max-width: 714px;
  }
  div img {
   display: none;
  }
}
```

Если изображение помещается как фоновое, его размеры будут меняться в соответствии с размерами окна Web-обозревателя.

## Обсуждение

Для создания панорамного представления вам потребуется широкая фотография. Затем вы должны поместить элемент изображения в правый верхний угол элемента блочного уровня, чтобы изображение растягивалось или сжималось в зависимости от размеров окна Web-обозревателя. Применение свойства max-width ограничивает ширину элемента div, не давая ей становиться больше ширины самого изображения.

В приведенном решении одно и то же изображение используется и в HTML-коде, и в таблице CSS. Основная причина такого подхода заключается в гарантии вывода изображения (контента) даже в том случае, когда пользовательский агент, визуализирующий страницу, не поддерживает технологию CSS.

## См. также

Свойство max-width в спецификации CSS 2.1 на http://www.w3.org/TR/CSS21/ visudet.html#propdef-max-width.

# 4.22. Комбинирование разных графических форматов

## Проблема

Вы хотите скомбинировать два изображения с разными форматами в одной картинке. Например, вы хотите объединить GIF- и JPEG-изображение в одном графическом представлении (рис. 4.24).



Рис. 4.24. Два изображения с разными форматами, объединенные в одно

#### Решение

Поместите изображение внутрь элемента блочного уровня, такого как div или h2:

```
<h2><img src="headline_text.gif" alt="Headline image set in GIF format" /></h2>
```

Воспользуйтесь графическим редактором для разделения изображения на файлы разных форматов (рис. 4.25).



Рис. 4.25. Два изображения, которые будут использованы для создания единого изображения

Одно из изображений назовите так же, как изображение, упомянутое в атрибуте src элемента img. Для объединения двух изображений в одну картинку поместите второе изображение в фон элемента блочного уровня:

```
h2 {
  background-image: url(headline_bkgd.jpg);
  background-repeat: none;
  width: 587px;
  height: 113px;
}
```

## Обсуждение

Два преобладающих в Web-пространстве формата — GIF и JPEG (и PNG, создающий устойчивую тройку). Оба разными методами сжимают файлы изображений. Обычно изображения с однотонными цветовыми областями сжимаются лучше в GIF-формате, в то время как JPEG-формат больше подходит для фотографий или изображений, содержащих тонкие оттенки цвета.

В примере, показанном на рис. 4.24 и 4.25, размер файлов двух отдельных изображений в сумме меньше размера файла окончательного комбинированного изображения. Это происходит потому, что та или иная часть изображения плохо поддается методу сжатия одного из форматов. Если вы сохранили окончательное изображение как один GIF-файл, фотофрагменты изображения создают завышенный размер файла. А если вы сохранили комбинированное изображение как JPEG-файл, завышать размер файла будут однотонные области. Разделив изображения на два разных формата, выгодно использующих соответствующие методы сжатия, вы уменьшите общий размер файлов.

Несмотря на то, что в данном решении используются свойства CSS background, вы можете достичь того же эффекта, позиционируя блочные элементы, содержащие встроенные изображения. Например, на рис. 4.26 можно увидеть, что штриховой рисунок лодки был наложен на фотографию двух мальчиков.



Рис. 4.26. Замысловатая комбинация двух изображений разных форматов

Для того чтобы предложенный метод заработал, включите элементы image в элементы div блочного уровня, как показано в следующем HTML-коде:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
    <html xmlns="http://www.w3.org/1999/xhtml">
    <html xmlns="http://www.w3.org/1999/xhtml">
</normal>
```

```
<title>CSS Cookbook</title>
</head>
<body>
<img src="kids.jpg" width="360" height="304" alt="kids
playing" />
<div id="boat"><img src="boat.gif" width="207" height="123"
alt="boat" /></div>
<div id="water"><img src="landscape.gif" width="315"
height="323"
alt="landscape" /></div>
</body>
</html>
```

Затем, с помощью CSS установите значение absolute для свойства элементов position. Задав значение absolute в свойстве position, вы удаляете элементы из нормального потока Web-страницы и вместо этого для определения их нового местоположения задаете значения свойств left, top и z-index:

```
#boat {
position: absolute;
width: 207px;
height: 123px;
 z-index: 2;
 left: 264px;
 top: 0;
}
#water {
position: absolute;
width: 315px;
height: 323px;
 z-index: 1;
left: 359px;
top: -20px;
}
```

Свойства left и top указывают местоположение изображений в пределах ближайшего к ним родительского элемента или начального вмещающего блока. В данном случае это исходный вмещающий блок для элементов div. К тому же у свойства margin элемента body значение равно 0, указывая на то, что начальная точка находится в левом верхнем углу окна просмотра Web-обозревателя:

```
body {
  margin: 0;
}
```

Несмотря на то что этот метод работает, в случае если Web-документ будет позже изменен, точное позиционирование станет источником неприятностей. Например, вставка в HTML-код простого заголовка над изображениями в результате приводит к аномалии, показанной на рис. 4.27:

```
<h2>Kids Welcome New Boat!</h2>
<img src="kids.jpg" width="360" height="304" alt="kids
```

```
playing" />
    <div id="boat"><img src="boat.gif" width="207" height="123"
        alt="boat" /></div>
        <div id="water"><img src="landscape.gif" width="315" height="323"
        alt="landscape" /></div>
```



Рис. 4.27. Представление нарушается при вставке строки заголовка

Поскольку изображение детей не позиционировалось с помощью значения absolute, оно сдвигается вниз в потоке документа. Другое изображение остается на прежнем месте, т. к. оно позиционировано в исходном вмещающем блоке и остается там же, где было до вставки строки заголовка.

Применяя метод позиционирования фона (background-positioning) в пределах элементов блочного уровня, вы можете создать изолированный (self-contained) модуль. При этом, когда контент вставляется в Web-страницу или удаляется из нее, комбинированное изображение остается целым, как показано на рис. 4.28 и в приведенном далее коде:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>CSS Cookbook</title>
<style type="text/css">
body {
    margin: 5% 10% 0 10%;
    }
#content {
    background-image: url(landscape.gif);
    background-repeat: no-repeat;
}
```



Рис. 4.28. Другой подход к комбинированию изображений

```
background-position: bottom right;
height: 400px;
width: 674px;
}
h2 {
margin: 0;
padding: 0;
background-image: url(kids.jpg);
background-repeat: no-repeat;
background-position: bottom left;
height: 400px;
width: 600px;
}
#boat {
background-image: url(boat.gif);
background-repeat: no-repeat;
display: block;
width: 207px;
height: 123px;
margin-left: 250px;
margin-top: 75px;
}
</style>
 </head>
 <body>
  <div id="content">
   <h2>Kids Welcome New Boat!
    <span id="boat">
    </span>
```

</h2> </div> </body> </html>

#### См. также

*Разд. 13.2* о создании неожиданного несоответствия между двумя элементами; *разд. 13.3* о комбинировании непохожих элементов.

# 4.23. Скругленные углы колонок с фиксированной шириной

## Проблема

Вы хотите создать скругленные углы в колонках с фиксированной шириной.

#### Решение

Создайте два фоновых изображения: одно, содержащее верхние углы, а другое — нижние (рис. 4.29).

Включите информационное содержимое колонки в элемент div:

```
<div id="box">
    <h2>
        I Met a Girl I&#8217;d Like to Know Better
        </h2>
        Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam
nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.
Ut wisi enim ad minim veniam.
</div>
```

Поместите нижнее фоновое изображение в элемент div:

```
#box {
  width: 214px;
  background-image: url(bkgd_bottom.gif);
  background-position: bottom;
  background-repeat: no-repeat;
}
```

Затем поместите верхнее фоновое изображение в элемент h2 (рис. 4.30):

```
h2 {
    background-image: url(bkgd_top.gif);
    backgroung-position: left top;
    background-repeat: no-repeat;
    padding: 7px 7px 3px 7px;
    margin: 0;
```

```
border-bottom: 1px solid #999;
font-size: 1.3em;
font-weight: normal;
color: #eee;
}
```



для верхних углов, а другое для нижних

в верхнюю и нижнюю части колонки

## Обсуждение

Для компенсации изменений при разных размерах шрифта сделайте изображения больше величины, заданной в описании страницы. Например, высота изображений, использованных в данном решении, равна 600 пикселам, однако нельзя считать чем-то необычным высоты графических элементов более 1 000 пикселов, гарантирующие сохранение целостности дизайна страницы при использовании очень крупных шрифтов.

#### Переменная ширина

Установив ширину колонки в единицах длины, таких как пикселы, можно создать одно изображение, содержащее два угла. Если ширина колонки будет меняться, когда пользователь меняет размеры окна Web-обозревателя, решение, предложенное для колонок с фиксированной шириной, не годится.

## См. также

Разд. 4.24—4.26 о скруглении углов в колонках с переменной шириной.

# 4.24. Скругленные углы (метод раздвижных дверей)

## Проблема

Вы хотите скруглить углы в колонках с переменной шириной.

## Решение

Воспользуйтесь методом раздвижных дверей (Sliding Doors), ставшим популярным благодаря Web-дизайнеру Дугласу Боуману (Douglas Bowman).

100%

Corner\_tl.gif..

4 +

36 72 108 1

O torner\_bl.gi... 36 72 108 1

Создайте макет со скругленными углами (рис. 4.31).







001

R \varTheta \varTheta 🧑 🗟 corner\_br.gi...

36

36

100%

corner\_tr.gif ...

0 36

0 36

Рис. 4.32. Графический макет колонки разделен на четыре изображения

Затем создайте отдельные графические представления для каждого из четырех углов, такие как показанные на рис. 4.32.

Заключите информационное содержимое колонки в дополнительные элементы div:

```
<div id="box">
<div id="innerhead">
 <h2>
  I Met a Girl I'd Like to Know Better
 </h2>
</div>
```

#### <div id="content">

#### <div id="innercontent">

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam.

#### </div>

#### </div>

</div>



Рис. 4.33. Колонка выводится со скругленными углами

Затем с помощью таблицы стилей CSS вставьте фоновые изображения (рис. 4.33). Верхний левый угол помещается в селектор ID с именем innerhead, правый верхний угол соскальзывает в уже существующий элемент h2, селектор ID с именем content получает левый нижний угол, и селектор ID innercontent приютит рисунок правого нижнего угла.

```
#innerhead {
  background-image: url(corner_tl.gif);
  background-position: top left;
  background-repeat: no-repeat;
}
h2 {
  background-image: url(corner_tr.gif);
  background-position: top right;
  background-repeat: no-repeat;
```

```
margin: 0;
padding: 7px;
border-bottom: 1px solid #999;
font-size: 1.3em;
font-weight: normal;
color: #eee;
1
#content {
background-image: url(corner bl.gif);
background-position: bottom left;
background-repeat: no-repeat;
}
#innercontent {
background-image: url(corner br.gif);
background-position: bottom right;
background-repeat: no-repeat;
}
```

## Обсуждение

Элементы div и h2 используются как средства для вставки фоновых изображений всех четырех углов колонки. Если окно Web-обозревателя меняется, фоновые изображения остаются в соответствующих углах, как показано на рис. 4.34.



Рис. 4.34. Скругленные углы сохраняются при расширении колонки

Для того чтобы наверняка сохранить целостность дизайна при расширении колонки, потребуется дальнейшее редактирование изображения. Используйте одну из его боковых сторон (левую или правую) и растяните оба графических представления углов как



Рис. 4.35. Графическое представление для верхнего левого угла шириной 600 пикселов и высотой более 250 пикселов



Рис. 4.36. Графическое представление для нижнего левого угла шириной 600 пикселов и высотой 600 пикселов

по вертикали, так и по горизонтали. Например, в данном решении увеличены изображения для верхнего левого и нижнего левого углов (рис. 4.35 и 4.36).

## См. также

В разд. 4.25 простое решение для скругления углов колонки.

# 4.25. Скругленные углы (метод горной вершины)

# Проблема

Вы хотите создать один набор графических файлов для скругления углов и иметь возможность отображать разные цвета фона в колонке.

## Решение

Воспользуйтесь методом горной вершины (Mountaintop), популяризованным Webдизайнером Дэном Седерхольмом (Dan Cederholm).

Создайте маленький графический рисунок, который будет служить основой для скругления углов (рис. 4.37).

#### Примечание

Учтите, что черный цвет на рис. 4.37 будет установлен прозрачным при экспортировании изображения в GIF-файл.

Экспортируйте изображение в GIF-файл с именем corner\_tl.gif.

Затем поверните изображение на 90° (рис. 4.38) и экспортируйте это изображение как GIF-файл с именем corner\_tr.gif. Повторите два последних шага для создания нижних углов в файлах corner\_br.gif и corner\_bl.gif.

Вставьте информационное наполнение колонки в дополнительные элементы div:

```
<div id="box">

<div id="head_outer">

<div id="head_inner">

<h2>

I Met a Girl I'd Like to Know Better

</h2>

</div>

</div>

Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy

nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim

ad minim veniam.
```

Далее поместите четыре графических представления углов в селекторы ID и селектор элемента р (рис. 4.39):



Corner\_tl co...

Рис. 4.37. Рисунок верхнего левого угла

Рис. 4.38. Поворот изображения на 90°



Рис. 4.39. Пример использования метода горной вершины

Изображения

```
div#box {
 width: 55%;
 background-color: #999999;
background-image: url(corner bl.gif);
 background-repeat: no-repeat;
background-position: bottom left;
#head outer {
background-image: url(corner tl.gif);
background-repeat: no-repeat;
}
#head inner {
background-image: url(corner tr.gif);
background-repeat: no-repeat;
background-position: top right;
}
div p {
margin: 0;
padding: 7px;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: 1.1em;
background-image: url(corner br.gif);
background-position: bottom right;
background-repeat: no-repeat;
color: #333333;
font-size: .8em;
line-height: 1.5;
}
```

# Обсуждение

Красота метода горной вершины кроется в его простоте. Четыре маленьких изображения, хранящиеся, благодаря методам сжатия GIF-формата, в четырех файлах малого размера, помещаются в фон четырех элементов блочного уровня.

Кроме того, нет нужды растягивать пару из них для поддержки целостности дизайна при изменении размера колонки, как в решении из *разд. 4.24*.

К тому же метод горной вершины позволяет быстро изменять цвет фона колонки без корректировки графических файлов углов, как показано на рис. 4.40. Тем не менее графические представления углов придется откорректировать, если изменится цвет фона Web-страницы или родительского элемента колонки.

## См. также

*Разд. 4.26* об автоматическом скруглении углов колонки без использования созданных пользователем изображений.



Рис. 4.40. Колонка сохраняет целостность, даже если меняется ее цвет или размер

# 4.26. Скругление углов с помощью JavaScript

#### Проблема

Вы хотите вставлять скругленные углы элементов без затрат времени на новую разметку и создание изображений вручную.

## Решение

Воспользуйтесь кодом Nifty Corners Cube (куб со стильными углами) Алессандро Фулчинити (Alessandro Fulciniti).

Сначала загрузите со страницы http://www.html.it/articoli/niftycube/index.html компоненты решения Nifty Corners Cube, включающие один файл CSS и один файл JavaScript.

Затем свяжите файл JavaScript с Web-страницей с помощью атрибута src элемента script:

<script type="text/javascript" src="/\_assets/js/niftycube.js"></script>

#### Примечание

Вы не должны ссылаться непосредственно на файл CSS, поскольку это сделает файл JavaScript.

Далее измените разметку, в которой должны появиться скругленные углы, вставив уникальное значение в атрибут id:

```
<div id="box">
<h2>Marquee selectus</h2>
...
</div>
```

Затем сделайте отдельный вызов в Web-обозревателе функции на JavaScript, чтобы указать элемент, который получит скругленные углы, и определить размер скругленных углов (рис. 4.41):

```
<script type="text/javascript" src="niftycube.js"></script>
<script type="text/javascript">
window.onload=function() {
   Nifty("div#box", "big");
}
```

</script>



Рис. 4.41. Скругленные углы (слева) и стандартное отображение колонки (справа)

## Обсуждение

Поскольку этот метод создания скругленных углов не требует практически никаких усилий, решение Nifty Corners Cube следовало бы назвать скорее инструментом, чем методом.

#### Примечание

Данное решение основано на JavaScript. Если в Web-обозревателе пользователя нет JavaScript или он отключен, скругленные углы не появятся.

#### Разные цвета

Цвета определяются автоматически. Функция на JavaScript автоматически изменяет цвета в соответствии с цветом фона элемента и его родительского элемента (обычно элемента body Web-страницы). Это означает, что разработчик должен побеспокоиться только о том, чтобы определить, какому элементу передаются кривые и размер.

#### Разные размеры

Есть четыре ключевых слова для задания размера, которые указываются при вызове инструмента Nifty Corners Cube на JavaScript: none, small, normal (по умолчанию) и big. Размер small равен 2 пикселам, normal — 5 пикселам, а big — 10 пикселам.

Например, для установки углов с размером small вызов функции на JavaScript мог бы выглядеть следующим образом:

```
<script type="text/javascript">
window.onload=function() {
  Nifty("div#box", "small");
}
</script>
```

#### Разные элементы

Инструмент Nifty Corners Cube принимает разные селекторы, облегчая указание элемента, у которого следует скруглить углы (табл. 4.3).

Селектор	Пример
Селектор типа	"div"
	"h3"
Селектор ID	"div#box"
	"h3#main"
Селектор класса	"div.box"
	"h3.box"
Потомок селектора ID	"div#box h3"
	"h3#main div"
Потомок селектора класса	"div.box h3"
	"h3.main div"
Группировка	"div, h3"
	"div, h3.main div, p"

Таблица 4.3. Селекторы, распознаваемые Nifty Corners Cube на JavaScript

Например, при установке скругленных углов в нескольких элементах функция на JavaScript может выглядеть следующим образом:

```
<script type="text/javascript">
window.onload=function() {
   Nifty("div, h3.main div, p", "small");
}
</script>
```

#### Отдельные углы

Метод Nifty Corners Cube могут применять и те разработчики, кто, возможно, не хочет создавать скругленные края у всех углов. В табл. 4.4 перечислены ключевые слова, позволяющие выделить тот угол или углы, которые надо скруглить.

Ключевое слово	Значение
tl	Верхний левый угол
tr	Верхний правый угол
bl	Нижний левый угол
br	Нижний левый угол
top	Верхние углы
bottom	Нижние углы
left	Левые углы
right	Правые углы
all (default)	Все углы

Таблица 4.4. Ключевые слова, распознаваемые инструментом Nifty Corners Cube

Например, для скругления только верхних углов у нескольких элементов на Webстранице вызов функции JavaScript мог бы выглядеть следующим образом:

```
<script type="text/javascript">
window.onload=function() {
  Nifty("div, h3.main div, p", "small top");
}
</script>
```

#### Примечание

В настоящее время есть варианты предложенного решения для разных библиотек JavaScript. Вы можете найти одно такое решение для jQuery по адресу http:// www.malsup.com/query/corner/.

#### См. также

Дополнительную информацию о методе Nifty Corners Cube на странице http://www.html.it/articoli/niftycube/index.html.
# 4.27. Вставка тени от элемента средствами CSS

# Проблема

Вы хотите поместить на элемент тень-рамку средствами CSS.

# Решение

Примените свойство box-shadow вместе со специализированными свойствами CSS, разработанными поставщиками Web-обозревателей (рис. 4.42):

```
#header {
    min-width: 250px;
    text-shadow: 0 -1px 0 rgba(0,0,0,.8);
    box-shadow: 3px 3px 19px rgba(0,0,0,.8);
    -webkit-box-shadow: 3px 3px 19px rgba(0,0,0,.8);
    moz-box-shadow: 3px 3px 19px rgba(0,0,0,.8);
    background-image: -webkit-gradient(linear, left top, left bottom, from(#930),
    to(#6b3703), color-stop(0.5, #903000));
    background-image: -moz-linear-gradient(left top, left bottom,
    from(rgba(153,51,0,.3)), to(#6b3703), color-stop(0.5, #903000));
    margin: 7px;
    padding: 14px;
}
```



Рис. 4.42. Элемент с отбрасываемой тенью

## Обсуждение

Тени-рамки действуют так же, как тени от текста (см. разд. 3.32).

Первое значение в свойстве определяет расстояние по оси X, а второе — по оси Y. Положительные значения указывают на расположение тени справа и снизу соответственно. Отрицательные значения помещают тень слева и сверху.

Третье значение определяет радиус размытия тени.

Четвертое значение задает цвет тени. В данном решении цвет задан с помощью функции RGBA, допускающей прозрачность. Такой подход к заданию цвета (хотя и поддерживаемый не всеми Web-обозревателями) обеспечивает более плавный переход к мозаичному фону.

## См. также

*Разд. 4.28* о методе вставки изображения, работающем в разных Web-обозревателях; описание свойства box-shadow в спецификации CSS3 на странице http://www.w3.org/ TR/css3-background/#the-box-shadow.

# 4.28. Вставка тени за изображением

# Проблема

Вы хотите поместить отбрасываемую тень за изображением (рис. 4.43).



Рис. 4.43. Тень, расположенная за изображением

## Решение

Поместите элемент img (рис. 4.44) внутрь элемента div с атрибутом class, имеющим значение imgholder:

```
<div class="imgholder">
<img src="dadsaranick2.jpg" alt="Photo of Dad, Sara, Nick" />
</div>
```



Рис. 4.44. Стандартная визуализация изображения

В элементе div задайте выравнивание изображения left, так чтобы текст обтекал изображение. Далее задайте фоновое изображение падающей тени в двух свойствах background. В первом свойстве background используйте изображение с альфа-прозрачностью, например, в формате PNG:

```
div.imgholder {
  float:left;
  background: url(dropshadow.png) no-repeat bottom
  right !important;
  background: url(dropshadow.gif) no-repeat bottom right;
  margin: 10px 7px 0 10px !important;
  margin: 10px 0 0 5px;
}
```

Что касается самого изображения, задайте свойства margin-right и margin-bottom для того, чтобы определить, какая доля изображения тени будет видна. Задайте также свойства border и padding для создания более яркого визуального эффекта:

```
div.imgholder img {
  display: block;
  position: relative;
  background-color: #fff;
  border: 1px solid #666;
  margin: -3px 5px 5px -3px;
  padding: 2px;
}
```

# Обсуждение

Первый шаг — создание изображения тени в программе редактирования графических файлов, такой как Adobe Photoshop. Лучше всего создать фоновое изображение размером 600×600 пикселов или больше (рис. 4.45). Если изображение тени большого размера, описываемый метод может приспособить его практически к любому изображению, применяемому на Web-странице.



Рис. 4.45. Тень видна справа и снизу

Первое свойство background изображения использует правило !important для отображения PNG-файла как отбрасываемой тени. Применяя формат PNG, можно менять цвет фона или изображения Web-документа, не затрагивая изображения тени. В случае применения других Web-обозревателей, не поддерживающих это правило, например Internet Explorer для Windows, перейдите к следующему свойству background и используйте вместо этого GIF-изображение как отбрасываемую тень. Свойства margin-left и margin-bottom элемента img управляют расстоянием, на которое выступает изображение тени за изображение переднего плана. Если на вашей странице величина тени слева и справа больше пяти пикселов (используемых в данном решении), измените это значение.

### См. также

Разд. 4.29 о создании теней с плавными цветовыми переходами.

# 4.29. Вставка сглаженной тени за изображением

## Проблема

Вы хотите сформировать нерезкие границы в изображении тени.

## Решение

Вставка еще одного формального элемента-оболочки div, охватывающего еще одно фоновое изображение, позволяет создать плавные границы тени.

Сначала создайте новое изображение в программе Adobe Photoshop, которое будет действовать как маска для сглаживания изображения отбрасываемой тени. Используя те же размеры, что и у изображения тени, удалите в файле все графическое наполнение, оставив только прозрачный фон.

### Примечание

Если у вас нет доступа к Adobe Photoshop, попробуйте версию онлайн на сайте https://www/photoshop/com/ или загрузите бесплатный графический редактор, например, GIMP (http://www.gimp.org/).

Затем в инструменте Gradient (Градиент) выберите вариант градиента, создающий плавный переход от установленного цвета фона к прозрачному (Background Color to Transparent) (рис. 4.46).

Убедитесь в том, что цвет фона на панели инструментов совпадает с цветом фона, используемом на Web-сайте, создайте шестипиксельный переход в направлении от левого края канвы к правому краю изображения.

Повторите создание перехода, но теперь создайте переход в направлении от верхнего края канвы к ее нижнему краю.

Далее сохраните изображение как файл PNG-24 с прозрачностью (рис. 4.47).

Подготовив изображения, откорректируйте HTML-код, включив в него новый элементоболочку div:

```
<div class="imgholder">
    <div>
    <img src="dadsaranick2.jpg" alt="Photo of Dad, Sara, Nick" />
    </div>
</div>
```

		Gradier	nt Editor				
Presets					OK Cancel Load Save		
ame: Gradie Smool	Foregrounent Type:	nd to Transpare Solid	nt			Nev	v
Line (tra							
Sto	pacity:	1 % Los	ation:	7	0	oloto	5

Рис. 4.46. Выбор нужного градиентного перехода



Рис. 4.47. Сохранение изображения как PNG-файла с альфа-прозрачностью

Откорректировав элемент-оболочку CSS первого изображения, примените плавающую модель, переместите изображение влево, используйте тень и задайте отступы между изображением и HTML-контентом:

```
div.imgholder {
  float: left;
  background: url(dropshadow.gif) no-repeat bottom right;
  margin: 0 7px 7px 0;
}
```

Далее введите маску, которая сгладит фон тени, и выделите место для отображения тени и маски (рис. 4.48):

```
div.imgholder div {
  background: url(shadowmask.png) no-repeat;
  padding: 0 6px 6px 0;
}
```



Рис. 4.48. Отбрасываемые тени со сглаженными краями

Наконец, задайте некоторый отступ и рамку для изображения (рис. 4.49):

```
div.imgholder img {
    display: block;
    position: relative;
```

}

```
background-color: #fff;
border: 1px solid #666;
padding: 2px;
```



Рис. 4.49. Изображение с тенью и дополнительным стилевым оформлением

## Обсуждение

Трудная часть предложенного решения — создание PNG-файла с альфа-прозрачностью, который взаимодействует с изображением тени и согласуется с цветом фона Web-сайта.

Поскольку Internet Explorer 5.5—6 для Windows не имеет собственной поддержки изображений с альфа-прозрачностью в формате PNG, воспользуйтесь решением, приведенным в *разд. 4.18*.

### См. также

Разд. 4.27 о создании простой тени за изображением.

# 4.30. Создание выносок с текстом

# Проблема

Вы хотите создать выноску с текстом, как показано на рис. 4.50.



Рис. 4.50. Выноска с прямой речью

# Решение

Разметьте информационное наполнение выноски и включите в нее как текст, который появится внутри выноски, так и имя автора этих слов (рис. 4.51):

```
<blockquote>
<span>
Be bold, baby!
</span>
<cite>
Christopher Schmitt
</cite>
</blockquote>
```



Рис. 4.51. Структурированный контент для выноски с прямой речью

Оформите выноску с помощью свойств CSS border и background. Затем выровняйте текст с именем автора так, чтобы он опустился ниже хвостика контура:

```
blockquote {
width: 250px;
}
blockquote p {
background: url(balloontip.gif);
background-repeat: no-repeat;
background-position: bottom;
padding-bottom: 28px;
}
blockquote p span {
display: block;
padding: 0.25em 0.25em 0.5em 0.5em;
border: 1pt solid black;
border-bottom-width: 0;
 font-size: 3em;
 font-family: "Comic Sans MS", Verdana, Helvetica, sans-serif;
 line-height: 0.9em;
}
cite {
 text-align: right;
display: block;
width: 250px;
}
```

# Обсуждение

Для создания выноски вам понадобится как минимум одно изображение с хвостиком контура и одной его стороной (рис. 4.52). Это изображение можно загрузить с Webсайта данной книги, http://csscookbook.com/. Три остальные стороны контура выноски создайте с помощью задания рамки для тега span.



Рис. 4.52. Хвостик для выноски

Для того чтобы ваш дизайн напоминал комикс, убедитесь, что в свойстве font-family задан свободно распространяемый шрифт корпорации Microsoft Comic Sans MS:

font-family: "Comic Sans MS", Verdana, Helvetica, sans-serif;

Если ваш компьютер работает под управлением OC Windows, возможно, на нем уже установлен этот шрифт. Несмотря на то, что это распространенный шрифт, некоторые пользователи могли не установить его на свои системы. В данном случае Webобозреватель будет искать шрифт, следующий по порядку в списке названий шрифтов, пока не найдет имеющийся в наличии шрифт для визуализации страницы.

Вы можете создать более интересное отображение с использованием выносок, слегка подкорректировав разметку и таблицу CSS. Сначала поместите в элемент span с атрибутом class, имеющим значение no, имя из элемента cite:

Глава 4

```
<blockquote>
<span>
Be bold, baby!
</span>

<cite>
<span class="no">
Christopher Schmitt
</span>
</cite>
</blockquote>
```

Далее в таблицу стилей CSS добавьте следующее правило, которое помешает выводить текст в окне Web-обозревателя:

```
.no {
  display: none;
}
```



Рис. 4.53. Выноска с прямой речью персонажа, изображенного на фотографии

Поместите фотографию в элемент cite с помощью свойства background-image для завершения создания нужного эффекта (рис. 4.53):

```
cite {
  margin: 0;
  padding: 0;
  background-image: url(baby.jpg);
```

#### 262

```
background-position: 0 0;
height: 386px;
text-align: right;
display: block;
width: 250px;
}
```

## См. также

Сопроводительную информацию о шрифте Comic Sans MS на странице http://www.microsoft.com/typography/web/fonts/comicsns/default.htm; пропагандистский Web-сайт, призывающий к отказу от использования шрифта Comic Sans MS по адресу http://www.bancomicsans.com.

# 4.31. Как помешать людям похищать ваши изображения

# Проблема

Вы хотите затруднить для всех копирование изображений с вашей Web-страницы.

# Решение

Используйте однопиксельный прозрачный GIF-файл как заполнитель (place marker), включите тег img в элемент div:

```
<div class="slide">
<img src="singlepixel.gif" alt=-"" />
</div>
```

Затем вставьте в Web-страницу изображение с помощью свойства background-image и обязательно задайте ширину и высоту изображения в элементах, как div, так и img:

```
div.slide {
  width: 500px;
  height: 468px;
  background-image: url(face.jpg);
  background-repeat: no-repeat;
  }
  .slide img {
  width: 500px;
  height: 468px;
  }
```

# Обсуждение

Для отображения нет необходимости использовать однопиксельный GIF-файл как заполнитель; действительно вы можете совсем избавиться от элемента img и попрежнему сохранить отображение исходного изображения:

```
<div class="slide">
</div>
```

Назначение прозрачного изображения — служить ложным источником. Пользователи будут считать, что они загружают понравившееся им изображение, а на самом деле будут загружать безобидную, замещающую его картинку.

### Панель инструментов Image корпорации Microsoft

В Internet Explorer 6 для Windows корпорация Microsoft начала включать средство, именуемое панелью инструментов Image (Изображение).

С помощью этого средства посетитель вашего сайта легко может послать по электронной почте, загрузить на свой компьютер или напечатать ваше изображение простым щелчком кнопкой мыши. Для того чтобы помешать появлению этой панели инструментов на ваших Web-страницах, добавьте следующие метатеги между элементами head:

```
<meta http-equiv="imagetoolbar" content="no" /> <meta http-equiv="imagetoolbar" content="false" / >
```

Разработчикам нетрудно вставить код на их Web-страницы, чтобы помешать чьемулибо программному продукту похищать ваши изображения, но разработчик мало что может сделать, т. к. Microsoft производит самый популярный Web-обозреватель.

### Полностью защищенных изображений нет

Даже применив предложенное решение и обойдя вывод панели инструментов, нельзя полностью обезопасить ни одно изображение от копирования с вашего Web-сайта на компьютер пользователя.

Прежде всего, изображения автоматически запоминаются Web-обозревателем посетителя и сохраняются во временной папке для быстрой перезагрузки Web-страниц. Эти кэшированные изображения стандартно удаляются спустя определенный промежуток времени или в любой момент, когда пользователь очищает кэш своего обозревателя.

Однако Web-обозреватель часто автоматически переименовывает эти изображения, и большинство посетителей даже не знает, где на его компьютере хранятся кэшируемые файлы.

Наиболее очевидный шаг, который может предпринять пользователь, — простой захват экрана своего рабочего стола с вашим изображением, выведенным в окне Webобозревателя. Получив моментальный снимок экрана, они могут открыть его в своем графическом редакторе и вырезать изображение.

Несмотря на то, что предложенные препятствующие методы могут помешать некоторым посетителям, они не всегда помогают.

## См. также

Дополнительную информацию о панели инструментов Image на странице http:// www.microsoft.com/windows/ie/ie6/using/howto/customizing/imgtoolbar.mspx#EXE;

```
264
```

запускаемый JavaScript-метод, затрудняющий похищение изображений, на странице http://javascript.internet.com/page-details/no-right-click.html.

# 4.32. Автоматическая вставка отражений в изображения

# Проблема

Вы хотите вставить автоматическое отражение графического заголовка.

# Решение

Загрузите создающий этот эффект скрипт на JavaScript со страницы http:// cow.neondragon.net/stuff/reflection/.

После передачи его на Web-сервер свяжите файл на JavaScript с Web-страницей в элементе head:

```
<script type="text/javascript" src="scripts/reflection.js">
</script>
```

Вставьте в Web-страницу изображение, для которого хотите создать отражение (рис. 4.54):

```
<img src="christinaleaf.png" alt="christina m. huggins" />
```



Рис. 4.54. Отражение графического заголовка

Для активизации отражения добавьте атрибут class со значением reflect:

<img src="christinaleaf.png" alt="christina m. huggins" class="reflect" />

# Обсуждение

Когда страница визуализируется в Web-обозревателе посетителя сайта, скрипт отражения на JavaScript внимательно исследует элементы изображения вашей Web-страницы, пытаясь найти атрибуты class со значением reflect. Затем скрипт использует как источник изображение, которое вы хотите отразить, и создает новое изображение. Если скрипт находит какие-нибудь элементы с изображением, удовлетворяющие этим критериям, он копирует изображение, отражает его и затем применяет в новом отраженном изображении значения по умолчанию 50 % для непрозрачности (opacity) и высоты.

### Настраиваемые параметры

Скрипт отражения предоставляет два настраиваемых параметра: высоту отражения и непрозрачность отражения.

Для регулировки высоты отражения добавьте новое значение rheightxx в атрибут изображения class, при этом xx — это процент высоты изображения, который следует копировать в отражении:

```
<img src="christinaleaf.png" alt="christina m. huggins" class="reflect rheight99" />
```

При увеличении процентного значения увеличивается размер отражения. Например, значение rheight99 означает, что у отражения будет 99 % высоты исходного изображения.

Для регулировки непрозрачности отражения вставьте новое значение ropacityXX в атрибут изображения class, при этом XX задает процент прозрачности отраженного изображения:

```
img src="christinaleaf.png" alt="christina m. huggins"
class="reflect ropacity33" />
```

По мере уменьшения значения непрозрачности отражение становится все менее видимым. Например, значение ropacity33 означает, что в отражении будет использовано 33 % исходной высоты исходного изображения.

Оба параметра, и непрозрачность, и высота, могут применяться одновременно для создания более тонких визуальных эффектов:

```
<img src="christinaleaf.png" alt="christina m. huggins"
class="reflect rheight99 ropacity33" />
```

### Известные проблемы Web-обозревателей

Скрипт отражения поддерживают Internet Explorer 5.5 для Windows, Firefox 1.5, Opera 9 и более поздние версии этих Web-обозревателей, Chrome и Safari. Анимационные изображения в Web-обозревателях не работают со скриптом отражения за исключением Internet Explorer для Windows. Кроме того, в Internet Explorer для Windows масштабируемые изображения выглядят искаженными.

### См. также

Сообщение в блоге, знакомящее с эффектом отражения на странице http:// cow.neondragon.net/stuff/reflection/.

# 4.33. Применение изображений-спрайтов

# Проблема

Вы хотите сэкономить на трафике, поместив все или большинство ваших пиктограмм в одно изображение.

# Решение

Поместите наиболее часто используемые изображения в одно главное (master) изображение, убедившись в том, что вокруг каждого изображения достаточно свободного пространства.

Создайте достаточно свободного места для отображения каждой пиктограммы и задайте неповторяющееся фоновое изображение.

В этом примере каждая пиктограмма будет располагаться рядом со строкой заголовка:

```
h2 {
  margin: 0;
  font-family: Verdana, Arial, Helvetica, sans-serif;
  padding: 0 0 0 24px;
  font-weight: normal;
  background-image: url(sprite-source.gif);
  background-repeat: no-repeat;
}
```

Используя селекторы ID, поместите каждую пиктограмму рядом с соответствующим заголовком с помощью свойства background-position (рис. 4.55):

```
h2#warning {
background-position: -16px -15px;
}
h2#questions {
background-position: -16px -51px;
}
h2#comment {
background-position: -16px -87px;
}
h2#document {
background-position: -16px -123px;
}
h2#print {
background-position: -16px -159px;
}
h2#search {
background-position: -16px -195px;
}
```



Рис. 4.55. Пиктограммы выводятся из единого изображения

# Обсуждение

Очень похожим способом разработчики применяют одно и то же изображение снова и снова, пользуясь способностью Web-обозревателя кэшировать изображение, применение спрайтов помогает немного развить эту идею. Поместив отдельные графические элементы в одно изображение, Web-разработчики уменьшают количество серверных обращений Web-обозревателя к их машинам. Предложенное решение очень подойдет для сайтов со средним или большим объемом трафика.

# См. также

Статью "CSS Sprites" (Спрайты CSS) на странице http://www.alistapart.com/ articles/sprites.

# 4.34. Обрезка фоновых изображений

# Проблема

Вы хотите использовать в HTML-элементе только часть фонового изображения.

# Решение

Для отделения части фонового изображения, как показано на рис. 4.56, примените собственное свойство CSS Web-обозревателя Firefox moz-image-rect:

```
div {
  background-image: -moz-image-rect(url(bkgd.jpg), 0, 100, 100%, 0);
  background-repeat: no-repeat;
  width: 200px;
  height: 200px;
  border:1px solid #666;
  float: left;
  margin-right: 20px;
  }
  div + div {
  background-image: -moz-image-rect(url(bkgd.jpg), 0, 100, 100%, 0);
  background-repeat: repeat;
  }
}
```



## Обсуждение

Свойству -moz-image-rect требуется два вида данных. Сначала укажите изображение, которое нужно обрезать:

background-image: -moz-image-rect(url(bkgd.jpg));

А потом задайте видимую часть изображения с помощью последовательности из четырех значений, разделенных запятыми и выраженных в пикселах или процентах.

Эти значения представляют те же стороны изображения, что и в свойстве margin или padding с сокращенной формой записи: верхнюю, правую, нижнюю и левую:

background-image: -moz-image-rect(url(bkgd.jpg), 0, 100, 100%, 0);

### Примечание

Я не указал единицы измерения px после целочисленных значений, хотя бы потому что Web-обозреватель понимает, что здесь должны быть пикселы. Использование px не допускается и помешает отображению изображения. В последующих выпусках Web-обозревателя это может измениться, так что следите за будущими реализациями этого свойства.

Во время написания книги способность обрезать фоновые изображения была только в ночных сборках (nightly build) Firefox. Основная версия Web-обозревателя, следующая за Firefox 3.5, должна включать в себя поддержку этого свойства.

Для того чтобы фоновое изображение выводилось в других Web-обозревателях, не поддерживающих обрезку фона, вставьте в правило CSS свойство background-image со стандартными значениями. Другие обозреватели игнорируют свойство обрезки фона Firefox:

```
div {
  background-image: url(bkgd.jpg);
  background-image: -moz-image-rect(url(bkgd.jpg), 0, 1, 100%, 0);
  background-repeat: no-repeat;
  width: 200px;
  height: 200px;
  border:1px solid #666;
  float: left;
  margin-right: 20px;
}
```

### См. также

Сообщение в блоге CSS3.info о поддержке обрезки фонового изображения в Firefox 3.6 на странице http://www.css3.info/firefox-3-6-adds-background-clipping/.

# 4.35. Применение масок к изображениям и рамкам

# Проблема

Вы хотите применить маску к изображению и окружающей его рамке.

# Решение

Сначала создайте изображение в формате PNG с альфа-прозрачностью (рис. 4.57).



Затем примените маску с помощью правила CSS (рис. 4.58):

```
img {
  display: block;
  float: left;
  margin-right:20px;
  border: 10px solid #ccc;
  padding: 2px;
  background-color: #666;
  -webkit-mask-box-image: url(mask.png);
}
```

# Обсуждение

При создании маски все прозрачные части изображения становятся маской или частью, скрывающей фрагмент фонового изображения. Этот результат трудно объяснить, т. к. обычно альфа-прозрачность означает хорошую прозрачность.



Рис. 4.58. Маска, примененная к верхнему ряду изображений и их рамкам



Рис. 4.59. Комбинация градиентов и масок

Кроме изображений в формате PNG, в качестве масок можно применять изображения в формате SVG (рис. 4.59) так же, как и для формирования градиентов (см. *разд. 4.16*):

img {
 display: block;

```
float: left;
margin-right:20px;
border: 10px solid #ccc;
padding: 2px;
background-color: #666;
-webkit-mask-box-image: -webkit-gradient(linear, left bottom, left top,
from(rgba(0,0,0,1)), to(rgba(0,0,0,0)));
}
```

Во время написания книги создание масок средствами CSS поддерживал только Webобозреватель Safari.

## См. также

Сообщение в блоге Surfin's Safari о масках CSS на странице http://webkit.org/ blog/181/css-masks/.

# Элементы страницы

# 5.0. Введение

Все выбранные варианты стилевого оформления, начиная с наиболее очевидных, таких как типографика и изображения, и заканчивая часто пренебрегаемыми, например, межстрочным интервалом или цветовой схемой, оказывают сильное влияние на восприятие посетителем информации Web-сайта.

Эта глава посвящена элементам, обрамляющим Web-страницу, как рама вставленную в нее картину. Элементы страницы — это параметры, влияющие на внешний вид Web-страницы, но не обязательно воспринимаемые как часть ее дизайна. Например, выведенная полоса прокрутки — это элемент страницы.

Манипулирование такими элементами, как поля и рамки, окружающие контекст Webстраницы, позволяет разработчикам создать оформление информационного наполнения страницы без реального применения стилей к контексту.

Эти простые изменения могут оказать сильное влияние на общий дизайн страницы или стать последним штрихом, завершающим дизайн.

# 5.1. Удаление полей страницы

## Проблема

Вы хотите избавиться от пробелов по краям Web-страницы и между рамкой окна Webобозревателя и содержимым страницы (рис. 5.1).

# Решение

Задайте нулевые значения в свойствах margin и padding для элементов html и body:

```
html, body {
  margin: 0;
  padding: 0;
}
```

# Обсуждение

Установка нулевых значений (единицы измерения при нулевом значении указывать не нужно) в свойствах margin и padding элемента body создает эффект страницы обрезан-

ной в край (full-bleed) — другими словами, удаляет пробельную зону по краям страницы.



Рис. 5.1. Поля страницы видны как пробельные зоны по краям Web-страницы

Тем не менее иногда, в зависимости от информационного наполнения Web-страницы, изменения значений свойств margin и padding может оказаться недостаточно для создания впечатления обрезанной в край страницы. Стандартные свойства других элементов могут иметь неожиданные побочные эффекты, проявляющиеся при попытке изменить поля страницы.

Например, если элемент h1 — первый прямой наследник элемента body, то некоторая назначенная ему пробельная зона появится над строкой заголовка или под верхним краем области просмотра окна обозревателя. На рис. 5.2 показан этот нежелательный эффект; цвет фона у заголовков и абзацев серый, поэтому описываемый эффект особенно заметен.

В этом случае для создания впечатления обрезанной в край страницы задайте нулевые значения в свойствах margin и padding служащего камнем преткновения элемента (в данном случае h1, h2, h3), как и элемента body. Этот шаг установит все отступы элементов в нулевое значение. Если такая установка невозможна (например, если вам нужны нижние поле или отступ), задайте нулевые значения в свойствах margin-top и padding-top для создания впечатления обрезанной в край страницы:

```
html, body {
  margin: 0;
  padding: 0;
  position: absolute;
  top: 0;
  left: 0;
}
```

```
h1, h2, h3 {
  margin-top: 0;
  padding-top: 0;
  background-color: #666;
}
p {
  background-color: #999;
}
```



🔺 🐔 Home 🖹 Bookmarks 🛇 oreilly.com 🛇 christopher.org
Lorem ipsum dolor
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nullam feugiat, turpis at pulvinar vulputate, erat libero tristique tellus, nec bibendum odio risus sit amet ante. Aliquam erat volutpat. Nunc auctor. Mauris pretium quam et urna. Fusce nibh. Duis risus. Curabitur sagittis hendrerit ante. Aliquam erat volutpat. Vestibulum erat nulla, ullamcorper nec, rutrum non, nonummy ac, erat. Duis condimentum augue id magna semper rutrum. Nullam justo enim, consectetuer nec, ullamcorper ac, vestibulum in, elit. Proin pede metus, vulputate nec, fermentum fringilla, vehicula vitae, justo. Fusce consectetuer risus a nunc. Aliquam ornare wisi eu metus. Integer pellentesque quam vel velit. Duis pulvinar.
Marhi lao mi
Morbi leo mi, nonummy eget, tristique non, rhoncus non, leo. Nullam faucibus mi quis velit. Integer in sapien. Fusce tellus odio, dapibus id, fermentum quis, suscipit id, erat. Fusce aliquam vestibulum ipsum. Aliquam erat volutpat. Pellentesque sapien. Cras elementum. Nulla pulvinar eleifend sem. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Quisque porta. Vivamus portitior urpis ac leo.
Lorem ipsum dolor
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi gravida libero nec velit. Morbi scelerisque

Рис. 5.3. Над строкой заголовка удалена пробельная зона

Как видно на рис. 5.3, мы добились эффекта обрезанной в край страницы. Обратите внимание на то, что серый цвет фона первого заголовка теперь примыкает к верхней границе области просмотра окна обозревателя.

## См. также

В *разд. 11.1* разработку одноколоночных макетов с помощью установки ненулевых значений в свойствах margin и padding.

# 5.2. Сброс стилей, назначаемых Web-обозревателем по умолчанию

# Проблема

Вы хотите помешать Web-обозревателям задавать значения параметров для элементов на Web-странице.

## Решение

Примените отдельную таблицу стилей, в которой заданы все или большая часть HTML-элементов, обычно используемых в Web-документе, с параметрами, имеющими нулевые значения или исключающими автогенерируемое содержимое, например, таблицу стилей, созданную Yahoo! Developer Network (http://developer.yahoo.com/) и названную YUI Reset CSS:

```
/*
Copyright (c) 2009, Yahoo! Inc. All rights reserved.
Code licensed under the BSD License:
http://developer.yahoo.net/yui/license.txt
version: 2.7.0
*/
html{color:#000;background:#FFF;}body,div,dl,dt,dd,ul,ol,li,h1,h2,h3,h4,h5,h6,
pre, code, form, fieldset, legend, input, button, textarea, p, blockquote, th,
td{margin:0;padding:0;}table{border-collapse:collapse;border-
spacing:0; } fieldset, img{border:0; } address, caption, cite, code, dfn, em, strong, th,
var,optgroup{font-style:inherit;font-weight:inherit;}del,
ins{text-decoration:none;}li{list-style:none;}caption,th{text-align:left;}
h1, h2, h3, h4, h5, h6{font-size:100%; font-
weight:normal;}q:before,q:after{content:'';}abbr,acronym{border:0;font-
variant:normal; }sup{vertical-align:baseline; }sub{vertical-
align:baseline; }legend{color:#000; }input, button, textarea, select, optgroup,
option {font-family:inherit; font-size:inherit; font-style:inherit; font-
weight:inherit; }input, button, textarea, select { * font-size:100%; }
```

Для применения исходной настройки параметров используйте приведенный код, сославшись на него в элементе link, до задания всех остальных стилей в Web-документе:

# <link rel="stylesheet" type="text/css"href="http://yui.yahooapis.com/2.7.0/build/reset/reset-min.css"> <link rel="stylesheet" type="text/css" href="mywebsitestylesheet.css">

# Обсуждение

Web-обозреватели применяют разные подходы к отображению элементов на странице, начиная с размера заголовков h1 (рис. 5.4) и заканчивая использованием или отказом от использования свойств margin и padding для формирования отступов в нумерованном списке (см. *paзd*. 6.3).



Рис. 5.4. Главный заголовок с небольшими отличиями

Задача таблицы стилей, сбрасывающей стандартные значения свойств элементов (рис. 5.5), — помочь преодолеть некоторые преграды, связанные с Web-разработкой, пригодной для разных обозревателей.

Поскольку большая часть свойств элементов "сбрасывается" (как показано на рис. 5.6), Web-дизайнеру приходится выполнять дополнительную работу по настройке внешнего вида Web-страницы, не полагаясь на обработку стилей Web-обозревателем.

# Main Content

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Praesent aliquam, justo convallis luctus rutrum, erat nulla fermentum diam, at nonummy quam ante ac quam. Maecenas urna purus, fermentum id, molestie in, commodo porttitor, felis. Nam blandit quam ut lacus.

Quisque ornare risus quis ligula. Phasellus tristique purus a augue condimentum adipiscing. Aenean sagittis. Etiam leo pede, rhoncus venenatis, tristique in, vulputate at, odio. Donec et ipsum et sapien vehicula nonummy. Suspendisse potenti. Fusce varius urna id quam. Sed neque mi, varius eget, tincidunt nec, suscipit id, libero. In eget purus. Vestibulum ut nisl. Donec eu mi sed turpis feugiat feugiat. Integer turpis arcu, pellentesque eget, cursus et, fermentum ut, sapien. Fusce metus mi, eleifend sollicitudin, molestie id, varius et, nibh. Donec nec libero.

## H2 level heading

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Praesent aliquam, justo convallis luctus rutrum, erat nulla fermentum diam, at nonummy quam ante ac quam. Maecenas urna purus, fermentum id, molestie in, commodo portitor, felis. Nam blandit quam ut lacus. Quisque ornare risus quis ligula. Phasellus tristique purus a augue condimentum adipiscing. Aenean sagittis. Etiam leo pede, rhoncus venenatis, tristique in, vulputate at, odio. sidebar1 Content

The background color on this div will only show for the length of the content. If you'd like a dividing line instead, place a border on the right side of the #mainContent div if the #mainContent div will always contain more content than the #sidebar1 div.

Donec eu mi sed turpis feugiat feugiat. Integer turpis arcu, pellentesque eget, cursus et, fermentum ut, sapien.

Рис. 5.5. Стандартное отображение HTML-текста

#### Main Content

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Praesent aliquam, justo convallis luctus rutrum, erat nulla fermentum diam, at nonummy quam ante ac quam. Maecenas urna purus, fermentum id, molestie in, commodo portitor, felis. Nam blandit quam ut lacus. Quisque ornare risus quis ligula. Phasellus tristique purus a augue condimentum adipiscing. Aenean sagittis. Etiam leo pede, rhoncus venenatis, tristique in, vulputate at, odio. Donec et ipsum et sapien vehicula nonummy. Suspendisse potenti. Fusce varius urna id quam. Sed neque mi, varius eget, tincidunt nec, suscipit id, libero. In eget purus. Vestibulum ut nisl. Donec eu mi sed turpis feugiat feugiat. Integer turpis arcu, pellentesque eget, cursus et, fermentum ut, sapien. Fusce metus mi, eleifend sollicitudin, molestie id, varius et, nibh. Donec nec libero.

#### H2 level heading

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Praesent aliquam, justo convallis luctus rutrum, erat nulla fermentum diam, at nonummy quam ante ac quam. Maecenas urna purus, fermentum id, molestie in, commodo porttitor, felis. Nam blandit quam ut lacus. Quisque ornare risus quis ligula. Phasellus tristique purus a augue condimentum adipiscing. Aenean sagittis. Etiam leo pede, rhoncus venenatis, tristique in, vulputate at, odio. sidebar1 Content The background color on this div will only show for the length of the content. If you'd like a dividing line instead, place a border on the right side of the #mainContent div if the #mainContent div will always contain more content than the #sidebar1 div. Donec eu mi sed turpis feugiat feugiat. Integer turpis arcu, pellentesque eget, cursus et, fermentum ut, sapien.

Рис. 5.6. Примененный сброс таблицы CSS

### Расширенный сброс CSS

Эрик Мейер (Eric A. Meyer), автор книги "Cascading Style Sheets: the Definitive Guide" (Каскадные таблицы стилей: подробное руководство) (издательство O'Reilly Media), после изучения исходных файлов компании Yahoo! создал собственную версию сброса стилей CSS (http://meyerweb.com/eric/thoughts/2007/04/12/reset-styles/):

```
/* v1.0 | 20080212 */
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, font, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td {
    margin: 0;
    padding: 0;
    border: 0;
    outline: 0;
    font-size: 100%;
    vertical-align: baseline;
   background: transparent;
}
body {
    line-height: 1;
}
ol, ul {
    list-style: none;
}
blockquote, q {
    quotes: none;
}
blockquote:before, blockquote:after,
q:before, q:after {
    content: '';
    content: none;
}
/* remember to define focus styles! */
:focus {
    outline: 0;
}
/* remember to highlight inserts somehow! */
ins {
    text-decoration: none;
}
del {
    text-decoration: line-through;
/* tables still need 'cellspacing="0"' in the markup */
table {
    border-collapse: collapse;
    border-spacing: 0;
```

}

В таблице сброса стилей Мейера создана точка отсчета для размеров шрифтов с помощью задания в универсальном селекторе body свойства font-size, равного 100 %, и свойства line-height, равного 1 (или 100 %).

Для большинства элементов задан прозрачный фон, в то время как в таблице YUI Reset CSS задан белый (#fff) цвет фона для элемента html.

Свойству quote в элементах blockquote и q, применяемых для задания вариантов выделения цитат и вложенных цитат, присвоено значение none. Этого свойства нет в файле YUI Reset CSS.

В файле YUI Reset CSS также отсутствуют правила для псевдокласса : focus, определяющего способ отображения элемента, активизируемого пользователем (например, щелчком мыши в текстовом поле ввода элемента-формы).

Какой из файлов использовать вам? Испытайте оба и решите, какой вам подходит.

Может быть лучшее решение — создать собственную таблицу сброса стилей CSS.

У таблиц сброса стилей Yahoo! Developer Network и Эрика Мейера достаточно различий в верном подходе к решению проблемы, и в данном случае нельзя остановить выбор просто на самой свежей версии или на варианте с более компактным кодом. Создайте собственное средство сброса стилей CSS, основываясь на том, как вы работаете и что нужно сделать, чтобы выполнить вашу работу наилучшим образом.

### См. также

Таблица стилей YUI Reset CSS на странице http://developer.yahoo.com/yui/reset; сообщение в блоге Эрика Мейера (Eric Meyer), названное "Reset Reloaded" (повторная загрузка сброса стилей) по адресу http://meyerweb.com/eric/thoughts/2007/05/01/resetreloaded/.

# 5.3. Раскрашивание полосы прокрутки в IE

# Проблема

Вы хотите откорректировать цвет полосы прокрутки в области просмотра Webобозревателя или в окне обозревателя Internet Explorer.

# Решение

Используйте свойства, управляющие цветами полосы прокрутки в Web-обозревателях:

```
body,html {
  scrollbar-face-color: #99ccff;
  scrollbar-shadow-color: #ccccff;
  scrollbar-highlight-color: #ccccff;
  scrollbar-3dlight-color: #99ccff;
  scrollbar-darkshadow-color: #ccccff;
  scrollbar-track-color: #ccccff;
  scrollbar-arrow-color: #000033;
}
```

### Предупреждение

Поскольку эти свойства не являются частью рекомендаций консорциума W3C для CSS, разработчики Web-обозревателей не обязаны включать поддержку этих свойств. Предложенное решение работает только в Web-обозревателях KDE Konqueror и Internet Explorer. Другие обозреватели будут просто пропускать эти правила. Корректность данных правил не будет подтверждаться сервисами, такими как http://jigsaw.w3.org/css-validator/#validate\_by\_uri.

## Обсуждение

Несмотря на то, что полоса прокрутки может показаться простым инструментом, на самом деле она состоит из нескольких элементов, которые создают управляемый 3D-объект. Рисунок 5.7 акцентирует внимание на различных свойствах полосы прокрутки. Как видите, для создания действительно разнообразной цветовой схемы этого элемента управления вам придется изменить значения семи свойств.



Помимо настройки полосы прокрутки в области просмотра окна Web-обозревателя, вы также можете отрегулировать цвета полосы прокрутки в текстовом поле (textarea) Web-формы, в элементах frameset, iframe и практически в любых элементах с полосой прокрутки:

```
highlight {
  scrollbar-face-color: #99ccff;
  scrollbar-shadow-color: #ccccff;
  scrollbar-highlight-color: #ccccff;
  scrollbar-3dlight-color: #99ccff;
  scrollbar-darkshadow-color: #ccccff;
```

```
scrollbar-track-color: #ccccff;
scrollbar-arrow-color: #000033;
}
<form>
<textarea class="highlight"></textarea>
</form>
```

### Работа в режиме обратной совместимости

Когда визуализируется страница, не имеющая корректного **DOCTYPE**, версии IE, предшествующие IE8, переходят в режим обратной совместимости (quirks mode) (см. *разд. 1.3*) и ищут свойства полосы прокрутки в селекторе элемента **body**.

Если у страницы есть **DOCTYPE**, допустимый в предыдущих версиях IE, IE8 остается в стандартном режиме и обрабатывает селектор элемента html.

Таким образом, на случай изменения **DOCTYPE** лучше всего сгруппировать селекторы элементов body и html и применить к ним одно правило CSS:

```
html .highlight, body .highlight {
  scrollbar-face-color: #99ccff;
  scrollbar-shadow-color: #ccccff;
  scrollbar-highlight-color: #ccccff;
  scrollbar-3dlight-color: #99ccff;
  scrollbar-darkshadow-color: #ccccff;
  scrollbar-track-color: #ccccff;
  scrollbar-arrow-color: #000033;
}
```

### Цветные полосы прокрутки в IE8

Для Internet Explorer 8 корпорация Microsoft преобразовала свои специализированные свойства CSS в расширения разработчиков с префиксом -ms-.

#### Примечание

IE8 позволяет Web-обозревателю замаскироваться под версию Internet Explorer 7, сохраняя действующим собственный уникальный набор свойств отображения. В режиме эмуляции Emulate IE7 Web-обозревателю понятен исходный набор специализированных свойств полосы прокрутки.

Таким образом, для обеспечения универсальной поддержки в разных версиях Internet Explorer убедитесь в том, что включены оба набора свойств:

```
highlight {
  scrollbar-face-color: #99ccff;
  scrollbar-shadow-color: #ccccff;
  scrollbar-highlight-color: #ccccff;
  scrollbar-3dlight-color: #99ccff;
  scrollbar-darkshadow-color: #ccccff;
  scrollbar-track-color: #ccccff;
  scrollbar-arrow-color: #000033;
  -ms-scrollbar-face-color: #99ccff;
  -ms-scrollbar-shadow-color: #ccccff;
  -ms-scrollbar-highlight-color: #ccccff;
  -ms-scrollbar-highligh
```

```
-ms-scrollbar-3dlight-color: #99ccff;
-ms-scrollbar-darkshadow-color: #ccccff;
-ms-scrollbar-track-color: #ccccff;
-ms-scrollbar-arrow-color: #000033;
}
```

Для точного указания свойств CSS, относящихся к конкретной версии IE, применяйте условные комментарии (см. *разд 12.7*).

### Примечание

У Web-обозревателя Safari также есть собственные правила CSS для раскрашивания полосы прокрутки. Дополнительную информацию см. на странице http://webkit.org/blog/363/ styling-scrollbars/.

### См. также

Специфические функциональные возможности Internet Explorer на странице http://msdn.microsoft.com/en-us/library/cc304082(VS.85,loband).aspx#ie\_specific; "IE Colour scrollbar marker" (Генератор цветной полосы прокрутки в IE) по адресу http://www.sean.co.uk/a/webdesign/color\_scrollbar\_maker\_ie.shtm.

# 5.4. Методы центрирования элементов на Web-странице

## Проблема

Вы хотите отцентрировать фрагменты Web-страницы, как показано на рис. 5.8.



Рис. 5.8. Центрированный текст заголовка

## Решение

Для центрирования текста в элементе блочного уровня используйте свойство textalign:

```
h1, h2, h3 {
   text-align: center;
}
```

## Обсуждение

С помощью свойства text-align можно центрировать текст в элементах блочного уровня. Но в данном примере заголовок занимает всю ширину элемента body, и если не применить к элементу цвет фона, можно просто не заметить этого центрирования. Серый цвет фона на рис. 5.9 показывает реальную ширину центрированных элементов.



Рис. 5.9. Реальная ширина элементов, показанная серым цветом фона

Альтернативный подход — применение полей для центрирования текста в контейнере:

```
h1, h2, h3 {
  margin-left: auto;
  margin-right: auto;
  width: 300px;
}
```

Когда задается значение auto в свойствах margin-left и margin-right (наряду со значением свойства width), элемент центрируется в родительском элементе.

### Таблицы

Для центрирования таблицы задайте атрибут class с каким-либо значением:

```
<<div class="center">

        This is the first cell
        This is the first cell
        This is the second cell
        This is the second cell
        This is the third cell, it's under the first cell
        This is the fourth cell, it's under the second cell.
        This is the fourth cell, it's under the second cell.
        This is the fourth cell, it's under the second cell.
        This is the fourth cell, it's under the second cell.
        This is the fourth cell, it's under the second cell.
        This is the fourth cell, it's under the second cell.
        This is the fourth cell, it's under the second cell.
        This is the fourth cell, it's under the second cell.
        This is the fourth cell, it's under the second cell.
```

Далее напишите следующее правило CSS:

```
.center {
  width: 50%;
  margin-left: auto;
  margin-right: auto;
}
```

### Изображения

Если вы хотите отцентрировать изображение, сначала вставьте элемент img в элемент div. Этот прием необходим, т. к. элемент img строчный (inline), как и элементы ем и strong. Он находится в нормальном потоке Web-страницы и не разделяет окружающее свободное пространство, как делают элементы блочного уровня, такие как р или blockquote. Разметка выглядит следующим образом:

```
<div class="flagicon"><img src="flag.gif" alt="Flag" width="160" height="60" /></div>
```

А правило CSS так:

```
.flagicon {
  text-align: center;
}
```

Для центрирования элементов с фиксированной шириной, таких как изображения, сначала задайте значение свойства padding-left родительского элемента равным 50 %. Затем определите половину ширины элемента, который вы центрируете, и задайте ее как отрицательное значение в свойстве margin-left. Это помешает левому краю элемента оставаться на позиции, равной 50 % ширины строки в соответствии с его отступом, и переместит элемент в середину страницы.

Разметка для изображения на Web-странице, использующая этот прием, будет выглядеть примерно следующим образом:

```
<img src="wolf.jpg" width="256" height="192" alt="Photo of wolf.">
```

Правило CSS, формирующее результат, показанный на рис. 5.10, выглядит следующим образом:

```
body {
    padding-left: 50%;
}
img {
    /*поле равно половине ширины изображения с минусом */
    margin-left: -138px;
}
```



Рис. 5.10. Изображение центрировано без применения нерекомендованного элемента center

### Вертикальное центрирование

Отцентрировав элемент горизонтально, вы можете с помощью этого метода продвинуться на один шаг дальше и отцентрировать изображение (или любой другой элемент) вертикально.

Разница заключается в том, что для выполнения этой задачи используется свойство position. Применяется та же разметка для элемента img, что и в предыдущем примере, но в данном случае правило CSS предназначено только для одного селектора (рис. 5.11):

```
img {
  position: absolute;
  top: 50%;
  left: 50%;
  margin-top: -96px;
  margin-left: -138px;
  height: 192px;
  width: 256px;
}
```

Благодаря абсолютному позиционированию (см. *разд. 2.23*) вы извлекаете элемент из нормального потока документа и помещаете его, куда захотите.

Если вы хотите центрировать и текст, и изображение (или другие изображения), а не одно изображение, заключите весь контент в элемент div:

```
<div id="centerFrame">
Epsum factorial non deposit quid pro quo hic escorol. Olypian
```


Рис. 5.11. Изображение центрируется на Web-странице горизонтально и вертикально

Затем в правиле CSS удалите свойство height и откорректируйте отрицательное значение верхнего поля для того, чтобы скомпенсировать появление дополнительных элементов на странице:

```
#centerFrame {
  position: absolute;
  top: 50%;
  left: 50%;
  /* изменяйте отрицательное значение, пока контент не отцентрируется */
  margin-top: -150px;
  margin-left: -138px;
  width: 256px;
}
```

Контент, предназначенный для центрирования, должен быть небольшим. Данное решение подходит только для приблизительной центровки текста и изображений, поскольку на разных компьютерах будет выводиться текст разной высоты. Если у вас несколько изображений и большой объем HTML-текста, пользователи, работающие на компьютерах с низким разрешением, будут вынуждены прокручивать страницу, чтобы увидеть отцентрированное вами содержимое.

#### См. также

В *главе 10* информацию о многоколоночной верстке с использованием позиции элементов на Web-странице.

## 5.5. Создание рамки страницы

## Проблема

Вы хотите поместить рамку или создать обрамление Web-страницы (рис. 5.12).



Рис. 5.12. Заключенная в рамку Web-страница

#### Решение

Воспользуйтесь свойством border элемента body:

```
body {
  margin: 0;
  padding: 1.5em;
  border: 50px #666 ridge;
}
```

## Обсуждение

Свойство border — это свойство с сокращенной формой записи, что позволит вам задать ширину, цвет и стиль рамки вокруг элемента с помощью одной строки, а не трех.

Если не пользоваться свойством с сокращенной формой записи в приведенном решении, вам придется заменить строку

border: 50px #666 ridge;

следующими тремя строками:

```
border-width: 50px;
border-color: #666;
border-style: ridge;
```

Создать обрамление можно и с помощью других стилей рамки, таких как пунктирная линия (dotted), штриховая линия (dashed), сплошная линия (solid), двойная линия (double), псевдотрехмерная рамка с углублением (groove), рамка в виде трехмерной врезки (inset), рамка в виде трехмерной вырезки (outset) (рис. 5.13).



Рис. 5.13. Имеющиеся в спецификации CSS стили рамки

Имейте в виду, что у стиля groove инверсные градации тени по сравнению с приведенным решением, в котором используется значение ridge (рамка в виде трехмерного выступа).

Вызывает беспокойство лишь то, что в Internet Explorer для Windows стиль dotted (пунктирный) отображается как ступенчатые окружности (aliased circles), тогда как в Firefox, Opera и Safari этот стиль выводится в виде квадратиков.

#### Рамки вокруг изображений

Вы также можете заключить в стилизованную рамку изображение (см. *разд. 4.2*). Вместо применения стандартной сплошной линии попробуйте поэкспериментировать с рамками, имеющими стиль groove или double, как показано на рис. 5.14:

```
img.left {
  float: left;
  margin-right: 7px;
  margin-bottom: 3px;
  border: 4px double #666;
}
```



#### См. также

В разд. 3.21 создание врезок с разными стилями рамок.

# 5.6. Вывод рамки вокруг окна просмотра Web-обозревателя

### Проблема

Вы хотите поместить рамку вокруг окна или области просмотра Web-обозревателя.

### Решение

Сначала задайте последовательность из восьми элементов div, которые помещаются под контентом Web-страницы, но до закрытия элемента body:

```
<div id="top"></div>
<div id="topright"></div>
<div id="right"></div>
<div id="bottomright"></div>
<div id="bottom"></div>
<div id="bottomleft"></div>
<div id="left"></div>
<div id="left"></div>
```

Задайте одинаковые ширину и высоту углов рамки и присвойте их свойству position значение fixed:

```
#topleft, #topright, #bottomleft, #bottomright {
  height: 24px;
  width: 24px;
  position: fixed;
  display: block;
  z-index: 20;
}
```

Для сторон рамки также задайте фиксированную позицию. Кроме того, для верхней и нижней сторон рамки укажите высоту, равную 24 пикселам, а для левой и правой сторон — ширину, равную 24 пикселам:

```
#top, #bottom {
height: 24px;
position: fixed;
left: 0;
right: 0;
display: block;
background-color: #ccff00;
z-index: 30
}
#left, #right {
width: 24px;
position: fixed;
top: 0;
bottom: 0;
display: block;
background-color: #ccff00;
z-index: 50;
}
```

Далее свяжите каждый элемент рамки с соответствующими углом или стороной окна просмотра:

```
#top {
top: 0;
}
#bottom {
bottom: 0;
}
#left {
left: 0;
}
#right {
 right: 0;
}
#topleft {
top: 0;
left: 0;
}
#topright {
top: 0;
right: 0;
}
 #bottomleft {
bottom: 0;
left: 0;
}
 #bottomright {
bottom: 0;
 right: 0;
}
```

#### Обсуждение

Особенность подхода, предложенного в данном разделе, заключается в том, что рамка в элементе body расширяется на высоту контента. Для постоянного отображения видимой рамки вокруг всего окна просмотра независимо от размера контента применяйте фиксированное позиционирование (см. *разд. 4.10*).

Вместо использования цветов фона для отображения полос можно применить похожий метод, использующий изображения в формате PNG (или даже градиенты CSS с прозрачностью, как в *разд. 4.16*) для получения эффекта рассеивания. По мере прокручивания страницы пользователем текст вдоль краев окна просмотра обозревателя тускнеет.

#### См. также

Фиксированное позиционирование в спецификации CSS 2 на странице http:// www.w3.org/TR/CSS2/visuren.html#fixed-positioning.

## 5.7. Настройка горизонтального разделителя

### Проблема

Вы хотите изменить вид горизонтальной линии или разделителя (элемент hr), используя вместо сплошной линии (рис. 5.15) что-то более интересное, например изображение, показанное на рис. 5.16.





Рис. 5.16. Стилизованный горизонтальный разделитель

#### Решение

Для получения желаемого эффекта примените комбинацию свойств CSS элемента hr:

```
<style type="text/css">
hr {
border: 0;
height: 43px;
background-image: url(hr.gif);
background-position: 50% 0;
background-repeat: no-repeat;
margin: .66em 0;
}
</style>
<!--[if lt IE 8]>
<style type="text/css">
hr {
 display: list-item;
 list-style: url(hr.gif) inside;
 filter: alpha(opacity=0);
width: 0;
}
</style>
```

#### Обсуждение

До появления версии языка HTML 4.0 отображением горизонтальных разделителей можно было управлять с помощью набора из четырех атрибутов: align, width, size и noshade. Поскольку язык HTML предназначен для разметки информационного наполнения, а не его внешнего вида, эти значения больше не входят в спецификацию HTML. (Разработчики Web-обозревателей могут поддерживать эти значения, но выгода от их применения может быть разной.) Управляя отображением с помощью правил CSS, вы получаете гораздо больше возможностей контролировать внешний вид горизонтальных разделителей.

Для создания стиля горизонтальных разделителей, подходящего для разных Webобозревателей, задайте нулевую рамку и затем с помощью свойства background вставьте изображение. Отрегулируйте по своему вкусу поля сверху и снизу.

Для поддержки стилей в старых версиях IE используйте условные комментарии, чтобы добавить альтернативный метод вставки фонового изображения:

```
<!--[if lt IE 8]>
<style type="text/css">
hr {
  display: list-item;
  list-style: url(hr.gif) inside;
  filter: alpha(opacity=0);
  width: 0;
}
</style>
<![endif]-->
```

Поскольку старые версии IE не могут вставлять фоновое изображение через элемент hr, задайте значение list-item в свойстве display. Это позволит вставить изображение с помощью свойства list-style.

Для удаления рамки вокруг элемента hr примените нулевую непрозрачность в фильтре CSS корпорации Microsoft.

#### См. также

В спецификации HTML 4.01 описание элемента hr на странице http://www/w3/ org/TR/html401/present/graphics.html#edef-HR; обзор стилей элемента hr на http://www.sovavsiti.cz/css/hr.html.

# 5.8. Добавление просмотра изображения в режиме *Lightbox*

## Проблема

Вы хотите выводить изображения поверх текущей Web-страницы (рис. 5.17) без распахивания нового окна Web-обозревателя.



Рис. 5.17. Текущая страница

### Решение

Загрузите исходный код для вывода изображений поверх страницы с Web-сайта http://www.huddletogether.com/projects/lightbox2/#download.

Вместе с библиотеками JavaScript Prototype Framework и Scriptaculous Effects вставьте специальный скрипт на языке JavaScript для вывода полноразмерных изображений поверх страницы:

```
<title>Mr. McCool's Homepage</title>
<!-- Structure for Lightbox effect -->
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript" src="scriptaculous.js?load=effects"></script>
<!-- Script for Lightbox -->
<script type="text/javascript" src="lightbox.js"></script>
```

Далее присоедините таблицу стилей, которая визуализирует внешнее оформление выводимого поверх страницы изображения:

```
<title>Mr. McCool's Homepage</title>
<script type="text/javascript" src="prototype.js"></script>
<script type="text/javascript" src="scriptaculous.js?load=effects"></script>
<script type="text/javascript" src="lightbox.js"></script>
<link rel="stylesheet" href="lightbox.css" type="text/css" media="screen" />
```

В контент Web-страницы включите ссылку на изображение, убедившись в том, что у нее есть атрибут rel со значением lightbox. Обычно ссылка включает в себя и свернутое в пиктограмму изображение:

<a href="trammel\_shoes.jpeg" rel="lightbox" title="Trammel shows off his happy shoes."><img src="trammel\_shoes\_tn.jpeg" alt="Mark Trammel is happy with his shoes." /></a>

Щелчок кнопкой мыши по ссылке активизирует вывод полноразмерного изображения поверх страницы, как показано на рис. 5.18.



Рис. 5.18. Полноразмерное изображение, появляющееся поверх страницы

## Обсуждение

Эффект вывода полноразмерного изображения поверх страницы базируется на двух основных компонентах JavaScript: Prototype Framework и Scriptaculous.

Prototype создает объектно-ориентированную оболочку, позволяющую разработчикам быстро формировать Web-приложения на основе языка JavaScript. Дополнительную информацию о Prototype смотрите на официальном Web-сайте http://prototypejs.org/.

Scriptaculous — это коллекция библиотек на JavaScript. Используя ее в сочетании с Prototype, разработчики могут устанавливать динамическое взаимодействие языков JavaScript и XML (технология Asynchronous JavaScript + XML, Ajax). Более подробную информацию о Scriptaculous можно найти на сайте http://script.aculo.us/.

С помощью имеющегося базового инструментария на JavaScript Web-разработчик Локеш Дакар (Lokesh Dhakar) (см. http://www.lokeshdhakar.com/projects/lightbox2/) разработал интеллектуальное средство просмотра изображений, отображающее полноразмерное изображение и позволяющее не покидать страницу, выводящую уменьшенные варианты изображений.

#### Примечание

Существуют также другие библиотеки JavaScript и галерея подключаемых модулей (плагинов). Например, познакомьтесь с jQuery (см. *главу 14*) и галереей изображений (http://code.google.com/p/galleria/).

#### Задание файлов

После того как вы загрузили из Интернета файлы на JavaScript и связали их с таблицей стилей и Web-страницей, проверьте корректность ссылок на эти файлы. Например, если вы поместили файлы на JavaScript и таблицу стилей в отдельные папки, убедитесь в том, что в коде указано их местонахождение:

```
<script type="text/javascript" src="/_assets/js/prorotype.js"></script>
<script type="text/javascript"
src="/_assets/js/scriptaculous.js?load=effects"></script>
<script type="text/javascript" src="/_assets/js/lightbox.js"></script>
<link rel="stylesheet" href="/_assets/css/lightbox.css"
type="text/css" media="screen" />
```

В файле скрипта на JavaScript, выводящего полноразмерные изображения, проверьте правильность ссылок на изображения. Если необходимо откорректировать местонахождение файлов с изображениями, найдите в начале файла на JavaScript строки, которые нужно исправить:

```
var fileLoadingImage = "/_assets/img/loading.gif";
var fileBottomNavCloseImage = "/_assets/img/closelabel.gif";
```

В таблице стилей для вывода полноразмерного изображения свойство фонового изображения применяется трижды. Убедитесь в том, что в ссылках на изображения, используемых в этих свойствах, указано правильное местонахождение файлов:

```
#prevLink, #nextLink {
  width: 49%;
  height: 100%
/* Искусственный прием в IE для визуального эффекта наведения мыши на ссылку */
  background: transparent url(/_assets/img/blank.gif) no-repeat;
  display: block;
  }
  #prevLink:hover, #prevLink:visited:hover {
   background: url(/_assets/img/prevlabel.gif) left 15% no-repeat;
  }
  #nextLink:hover, #nextLink:visited:hover {
   background: url(/_assets/img/nextlabel.gif) right 15% no-repeat;
  }
```

#### Создание покадрового просмотра

Помимо показа одного изображения, средство просмотра полноразмерного изображения можно настроить на вывод изображений в режиме слайд-шоу, как показано на рис. 5.19.

Для достижения этого эффекта исправьте значение атрибута rel, добавив в него квадратные скобки после слова lightbox и имя набора изображений. В примере кода я использовал имя набора austin, поскольку фотографии были сделаны в г. Остин, штат Texac:

```
<a href="trammell_shoes.jpg" rel="lightbox[austin]"</li>
title="Trammell shows off his happy shows."><img src="tremmell_shoes_tn.jpg"</li>
alt="Mark Trammel is happy with his shoes." /></a>
<a href="molly_andy.jpg" rel="lightbox[austin]" title="Molly and Andy pose for a shot."><img src="molly_andy.jpg" alt="Molly and Andy pose for a shot."</li>
<a href="molly_andy_tn.jpg" alt="Molly and Andy pose for a shot."</a>
<a href="msjen.jpg" rel="lightbox[austin]" title="Ms. Jen at breakfast."><img src="msjen_tn.jpg" alt="Ms. Jen at breakfast." />/a>/li>
```

Имя набора должно быть одинаковым, для того чтобы связанные изображения были помещены в одну презентацию.



Рис. 5.19. Средство просмотра полноразмерного изображения может выводить ряд изображений

#### Известные проблемы Web-обозревателей

Поскольку средство вывода полноразмерного изображения поверх страницы построено на базе оболочки Prototype, его действие в Web-обозревателе зависит от поддержки обозревателями Prototype. Во время написания книги к ним относились следующие Web-обозреватели:

- Microsoft Internet Explorer 6 для Windows и более поздние версии;
- Firefox 1.0 и более поздние версии;
- Safari 1.2 и более поздние версии;
- Opera 9.25 и более поздние версии;
- Chrome.

В случае отсутствия поддержки средство вывода полноразмерного изображения изящно самоустраняется. Если Web-обозреватель посетителя его не поддерживает, он будет руководствоваться значением атрибута href.

```
<a href="trammell_shoes.jpg" rel="lightbox"
title="Trammell shows off his happy shows.">
<img src="tremmell_shoes_tn.jpg" alt="Mark Trammel is happy with his shoes." /></a>
```

В этом примере Web-обозреватель выводит на экран файл trammell\_shoes.jpg.

#### См. также

Статью "Learn 3 Excellent JavaScript Libraries at Once" (Изучите 3 отличные библиотеки JavaScript одновременно) на странице http://net.tutsplus.com/tutorials/javascript-ajax/learn-3-excellent-javascript-libraries-at-once/.

## 5.9. Изменение непрозрачности элементов

#### Проблема

Вы хотите изменить непрозрачность или прозрачность элемента.

#### Решение

В спецификации CSS есть свойство орасіту, которое очень просто применять (рис. 5.20):

```
#number4 {
    opacity: .4; /* .4 = 40% прозрачности */
    filter: alpha(opacity=40); /* 40 = 40% прозрачности */
}
```

#### Обсуждение

Значение .4 в свойстве opacity означает элемент на 40 % непрозрачный. Значение 0 указывает на невидимый элемент, а значение 1 — на отсутствие прозрачности.

В собственном свойстве Internet Explorer filter необходимо задавать значение, равное степени прозрачности в процентах. Непрозрачность для фильтра alpha принимает значения от 0 до 100. 0 означает невидимый элемент, а 100 — отсутствие прозрачности.

#### Примечание

Непрозрачность меняет все содержимое блочного элемента, в то время как задание непрозрачности с помощью функции RGBA (см. *разд. 5.10*) изменяет непрозрачность самого элемента.



Рис. 5.20. Реализация прозрачности цифры 4 и квадрата

#### Поддержка Web-обозревателями

В настоящее время фильтр непрозрачности поддерживается в Firefox 1.5, Opera 9, Safari 1.2 и более поздних версиях этих обозревателей, а также в Chrome.

В Internet Explorer 5.5 для Windows и его более поздних версиях для создания эффекта, совместимого с другими Web-обозревателями, требуется применение его собственного фильтра alpha.

Поскольку свойство filter частное, правило CSS некорректно и таблица стилей, содержащая его, не пройдет проверку на допустимость. Выход — перенести специфичные для IE стилевые правила в условные комментарии и применять эти правила только в Internet Explorer.

#### Предупреждение

Недостаток применения фильтра непрозрачности состоит в том, что значение фильтра наследуется. Если родительский элемент прозрачен на 10 %, прозрачность дочерних элементов также будет равна 10 %. Внимательно следите за разборчивостью вашей Webстраницы.

#### См. также

Свойство opacity в спецификации CSS3 на странице http://www.w3.org/TR/css-3color/#transparency; *paзd 5.10* о задании непрозрачности цвета фона элемента; *paзd. 4.14* об установке параметров визуализации изображений Web-обозревателем.

## 5.10. Регулировка непрозрачности цвета фона

#### Проблема

Вы хотите задать непрозрачность цвета фона элемента.

#### Решение

Задайте прозрачность цвета фона элемента с помощью значения RGBA (рис. 5.21):

```
#number4 {
   background-color: rgba(255, 255, 0, .4);
}
```



#### Обсуждение

Web-обозреватели Firefox 3, Opera 10 и их более поздние версии, а также Safari поддерживают функцию гgba для задания цвета фона вместе со значением прозрачности. При разработке, предполагающей работу в разных Web-обозревателях, первым задайте свойство background-color с традиционным кодированием цвета (RGB, шестнадцатеричное значение и т. п.), а затем используйте следом другое свойство background-color со значением RGBA:

```
#number4 {
  background-color: rgb(255, 255, 0);
  background-color: rgba(255, 255, 0, .4);
}
```

Это позволит Web-обозревателям Internet Explorer и Firefox 2 по крайней мере отобразить цвет фона, в то время как пользователи Safari, Firefox 3, Opera 10 и их более поздних версий увидят прозрачность. Другая тактика — не применять значения цвета, а вместо них использовать маленькое изображение в формате PNG, обработанное в программе графического редактора, такой как Adobe Photoshop или Adobe Fireworks, и заданное в свойстве background-image. Дополнительную информацию об этом приеме см. в *разд. 4.5*.

#### Поддержка в Internet Explorer

В Internet Explorer 5.5 и более поздних версиях можно создать прозрачность фонового цвета с помощью свойства filter, предназначенного для формирования градиента.

Сначала преобразуйте RGB-значение цвета в шестнадцатеричное. В данном примере значение rgb(255, 255,0) преобразуется в #FFFF00.

Затем превратите значение альфа-прозрачности в строку с шестнадцатеричным значением (табл. 5.1). В примере это значение 66.

Значение альфа-прозрачности	Шестнадцатеричное значение
0	0
0.1	1A
0.2	33
0.3	4D
0.4	66
0.5	7F
0.6	99
0.7	В3
0.8	СС
0.9	E5
1	FF

Таблица 5.1. Таблица преобразования альфа-прозрачности

Далее сформируйте одну строку из шестнадцатеричных значений прозрачности и цвета фона, начав со значения прозрачности: #66FFFF00.

Создайте для элемента отдельное правило CSS, задав для цвета фона значение transparent:

```
#number4 {
   background-color: transparent;
}
```

Затем примените строку с шестнадцатеричными значениями прозрачности и цвета в свойстве создания градиента filter:

```
#number4 {
   background-color: transparent;
   filter:progid:
   DXImageTransform.Microsoft.gradient(startColorstr=#66FFFF00,
   endColorstr=#66FFFF00);
}
```

Поскольку это градиент, вы можете задать переход от одного значения цвета к другому. Но у этого специализированного свойства есть и другое применение. Указав один и тот же начальный и конечный цвет и значение прозрачности, можно получить прозрачный цвет, отображающийся в разных Web-обозревателях.

Далее добавьте свойство zoom со значением 1, чтобы заставить IE отображать эффект или показывать элемент с псевдосвойством "hasLayout" (рис. 5.22):

```
#number4 {
  background-color: transparent;
  filter:progid:DXImageTransform.Microsoft.gradient(startColorstr=#66FFFF00,
  endColorstr=#66FFFF00);
  zoom: 1;
}
```

#### Примечание

Характеристика hasLayout применяется только в Internet Explorer 7 и более ранних версиях. Некоторые элементы ведут себя по-разному при наличии или отсутствии "layout" (схемы размещения).

Для решения возникающих проблем это псевдосвойство активизируется с помощью ряда селекторов CSS, один из них — свойство zoom. Применение свойства zoom для установки псевдосвойства hasLayout характерно для IE и игнорируется другими Webобозревателями. В некоторых примерах таблиц стилей можно найти свойство zoom со значением 1, используемое только для того, чтобы заставить более ранние версии IE отображать элементы так, как будто у них есть "layout".

Дополнительную информацию о псевдосвойстве hasLayout см. на странице http://msdn.microsoft.com/en-us/library/bb250481(VS.85,loband).aspx.

При наличии правила CSS, применяющего специализированное свойство, мы можем поместить код в условный комментарий, что позволит обрабатывать это правило только Web-обозревателям IE:

```
<!--[if IE]>
<style type="text/css">
#number4 {
    background-color: transparent;
    filter:progid:DXImageTransform.Microsoft.gradient(startColorstr=#66FFFF00,
    endColorstr=#66FFFF00);
    zoom: 1;
}
</style>
```

```
<![endif]-->
```



Рис. 5.22. Прозрачность фона в IE6

#### См. также

Описание фильтра градиента в спецификации MSDN на странице http:// msdn.microsoft.com/en-us/library/ms532997(VS.85).aspx; конвертер RGB-цвета в строку с шестнадцатеричным значением на странице http://www.javascripter.net/ faq/rgbtohex.htm; *pa3d*. 4.14 об установке параметров Web-обозревателя для визуализации изображений; *pa3d*. 5.9 об изменении непрозрачности или прозрачности элемента.

## глава 6 Списки

## 6.0. Введение

Списки помогают людям оставаться собранными и организованными и в том случае, когда жена вручает мужу на пороге дома список продуктов для покупки и когда музы-кальный канал представляет 100 худших песен всех времен.

В Web-дизайне происходит то же самое.

HTML-списки, группируя ключевые элементы, облегчают представление организованной информации посетителям нашего сайта. Они привлекательны отчасти благодаря способу их отображения на странице.

Стандартно элементы списка имеют абзацные отступы и помечаются маркерами, как правило, закрашенными кружками в случае маркированного списка или числами в нумерованных списках (рис. 6.1).

С помощью нескольких строк на языке HTML Web-программист может создать маркированный список на Web-странице, не открывая графического редактора. Технология CSS позволяет создавать гораздо более привлекательные списки.



С помощью нескольких простых правил CSS Web-разработчики могут моделировать представление списка, добиваясь совершенства дизайна Web-страницы, а не полагаться на стандартное стилевое оформление Web-обозревателя.

В этой главе показано, как изменить нумерацию элементов списка, применить собственное изображение в качестве маркера элементов списка, создать обратный отступ, не использующий маркеры списка, и т. д.

## 6.1. Изменение формата списка

## Проблема

Вы хотите изменить стандартный стиль списка; например, изменить маркер или нумерацию (рис. 6.2).



Рис. 6.2. Маркеры списка заменены строчными римскими цифрами

## Решение

Используйте свойство list-style-type для изменения маркера или типа счетчика:

```
li {
   list-style-type: lower-roman;
}
```

## Обсуждение

В спецификации CSS 2.1 предлагается несколько стилей нумерации списка, приведенных в табл. 6.1. Web-обозреватели обычно изменяют тип маркера при переходе от одного уровня списка к другому, вложенному уровню. Для того чтобы помешать отображению традиционной системы маркировки списка, измените значение свойства list-style-type в каждом списке-потомке.

Таблица 6.1. Стили маркеров

Стиль/значение	Описание	Поддержка в Web-обозревателях
square	Обычно закрашенный квадратик, хотя точное представление не определено	Во всех основных Web-обозревателях
disc	Обычно закрашенный кружок, хотя точное представление не определено	Во всех основных Web-обозревателях
circle	Обычно не заполненный кружок, хотя точное представление не определено	Во всех основных Web-обозревателях
decimal	Начинается с 1, затем идет 2, 3, 4 и т. д.	Во всех основных Web-обозревателях
decimal-leading-zero	Начинается с 01, затем идет 02, 03, 04 и т. д. Количество ведущих нулей может равняться числу значащих цифр, используемых в номерах элементов списка. Например, 0001 может применяться в списке из 5 876 элементов	Во всех основных Web-обозревателях
lower-roman	Начинается со строчных римских цифр	Во всех основных Web-обозревателях
upper-roman	Начинается с заглавных римских цифр	Во всех основных Web-обозревателях
lower-alpha	Начинается со строчных букв таблицы кодировки ASCII	Во всех основных Web-обозревателях
upper-alpha	Начинается с заглавных букв таблицы кодировки ASCII	Во всех основных Web-обозревателях
lower-latin	Начинается со строчных букв таблицы кодировки ASCII	Во всех основных Web-обозревателях
upper-latin	Начинается с заглавных букв таблицы кодировки ASCII	Во всех основных Web-обозревателях
lower-greek	Начинается с классических греческих букв: альфа, бета, гамма и т. д.	Safari, Firefox, IE8, Opera
hebrew	Начинается с символов традиционного иврита	Safari, Firefox
hiragana	Начинает подсчет в японской системе hiragana	Firefox
katakana	Начинает подсчет в японской системе katakana	Firefox, Safari, Chrome
hiragana-iroha	Начинает подсчет в японской системе hiragana-iroha	Firefox, Safari, Chrome
none	Маркер не отображается	Во всех основных Web-обозревателях

#### См. также

В разд. 6.9 использование пользовательских изображений в качестве маркеров списка; главу 12 "Lists and Generated Content" (Списки и генерируемый контент) в книге Эрика Мейера (Eric Meyer) "Cascading Style Sheets: The Definitive Guide" (Каскадные таблицы стилей: подробное руководство) (O'Reilly).

## 6.2. Изменение цвета маркера списка

## Проблема

Вы хотите изменить цвет маркера списка без добавления графики.

## Решение

Сначала вставьте текст из элементов списка в отдельные теги span:

```
<span>I'm Not the Same Person I was in the Database</span>
<span>Past Breaches of Our Privacy</span>
The Best of Intentions
Whatever Happened to Automation?
The Smart Choice is Not Needing to Make One
```

Задайте цвет списка с помощью универсального селектора типа:

```
ul {
color: #F00;
}
```

Затем для задания цвета текста в элементе span, содержащем пункт списка, примените селектор потомка (рис. 6.3):

```
ul span {
  color: black;
}
```

## Обсуждение

Когда вы с помощью свойства color задаете цвет маркера, присоединенного к элементу списка, текст в этом пункте списка наследует цвет, как показано в нижних пунктах списка на рис. 6.3.

Для предотвращения наследования вы должны вставить элемент span со своим собственным свойством color.

#### См. также

*Разд. 6.6* о вставке маркера в виде пользовательского изображения, в котором предложено альтернативное решение, не требующее дополнительных элементов span и добивающееся того же эффекта.

				s/cniisto	pher/besktop/ T	000	91
CS	S Cookbook	0	Christopher Schmitt : D	) ©	ද්ය Technology Books, Tech	0	+
hla	of Co	ntont	e				
IDIC		nun	.5				
propose	d, the contents	s of the pap	per will contain the fo	llowing	sections:		
• I'm N	Not the Same F	erson I wa	as in the Database				
Past	Breaches of O Best of Intentio	ur Privacy	'				
• What	tever Happene	d to Autor	mation?				
• The s	Smart Choice	is Not Nee	eding to Make One				
nere are	any objection	s to how th	hese sections are divid	led, plea	ase let <u>Nicholas</u> know abou	t it.	

Рис. 6.3. Теперь маркеры выделены цветом, но только два первых пункта списка отображаются другим цветом

## 6.3. Задание в списке отступов, одинаково отображаемых в разных Web-обозревателях

#### Проблема

Разные Web-обозреватели применяют различные методы для формирования отступов в списках. Вы хотите задать в вашем списке левые поля, которые будут отображаться во всех Web-обозревателях одинаково.

#### Решение

Задайте свойства margin-left И padding-left в Элементе ul:

```
ul {
  margin-left: 40px;
  padding-left: 0px;
}
```

### Обсуждение

Разные Web-обозреватели применяют различные методы для формирования отступов, или пробельных зон, в списке.

Web-обозреватели Firefox, Chrome и Safari создают пробельные зоны в списке с помощью свойства padding, a Internet Explorer и Opera — на основе свойства списка margin. Для повышения универсальности отображения в разных Web-обозревателях необходимо задать значения как для левого поля, так и для левого отступа списка. Укажите конкретное значение отступа в одном из этих свойств. Если разделить эту величину между значениями разных свойств, отображение в разных Web-обозревателях будет несогласованным.

#### Отсутствующие маркеры

Если задать нулевые значения поля и отступа в списке, содержащемся непосредственно в элементе body, Web-обозреватель отобразит маркеры за пределами окна просмотра обозревателя, сделав их невидимыми для пользователя. Для того чтобы гарантировать видимость маркеров, задайте левое поле или левый отступ элемента ul, по крайней мере, равным 1 ет.

#### См. также

*Paзд. 6.11* о создании обратных отступов; свойство padding в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/box.html#propdef-padding; свойство margin в спецификации CSS 2.1 на http://www.w3.org/TR/CSS21/box.html#propdef-margin.

# 6.4. Установка разделителей между элементами списка

#### Проблема

Вы хотите создать разделители между пунктами списка.

#### Решение

Воспользуйтесь свойством border для формирования видимого разделителя:

```
li {
    border-top: 1px solid black;
    padding: .3em 0;
}
```

Затем примените свойство border для создания нижней границы элемента ul (рис. 6.4):

```
ul {
  margin-left: 40px;
  padding-left: 0px;
  border-bottom: 1px solid black;
  list-style: none;
  width: 36%;
}
```

00	CSS Cookbook	C
🕩 🕞 , C X 🍙 🖨	1Parsword Google	Q
CSS Cookbook 😵 🚺 Christop	oher Schmitt : De 😧 🎘 Technology Books, Tech 😵 🗧	H.
able of Contents		
proposed, the contents of the paper will co	ontain the following sections:	
I'm Not the Same Person I was in the		
I'm Not the Same Person I was in the Database		
Past Breaches of Our Privacy		
Past Breaches of Our Privacy The Best of Intentions		
The Best of Intentions Whatever Happened to Automation?		
The Best of Intentions Whatever Happened to Automation? The Smart Choice is Not Needing to		

Рис. 6.4. Разделители, помещенные между пунктами списка

#### Обсуждение

В соответствии с блочной моделью отступы находятся внутри рамки и полей. Задавая величину отступа, вы делаете нижнюю границу маркированного списка охватывающей пустое пространство слева от пунктов списка и ширину списка.

Для согласования длин разделителей задавайте значение только в свойстве margin-left маркированного списка. В противном случае длина разделителей элементов списка li и маркированного списка ul будет разной.

Например, если отступы для списка задаются с помощью свойства padding-left, нижняя граница списка будет длиннее разделителей между элементами (рис. 6.5).

```
li {
  border-top: 1px solid black;
  padding: .3em 0;
}
ul {
  margin-left: 0px;
  padding-left: 40px;
  border-bottom: 1px solid black;
  list-style: none;
  width: 36%;
}
```

#### См. также

*Разд. 6.3* о создании отступов в списке, отображаемых одинаково в разных Webобозревателях.



Рис. 6.5. Нижний разделитель длиннее всех остальных

# 6.5. Создание пользовательских текстовых маркеров в списках

### Проблема

Вы хотите использовать собственные текстовые маркеры в списке.

### Решение

Создайте абзацный отступ, вставьте пользовательский текст и как автоматически генерируемое содержимое (autogenerated content) закрывающие кавычки "елочки", играющие роль маркеров (рис. 6.6):

```
ul {
    list-style: none;
    margin: 0;
    padding: 0 0 0 1em;
    text-indent: -1em;
}
li {
    width: 33%;
    padding: 0;
    margin: 0 0 0.25em 0;
}
li:before {
    content: "\00BB \0020";
}
```



Рис. 6.6. Текстовый маркер для списка

#### Обсуждение

Задание значения none в свойстве list-style скрывает маркер списка, связанный с ним по умолчанию. Обычно маркер добавляется слева от каждого элемента списка.

Вместо присоединения маркера к элементу списка, пользовательский текстовый маркер помещается внутристрочно в текст элемента списка. Поскольку текстовый маркер находится внутри элемента списка, вам нужно каким-то образом вытолкнуть стандартный маркер из блока элемента списка. Эту задачу выполняет установка отрицательного значения отступа первой строки.

Отрицательная величина в свойстве text-indent сдвигает первую строку влево, а положительная — вправо:

```
ul {
  list-style: none;
  margin: 0;
  padding: 0 0 0 1em;
  text-indent: -1em;
}
```

Псевдоэлемент :before генерирует текстовый маркер. Информационное содержимое, состоящее из простых клавиатурных символов, может быть легко вставлено следующим образом:

```
li:before {
  content: "> ";
}
```

Однако для встраивания специальных символов спецификация CSS 2.1 требует указания специальных значений Unicode (ISO 10646). Таким образом, вы должны задать символ с помощью его *шестнадцатеричного escape-эквивалента Unicode*, а не обычного символа HTML 4, такого как » (см. http://www.alanwood.net/demos/ ansi.html).

Для обособления этих значений в CSS используют знак обратного слэша (\) перед каждым шестнадцатеричным значением Unicode:

```
li:before {
  content: "\00BB \0020";
}
```

Когда книга готовилась к печати, это решение действовало в Web-обозревателях Firefox, Safari, Chrome и Opera, т. к. они могут поддерживать создание автоматически генерируемого содержимого. К сожалению, в этот перечень не вошли версии Internet Explorer для Windows, предшествующие IE8, поскольку они не способны обрабатывать автоматически генерируемый контент.

Для того чтобы иметь возможность использовать решение в разных Webобозревателях, не применяйте автоматически генерируемое содержимое. Вместо него вставьте текстовый маркер вручную перед элементом списка:

```
» I'm not the Same Person I was in the Database
» Past Breaches of Our Privacy
» The Best of Intentions
» Whatever Happened to Automation?
» The Smart Choice is Not Needing to Make One
```

Главный недостаток такого подхода состоит в том, что у вас будет два маркера для каждого элемента списка (маркер списка, генерируемый Web-обозревателем, и маркер, включенный в текст вручную), если в Web-обозревателе отключена поддержка CSS и пользователь видит только информационное содержимое страницы. Но эту проблему нельзя считать критически важной, просто на Web-странице отображается ненужный элемент оформления.

#### См. также

В спецификации CSS 2.1 описание escape-символов на странице http://www.w3.org/ TR/REC-CSS2/syndata.html#escaped-characters.

## 6.6. Создание пользовательских маркеров-изображений в списках

#### Проблема

Вы хотите создать свои графические маркеры списка. Например, на рис. 6.7 применяется изображение ромба.

	0				CSS C	ookbook		(
<b>Back</b>	Forward I	C Reload	Stop	Home	Print	1P 1Password		gle 🕻
]	CSS Cookboo	ok (	8 🚺	Christe	opher Sc	hmitt : 🤅	Technology Books, Te 😒	+
[a]	ble of (	Con	ten	ts				
	posed the ea	ntanta	of the r	onor W	ill cont	ain the foll	wing sections:	
s pro	poseu, me co	ments	or the p	aper w	in cond		Jwing sections.	
-	I'm Not the S	ame Per	rson I	was in t	he Data	abase		
+ 1	Past Breaches	s of Our	r Privad	су				
* '	The Best of I	ntentior	ns					
*	Whatever Ha	ppened	to Aut	omatio	n?			
	The Smart Ch	noice is	Not N	eeding	to Mak	e One		
*				these s	actions	ore divide	l alana lat Niakalas kasu akau	
f there	e are any obje	ections 1	to how	these s	ecuons	are urviue	, please let <u>inicholas</u> know abou	IT 11.
f there	e are any obje	ections	to how	these s	ecuons	are divide	i, please let <u>Micholas</u> know abou	it it.
f there	e are any obje	ections	to how	uiese s	ecuons	are divide	i, please let <u>ivienolas</u> know abou	it 1t.

#### Решение

Для применения графического маркера пунктов списка используйте свойство liststyle-image:

```
ul {
   list-style-type: disc;
   list-style-image: url(bullet.gif);
}
```

### Обсуждение

Задайте местоположение изображения, которое вы хотите использовать в качестве маркера, как значение в свойстве list-style-image. У вас нет возможности в таблице стилей CSS управлять размером изображения, используемого как маркер, поэтому у задаваемого изображения уже должен быть подходящий размер.

Слишком крупные изображения могут ухудшать удобочитаемость элемента списка или маркер может не выводиться целиком в окне просмотра (рис. 6.8). Создавая собственные маркеры, убедитесь в том, что у них подходящий размер, и они сочетаются с дизайном вашей Web-страницы.

#### Блокирование наследования

Значение графического маркера *наследуется*, т. е. вложенные списки выбирают изображение в качестве маркера так же, как родительский элемент.

Отказаться от наследования можно, задав значение none в дочерних списках.

```
ul {
   list-style-type: disc;
```

```
list-style-image: url(bullet.gif);
}
ul ul {list-style-image: none;}
```

Всегда вставляйте свойство list-style-type на случай невозможности использования изображения. В данном решении применяется маркер списка disc, если изображение bullet.gif не может быть отображено.



Рис. 6.8. Большое изображение, использующееся как маркер, выводится не полностью

#### См. также

*Разд. 6.5* о создании пользовательских текстовых маркеров; свойство list-style-image в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/generate.html# propdef-list-style-image.

# 6.7. Вставка больших пользовательских маркеров-изображений в списки

## Проблема

Вы хотите применять пользовательские изображения большого размера в качестве маркеров списка, не сталкиваясь с ограничениями, возникающими при использовании свойства list-style-image.

### Решение

Сначала удалите маркер, используемый по умолчанию, с помощью свойства list-style (рис. 6.9):

```
ul {
  margin-left: 40px;
  padding-left: 0px;
  list-style: none;
}
```



Рис. 6.9. Стандартные маркеры списка удалены

Отведите достаточно свободного пространства слева от элемента списка для вставки нового маркера-изображения (рис. 6.10):

```
ul {
  margin-left: 40px;
  padding-left: 0px;
  list-style: none;
}
```

```
li {
padding: .3em 0 1em 40px;
 font: 1.1em/1.2 Verdana, Arial, Verdana, sans-serif;
ł
       000
                                             CSS Cookbook
        .
              (C)
                             X
                                   ( 1
                                         (1P)
                                                         📋 (file:///Usi 😭 🔻 ) - 🚱 🖌 Google
                              Stop
                                               1Password
        Back
             Forward
                       Reload
                                   Home
                                          Print
                               0
                                      Christopher Schmitt : De...
                                                              🚑 Technology Books, Tech ...
               CSS Cookbook
```

## **Table of Contents**

As proposed, the contents of the paper will contain the following sections:

I'm Not the Same Person I was in the Database

Past Breaches of Our Privacy

The Best of Intentions

Whatever Happened to Automation?

The Smart Choice is Not Needing to Make One

If there are any objections to how these sections are divided, please let Nicholas know about it.

Рис. 6.10. Слева от списка установлен дополнительный отступ

Затем вставьте новый пользовательский маркер с помощью свойства background (рис. 6.11):

```
ul {
margin-left: 40px;
padding-left: 0px;
list-style: none;
}
li {
padding: .3em 0 1em 40px;
font: 1.1em/1.2 Verdana, Arial, Verdana, sans-serif;
background: url(search 32.gif) no-repeat;
}
```

#### Обсуждение

Применение свойства background для создания улучшенного отображения списка предоставляет больше возможностей, чем использование свойства list-style-image. С помощью этого метода можно применять маркер списка любого размера до тех пор, пока слева от элемента списка отведено достаточно свободного пространства.

0

Q

0



Рис. 6.11. Список с большим изображением в качестве маркера

#### Примечание

Если используется прозрачное изображение, может потребоваться задание цвета фона.

#### См. также

Разд. 6.8 с более сложным вариантом предложенного решения.

## 6.8. Улучшение представления списка с помощью графики

#### Проблема

Вы хотите привлечь дополнительное внимание к списку с помощью ярких изображений.

#### Решение

Вставьте фоновые изображения в элементы ul и li.

Сначала создайте фоновое изображение для набора элементов маркированного списка и изображение для маркера списка (рис. 6.12).

Далее задайте маркированный список, чтобы ввести в него фоновое изображение. Также включите в него свойство width со значением, равным ширине фонового изображения (рис. 6.13):

```
ul {
  background: url(list-bkgd.gif) bottom;
  width: 298px;
  list-style: none;
  padding: 0 0 12px;
  margin: 0;
}
```



Рис. 6.12. Два изображения для пользовательского представления списка

Затем вставьте маркер списка в элемент списка. Задайте нижнюю границу как разделитель между элементами списка (рис. 6.14):

```
ul {
  background: url(list-bkgd.gif) bottom;
  width: 298px;
  list-style: none;
  padding: 0 0 12px;
  margin: 0;
  }
li {
  color: #eee;
  font-family: Verdana, Arial, Verdana, sans-serif;
```

```
padding: 7px 7px 7px 20px;
border-bottom: 1px solid #888;
background: url(list-marker.gif) no-repeat 5px .8em;
}
```



Рис. 6.13. Задано фоновое изображение для списка в целом



#### Обсуждение

Это решение объединило несколько разных приемов.

В первом фрагменте решения описана вставка фонового изображения в элемент ul. Поскольку у изображения есть установленные ширина и высота, обязательно задайте ширину в таблице CSS.

Что касается высоты, то у Web-разработчиков есть много причин не задавать эту характеристику. Пользователь может увеличить размер стандартного шрифта, сделав текст крупнее. Кроме того, стиль списка может применяться к спискам с малым или большим числом элементов.

Для того чтобы обеспечить удовлетворительный результат в любой ситуации, фоновое изображение должно иметь большую высоту.

В данном решении у фонового изображения высота равна 465 пикселов и ее более чем достаточно для нормального отображения нескольких пунктов списка. Даже если в каком-либо Web-обозревателе установлены шрифты большего размера, дизайн страницы не разрушится, как показано на рис. 6.15.

Поскольку у фонового изображения имеются скругленные углы на нижнем крае, задан нижний отступ 12 пикселов, чтобы элементы списка не накладывались на углы.


Кроме того, изображение прижато к нижнему краю списка. Это позволяет всегда выводить скругленные нижние углы изображения, даже если увеличены размер шрифта или возросло количество элементов списка.

Далее для отображения элементов списка использовано два приема. Во-первых, между элементами списка помещены разделители. В отличие от решения в *разд. 6.3*, не нужен разделитель для нижней границы элемента ul. Во-вторых, вставлены маркеры списка методом, описанным в *разд. 6.7*.

#### См. также

В главе 7 способы преобразования этого текста в работающее навигационное меню.

# 6.9. Создание внутристрочных списков

#### Проблема

Вы хотите перечислять пункты в одном абзаце и представить их в виде HTML-списка ul с выводом элементов полужирным шрифтом и разделением пунктов списка запятыми, как показано на рис. 6.16.



Рис. 6.16. Список, отображаемый внутри абзаца

#### Решение

Задайте абзацы перед списком (и если нужно, после списка):

```
<h3>
Table of Contents
</h3>
As proposed, the contents of the paper will contain the
following sections:
```

#### Списки

```
I'm not the Same Person I was in the Database
Past Breaches of Our Privacy
Past Breaches of Intentions
The Best of Intentions
Ii>Whatever Happened to Automation?
class="last">The Smart Choice is Not Needing to Make One

If there are any objections to how these sections are divided,
please let <a href="mailto:nick@example.com">Nicholas</a> know about
it.
```

В таблице CSS задайте отображение абзацев как внутристрочных элементов, а затем воспользуйтесь автоматически генерируемым содержимым для отображения запятых между пунктами списка и точки в конце списка:

```
ul, li {
 display: inline;
 margin: 0;
 padding: 0;
 font-weight: bold;
 font-style: italic;
ļ
li:after {
 content: ", ";
}
li.last:after {
 content: ".";
}
p {
 display: inline;
}
```

#### Обсуждение

В этом методе сохраняется структура абзацев и списков, но шире используются возможности технологии CSS, позволяющие представить список внутри абзаца. Однако, выбрав размещение списка внутри абзаца, вы жертвуете наглядностью представления списка.

Важная часть данного решения — установка значения inline в свойстве display для элементов списка и абзацев. Это значение позволяет разместить элементы в одной строке вместо отделения их друг от друга пробельными зонами сверху и снизу.

#### Предупреждение

Web-обозреватель Internet Explorer 7 для Windows и его более ранние версии не поддерживают автоматически генерируемое содержимое.

Свойство display в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/ visuren.html#propdef-display.

# 6.10. Создание обратных отступов в списке

#### Проблема

Вы хотите, чтобы первая строка элемента списка начиналась левее остальных строк, создавая таким образом обратный отступ, показанный на рис. 6.17.



Рис. 6.17. Обратные отступы в списке

#### Решение

Используйте отрицательное значение в свойстве text-indent:

```
ul {
  width: 30%;
  padding: 0 0 0.75em 0;
  margin: 0;
  list-style: none;
  }
  li {
   text-indent: -0.75em;
  margin: 0.33em 0.5em 0.5em 1.5em;
  }
```

#### Обсуждение

Несмотря на то, что маркеры (числовые, графические или текстовые) помогают привлечь внимание к имеющемуся списку, иногда возникает желание избавиться от этих элементов оформления. Для того чтобы не зависеть от маркеров при стилевом оформлении списка, воспользуйтесь обратным отступом.

В данном решении вы уменьшаете отступ списка на <sup>3</sup>/<sub>4</sub> em, создавая видимый, но незначительный выступ. Вы можете сделать этот стилевой прием едва заметным или определяющим, уменьшив в дальнейшем значение свойства text-indent или увеличив размер шрифта в элементе списка.

#### См. также

*Разд. 3.24* о задании отступов абзаца; свойство text-indent в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/text.html#propdef-text-indent.

# 6.11. Смещение маркера внутрь списка

## Проблема

Вы хотите сместить маркер внутрь блока элемента списка, как показано на рис. 6.18. Это создает эффект обтекания маркера текстом.



#### Решение

Воспользуйтесь свойством list-style-position и задайте значение inside:

```
li {
    list-style-position: inside;
    width: 33%;
    padding: 0;
    margin: 0;
}
ul {
    margin: 0;
    padding: 0 0 0 1em;
}
```

## Обсуждение

Обычно маркер стоит снаружи от текста, и в результате получается четкий список. В некоторых дизайнерских решениях иногда требуется вывод маркера как части текста. Дизайнер может выбрать такой вариант, например, для того чтобы устранить нехватку свободного пространства слева от списка.

Замена стандартного маркера списка своим собственным маркером в этом случае может улучшить визуальное представление данного решения. Например, на рис. 6.19 показаны ромбы вместо стандартного знака маркера.



Свойство list-style-position в спецификации CSS 2.1 на странице http://www.w3.org/ TR/CSS21/generate.html#propdef-list-style-position.

# 6.12. Стилевое оформление списка определений

#### Проблема

Вы хотите разместить в одной строке термины и начальные строки их определений из стандартного списка определений.

#### Решение

Создайте корректный список определений (рис. 6.20) средствами языка HTML:



Рис. 6.20. Стандартное отображение списка определений

Затем создайте поле слева от всего списка определений:

```
dl {
  margin-left: 5em;
}
```

Задайте ширину для вывода определяемых терминов (элементов dt), меньшую чем заданная величина левого поля (рис. 6.21):

```
dt {
  width: 4em;
}
```



Рис. 6.21. Ширина определяемых терминов, равная 4 ет

Далее задайте свойства float и clear для определяемого термина, а также используйте отрицательное значение для левого поля, чтобы расположить термин слева от его определения, как показано на рис. 6.22:

```
dt {
  width: 4em;
  float: left;
  clear: left;
  margin:0 0 1em -5em;
  font-weight: bold;
}
```

Для определений (элементов dd) задайте плавающее смещение влево и ширину, равную 100 % (рис. 6.23):

```
dd {
  float: left;
  width: 100%;
}
```



Рис. 6.22. Сдвиг термина влево от его определения



Рис. 6.23. Корректировка ширины вывода для определений

Затем откорректируйте поле и отступ для усиления визуального разделения определений, как показано на рис. 6.24:

```
dd {
  float:left;
  width:100%;
  padding: .2em 0 0 0;
  margin: 0 0 1em 0;
}
```

00	CSS Cookbook				
	C X 👚 1P 🗋 file:///Volumes/christopher/Docume 🏠 🔻 - 🚷 Google Q				
	CSS Cookbook 🛛 🕄 Christopher Schmitt : De 🥹 🖉 Technology Books, Tech 😒 🕇 🗮				
Defi	nition Terms				
CSS	Cascading Style Sheets				
HTML	HyperText Markup Language				
	How To Meet Ladies				
	Epsum factorial non deposit quid pro quo hic escorol. Olypian quarrels et gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum e pluribus unum. Defacto lingo est igpay atinlay. Marquee selectus non provisio incongruous feline nolo contendre.				
Done	1				

Рис. 6.24. Корректировка полей и отступов для определений

После этого добавьте стилевое оформление элементов для улучшения внешнего вида списка (рис. 6.25):

```
dt {
width: 4em;
 float: left;
 clear: left;
margin:0 0 1em -5em;
 font-weight: bold;
border-top: 1px solid #000;
padding: .2em 0 0 0;
}
dd {
 float: left;
width: 100%;
padding: .2em 0 0 0;
margin: 0 0 1em 0;
color: #333;
}
dt+dd {
border-top: 1px solid #000;
}
```

#### Обсуждение

Расположение термина рядом с его определением — очень распространенное решение. Задавая поле слева от всего списка определений, вы можете заставить термины сместиться в свободную зону. После этого использование плавающей модели (наряду с продуманным значением отступа) завершает обработку списка.

į	CSS Cookbook 💿 💽 Christopher Schmitt : De 😨 🛵 Technology Books, Tech 😨			
efi	nition Terms			
20	Concoding Style Shoots			
55				
TML	HyperText Markup Language			
	How To Meet Ladies			
	Epsum factorial non deposit quid pro quo hic escorol. Olypian quarrels et gorilla congoli sic ad nauseum. Souvlaki ignitus carborundum e pluribus unum. Defacto lingo est igpay atinlay. Marquee selectus non provisio incongruous feline nolo contendre.			

Рис. 6.25. Шлифовка внешнего вида списка определений

#### Применение генерируемого содержимого

Для того чтобы показать, что за термином последует определение, используйте в определяемом термине псевдоэлемент :after:

```
dt:after {
   content: ":";
}
```

Поскольку термины могут иметь несколько определений, можно каждому определению присвоить номер. В спецификации CSS есть средство подсчета, подходящее для этой цели.

Сначала в элементе dt примените свойство counter-reset:

```
dt {
  counter-reset: item;
}
```

Свойство counter-reset создает или сбрасывает в начальное состояние счетчик. По мере отображения элементов dt и при установленной связи CSS с элементом счетчик инициализируется и затем последовательно возвращается в исходное состояние при каждом отображении данного элемента в документе.

Следующий шаг — заставить Web-обозреватель выводить номер перед каждым определением с помощью функции counters ():

```
dd:before {
   content: counters(item, "") ". ";
}
```

В функции counters() передаются два параметра: счетчик, который будет использоваться, и строка. Строка применяется для разделения списка на уровни или подсписки. Примерами разделителей в счетчике могут быть точка или дефис. В данном решении нет подсписков, поэтому строка остается пустой.

Для вставки после номера точки и пробела после функции counters() приводится строка в кавычках.

Для того чтобы в документе выводился нужный номер, необходимо, чтобы счетчик помечал каждое новое определение. Делается это с помощью свойства counter-increment, которое принимает значение счетчика с именем, заданным в свойстве counter-reset:

```
dd:before {
   content: counters(item, "") ". ";
   counter-increment:item;
}
```

Окончательный результат показан на рис. 6.26.



Рис. 6.26. Применение генерируемого содержимого в списке определений

#### Примечание

Генерируемое содержимое не поддерживается в версиях Internet Explorer для Windows, предшествующих IE8. Все остальные современные Web-обозреватели поддерживают генерируемое содержимое.

#### См. также

Оригинальную работу Роберта O'Рурка (Robert O'Rourke) о представлении списка определений в виде таблицы на странице http://www.sanchothefat.com/dev/layouts/d efinition-lists-ugly.html, вдохновителем которой стал раздел CSS Challenge (проблемы CSS) на сайте Брюса Лоусона (Bruce Lawson) по адресу http://www.brucelawson.co.uk/ 2009/css-challenge/.

# 6.13. Традиционное отображение киносценария с помощью элемента языка HTML5 *dialog*

#### Проблема

Вы хотите вывести на экран киносценарий в традиционном виде.

#### Решение

Pазметьте контент сценария с помощью элемента языка HTML5 dialog:

```
<div id="screenplay">
<h3>Cut to</h3>
Int. Kitchen - Continuous
<dialog>
<dt>Beth</dt>
<dd> I told you the one about Salma Hayek?</dd>
</dialog>
Beth walks closer to John.
The innocuous baby monitor gets <strong>louder</strong>.
<dialog>
<dt>Beth</dt>
<dd>Nursing a hungry baby in some destitute African village?</dd>
 <dt>John</dt>
 <dd><span class="how">(gasps)</span>No.</dd>
<dt>Beth</dt>
 <dd>This actually happened, but the commentator, I forget who, ended the piece with
     "your move, Jolie"</dd>
```

</dialog>

```
</div><!-- /#screenplay -->
```

Затем примените стилевые правила для форматирования контента и представления его в виде традиционного киносценария:

```
body {
  font-size: 62.5%;
  font-family: "Courier New", Courier, monospace;
  margin: 0 auto;
  width: 612px;
  }
  #screenplay {
   padding: 0 10.9em;
  }
```

```
#screenplay h3 + p {
 text-transform: uppercase;
}
#screenplay h3 {
 text-transform: uppercase;
text-align: right;
background: white;
}
#screenplay h3:after {
 content: ":";
}
dialog {
 font-size: 1.2em;
}
dt {
text-transform: uppercase;
text-align: center;
margin-top: 1.6em;
ł
dd {
margin-left: 7.2em;
}
span.how {
display: block;
text-align: center;
margin-right: 7.2em;
padding-right: 5em;
}
#screenplay strong {
text-transform: uppercase;
}
```

# Обсуждение

В спецификацию HTML5 включен новый элемент dialog, прежде всего для отображения разговора. В формате разметки используются те же элементы dt и dd, но элемент dl заменен элементом dialog.

# См. также

Элемент dialog в спецификации HTML5 на странице http://www.w3.org/TR/2008/WDhtml5-20080122/#the-dialog.

# 6.14. Превращение списка в дерево каталогов

#### Проблема

Вы хотите из списка создать структуру дерева каталогов.

#### Решение

Сначала создайте последовательность вложенных маркированных списков, которые послужат основой структуры дерева каталогов (рис. 6.27):

```
Morning Sessions
 Troubleshooting IE6
 Object Oriented CSS
 Fluid Typography
 Tomorrow's CSS3 Today
 Afternoon Sessions
 <111>
 Web Form Elements
 Flexible Layouts
 Coding Layouts
 Future CSS & amp; Markup
```



Рис. 6.27. Стандартное отображение маркированных списков

Создайте три маленькие геометрические фигуры: вертикальную линию или ствол, ветвь и концевую ветвь. Отобразите вертикальную линию сбоку от маркированных списков, как показано на рис. 6.28:

```
.itinerary, .itinerary ul {
   list-style-type: none;
```

```
background-image: url(pipe.gif);
background-repeat: repeat-y;
margin: 0;
padding: 0;
}
.itinerary ul {
margin-left: 12px;
}
```

Отобразите ветвь рядом с каждым пунктом списка:

```
.itinerary li {
    margin: 0;
    padding: 0 12px 0 28px;
    background-image: url(branch.gif);
    background-repeat: no-repeat;
    line-height: 1.5;
}
```

```
CSS Cookbook
\Theta \circ \circ
                                                                                           0
        - (C) (X) (A) (1P) ( ) (file:///Volumes/christopher/Docume 😭 🔻 ) - 🚷 Google Q
                                 Christopher Schmitt : De... 😧 🛵 Technology Books, Tech ...
                                                                                   0
        CSS Cookbook
The CSS Summit
Morning Sessions
  Troubleshooting IE6
  Object Oriented CSS
  Fluid Typography
  Tomorrow's CSS3 Today
Afternoon Sessions
  Web Form Elements
  Flexible Layouts
  Coding Layouts
 Future CSS & Markup
So Done
```

Рис. 6.28. Набор вертикальных линий

Затем в HTML явно разметьте с помощью атрибута class последний пункт в каждом маркированном списке:

```
        Morning Sessions

        Troubleshooting IE6
        Object Oriented CSS
        Fluid Typography
        class="branchend">Tomorrow's CSS3 Today
```

Теперь для отображения концевой ветви (рис. 6.29) примените селектор класса:

```
.itinerary li.branchend {
    /* соответствует цвету фона */
    /* родительского элемента или страницы */
    background-color: #fff;
    background-image: url(branchend.gif);
}
```



Рис. 6.29. Отображение концевых ветвей

#### Обсуждение

Метод, примененный в решении, основан на *разд. 6.8*, в котором используются пиктограммы как фоновое изображение пунктов списка. В данном решении вызываются три разных маленьких изображения, помещаемые в определенные области маркированных списков для получения нужного эффекта.

#### Применение CSS3

Для вставки концевой ветви в дерево каталогов мы вынуждены были включить в разметку атрибут class.

В спецификации CSS3 псевдокласс :last-of-type может избавить от использования атрибута class:

```
.itinerary li:last-of-type {
   /* соответствует цвету фона */
   /* родительского элемента или страницы */
   background-color: #fff;
   background-image: url(branchend.gif);
}
```

Во время написания книги псевдокласс :last-of-type поддерживался в Safari 3, Орега 9.5 и более поздних версиях этих Web-обозревателей.

Перечень селекторов спецификации CSS3 см. в приложении 4.

#### См. также

Статью Михала Войцеховского (Michal Wojciechowski) "Turning Lists into Trees" (Превращение списков в деревья) на странице http://odyniec.net/articles/turning-lists-into-trees.

# 6.15. Создание системы "звездного" рейтинга

#### Проблема

Вы хотите отображать систему рейтинга в виде звездочек, которая позволит пользователям выбирать на экране их собственные рейтинги.

#### Решение

Первый шаг — HTML-разметка, включающая маркированный список из пяти пунктов (рис. 6.30):

```
<div class="product" id="prod345781">
<h1>CSS Cookbook</h1>
Submit your review:
<a href="#">1 Star</a>
<a href="#">2 Stars</a>
<a href="#">3 Stars</a>
<a href="#">4 Stars</a>
<a href="#">5 Stars</a>
<a href="#">5 Stars</a>
```



Рис. 6.30. Стандартное отображение HTML-разметки для "звездного" рейтинга

Далее создайте изображение, содержащее все возможные варианты рейтингов, а также варианты изображения, получаемые при наведении курсора мыши (рис. 6.31). (Вы можете представить каждую звездочку как вписанную в квадрат, т. к. это несколько облегчит кодирование стилей CSS.)

Имея изображение с набором звездочек, примените правила CSS для ограничения ширины и высоты маркированного списка и вставки матрицы из звездочек:

```
.rating {
  margin: 0;
  padding: 0;
  list-style: none;
  clear: both;
  width: 75px;
  height: 15px;
  background-image: url(stars.gif);
  background-repeat: no-repeat;
  position: relative;
}
```

Далее задайте плавающую модель для всех пунктов списка (для поддержки IE6), одновременно удалив из них текст с помощью отрицательного значения в свойстве textindent:

```
.rating li {
  text-indent: -9999em;
  float: left; /* for IE6 */
}
```

Следующий шаг — абсолютное позиционирование в строке ссылки из каждого пункта списка в пределах 75 пикселов, соответствующих ширине маркированного списка, заданной ранее:

```
.rating li a {
position: absolute;
top: 0;
left: 0;
 z-index: 20;
height: 15px;
width: 15px;
display: block;
}
.rating .one a {
left: 0;
1
.rating .two a {
left: 15px;
1
.rating .three a {
left: 30px;
}
.rating .four a {
left: 45px;
1
.rating .five a {
left: 60px;
}
```

Поместив блоки на место, вы сможете задать средствами CSS рейтинг продукта по умолчанию. Например, для оценки "2 звездочки из пяти" потребуется просто блок объявления свойства background-position (рис. 6.32):

```
#prod345781 .rating {
    /* background-position: 0 0px; 0 out of 5 */
    /* background-position: 0 -15px; 1 out of 5 */
    background-position: 0 -30px; /* 2 out of 5 */
    /* background-position: 0 -45px; 3 out of 5 */
    /* background-position: 0 -60px; 4 out of 5 */
    /* background-position: 0 -75px; 1 out of 5 */
}
```

Для того чтобы добавить в систему ранжирования возможность задания рейтинга пользователем, установите расширение ссылки из каждого пункта списка до 75 пикселов в ширину и еще раз вставьте изображение звездочек как фоновое:

```
#prod345781 .rating li a:hover {
  z-index: 10;
  width: 75px;
  height: 15px;
  overflow: hidden;
  left: 0;
```

}

Рис. 6.32. Двухзвездочный рейтинг

Затем напишите специальные правила, сдвигающие фоновое изображение так, чтобы отображался второй набор закрашенных звездочек. Насколько сдвинется вверх фоновое изображение зависит от того, какой рейтинг пользователь укажет мышью (рис. 6.33):

```
#prod345781 .rating .one a:hover {
   background-position: 0 -105px; /* 1 out of 5 */
}
#prod345781 .rating .two a:hover {
   background-position: 0 -120px; /* 2 out of 5 */
}
#prod345781 .rating .three a:hover {
   background-position: 0 -135px; /* 3 out of 5 */
}
#prod345781 .rating .four a:hover {
   background-position: 0 -150px; /* 4 out of 5 */
}
#prod345781 .rating .five a:hover {
   background-position: 0 -165px; /* 5 out of 5 */
}
```

#### Обсуждение

Это решение использует несколько методов.

Первый важный метод — применение *спрайта CSS*, изображения, содержащего несколько пиктограмм (дополнительную информацию см. в *разд. 4.33*). Начальный рейтинг устанавливается с помощью позиционирования изображения звездочек, которое служит фоновым изображением маркированного списка как такового.



Рис. 6.33. Наведение указателя мыши на строку со звездочками, показывающую рейтинг товара или сервиса, назначенный непосредственно пользователем

Следующий метод — изменение способа позиционирования, называемое сцеплением (см. *разд. 2.25*). Благодаря абсолютному позиционированию в ряд каждой ссылки в маркированном списке пользователь может щелкнуть кнопкой мыши на том количестве звездочек, которое хочет назначить продукту или сервису.

И наконец, последний метод повторно использует изображение-спрайт. Когда пользователь выбирает, какой звездный рейтинг задать продукту или сервису, ширина ссылки изменяется и становится равной ширине строки со всеми звездочками. Это позволяет отобразиться фоновому изображению ссылки поверх собственного фонового изображения маркированного списка.

Несмотря на то, что оба эти элемента применяют одно и то же фоновое изображение, ссылка помещается выше или поверх фона элемента 11. Следовательно, фоновое изображение ссылки видимо, а фоновое изображение 11 нет.

Задание в качестве значения свойства background-position большей отрицательной величины смещает в нужную позицию другой набор закрашенных звездочек.

#### См. также

Содержательную статью Пола О'Брайена (Paul O'Brien) о методе создания подобной матрицы из звездочек на странице http://www.search-this.com/2007/05/23/css-the-star-matrix-pre-loaded/.

# Ссылки и навигация

# 7.0. Введение

Без ссылок Web-пространство потеряет смысл.

Ссылки позволяют вам отслеживать информацию, переходя с одной Web-страницы на другую и от одной идеи к другой независимо от местонахождения сервера информационного сайта.

В 1996 г. эксперт по использованию Web-пространства Якоб Нильсон (Jakob Nielson) назвал применение нестандартных цветов ссылок одной из десяти основных ошибок Web-дизайна (см. http://www.useit.com/alertbox/9605.html). Однако его совет использовать синий цвет для ссылок на страницы, на которых пользователь еще не побывал, и пурпурный или красный для отображения посещенных страниц объясняется соображениями логической упорядоченности, а не эстетическими представлениями.

К счастью, в новом тысячелетии он пересмотрел свои взгляды на цвета ссылок (см. http://www.useit.com/alertbox/link-list-color.html). Ссылки, будучи важной частью "Всемирной паутины", теперь могут быть одновременно и логически согласованными, и визуально привлекательными.

В этой главе показано, как улучшить эстетическое представление, изменяя стили ссылок. Вы научитесь всему, начиная от удаления подчеркивания ссылок и заканчивая изменением курсоров, созданием визуальных эффектов без применения JavaScript при наведении указателя мыши на ссылку (rollover) и формированием горизонтального меню из закладок.

# 7.1. Легкое создание текстовых меню и подменю

#### Проблема

Вы хотите быстро сгенерировать разметку для навигационного списка вместе с заранее заготовленными стилями.

#### Решение

Используйте List-O-Matic, построитель меню с Web-сайта Accessify (http:// accessify.com/tools-and-wizards/developer-tools/list-o-matic/).

Введите метки для навигационных меню, адреса ссылок и необязательные заголовочные атрибуты, как показано на рис. 7.1.

Hide the help pane	el (for this page only) and don't s	now it again
2004 If you have	(hopefully much improved!) ver a hankering for the original, it's	archived here
	· · · · · · · · · · · · · · · · · · ·	
Just give it a go - i	ou choose the content, layout ar t's child's play!	a presentation of your list-based navigation menus.
ouse give it a go = 1	a o cana o piay.	
If you're not sure l	how this works, check out the vi	deo how-to tutorial here.
Step 1 - Enter the	navigation details	
	(Preview the markup)	'm done. Show me the styles
Link text:	Address/URL:	Title attr (optional):
Home	1	Home page
This link con	tains sub-level navigation (option	al):
Link text:	Address/URL:	Title attr (optional):
About	/about/	About us
This link con	tains sub-level navigation (option	al):
Link text:	Address/URL:	Title attr (optional):
Archives	/archives/	Past diatribes
This link con	tains sub-level navigation (option	al):
Link text:	Address/URL:	Title attr (optional):
Writing	/writing/	Published works
This link con	tains sub-level navigation (option	al):
Link text:	Address/URL:	Title attr (optional):
1	/speaking/	Training and conferences
Speaking	tains sub-level navigation (option	al):
Speaking This link con		
Speaking This link con Link text:	Address/URL:	Title attr (optional):

Рис. 7.1. Интерактивное Web-приложение для генерации удобных меню с помощью маркированных списков

Далее выберите стиль навигационного меню (рис. 7.2).

Щелкните мышью кнопку Show me the markup (Показать разметку) для получения разметки и правил CSS, которые можно вставить непосредственно в вашу Webстраницу.

# Обсуждение

Применяя маркированные списки и ссылки (см. *разд. 1.10*), List-O-Matic с Web-сайта Accessify выполняет тяжелую часть работы по кодированию и стилевому оформлению навигационного меню. Для согласования с дизайном вашего сайта непременно настройте правила CSS в соответствии с вашим проектом.

-O-Matic - Generate CSS-styl		
Step 2 - Pick a style for your	navigation	
Edit the list	t items) (Reset page & start again) (Show me the markup	
💿 Horizontal styles 🔘 Vertica	l styles 🔨	4
Home Products Ou	r Services Contact Us Privacy Policy	Ì
Home Products Ou	r Services Contact Us Privacy Policy	
Home Products Our	Services Contact Us Privacy Policy	
Edit the list	titems Reset page & start again Show me the markup $\rightarrow$	

Рис. 7.2. Примеры стилевого оформления меню

Видеоуроки о приемах использования List-O-Matic на странице (http://accessify.com/ screencasts/list-o-matic/.

# 7.2. Удаление подчеркивания ссылок и другие приемы стилевого оформления

#### Проблема

Вы хотите убрать стандартное подчеркивание ссылок (рис. 7.3).

#### Решение

Используйте свойство text-decoration с селектором псевдокласса для непосещенных и посещенных ссылок:

```
a:link, a:visited {
  text-decoration: none;
}
```

#### Обсуждение

Воспользуйтесь псевдоклассами :link и :visited для применения стилей ссылок в Web-документе. Псевдокласс :link применяется к ссылкам, которые пользователь еще не посетил. Псевдокласс :visited соответствует ссылкам, уже посещенным пользователем.

\varTheta 🔿 🔿 CSS Cookbook	)
Image: Back Forward     Image: Complex Compl	>
HTML Validation	
CSS Cookbook 🙁 🏹 Christopher Schmitt 😒 🖉 Technology Books, 😒 🕇 🕇	
Site navigation:	
<ul> <li>Home</li> <li>About</li> <li>Archives</li> <li>Writing <ul> <li>Releasing CSS</li> <li>CSS Conskbook</li> </ul> </li> <li>Speaking</li> <li>Contact</li> </ul>	
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus vitae ligula ut metus interdum accumsan. Vivamus commodo. Etiam eros massa, hendrerit hendrerit, ultricies facilisis, porttitor eget, magna. Fusce lacus ligula, dapibus sed, mattis id, tristique non, arcu. Vivamus feugiat. Etiam erat pede, posuere placerat, porttitor vitae, sollicitudin ut, tellus. Nunv volutpat velit blandit enim. Nulla luctus. In viverra nibh ac massa. Curabitur vel tellus a velit feugiat tristique. Mauris ultricies cursus diam. Nullam scelerisque blandit pede. Maecenas tellus velit, pellentesque placerat, tincidunt nec, tempus at, tellus. Quisque sapien. Nullam eget ante at mi consequat mattis. Nunc accumsan tincidunt ipsum. Sed adipiscing tincidunt lorem. Mauris arcu. Vestibulum laoreet augue.	2

Рис. 7.3. Ссылки без подчеркивания

Свойство text-decoration может принимать пять значений, перечисленных в табл. 7.1.

Значение	Результат
underline	Линия располагается под текстом
overline	Линия располагается над текстом
blink	Тест мигает
line-through	Линия перечеркивает текст
none	Текст никак не выделяется

**Таблица 7.1. Значения свойства** text-decoration

Приведенные значения часто используются для улучшения визуального представления Web-страницы. Вместо подчеркивания всех ссылок в документе дизайнеры задают значение none в свойстве text-decoration, одновременно изменяя цвет фона ссылки, цвет текста ссылки или и то, и другое:

```
a:link, a:visited {
  text-decoration: none;
  background-color: red;
  color: white;
```

}

Для того чтобы сделать более удобным использование страницы для посетителейдальтоников, не способных отличать выделенные цветом ссылки от остального HTMLтекста, окрашенного в стандартный цвет, дизайнеры задают также начертание текста ссылок полужирным шрифтом:

```
a:link, a:visited {
  font-weight: bold;
  text-decoration: none;
  color: red;
}
```

Значение line-through может стать интересным элементом дизайна страницы и применяться для обозначения ссылки, которую пользователь уже посетил, напоминая вычеркнутую строчку в списке намеченных дел (рис. 7.4):

```
a:link {
  font-weight: bold;
  text-decoration: none;
  color: red;
}
a:visited {
  font-weight: bold;
  text-decoration: line-through;
  color: black;
}
```



volutpat velit blandit enim. Nulla luctus. In viverra nibh ac massa. Curabitur vel tellus a velit feugiat tristique. Mauris ultricies cursus diam. Nullam scelerisque blandit pede. Maecenas tellus velit, pellentesque placerat, tincidunt nec, tempus at, tellus. Quisque sapien. Nullam eget ante at mi consequat mattis. Nunc accumsan tincidunt ipsum. Sed adipiscing tincidunt lorem. Mauris arcu. Vestibulum laoreet augue.

Свойство text-decoration в спецификации CSS 2.1 на странице http://www.w3.org/ TR/CSS21/text.html#propdef-text-decoration; обновленную статью Якоба Нильсона (Jakob Neilson) "Design Guidelines for Visualizing Links" (Рекомендации разработчикам по визуализации ссылок) на http://www.useit.com/alertbox/20040510.html.

# 7.3. Изменение цвета ссылок

# Проблема

Вы хотите создать разные стили для ссылок: один стиль для навигации, а другой для ссылок в основном тексте.

## Решение

Используйте псевдоклассы: :link, :visited, :hover и :active в следующей последовательности:

```
body {
   color: #9ff;
}
a:link {
   color: #3cf;
}
a:visited {
   color: #cecece;
}
a:hover {
   color: #366;
}
a:active {
   color: #399;
}
```

# Обсуждение

Псевдоклассы гиперссылок в каскадной таблице с точки зрения приоритета равны, поэтому вы избежите конфликтов, перечисляя их в следующем порядке: :link, :visited, :hover и :active. Для запоминания порядка перечисления часто пользуются следующим мнемоническим правилом: "LoVe/HAte".

Посещенная или непосещенная ссылки могут регистрировать статус hover (указатель наведен на ссылку) и active (момент открытия ссылки) одновременно. Поскольку у псевдоклассов равный приоритет, тот статус, который указан последним, и будет отображаться на экране, вот почему в некоторых случаях псевдокласс :hover не срабатывает. Если он появляется до псевдоклассов :active или :visited, они скрывают статус hover в соответствии с правилами каскадирования.

Динамические псевдоклассы :hover, :active и :focus в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/selector.html#dynamic-pseudo-classes; рассуждения Эрика Мейера (Eric Meyer) о специфичности ссылок на странице http:// www.meyerweb.com/eric/css/link-specificity.html.

# 7.4. Удаление пунктирных линий при щелчке мышью по ссылкам в Internet Explorer

Вы хотите удалить пунктирные линии, появляющиеся при щелчке кнопкой мыши ссылок в Internet Explorer.

## Решение

Задайте для ссылок свойство outline со значением none:

```
a {
  outline: none;
}
```

#### Обсуждение

Свойство outline не является частью блочной модели, как свойства margin, border и padding. Если для элемента задается ширина, то border пересчитывается, a outline нет.

#### Примечание

В отличие от рамки у контура (outline) нет специфических свойств CSS. Например, не существует свойства outline-top.

Пунктирные линии, характерные для Internet Explorer, облегчают чтение, позволяя посетителям сайта знать, где они щелкали мышью, и какие части страницы реагируют на щелчок кнопкой мыши. Однако возможны ситуации, в которых контуры ссылок могут испортить внешний вид страницы.

Для обеспечения взаимодействия с посетителями сайта (даже если свойство outline равно none или not) рекомендуется при стилевом оформлении ссылок задать псевдокласс :focus и эффекты наведения указателя мыши на ссылки (rollover):

```
a:hover, a:active, a:focus {
  color: #399;
}
```

Псевдокласс : focus применяется, когда пользовательской клавиатурой или другим устройством ввода активизируется элемент, такой как input.

#### Примечание

Для применения свойства outline Internet Explorer требуется наличие допустимого DOCTYPE (см. *разд. 1.3*).

Свойство outline в спецификации CSS 2 на странице http://www.w3.org/TR/CSS2/ ui.html#dynamic-outlines.

# 7.5. Изменение цвета ссылок в разных секциях страницы

# Проблема

Вы хотите применять по-разному оформленные ссылки в основном тексте и в элементах навигации.

# Решение

Сначала поместите секции страницы в элементы div с разными значениями атрибутов:

```
<div id="nav">
[...]
</div><!-- /#nav -->
<div id="content">
[...]
</div><!-- /#content -->
```

Затем используйте селекторы потомка вместе с селекторами ID и правило LV/HA, обсуждавшееся в *разд.* 7.3, для того чтобы отделить разные стили ссылок, предназначенные для разных частей Web-страницы:

```
/* дизайн навигационных ссылок */
#nav a:link {
  color: blue;
  }
#nav a:visited {
  color: purple;
  }
  /* дизайн контекстных ссылок*/
#content a:link {
  color: white;
  }
#content a:visited {
  color: yellow;
  }
```

# Обсуждение

Использование селектора ID для обозначения секций Web-страницы открывает возможность применения разных стилей к одним и тем же элементам или селекторам. Используйте одни и те же селекторы для формирования в различных секциях ссылок с разными стилями. Дополнительную информацию о селекторе ID можно найти в *разд. 2.2.* Задание мнемонического правила LV/HA для порядка следования статуса ссылок также обеспечивает предсказуемость их поведения.

#### См. также

Tutorial on CSS Pseudo-classes (руководство по псевдоклассам CSS) от W3Schools на странице http://www.w3schools.com/css/css\_pseudo\_classes.asp.

# 7.6. Вывод пиктограмм в конце ссылок разных типов

#### Проблема

Вы хотите выводить пиктограммы в конце внутристрочных ссылок, как показано на рис. 7.5.



Рис. 7.5. Пиктограммы, помещенные в конец ссылок

#### Решение

Задайте в документе ссылки:

```
Sed sed nisi. <a href="mailto:orders@csscookbook.com">Quote me on
an estimate</a> nulla ligula. Etiam pulvinar,
<a href="http://www.csscookbook.com/">CSS Cookbook web site</a> quisque
ante quam, ultricies quis, rutrum dignissim, <a href="sample-chapter.pdf">
fermentum</a> a, est. Nulla felis dolor, tempor non, posuere ut, ornare ac, eros.
Cras in metus. Sed rutrum lectus ut nibh. Pellentesque ipsum.
```

Затем примените селекторы атрибутов (см. разд. 2.5):

```
a {
  text-decoration: none;
  font-weight: bold;
}
```

```
a[href^="mailto:"] {
  padding-right: 20px;
  background: url(mail.gif) no-repeat right;
}
a[href^="http://"] {
  padding-right: 20px;
  background: url(website.gif) no-repeat right;
}
a[href$="pdf"] {
  padding-right: 20px;
  background: url(document.gif) no-repeat right;
}
```

#### Обсуждение

Применение селекторов атрибута — еще один метод, не требующий дополнительной разметки для включения в контекст элемента img.

В разд. 2.5 обсуждается много вариантов применения селекторов атрибута для помещения пиктограмм (или текста) в начало ссылок (и других элементов) или перед ними.

Поскольку в решении применяется генерация контента, оно не годится для IE7 и более ранних версий этого Web-обозревателя. IE8 поддерживает генерацию контента.

#### См. также

Презентацию Дэйва Ши (Dave Shea) о вставке пиктограммы как фонового изображения во внутристрочную ссылку на странице http://www.mezzoblue.com/presentations/2006/ sxsw/css/q1.html; объяснение того, почему этот вариант не работает в IE на странице http://www.brunildo.org/test/InlineBlockLayout.html.

# 7.7. Меняющиеся курсоры

#### Проблема

Вы хотите заменить стандартный курсор, отображаемый, когда указатель мыши наведен на ссылку, как показано на рис. 7.6.

#### Решение

Для замены вида курсора используйте свойство cursor:

```
a:link, a:visited {
  cursor: move;
}
```

#### Обсуждение

Свойство cursor может принимать разные значения, перечисленные в табл. 7.2. Однако поддержка этих значений в различных Web-обозревателях разная. Opera 7, Internet

0	0	CSS Cookbook		$\bigcirc$			
Back	Forward Reload	Stop Home Print 1Passy	word	God			
HTML \	/alidation						
	CSS Cookbook	Christopher Schmi 🕄	📇 Technology Books, 🕄	+ =			
Sed sed nisi. <u>Quote me on an estimate</u> nulla ligula. Etiam pulvinar, <u>CSS Cookbook web</u> <u>site</u> quisque ante quam, ultricies quis, rutrum dignissim, <u>fermentum</u> a, est. Nulla felis dolor, tempor non, posuere ut, ornare ac, eros. Cras in metus. Sed rutrum lectus ut nibh. Pellentesque ipsum.							

Рис. 7.6. Курсор изменяется на изображение руки

Explorer 5.5 для Windows и более поздние версии этих Web-обозревателей поддерживают свойство cursor. Несмотря на то, что Firefox поддерживает большую часть значений, он не поддерживает uri. Кроме того, в Firefox свойство cursor не наследуется от родительского дочерними элементами.

Значение	Описание	Образец
auto	Курсор меняется на изображение, определенное Web-обозревателем	ĸ
crosshair	Две перпендикулярные линии, пересекающиеся в середине; похож на увеличенный знак +	+
default	Зависящий от платформы курсор, визуализируемый в большинстве Web-обозревателей как стрелка. Разработ- чики Web-обозревателей или операционных систем могут определять разные стили курсоров	k
pointer	Применяется для иллюстрации того, что указатель мыши направлен на ссылку; иногда визуализируется как рука с поднятым указательным пальцем. Разработчики Web- обозревателей или операционных систем могут задавать разные стили курсоров	Ð.
move	Показывает, что элемент можно переместить; иногда ви- зуализируется как перекрестье со стрелками на концах линий или в виде руки с поднятыми пятью пальцами	۲
e-resize, ne-resize, nw-resize, n-resize, se-resize, sw-resize, s-resize, w-resize	Стрелка, показывающая в какую сторону можно двигаться; например, se-resize указывает на юго-восточное направ- ление	٨
text	Показывает, что можно выделять текст; иногда изображается как вертикальная линия, обычно при- меняемая в программах обработки текста	Ĩ
wait	Показывает, что компьютер занят; иногда отображается в виде песочных часов	0

#### Таблица 7.2. Значения свойства cursor

```
Таблица 7.2 (окончание)
```

Значение	Описание	Образец
progress	Показывает, что компьютер занят, но пользователь все еще может взаимодействовать с Web-обозревателем	<b>N</b> 00
help	Показывает, что доступна информация или справка, часто по адресу, на который указывает ссылка; иногда отобража- ется как знак вопроса или стрелка со знаком вопроса	<b>1</b> 2
<uri></uri>	Курсор можно заменить внешним курсором, например, изображением, курсором из файлов курсоров Windows, курсором SVG (Scalable Vector Graphics, масштабируемая векторная графика) и т. д.	Не опреде- лен

Код для вставки пользовательского курсора аналогичен коду, применяемому для задания фонового изображения элемента:

```
a.help:link, a.help:visited {
  cursor: url(bewildered.gif);
}
```

Используя необычные курсоры, большинство пользователей столкнется с непредсказуемым или даже сильно раздражающим изменением привычных методов навигации в Интернете, зависящим от того, насколько резко новый курсор отличается от прежнего. Поэтому изменять курсор, который пользователь уже привык видеть, рискованно.

#### См. также

Свойство cursor в спецификации CSS 2.1 на странице http://www.w3.org/ TR/CSS21/ui.html#propdef-cursor; примеры различных курсоров в действии на странице http://www.zimmertech.com/tutorials/css/20/changing-cursors-tutorial.php.

# 7.8. Изменение внешнего вида ссылки при наведении указателя мыши без применения JavaScript

#### Проблема

Вы хотите создать простой визуальный эффект при наведении на ссылку указателя мыши (rollover), не применяя для смены изображений язык JavaScript.

#### Решение

Для создания визуального эффекта воспользуйтесь псевдоклассами :hover и :active:

```
a:link {
  color: #777;
  text-decoration: none;
}
```

```
a:visited {
  color: #333;
  text-decoration: none;
}
a:link:hover, a:visited:hover {
  color: #777;
  background-color: #ccc;
}
a:link:active, a:visited:active {
  color: #ccc;
  background-color: #ccc;
}
```

## Обсуждение

Псевдокласс :hover имитирует обычное событие языка JavaScript onmouseover. Но вместо выполнения функции на JavaScript в тот момент, когда пользователь наводит указатель мыши на ссылку с псевдоклассом :hover, внешний вид ссылки меняется благодаря применению других стилей.

Если порядок указания селекторов с одинаковой специфичностью нарушается, это может помешать применению какого-либо из заменяющих стилей. Во избежание подобных проблем пользуйтесь правилом LV/HA (см. *разд.* 7.5).

Несмотря на то, что псевдоклассы :hover и :active могут применяться к любому элементу, обычно они используются в ссылках.

#### Известные проблемы

В данном решении два псевдокласса гарантируют появление визуального эффекта при наведении указателя мыши только на ссылки-якори. Без псевдоклассов :hover и :active современные Web-обозреватели могли бы на законных основаниях применять



Рис. 7.7. Нежелательный визуальный эффект при наведении указателя мыши на ссылку, примененный к заголовку

визуальные эффекты наведения указателя мыши к любым элементам, размеченным как закладки или якори (anchor), как показано в приведенном далее коде и на рис. 7.7:

<h2><a name="europan">Li Europan lingues</a></h2>

Однако для ссылок в документе рекомендуется вместо использования атрибутов name применять атрибуты id:

<h2 id="europan">Li Europan lingues</h2>

#### См. также

Псевдоклассы :active и :hover в спецификации CSS 2.1 на странице http:// www.w3.org/TR/CSS21/selector.html#x36; обсуждение ссылок и специфичности на http://www.meyerweb.com/eric/css/link-specificity.html.

# 7.9. Анимация ссылок с помощью свойств CSS3 *transition* при наведении указателя мыши

#### Проблема

Вы хотите отрегулировать время существования визуального эффекта при наведении указателя мыши на ссылку.

#### Решение

Для задания анимации при наведении указателя мыши используйте свойства CSS3 transition (puc. 7.8):

```
#navsite a {
 -webkit-transition-timing-function: linear;
 -webkit-transition-duration: .66s;
 -webkit-transition-property: background-color; }
a:link {
 color: #777;
 text-decoration: none;
}
a:visited {
color: #333;
text-decoration: none;
}
a:link:hover, a:visited:hover {
 color: #777;
background-color: #ccc;
}
a:link:active, a:visited:active {
 color: #ccc;
background-color: #ccc;
}
```

00	00	00	00
▲ ►	<ul> <li>►</li> <li>●</li> <li>●</li></ul>	<ul> <li>►</li></ul>	<ul> <li>►</li> <li>●</li> <li>●</li></ul>
CSS Cookbook	CSS Cookbook	CSS Cookbook	CSS Cookbook
Site navigation:	Site navigation:	Site navigation:	Site navigation:
<ul><li>Home</li><li>About</li><li>Archives</li><li>Writing</li></ul>	<ul> <li>Home</li> <li>About</li> <li>Archives</li> <li>Walling</li> </ul>	Home     About     Archives     Winne	<ul> <li>Home</li> <li>Arbout</li> <li>Archives</li> <li>Writing</li> </ul>
Reasing CSS     CSS Cookbook     Speaking	Releasing CSS     CSS Cookbook     Speaking	<ul> <li>Releasing CSS</li> <li>CSS Cookbook</li> <li>Speaking</li> </ul>	Releasing CSS     CSS Cookbook     Speaking
• Contact	• Contact	• Contact	• Contact
Li Europan Lingues	Li Europan Lingues	Li Europan Lingues	Li Europan Lingues
Sed sed nisi. Morbi gravida, odio	Sed sed nisi. Morbi gravida, odio	Sed sed nisi. Morbi gravida, odio	Sed sed nisi. Morbi gravida, odio
posuere tortor elit eget est. Nulla li ultricies quis, rutrum dignissim, fe	posuere tortor elit eget est. Nulla li ultricies quis, rutrum dignissim, fe	posuere tortor elit eget est. Nulla li ultricies quis, rutrum dignissim, fe	posuere tortor elit eget est. Nulla li ultricies quis, rutrum dignissim, fe
Go to "file:///writing/releasing-css"	Go to "file:///writing/"	Go to "file:///archives/"	Go to "file:///"

Рис. 7.8. Плавное уменьшение яркости фона при скольжении указателя по ссылкам

# Обсуждение

Свойство transition — дополнение в спецификации CSS3, предложенное разработчиками Web-обозревателя Apple's Safari. Во время написания книги оно поддерживалось только Safari. Свойству transition в Web-обозревателе Safari для получения видимого эффекта требуется специальное расширение -webkit-.

#### Timing function

Функция transition-timing-function задает тип анимации или переход, которого вы хотите добиться при отображении ссылки. В предложенном решении ее значение равно linear и означает, что каждый кадр анимации будет длиться одно и то же время.

К другим возможным значениям функции относятся ease, ease-in, ease-out cubic-bezier(x1, y1, x2, y2).

Значение cubic-bezier представляет переход, отражаемый кривой Безье третьего порядка (рис. 7.9). Точки  $P_0$  и  $P_3$  определяют начало и конец преобразования и всегда равны 0.0 и 1.0 соответственно. Скорость, с которой выполняется анимация, задается положением точек  $P_1$  (x1, y1) и  $P_2$  (x2, y2).

Если задана функция со значением ease-in, переход начинается медленно, а затем скорость растет. Это значение эквивалентно значению cubic-bezier(0.42, 0, 1.0,1).

В случае значения ease-out переход начинается с высокой скоростью, а затем она падает. Это значение эквивалентно значению cubic-bezier(0, 0, 0.58, 1.0).

Значение ease эквивалентно значению cubic-bezier(0.25, 0.1, 0.25, 1.0).

#### Продолжительность и задержка

Значение по умолчанию свойства transition-duration равно 0. Любое отрицательное значение трактуется как нулевое. Значение может задаваться в секундах (s) и миллисекундах (ms), но не только.


Рис. 7.9. Математическое представление переходов (источник: http://www.w3.org/tr/css3-transitions/#transition-function\_tag)

Свойство transition-delay задает величину промежутка времени перед началом перехода.

#### Свойство transition

Свойство transition-property определяет, к какому визуальному свойству CSS применяется переход. В предложенном решении переход применен к цвету фона.

#### Свойство с сокращенной формой записи

Вы можете задать характеристики перехода в одном свойстве transition. Например, вы можете сократить описание перехода в решении следующим образом:

```
#navsite a {
  -webkit-transition: background-color .66s linear;
}
```

#### Распространение переходов на другие свойства

На персональном Web-сайте Фарука Атеша (Faruk Ateş) (см. http://farukat.es/) свойство transition используется для смены цвета фона элемента, а также цвета элемента, ширины, тени, тени от текста и непрозрачности (рис. 7.10):

```
#web20 li a {
  background: transparent url(icons.png) 100% 0 no-repeat;
  border: none;
  color: #848484;
  display: block;
  font-size: 11px;
  font-weight: normal;
  height: 63px;
  left: -19px;
```

```
line-height: 18px;
margin: 3px 0;
padding-left: 19px;
position: relative;
text-decoration: none;
width: 220px;
text-shadow: rgba(0,0,0, 0) 1px 1px 2px;
-moz-box-shadow: rgba(0,0,0, 0) 2px 2px 2px;
-webkit-box-shadow: rgba(0,0,0, 0) 2px 2px 2px;
-moz-border-radius-topright: 31px;
-moz-border-radius-bottomright: 31px;
-webkit-border-top-right-radius: 31px;
-webkit-border-bottom-right-radius: 31px;
-webkit-transition: background-color .25s ease,
  color .5s ease,
  width .2s ease-out,
  -webkit-box-shadow .25s ease,
  text-shadow .2s ease,
  opacity .2s ease;
}
```



Рис. 7.10. Сложный визуальный эффект средствами CSS при наведении указателя мыши

#### См. также

Статью, знакомящую с переходами, в блоге Surfin's Safari на странице http://webkit.org/blog/138/css-animation/; описание переходов в спецификации CSS3 на странице http://www.w3.org/TR/css3-transitions/#transitions-.

# 7.10. Создание навигационных текстовых меню и визуальных эффектов при наведении указателя мыши на его пункты

# Проблема

У вас есть список ссылок, но вы хотите создать элегантное меню, например такое, как на рис. 7.11.



Рис. 7.11. Набор стилизованных ссылок

## Решение

Прежде всего, разметьте список ссылок в маркированном списке так, чтобы они были включены в элемент-контейнер div с атрибутом id:

#### <div id="navsite">

```
Site navigation:

<a href="/">Home</a>
<a href="/about/">About</a>
<a href="/archives/">Archives</a>
<a href="/writing/">Writing</a>
<a href="/writing/">Speaking</a>
<a href="/speaking/">Speaking</a>
<a href="/contact/">Contact</a>

<//div>
```

Далее используйте свойство border в элементах-якорях для создания основной структуры дизайна:

```
#navsite p {
  display: none;
}
```

```
a {
 text-shadow: 0 -1px 0px rgba(0,0,0,.8);
}
#navsite {
 font-family: Verdana, Helvetica, Arial, sans-serif;
 font-size: 1em;
 font-weight: bold;
border-right: 1px solid #666;
padding: 0;
margin-bottom: 1em;
color: #333;
width: 15em;
ļ
#navsite ul {
list-style: none;
margin: 0;
padding: 0;
}
#navsite ul li {
margin: 0;
border-top: 1px solid #003;
}
#navsite ul li a:link, #navsite ul li a:visited {
display: block;
padding: 4px 4px 4px 0.5em;
border-left: 10px solid #369;
border-right: 1px solid #69c;
border-bottom: 1px solid #369;
 color: #E5DEAC;
 text-decoration: none;
background-color: #495961;
background-image: url(title-glass.png);
background-position: 50%;
background-repeat: repeat-x;
html>body #navsite ul li a {
width: auto;
#navsite ul li a:hover {
border-left: 10px solid #036;
border-right: 1px solid #69c;
border-bottom: 1px solid #369;
background-color: #69f;
 color: #fff;
}
```

#### Обсуждение

Меню облегчает посетителям перемещение по вашему сайту. Для того чтобы помочь пользователю найти навигационное меню, создайте стилевое оформление ссылок, вы-

деляющее их из основного текста. Для этого воспользуйтесь селектором ID при написании правил CSS. Как видно из данного решения, для создания удачного меню потребуются способы обхода некоторых ошибок Web-обозревателей, а также непосредственная реализация дизайнерского решения с помощью технологии CSS.

#### Создание списка

Во фрагменте, помеченном div, одна строка текста обозначает набор ссылок как навигационный:

Site navigation:

Если у Web-обозревателя пользователя нет поддержки CSS, это строка текста будет видна. Для того чтобы скрыть ее в Web-обозревателях с поддержкой CSS, задайте свойство display со значением none:

```
#navsite p {
  display: none;
}
```

#### Примечание

Я предпочитаю размечать Web-документ с помощью секций и помечать навигационные элементы. Вам может нравиться или быть рекомендован другой метод. Некоторые Webразработчики решают отказаться от использования контейнера div и элемента p как метки и просто задают значение id в элементе ul. Затем они корректируют правила CSS, относящиеся к маркированному списку.

Следующий шаг — стилевое оформление элемента div, содержащего набор ссылок меню. В этом правиле CSS задаются стили ссылок, наследующие набор свойств элемента div. Кроме того, указываются значения свойств width, border-right, padding и margin-bottom, чтобы не скомкать меню:

```
#navsite {
  font-family: Verdana, Helvetica, Arial, sans-serif;
  font-size: lem;
  font-weight: bold;
  border-right: lpx solid #666;
  padding: 0;
  margin-bottom: lem;
  color: #333;
  width: 15em;
}
```

Следующее правило CSS устраняет любые потенциальные проблемы, связанные с отступами пунктов в списках, установкой нулевых значений свойств margin и padding, а также удалением любых маркеров списка:

```
#navsite ul {
    list-style: none;
    margin: 0;
    padding: 0;
}
```

В следующем правиле вы обеспечиваете отсутствие полей у всех элементов списка. Кроме того, это правило CSS задает однопиксельную верхнюю границу элемента списка. Такое оформление элемента помогает отделить его от других пунктов списка:

```
#navsite ul li {
  margin: 0;
  border-top: 1px solid #003;
}
```

#### Стилевое оформление ссылок

В приведенном далее правиле задаются стили ссылок. По умолчанию ссылки — строчные элементы. Нам необходимо визуализировать ссылки как блочные элементы, чтобы можно было щелкнуть кнопкой мыши, указывая на ссылку целиком, вместе с ее стилевым оформлением, а не только на текст. Такое преобразование выполняет свойство display со значением block.

Используйте следующие объявления для стилевого оформления и вывода рамок, задания цвета текста, шрифта и ширины:

```
#navsite ul li a:link, #navsite ul li a:visited {
    display: block;
    padding: 4px 4px 0.5em;
    border-left: 10px solid #369;
    border-right: 1px solid #69c;
    border-bottom: 1px solid #369;
    color: #E5DEAC;
    text-decoration: none;
    background-color: #495961;
    background-image: url(title-glass.png);
    background-repeat: repeat-x;
}
```

#### Примечание

В последнем объявлении для ссылок задается ширина, равная 100 %. Это правило было введено для того, чтобы можно было щелкнуть кнопкой мыши всю область ссылки в Internet Explorer для Windows. Недостаток данного правила заключается в том, что оно создает проблемы в более старых версиях Web-обозревателей IE Mach и Netscape Navigator (которые вам быть может придется поддерживать). Для того чтобы переустановить ширину ссылки, воспользуйтесь селектором потомка (который не умеет обрабатывать IE7):

```
html>body #navsite ul li a {
  width: auto;
}
```

Последнее правило CSS задает стили для создания визуального эффекта при наведении указателя мыши на ссылки:

```
#navsite ul li a:hover {
   border-left: 10px solid #036;
   border-right: 1px solid #69c;
```

```
border-bottom: 1px solid #369;
background-color: #69f;
color: #fff;
}
```

#### Маркированные списки как основа навигационных меню

Маркированный список — наилучший способ структурирования меню ссылок как теоретически, так и практически. С одной стороны, набор ссылок — это набор элементов маркированного списка. Применение маркированных списков для навигации по сайту создает строгую структуру вашего Web-документа, основанную на логически и семантически корректной разметке.

С другой стороны, если ссылки организованы как маркированный список, легче задать их стилевое оформление в виде меню, чем описывать стили последовательности элементов div:

```
<div id="navsite">
  Site navigation:
  <div><a href="/">Home</a></div>
  <div><a href="/about/">About</a></div>
  <div><a href="/archives/">Archives</a></div>
  <div><a href="/writing/">Writing</a></div>
  <div><a href="/speaking/">Speaking</a></div>
  <div><a href="/speaking/">Speaking</a></div>
  <div><a href="/contact/">Contact</a></div>
  <div><a href="/contact/">Contact</a></div>
  <div><a href="/contact/">Contact</a></div>
  </div>
```

#### См. также

Статью "CSS Design: Taming Lists" (CSS-дизайн: укрощение списков) Марка Ньюхауза Newhouse) странице http://www.alistapart.com/articles/taminglists/; (Mark на статью/руководство "Semantics, HTML, XHTML, and Structure" (Семантика, HTML, XHTML И структура) Ширли Е. Кайзер (Shirley E. Kaiser) на http:// brainstormsandraves.com/articles/semantics/structure/.

# 7.11. Вставка подменю в вертикальные меню

# Проблема

Вы хотите вставить дополнительное меню под основной набор навигационных ссылок, как показано на рис. 7.12.

Дополните основное навигационное меню, включив в нужный пункт списка вложенный маркированный список. В следующем примере в пункт списка Writing помещены две публикации:

```
<div id="navsite">
Site navigation:
<a href="/">Home</a>
```

Задайте некоторое поле слева от вложенного маркированного списка с помощью селектора потомка:

```
#navsite ul ul {
   background-color: white;
   margin-left: 10px;
}
```



Рис. 7.12. Подменю, вставленное в вертикальное меню

#### Решение

Затем задайте стили ссылок и визуальный эффект при наведении указателя мыши на ссылку в соответствии с рис. 7.12:

```
#navsite ul ul li a:link, #navsite ul ul li a:visited {
   border-left: 10px solid #69c;
   border-right: 1px solid #9cF;
```

```
border-bottom: 1px solid #69c;
background-color: #888;
}
#navsite ul li a:hover, #navsite ul ul li a:hover {
border-left: 10px solid #036;
border-right: 1px solid #69c;
border-bottom: 1px solid #369;
background-color: #69f;
color: #fff;
}
```

#### Обсуждение

Применение селекторов потомка (см. *разд. 2.2*) позволит вам определить стили вложенных ссылок. Сдвинув список с помощью поля, вы заставите ссылки появляться под их родительской ссылкой.

#### Ограничение применения стилей

Основной недостаток применения селекторов потомка в данном решении состоит в том, что если у вас есть несколько вложенных списков (т. е. список в списке, который тоже вложен в список), селекторы потомка применяются и в третьем вложенном списке. Таким образом получается, что у нескольких вложенных списков один и тот же внешний вид.

Один вариант решения этой проблемы — для ограничения применения стилей в нескольких вложенных списках использование селектора прямого потомка:

```
#navsite ul li > ul li a:link, #navsite ul li > ul li a:visited {
   border-left: 10px solid #69c;
   border-bottom: 1px solid #9cF;
   border-bottom: 1px solid #69c;
   background-color: #888;
}
#navsite ul li a:hover, #navsite ul li > ul li a:hover {
   border-left: 10px solid #036;
   border-right: 1px solid #69c;
   border-bottom: 1px solid #369;
   background-color: #69f;
   color: #fff;
}
```

#### См. также

Web-сайт Listamatic с многочисленными примерами и кодом навигационных меню на странице http://css.maxdesign.com.au/listamatic/.

# 7.12. Создание горизонтальных навигационных меню

## Проблема

Вы хотите создать горизонтальное навигационное меню из маркированного набора ссылок; на рис. 7.13 показан стандартный набор.

0	0				CS	S Cookbo	ok					3	$\bigcirc$
Back	Forward	C Reload	Stop (	Home	Print	1P 1Password		file:///	י ג י	) - 🚷	Googl	e (	٩)
HTML V	alidation												0.23
	CSS Cookbo	ook	0	Christ	opher S	Schmitt : D	e 🕲	E Tech	nology B	ooks, Tecl	h 🕲	+	=
Site na	avigation: <u>Home</u> <u>About</u> <u>Archives</u> <u>Writing</u> <u>Releas</u> <u>CSS C</u> <u>Speaking</u> <u>Contact</u>	ing CSS ookboo	<u>k</u>										
Lorem accum	ipsum dolor san. Vivamus Eusce lacus	sit ame s comm	et, conse odo. Eti	am eros	dipiso massa ttis id	cing elit. a, hendren tristique	Vivamu it hend	s vitae lig rerit, ultri	gula ut r icies fac	netus int ilisis, po giat Etis	erdum rttitor e	get,	•

Рис. 7.13. Стандартное отображение ссылок

## Решение

Сначала создайте правильно сформированный набор маркированных ссылок:

```
<div id="navsite" class="clearfix">
  Site navigation:

    <a href="/">Home</a>
    <a href="/about/">About</a>
    <a href="/archives/">Archives</a>
    <a href="/writing/">Writing</a>
    <a href="/writing/">Writing</a>
    <a href="/speaking/" id="current">Speaking</a>
    <a href="/contact/">Contact</a>
    <a href="/contact/">Contact</a>
```

Затем задайте правила CSS для навигационной структуры, обязательно указав свойство float для пунктов списка (рис. 7.14):

```
#navsite p {
  display: none;
}
```

```
#navsite ul {
    width: 100%;
    float: left;
padding: 0;
margin-left: 0;
border-bottom: 1px solid #778;
 font: bold 12px Verdana, sans-serif;
}
#navsite ul li {
list-style: none;
margin: 0;
 float: left;
ļ
#navsite ul li a {
padding: 12px 0.5em;
margin-left: 3px;
border: 1px solid #778;
border-bottom: none;
background-color: #666;
 text-decoration: none;
background-image: url(title-glass.png);
background-position: 50%;
background-repeat: repeat-x;
display: block;
width: 7em;
}
#navsite ul li a:link {
 color: white;
ļ
#navsite ul li a:visited {
 color: #667;
}
#navsite ul li a:link:hover, #navsite ul li a:visited:hover {
color: #000;
background-color: #aae;
border-color: #227;
}
#navsite ul li a#current {
background-color: white;
border-bottom: 1px solid white;
color: #448;
margin-bottom: -1px;
}
#navsite ul li a#current:hover {
background-image: url(title-glass.png);
background-position: 50%;
background-repeat: repeat-x;
}
.clearfix:after {
 content: ".";
```

```
display: block;
```

```
height: 0;
clear: both;
visibility: hidden;
}
/* for IE6 */
* html .clearfix {
height: 1%;
}
/* for IE7 */
*:first-child+html .clearfix {
min-height: 1px;
}
```



Рис. 7.14. Навигационное меню с закладками-ярлыками

## Обсуждение

Первый фрагмент решения скрывает заголовок. Это делается потому, что визуального представления в виде навигационного меню из закладок-ярлыков достаточно, чтобы пользователь понял, что это навигационные ссылки:

```
#navsite p {
  display: none;
}
```

В следующем правиле определены поле и отступ для блока, который создан элементом маркированного списка ul. Линия, протянувшаяся вдоль нижнего края закладокярлыков, нарисована благодаря свойству border-bottom (рис. 7.15):

```
#navsite ul {
  width: 100%;
  float: left;
  padding: 0;
```

```
margin-left: 0;
border-bottom: 1px solid #778;
font: bold 12px Verdana, sans-serif;
}
```



Рис. 7.15. Линия, на которой будут повторно установлены навигационные закладки

Далее следует объявление, которое делает возможным отображение горизонтального навигационного меню с помощью маркированного списка, — свойство float для пунктов списка:

```
#navsite ul li {
    list-style: none;
    margin: 0;
    float: left;
}
```

#### Примечание

Другой способ построения горизонтальных меню — применение свойства inline. Несмотря на то, что оба варианта в результате позволяют получить горизонтальное меню, задавая для пункта списка свойство float, вы сохраняете у элемента свойства блочного уровня, что позволит указать для последовательных пунктов меню такие свойства, как margin и width. Если вы не хотите, чтобы пункты были одной ширины или предпочитаете задать одинаковые пробельные промежутки между ними, свойство inline — приемлемая альтернатива вместе со свойствами padding-left и padding-right.

Вместо того чтобы располагать элементы списка по умолчанию, один над другим, Web-обозреватель размещает элементы списка так же, как текст, изображения и другие строчные элементы (рис. 7.16).

Для создания пункта меню в виде закладки-ярлыка воспользуйтесь свойством border в следующем правиле CSS:

```
#navsite ul li a {
  padding: 12px 0.5em;
  margin-left: 3px;
  border: 1px solid #778;
  border-bottom: none;
  background-color: #666;
  text-decoration: none;
  background-image: url(title-glass.png);
  background-position: 50%;
  background-repeat: repeat-x;
  display: block;
  width: 7em;
}
```



Рис. 7.16. Горизонтальное расположение списка

Первое свойство border с сокращенной формой записи задает сплошную однопиксельную рамку вокруг ссылки. Но следующее сразу за ним свойство border-bottom предписывает обозревателю не отображать границу под ссылкой.

Значение свойства border-bottom отображается поверх свойства border с сокращенной формой записи (рис. 7.17). Объявление в свойстве border-bottom переопределяет объявление в свойстве border благодаря порядку их следования.

После создания рамки вокруг закладки задайте цвет текста ссылок и визуальное отображение наведения указателя мыши на ссылку:

```
#navsite ul li a:link {
  color: white;
}
#navsite ul li a:visited {
  color: #667;
}
```

```
#navsite ul li a:link:hover, #navsite ul li a:visited:hover {
 color: #000;
 background-color: #aae;
 border-color: #227;
}
          CSS Cookbook
                            C
                                    X
                                           A
                                                 1P
                   1
                                                                    📋 (file:///U 😭 🔻 ) - (🎦 🕻 Google
                                                                                                           Q
                                                        1Password
          Back
                 Forward
                            Reload
                                          Home
                                                  Print
         HTML Validation
                                            Christopher Schmitt : De.
                                                                    0
                                                                                                   0
                                                                                                        +
                  CSS Cookbook
                                     0
                                                                        🚑 Technology Books, Tech
                                          rchives
                                                        Nriting
            Home
                           bout
                                                                        Speaking
                                                                                      Contact
         Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vivamus vitae ligula ut metus interdum
          accumsan. Vivamus commodo. Etiam eros massa, hendrerit hendrerit, ultricies facilisis, porttitor eget,
         magna. Fusce lacus ligula, dapibus sed, mattis id, tristique non, arcu. Vivamus feugiat. Etiam erat pede,
          posuere placerat, porttitor vitae, sollicitudin ut, tellus. Nunc volutpat velit blandit enim. Nulla luctus. In
          viverra nibh ac massa. Curabitur vel tellus a velit feugiat tristique. Mauris ultricies cursus diam. Nullam
          scelerisque blandit pede. Maecenas tellus velit, pellentesque placerat, tincidunt nec, tempus at, tellus.
          Quisque sapien. Nullam eget ante at mi consequat mattis. Nunc accumsan tincidunt ipsum. Sed adipiscing
          tincidunt lorem. Mauris arcu. Vestibulum laoreet augue.
```

Рис. 7.17. Видимые закладки

Заключительное правило CSS определяет, как будет выглядеть "находящаяся в обращении" ссылка. Этот стиль применяется к ссылке, представляющей страницу, просматриваемую пользователем в настоящий момент (рис. 7.18):

```
#navsite ul li a#current {
  background: white;
  border-bottom: 1px solid white;
}
```

374

Затем добавьте самоочищающее свойство float (см. *разд. 2.22*) для навигационного меню в целом в элементе div. Оно гарантирует, отсутствие обтекания навигационного меню любым текстом или контентом из остальной части страницы:

```
.clearfix:after {
  content: ".";
  display: block;
  height: 0;
  clear: both;
  visibility: hidden;
 }
 /* for IE6 */
 * html .clearfix {
  height: 1%;
 }
 /* for IE7 */
 *:first-child+html .clearfix {
  min-height: 1px;
 }
```

	), C			file:///U	<u>☆</u> ▼) - 🚷 - (	Google
Back Forw	ard Reload St	top Home Print	t 1Password			-
	on Cookhook	Christenh	ar Sahmitta Da	O / Tashua	lanu Baalus Tash	0 1
	COORDOOK	Christophi	er seminte. De	Con recimo	nogy books, recir.	
Home	About	Archives	Writing	Speaking	Contact	
			2-12			100
orem insu	m dolor sit amet (	consectetuer adin	hiscing elit. Viv	amus vitae ligu	la ut metus inter	dum
orem ipsur	n dolor sit amet, o	consectetuer adip	biscing elit. Viv	vamus vitae ligu	la ut metus inter	dum
orem ipsur	n dolor sit amet, d /ivamus commod	consectetuer adip lo. Etiam eros ma	biscing elit. Viv	vamus vitae ligu hendrerit, ultrici	la ut metus inter es facilisis, porti	dum titor eget,
orem ipsur ccumsan. V nagna, Fuse	n dolor sit amet, d /ivamus commod ce lacus ligula, da	consectetuer adip lo. Etiam eros ma pibus sed, mattis ae. sollicitudin ut	biscing elit. Viv assa, hendrerit l i id, tristique no	vamus vitae ligu hendrerit, ultrici on, arcu. Vivamu volutnat velit bli	la ut metus inter es facilisis, portt us feugiat. Etiam andit enim Null	dum titor eget, a erat pede
orem ipsur ccumsan. V nagna. Fuse osuere plac	n dolor sit amet, o /ivamus commod ce lacus ligula, da cerat, porttitor vita ac massa. Curab	consectetuer adip lo. Etiam eros ma pibus sed, mattis ae, sollicitudin ut inr yel tellus a y	biscing elit. Viv assa, hendrerit l a id, tristique no t, tellus. Nunc v	vamus vitae ligu hendrerit, ultrici on, arcu. Vivamu volutpat velit bla tique. Mauris ul	la ut metus inter es facilisis, portu us feugiat. Etiam andit enim. Nulla tricies cursus dis	dum titor eget, a erat pede a luctus. In am Nullar
orem ipsur ccumsan. V nagna. Fuso osuere plac iverra nibh	n dolor sit amet, /ivamus commod ze lacus ligula, da zerat, portitor vita ac massa. Curab blandit pede. Mas	consectetuer adip lo. Etiam eros ma pibus sed, mattis ae, sollicitudin ut itur vel tellus a ve	biscing elit. Viv assa, hendrerit l 5 id, tristique no t, tellus. Nunc v elit feugiat tristi t, pellentecque	vamus vitae ligu hendrerit, ultrici on, arcu. Vivamu volutpat velit bla tique. Mauris ul	la ut metus inter es facilisis, port us feugiat. Etiam andit enim. Null tricies cursus dia	dum titor eget, a erat pede a luctus. In am. Nullar
orem ipsur ccumsan. V nagna. Fuso osuere plac iverra nibh celerisque	n dolor sit amet, o /ivamus commod ce lacus ligula, da cerat, porttitor vita ac massa. Curab blandit pede. Mac	consectetuer adip lo. Etiam eros ma pibus sed, mattis ae, sollicitudin ut itur vel tellus a ve ccenas tellus velit	biscing elit. Viv assa, hendrerit h s id, tristique no t, tellus. Nunc v elit feugiat trist t, pellentesque	vamus vitae ligu hendrerit, ultrici on, arcu. Vivamu volutpat velit bla tique. Mauris ul placerat, tincidu	la ut metus inter es facilisis, porti us feugiat. Etiam andit enim. Nulli tricies cursus dia unt nec, tempus a	dum titor eget, a erat pede a luctus. In am. Nullar at, tellus.
orem ipsur ccumsan. V nagna. Fuso osuere plac iverra nibh celerisque l Quisque sap	n dolor sit amet, o /ivamus commod ce lacus ligula, da cerat, porttitor vita ac massa. Curab blandit pede. Mac ien. Nullam eget	consectetuer adip lo. Etiam eros ma pibus sed, mattis ae, sollicitudin ut itur vel tellus a ve ccenas tellus velit ante at mi consec Vartibulura horse	biscing elit. Viv assa, hendrerit h s id, tristique no t, tellus. Nunc v elit feugiat trist t, pellentesque quat mattis. Nu	vamus vitae ligu hendrerit, ultrici on, arcu. Vivami volutpat velit bla tique. Mauris ul placerat, tincidu inc accumsan tin	la ut metus inter es facilisis, porti us feugiat. Etiam andit enim. Nulli tricies cursus dia unt nec, tempus a ncidunt ipsum. S	dum titor eget, a erat pede a luctus. In am. Nullar at, tellus. ded adipisc

Рис. 7.18. Внешний вид текущей ссылки

#### См. также

Прототип полосы меню закладок (как и другие стили навигационных элементов) на странице http://css.maxdesign.com.au/listmatic/horizontal05.htm.

# 7.13. Построение горизонтальных навигационных меню с раскрывающимися меню

## Проблема

Вы хотите вставить раскрывающееся меню в горизонтальное навигационное меню, как показано на рис. 7.19.

## Решение

Как в *разд.* 7.11, дополните основное навигационное меню, включив в соответствующий пункт меню вложенный маркированный список:

```
<div id="navsite">
Site navigation:

<a href="/">Home</a>
<a href="/about/">About</a>
<a href="/archives/">Archives</a>
<a href="/writing/">Writing</a>

<a href="/writing/releasing-css">Releasing CSS</a>
<a href="/writing/css-cookbook">CSS Cookbook</a>
```

```
a href="/speaking/">Speaking</a>a href="/contact/">Contact</a></div>
```



Рис. 7.19. Внешний вид текущей ссылки

Затем, используя селекторы потомка для определения стилей вложенного маркированного списка, задайте отрицательное значение, определяющее положение пунктов раскрывающегося меню:

```
#navsite ul li ul {
    position: absolute;
    width: 7em;
    left: -999em;
    float: none;
    border-bottom: none;
}
```

Для их вызова используйте псевдокласс : hover:

```
#navsite ul li:hover ul
{left: auto;
}
#navsite ul li ul li a {
    background: #999;
    border: 1px solid black;
    margin-top: 4px;
}
```

#### Обсуждение

В современных Web-обозревателях решение действует хорошо. Но для поддержки в разных Web-обозревателях можно применить дополнительный код на JavaScript и модифицированные правила CSS, названные Son of Suckerfish Dropdowns (http://htmldog.com/articles/suckerfish/dropdowns/).

Легкий метод преобразования Son of Suckerfish Dropdowns в простые функции jQuery можно найти по адресу http://nederdev.com/artickles/suckerfish-meets-iquery.

#### Раскрывающиеся меню поверх флэш-дизайна

Если у вас есть раскрывающееся меню, появляющееся в Web-документе поверх флэшанимации, то вы знаете, что такое меню может иногда не отображаться или его может скрыть флэш-анимация. Для исправления такого поведения попробуйте задать параметр wmode со значением transparent, позволяющим отображать раскрывающееся меню:

#### См. также

Web-сайт Listamatic с многочисленными примерами и кодом навигационных меню на странице http://css.maxdesign.com.au/listamatic/.

# 7.14. Создание навигационного меню с клавишами доступа

## Проблема

Вы хотите создать навигационное меню с клавишами доступа.

## Решение

Создайте набор ссылок в виде маркированного списка с атрибутом accesskey в элементах-якорях:

```
div id="navsite">

        <a href="/" accesskey="h">Home</a>
        <a href="/about/" accesskey="b">About</a>
        <a href="/archives/" accesskey="a">Archives//a>
        <a href="/writing/" accesskey="w">Writing</a>
        <a href="/writing/" accesskey="w">Writing</a>
        <a href="/speaking/" accesskey="s">Speaking</a>
        <a href="/contact/" accesskey="s">Contact</a>
```

```
</div>
```

Затем вставьте в элементы span буквы, которые вы хотите указать в качестве клавиш доступа:

```
<div id="navsite">

<a href="/" accesskey="h"><span class="akey">H</span>ome</a>
<a href="/about/" accesskey="b">A<span class="akey">b</span>out</a>
<a href="/archives/" accesskey="a"><span
class="akey">A</span>rchives/" accesskey="a"><span
class="akey">A</span>rchives/" accesskey="a"><span
class="akey">A</span>rchives//a>
<a href="/writing/" accesskey="w"><span class="akey">W</span>riting</a>
<a href="/speaking/" accesskey="s"><span
class="akey">S</span>peaking/" accesskey="s"><span
class="akey">S</span>riting</a>
<a href="/contact/" accesskey="s"><span
class="akey">C</span>ontact</a>
<a href="/contact/" accesskey="c"><span class="akey">C</span>ontact</a>
<a href="/contact/" accesskey="c"><span class="akey">C</span>ontact</a>
```

Далее задайте стилевое оформление клавиш доступа с помощью селектора класса (рис. 7.20):

```
.akey {
  text-decoration: underline;
}
```

## Обсуждение

Клавиши доступа позволяют посетителям легко перемещаться по Web-сайту без помощи мыши. В данном решении клавиши доступа назначены навигационным элементам. Нажав одну из них, пользователь переходит на страницу, заданную в ссылке.



При единообразном использовании клавиш доступа посетитель сайта может применять для навигации одну и ту же последовательность клавиш и таким образом сформировать прочные пользовательские навыки.

#### Известные проблемы Web-обозревателей

Клавиши доступа должны работать в Web-обозревателях IE 4, Firefox, Safari, Chrome и Opera 7 и их более поздних версиях.

Одно из препятствий использования клавиш доступа — отсутствие стандартного набора клавиш, соответствующих определенным ссылкам. Например, не лучше ли применять букву h для "Home Page" (как сделано в данном примере) или букву m для "Main Page"?

# См. также

Клавиши доступа в спецификации HTML 4 на странице http://www.w3.org/ TR/html4/interact/forms.html#h-17.11.2; статью Стюарта Робертсона (Stuart Robertson) "Accesskeys: Unlocking Hidden Navigation" (Клавиши доступа: разблокировка скрытой навигации) на http://alistapart.com/articles/accesskeys/.

# 7.15. Создание переходов с помощью ссылок-цепочек

# Проблема

Вы хотите создать вложенный список, наподобие показанного на рис. 7.21, для формирования линейки *ссылок-цепочек* (breadcrumb navigation), представляющей собой набор ссылок, возвращающих на начальную страницу (рис. 7.22).

# Решение

Прежде всего, создайте правильно структурированный набор вложенных маркированных ссылок, которые указывают местонахождение страниц на сайте:

```
<div id="crumbs">
<h3>Location:</h3>
<a href="/">Home</a>
<a href="/writing/">Writing</a>
<a href="/writing/books/">Books</a>
<a href="/writing/books/">CSS Cookbook</a>
```

 </div>



Рис. 7.21. Стандартная визуализация вложенного списка

O CSS Cookbook	$\bigcirc$
	sers/chr 😭 🔻 - 🚷 - Google 🛛 🔾
Back Forward Reload Stop Home Print IPassword	
	Tashaslasu Basha Tash Ca. O di 🗮
CSS Cookbook	Technology Books, Tech Co 🥹 🕂 🗨
	÷
	^
Home $\Rightarrow$ Writing $\Rightarrow$ Books $\Rightarrow$ CSS Cookbook	
Lorom incum dolor sit amot	
Loreni ipsuni uoior sit amet	
consectetuer adipiscing elit. Vivamus vitae ligula ut metus interdu	um accumsan. Vivamus
commodo. Etiam eros massa, hendrerit hendrerit, ultricies facilisi	is, porttitor eget, magna.
Fusce lacus ligula, dapibus sed, mattis id, tristique non, arcu. Viv	amus feugiat. Etiam erat
enim Nulla luctus. In viverra nibh ac massa. Curabitur val tellus	a velit fengiat tristique
Mauris ultricies cursus diam Nullam scelerisque blandit pede. M	aecenas tellus velit
pellentesque placerat, tincidunt nec, tempus at, tellus. Quisque sa	apien. Nullam eget ante at
mi consequat mattis. Nunc accumsan tincidunt ipsum. Sed adipis	scing tincidunt lorem.
Mauris arcu. Vestibulum laoreet augue.	
	*

Теперь задайте свойство display как для списков ul, так и для пунктов списка li:

```
#crumbs {
background-color: #eee;
padding: 4px;
#crumbs h3 {
display: none;
}
#crumbs ul {
display: inline;
padding-left: 0;
margin-left: 0;
}
#crumbs ul li {
display: inline;
}
#crumbs ul li a:link {
padding: .2em;
}
```

В каждый вложенный список поместите маленькое фоновое изображение стрелки слева от ссылки:

```
#crumbs ul ul li{
   background-image: url(arrow_r.gif);
   background-repeat: no-repeat;
   background-position: left;
   padding-left: 20px;
}
```

#### Обсуждение

Как в сказке "Гензель и Гретель", дорожка из хлебных крошек (breadcrumb trail) используется для того, чтобы помочь людям найти дорогу домой, в Web-пространстве дорожка ссылок-цепочек показывает путь к странице, которую пользователь просматривает в данный момент (рис. 7.23).

Из решения можно было бы удалить свойство background-image, если бы большая часть Web-обозревателей поддерживала псевдоэлемент :before. В решение в этом случае можно было бы включить другое правило CSS, похожее на следующее:

```
#crumbs ul ul li:before {
  content: url(arrow.gif);
}
```

Во время написания книги все основные Web-обозреватели за исключением IE7 и более ранних версий поддерживали псевдоэлемент :before.

	outers < Publishers in the Yahoo! Directory	0
Back Forward Reload Stop Home	Print 1Password 1P	☆▼ · 🚷 Google Q
HTML Validation		C.0. 1941
🧤 Computers < Publishers in t 🕄 🚺	Christopher Schmitt : Desig 🕲 📗 🚑 Technol	ogy Books, Tech Co 🕲 🕇 🛒
Yahoo! My Yahoo! Mail Welcome, christopher	schmit [Sign Out, My Account]	Directory Home Help
YAHOO! DIRECTOR	Y ◯ Web   ⊙ Directory   ◯ Category	Search Cove the Add it to
Publishers > Computers	Email this (	page Suggest a Site Advanced Search
Computers	Second Services > Toblishers > Second and reconnect	
	N	
	SPONSOR RESULTS	SPONSOR RESULTS
	See your site here	
CATEGORIES (What's This?)		YourinternationSulde
● <u>Game Guides@</u>		↓ ▼ ● ● ● ● ●

Рис. 7.23. Пример дорожки ссылок-цепочек

#### См. также

Аннотированную версию сказки "Гензель и Гретель" на странице http:// www.surlalunefairytales.com/hanselgretel/index.html; научную статью об эффективности навигации с помощью ссылок-цепочек на http://psychology.wichita.edu/ surl/usabilitynews/52/breadcrumb.htm.

# 7.16. Создание с помощью изображений визуальных эффектов наведения указателя мыши на ссылку

#### Проблема

Вы хотите заменить изображениями текстовые ссылки при наведении на них указателя мыши.

#### Решение

Сначала заключите текст, находящийся в якоре, в элемент span:

```
<a href="/" id="linkhome">
<span> Homepage</span>
</a>
```

Далее вместо применения языка JavaScript используйте для смены изображений свойство background-image в селекторах псевдоклассов :hover и :active (рис. 7.24):

```
a span {
display: none;
}
a:link {
display: block;
width: 100px;
height: 50px;
background-image: url(submit.png);
background-repeat: no-repeat;
background-position: top left;
}
a:link:hover {
display: block;
width: 100px;
height: 50px;
background-image: url(submit-roll.png);
background-repeat: no-repeat;
background-position: top left;
}
a:link:active {
display: block;
width: 100px;
height: 50px;
background-image: url(submit-on.png);
background-repeat: no-repeat;
background-position: top left;
}
```



Рис. 7.24. Ссылка в стандартном, активном состояниях и при наведении на нее указателя мыши

### Обсуждение

У замены текста изображением есть пять достоинств. Во-первых, она разделяет текстовое содержимое страницы и ее представление. Изображение, содержащее более искусно отформатированный текст, — это часть представления страницы, и следовательно, оно управляется стилевыми установками, в то время как в содержимом разметки остается чистый текст. Второе достоинство состоит в том, что заголовок-изображение можно модифицировать на всем сайте, внеся одно изменение в таблицу стилей. В-третьих, этот метод действует при применении альтернативных стилей или при переключении таблиц стилей.

Включение элемента span в другой элемент позволяет скрыть HTML-текст и дает возможность элементу дизайна страницы, такому как изображение, появляющееся при наведении указателя мыши, выводиться в качестве фонового изображения. В-четвертых, если в Web-обозревателе пользователя не включена поддержка CSS, предлагаемое решение будет выводить стандартный HTML-текст, избавляя пользователя от необходимости загружать ненужные изображения. Пятое достоинство состоит в том, что данное решение яснее и проще, чем вариант с применением JavaScript.

Вы также можете использовать этот метод для элементов страницы, не требующих создания визуального эффекта при наведении указателя мыши, например, для вставки изображения, заменяющего текст заголовка, что гарантирует вывод специфического шрифта, которого обычно нет на компьютерах пользователей, в виде изображения. Для этого, прежде всего, выполните следующую разметку (рис. 7.25):

<h2 id="headworld"><span>Hello, world!</span></h2>



Рис. 7.25. Стандартное отображение заголовка

Затем задайте следующие правила CSS для вставки изображения (рис. 7.26):

```
h2#headworld span {
  display: none;
}
h2#headworld {
  width: 395px;
  height: 95px;
  background-image: url(heading.gif);
  background-repeat: no-repeat;
  background-position: top left;
}
```



Рис. 7.26. Замена текста изображением

Многие называют этот метод FIR-методом (Fahrner Image Replacement, замещение текста изображением Фарнера), названным в честь Тодда Фарнера (Todd Fahrner) (см. *разд. 4.20*).

Недостаток данного решения проявляется при использовании экранных считывателей или дикторов (screen readers), программ, делающих компьютеры доступными для слепых и слабовидящих людей. Некоторые экранные считыватели не читают элементы с установкой display: none. Дополнительную информацию об этом можно найти в статье "Facts and Opinion About Fahrner Image replacement" (Факты и мнения о методе замещения текста изображением Фарнера) на сайте http://www.alistapart.com/ articles/fir/.

#### Замещение изображением Лихи — Лэнгриджа

Альтернативой этому решению служит LIR-метод (Leahy-Langridge Image Replacement, замещение текста изображением Лихи — Лэнгриджа). Разработанный независимо Симусом Лихи (Seamus Leahy) и Стюартом Лэнгриджем (Stuart Langridge) LIR-метод убирает текст из поля зрения. Достоинство этого метода состоит в том, что для скрытия текста не нужен дополнительный элемент span. Допустим, что у заголовка стандартная HTML-разметка:

<h2 id="headworld">Hello, world!</h2>

Изображение для замены заголовка появляется как фоновое, поскольку правило CSS задает отступ с высотой, равной высоте заголовка-изображения. Поэтому свойство height задается со значением 0:

```
h2#headworld {
/* Ширина изображения */
width: 395px;
/* Высота изображения — первая величина в свойстве padding */
padding: 95px 0 0 0;
overflow: hidden;
background-image: url(heading.gif);
background-repeat: no-repeat;
voice-family: "\"}\"";
voice-family: inherit;
```

```
height /**/: 95px;
height: 0px !important;
}
```

Последние четыре строки правила CSS необходимы для того, чтобы скомпенсировать слабую поддержку блочной модели в IE7 и его более ранних версиях (см. *разд. 2.10*). Таким образом, более ранние версии IE получат значение высоты 95 px, а остальные Web-обозреватели — нулевое значение.

#### Примечание

Другой способ — применение условных комментариев для передачи особых значений в Web-обозреватели IE. Дополнительную информацию см. в *разд. 12.7*.

#### Ріху-метод

Еще один способ создания визуального эффекта наведения указателя мыши с помощью изображения реализуется свойством background-position. Этот метод, называемый Pixy-методом (также известный как CSS-спрайты, описанные в *разд. 4.33*), предусматривает включение отображений всех трех состояний ссылки в одно изображение и в дальнейшем перемещение позиции изображения с помощью свойства background-position (рис. 7.27):

```
a span {
display: none;
}
a:link, a:visited {
display: block;
width: 125px;
height: 30px;
background-image: url(btn_omni.gif);
background-repeat: no-repeat;
background-position: 0 0;
}
a:link:hover, a:visited:hover {
display: block;
width: 125px;
height: 30px;
background-image: url(btn omni.gif);
background-repeat: no-repeat;
/* Сдвиг изображения на 30 пикселов вверх */
background-position: 0 -30px;
}
a:link:active, a:visited:active {
display: block;
width: 125px;
height: 30px;
background-image: url(btn_omni.gif);
background-repeat: no-repeat;
/* Сдвиг изображения на 60 пикселов вверх */
background-position: 0 -60px;
}
```



Рис. 7.27. Отображение порции изображения, создающей визуальный эффект при наведении на ссылку указателя мыши

#### Предупреждение

Недостаток почти всех современных методов замещения текста изображением заключается в том, что пользователи не видят на экране ничего, если вывод изображений отключен или блокирован или просто изображения не загружаются при сохранении поддержки CSS. Важно исследовать и применить метод, наиболее подходящий для вашей ситуации. Избегайте замещения изображением важных заголовков. Дополнительную информацию о методах замещения изображением см. в *разд. 4.20*.

#### См. также

*Разд. 4.20* о замещении HTML-текста привлекательной графикой или типографикой; еще один пример LIR-метода Симуса Лихи (Seamus Leahy) на сайте http:// www.moronicbajebus.com/playground/cssplay/image-replacement/; объяснение того, как создавать при подключении CSS более быстрые изменения отображения ссылок при наведении указателя без предварительной загрузки изображений на страницу http://wellstyled.com/css-nonpreload-rollovers.html; краткое описание FIR-метода на http://www.stopdesign.com/also/articles/replace\_text/.

# 7.17. Создание сворачиваемых меню

#### Проблема

Вы хотите скрыть набор ссылок и дать пользователю возможность вывести на экран эти ссылки при необходимости. Например, вместо вывода двух маркированных списков ссылок один скрыть (как показано на рис. 7.28) и затем позволить пользователю вывести его, щелкнув кнопкой мыши на знаке "+" (как показано на рис. 7.29).

#### Решение

Сначала разметьте HTML-ссылки как сворачиваемые с помощью атрибута id в элементе ul:

```
<h5>Interesting Links (+/-)</h5>
```

```
<a href="http://www.oreilly.com/">O'Reilly</a><a href="http://www.slashdot.org/">Slashdot</a><a href="http://www.apple.com/">Apple</a><a href="http://www.microsoft.com/">Microsoft</a><a href="http://www.mozilla.org/">Mozilla</a>
```

#### 



Рис. 7.28. Запрещение вывода на экран второго набора ссылок



Затем создайте правило CSS, препятствующее выводу на экран второго набора ссылок при первоначальной загрузке страницы:

```
#menulink {
  display: none;
}
```

Теперь добавьте следующую функцию на языке JavaScript, переключающую список ссылок с помощью замены значения block свойства display на значение none и наоборот:

```
function kadabra(zap) {
  if (document.getElementById) {
   var abra = document.getElementById(zap).style;
   if (abra.display == "block") {
     abra.display = "none";
     } else {
     abra.display = "block";
   }
   return false;
   } else {
    return true;
   }
}
```

Вставьте якорь с событием JavaScript onclick в разметку заголовка. Когда пользователь щелкнет ссылку кнопкой мыши, этот щелчок запустит функцию на языке JavaScript:

<h5><a href="#" onclick="return kadabra('menulink');"> Interesting Links (+/-)</a></h5>

#### Примечание

Оболочки JavaScript, такие как jQuery (см. *главу* 14), могут переработать предложенное решение, не вставляя в HTML-элементы событий, относящихся к JavaScript (техника программирования, называемая "ненавязчивым" JavaScript).

# Обсуждение

В приведенной функции на JavaScript используется метод getElementById для переключения вывода списка ссылок из меню. Данный метод можно настроить без включения дополнительных строк на JavaScript для отображения многочисленных меню или фрагментов Web-документа:

```
Are you sure you want to know the truth? If so,
follow <a href="#" onclick="return kadabra('spoiler'); ">this link.</a>
Darth Vadar was Luke's father!
```

Данный метод работает в Web-обозревателях Netscape Navigator 6, Opera 7.5, Internet Explorer 5 для Windows и более поздних версиях этих обозревателей, а также в Safari.

#### См. также

Дополнительную информацию о методе getElementById на странице http:// www.mozilla.org/en/DOM/document.getElementById.

# 7.18. Создание контекстных меню

#### Проблема

У вас есть навигационное меню (рис. 7.30), созданное в *разд. 7.10*, и вы хотите выделить цветом в меню местонахождение текущей страницы.

#### Решение

Поместите атрибут id в элемент body Web-документа:

<body id="pagespk">

Кроме того, вставьте в якори атрибуты id для каждой ссылки меню:

```
<div id="navsite">
<h5>Site navigation:</h5>
<a href="/" id="linkhom">Home</a>
<a href="/about/" id="linkabt">About</a>
```

```
<a href="/archives/" id="linkarh">Archives</a>
<a href="/writing/" id="linkwri">Writing</a>
<a href="/speaking/" id="linkspk">Speaking</a>
<a href="/contact/" id="linkcnt">Contact</a>

</div>
```



Рис. 7.30. Набор навигационных ссылок

С помощью правил CSS поместите два селектора id в один селектор потомка для завершения меню (рис. 7.31):

```
#pagespk a#linkspk {
  border-left: 10px solid #f33;
  border-right: 1px solid #f66;
  border-bottom: 1px solid #f33;
  background-color: #fcc;
  color: #333;
}
```

#### Обсуждение

Если у вас маленький сайт, вы можете показать в наборе навигационных ссылок ссылку, представляющую текущую страницу, убрав якорь-ссылку для этой страницы:

```
<div id="navsite">
  <h5>Site navigation:</h5>

  <a href="/" id="linkhom">Home</a>
  <a href="/about/" id="linkabt">About</a>
```

```
<a href="/archives/" id="linkarh">Archives</a><a href="/writing/" id="linkwri">Writing</a><a href="/contact/" id="linkcnt">Contact</a></u></div>
```



Рис. 7.31. Текущая ссылка, отличающаяся от остальных ссылок

В случае сайтов большего объема, у которых могут быть дополнительные меню, удаление тегов-ссылок для каждой страницы увеличит время создания и сопровождения. Разметив ссылки соответствующим образом, их можно вызывать на серверной стороне, и затем вы можете редактировать правила CSS, управляющие стилем навигационных ссылок так, как нужно.

Для расширения правила CSS и включения в него всех ссылок навигационного меню сгруппируйте селекторы потомка с помощью запятой и, по крайней мере, одного пробела:

```
#pagehom a#linkhom:link,
#pageabt a#linkabt:link,
#pagearh a#linkarh:link,
#pagewri a#linkwri:link,
#pagespk a#linkspk:link,
#pagecnt a#linkcnt:link {
border-left: 10px solid #f33;
border-right: 1px solid #f66;
border-bottom: 1px solid #f63;
background-color: #fcc;
color: #333;
}
```

В элемент body любого Web-документа обязательно вставляйте соответствующий атрибут id. Например, для начальной или основной страницы сайта элемент body выглядит так:

<body id="pagehom">

#### См. также

Селекторы потомка в спецификации CSS 2.1 на странице http://www.w3.org/ TR/CSS21/selector.html#descendant-selectors.

# 7.19. Создание всплывающих подсказок с помощью атрибута *title*

#### Проблема

Вы хотите, чтобы появлялась всплывающая подсказка при наведении указателя мыши на ссылку.

#### Решение

Используйте в теге ссылки атрибут title для создания всплывающей подсказки, подобной той, что показана на рис. 7.32:

```
<a href="http://www.google.com/" title="Search the Web">...</a>
```



Рис. 7.32. Значение атрибута title, отображаемое как всплывающая подсказка

#### Обсуждение

Всплывающая подсказка может применяться почти к любому элементу на Webстранице для облегчения его использования. Попробуйте воспользоваться всплывающими подсказками не только для ссылок, но и для ячеек таблицы и элементов ввода формы.

#### См. также

Атрибут title в спецификации языка HTML 4.1 на странице http://www.w3.org/ TR/html4/struct/global.html#h-7.4.3.

# 7.20. Разработка динамического меню из закладок

# Проблема

Вы хотите создать навигационное меню из закладок-ярлыков со скругленными углами, которое бы работало даже при изменении размера шрифта; на рис. 7.33 показан стандартный вариант меню.



Рис. 7.33. Динамическая навигация с помощью закладок-ярлыков

#### Решение

Сначала напишите разметку для навигационного меню:

```
<div id="header">
<h2>Personal Site dot-com</h2>
<h5>Site navigation:</h5>
<a href="/">Home</a>
<a href="/about/">About</a>
<a href="/archives/">Archives</a>
<a href="/writing/">Writing</a>
```

```
id="current"><a href="/speaking/">Speaking</a><a href="/contact/">Contact</a></div>
```

Затем создайте два изображения закладки-ярлыка: одно для ссылок-якорей, а другое для представления текущей страницы, просматриваемой пользователем. Разделите каждое изображение закладки-ярлыка на два изображения (рис. 7.34).



Рис. 7.34. Разделенное на две части изображение закладки-ярлыка (обратите внимание на скругления верхних углов изображений)

Затем поместите правую часть закладки-ярлыка в фон элемента списка:

```
#header li {
  float: left;
  background-image: url(tab_right.gif);
  background-repeat: no-repeat;
  background-position: right top;
  margin:0;
  padding: 0;
}
```

Поместите левую часть изображения закладки-ярлыка в фон элемента-якоря:

```
#header a {
  display: block;
  background-image: url(tab_left.gif);
  background-repeat: no-repeat;
  background-position: left top;
  padding: 5px 15px;
  color: #ccc;
  text-decoration: none;
  font-family: Georgia, Times, "Times New Roman", serif;
}
```

Назначьте специальную закладку-ярлык для представления текущего Web-документа, просматриваемого в данный момент:

```
#header #current {
   background-image:url("tab_right_current.gif");
}
#header #current a {
   background-image:url("tab_left_current.gif");
   color: black;
}
```

Поместите в нижнюю часть группировки изображение линии толщиной в один пиксел.

## Обсуждение

Сохранение текста в навигационных ссылках облегчает использование, дизайн и сопровождение Web-сайта. Например, пользователи со слабым зрением могут отрегулировать размер шрифта и закладок без нарушения дизайна страницы.

Поскольку пользователи могут задавать очень большие размеры шрифта, фоновые изображения закладок-ярлыков должны быть также достаточно большими; в противном случае они будут отображаться разорванными. В данном решении у изображений закладок-ярлыков высота — 450 пикселов.

Web-разработчики предпочитают этот метод, т. к. он позволяет с легкостью поддерживать список ссылок. Для изменения навигационной метки или устранения опечатки разработчики могут просто отредактировать HTML-текст без необходимости повторного обращения к программе редактирования изображений для создания изображений закладок-ярлыков.

Еще одно достоинство этого метода — возможность разработки более эстетически привлекательного отображения закладки-ярлыка. В *разд. 7.12* показано, как создавать навигационное меню из закладок-ярлыков с помощью свойства border. Этот подход позволяет сформировать прямоугольные или квадратные рамки закладок. В данном решении Web-разработчики могут скруглить углы закладок и ввести комбинирование цветов для улучшения эстетического представления.

## См. также

*Разд. 3.22* о применении аналогичного приема "резиновой" верстки для создания врезок (pull quotes) с помощью изображений; статью "Sliding Doors of CSS, Part II" (Pаздвижные двери CSS, часть 2) на странице http://www.alistapart.com/articles/slidingdoors2/, расширяющую концепцию применения навигационного меню из закладок-ярлыков.

# 7.21. Изменяющиеся стили ссылок-якорей

## Проблема

Вы хотите изменять стиль отображения элементов Web-страницы, когда пользователь щелкает ссылку кнопкой мыши.

# Решение

Прежде всего, создайте в документе разметку с помощью обычных ссылок-якорей. В данном решении ссылки-якори (называемые *идентификаторами фрагментов* (fragment identifier)) помещаются в изображение-карту:
```
<area shape="circle" coords="340,88,58" href="#trueman" />
<area shape="circle" coords="480,287,79" href="#katrina" />
</map>
<div class="bios">
<dl id="katrina">
 <dt>Katrina</dt>
 <dd>...</dd>
</dl>
<dl id="jessica">
 <dt>Jessica</dt>
 <dd>...</dd>
</dl>
<dl id="trueman">
 <dt>Trueman</dt>
 <dd>...</dd>
</dl>
<dl id="mark">
 <dt>Mark</dt>
 <dd>...</dd>
</d1>
</div><!-- end /#bios -->
```

Затем с помощью правил CSS задайте на Web-странице стили по умолчанию (рис. 7.35):

```
.bios dt {
 font-weight: bold;
}
.bios dd {
 margin: 0;
 padding: 0;
}
```

Далее воспользуйтесь псевдоклассом :target для определения внешнего вида элементов при щелчке кнопкой мыши по ссылке-якорю (рис. 7.36):

```
.bios dl:target {
background-color: #999999;
border: 1px solid black;
padding: lem;
font-weight: bold;
line-height: 1.5;
}
.bios dl:target dt {
font-style: italic;
color: white;
font-size: 1.5em;
background-color: #cccccc;
margin-right: 20px;
}
.bios dl:target dd {
margin-right: 20px;
background-color: #cccccc;
padding: 0 1em 1em 1em;
}
```





#### Рис. 7.35. Стандартная визуализация Web-страницы

Для того чтобы вернуть элементам псевдокласса :target стиль по умолчанию, когда пользователь щелкает кнопкой мыши другую ссылку-якорь, воспользуйтесь псевдоклассом :not (рис. 7.37):

```
.bios dl:not(:target) {
  border: none;
  padding: 0;
  font-size: .8em;
}
```



Рис. 7.37. Фрагмент Katrina с восстановленным стилем по умолчанию при активизации другой ссылки

## Обсуждение

Псевдоклассы :target и :not — часть спецификации CSS3 и поэтому не слишком хорошо известны большинству Web-дизайнеров. Тем не менее эти селекторы могут выполнить бо́льшую часть тяжелой работы.

#### Сворачиваемые меню только средствами CSS

Благодаря применению данных селекторов можно заменить основанное на языке JavaScript решение несколькими дополнительными правилами CSS. Во-первых, измените разметку, вставив ссылку-якорь:

```
<h5>
<a href="#menulink">Interesting Links </a>
</h5>
<a href="http://www.ora.com/">O'Reilly</a>
<a href="http://www.slashdot.org/">Slashdot</a>
<a href="http://www.apple.com/">Apple</a>
<a href="http://www.microsoft.com/">Microsoft</a>
<a href="http://www.microsoft.com/">Microsoft</a>
<a href="http://www.mozilla.org/">Mozilla</a>
```

Затем задайте следующие правила CSS:

```
/* визуализация по умолчанию */
ul#menulink {
    display: none;
  }
  /* когда ссылка активирована */
ul:target {
    display: block;
  }
  /* возврат к визуализации по умолчанию */
ul:not(:target) {
    display: none;
  }
```

#### Известные проблемы Web-обозревателей

В настоящий момент псевдоклассы :target и :not поддерживаются в Web-обозревателях Firefox, Safari, Chrome, Opera и Internet Explorer 7 для Windows.

#### См. также

Псевдокласс :target в спецификации CSS3 на странице http://www.w3.org/TR/css3-selectors/#target-pseudo.

# глава 8 Формы

# 8.0. Введение

Без HTML-форм вы не сможете войти в систему основанных на Web-технологии учетных записей электронной почты, заказать книги одним щелчком мыши или торговать акциями в интерактивном режиме. Несмотря на то, что формы способствуют повсеместному проникновению "Всемирной паутины", отображение их Web-обозревателями остается ужасным.

Стандартная визуализация интерактивных форм обычно включает в себя однострочные и многострочные поля ввода с эффектом объемности и скучно выглядящие кнопки. С таким внешним видом и впечатлением можно смириться в случае применения формы в маленькой внутренней сети или на небольшом Web-сайте, но не при разработке профессионального проекта.

Даже Google, прославлявший минимализм, для создания более реалистичных элементов управления формы занялся изменением своей хваленой поисковой формы, используя специальные свойства CSS, входящие в WebKit.

К счастью, с помощью нескольких правил CSS вы можете создавать формы, выделяющиеся из общей массы. Эта глава поможет освоить методы создания форм высокого качества.

Вы познакомитесь с установкой параметров пользовательских элементов ввода языка HTML, таких как кнопки, многострочные текстовые области и поля. В главу включен метод, который описывает создание кнопки однократного предъявления формы (submit-once-only), избавляющей посетителей сайта от ошибочной отправки нескольких процессов на сервер. В конце главы предлагается два примера проектов: простая регистрационная форма без таблиц и длинная регистрационная форма с таблицами.

#### Примечание

Приложение 4 может служить отличным дополнением к этой главе. Кроме того, см. на Webсайте http://webformelements.com наглядный справочник, подробно описывающий действие большинства визуальных свойств CSS, предназначенных для элементов форм, применяемых в 10 современных Web-обозревателях.

# 8.1. Изменение пробельной зоны вокруг формы

## Проблема

Вы хотите изменить объем свободного пространства вокруг формы.

#### Решение

Задайте нулевое значение поля и одновременно откорректируйте значения отступа (padding) у элементов формы (рис. 8.1):

```
form {
    margin: 0;
    padding: lem 0;
    border: lpx dotted red; /* задается для того, чтобы увидеть эффект от изменения
    oтступа */
}
```

00	CSS Cookbook	$\bigcirc$
Back Forward Reload	X     Image: Constraint of the constrain	GIQ
CSS Cookbook	😧 🚺 Christopher Schmitt : Design 🕲 🖉 Technology Books, Tech Con 🕲	+ =
e pluribus unum. Defacto lingo est igpay atinlay. Marquee selectus non provisio incongruous feline nolo contendre. Gratuitous octopus niacin, sodium glutimate.	necessi far uniform grammatica, pronunciation e plu sommun paroles. Please fill out our form below in order to contact me. Title	A V
non interruptus stadium. Sic tempus fugit esperanto hiccup estrogen. Glorious baklava ex librus hup hey ad infinitum. Non sequitur condominium	Name Email	
facile et geranium incognito. Epsum factorial non deposit	State/Province	
quid pro quo nic escorol. Marquee selectus non provisio incongruous feline nolo	Country	U.
contendre Olypian quarrels et gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum e pluribus unum.	Message	▲ ▼

Рис. 8.1. Отступ, заданный внутри рамки формы

## Обсуждение

Помещая форму в макет Web-страницы, необходимо откорректировать свободное пространство между формой и другими элементами макета. Обычно наиболее частая корректировка — изменение верхнего и нижнего отступов формы.

## См. также

В разд. 8.3 стилевое оформление элементов ввода.

# 8.2. Удаление пробельной зоны вокруг формы

## Проблема

Вы хотите удалить пробельную зону вокруг маленькой формы.

#### Решение

Задайте элемент form, отображаемый как внутристрочный или строчный, а не блочный элемент:

```
form {
  display: inline;
  padding: 0
}
```

## Обсуждение

По умолчанию элемент form — это блочный элемент, что означает инициирование перевода строки над и под ним. Для удаления этой пробельной зоны измените значение свойства по умолчанию на inline.

#### См. также

В разд. 1.5 обсуждение блочных и строчных элементов.

## 8.3. Задание стилей для элементов ввода

## Проблема

Вы хотите изменить цвета фона элементов input (перейти от рис. 8.2 к рис. 8.3).

#### Решение

Используйте селектор класса при разработке элементов ввода формы:

```
<h2>Simple Quiz</h2>
<form action="simplequiz.php" method="post">
Are you
  <input type="radio" value="male" name="sex"
class="radioinput" />
 Male or
  <input type="radio" value="female" name="sex"
class="radioinput" />
 Female?
What pizza toppings do you like? <input type="checkbox" name=""
value="1" class="checkbxinput"> Pepperoni <input type="checkbox"
name="" value="mushrooms" class="checkbxinput"> Mushrooms <input
type="checkbox" name="" value="pineapple" class="checkbxinput">
Pineapple
<label for="question1">Who is buried in Grant's tomb?</label>
<input type="text" name="question1" id="question1"
```

#### Формы

```
class="textinput"
value="Type answer here" /><br />
<label for="question2">In what country is the Great Wall of
China Located?</label>
    <input type="text" name="question2" id="question2"
class="textinput"
value="Type answer here" /><br />
    <label for="password">What is your password?</label>
    <input type="password" name="password" id="password"
class="pwordinput"
value="" /><br />
    <input name="reset" type="reset" id="reset" value="Reset" />
    <input type="submit" name="Submit" value="Submit"
class="buttonSubmit" />
    </form>
```

Затем для изменения отображения элементов ввода примените следующие правила CSS:

```
.textinput {
margin-bottom: 1.5em;
width: 50%;
color: #666;
background-color: #ccc;
.pwordinput {
color: white;
background-color: white;
}
.radioinput {
color: green;
background-color: #ccc;
}
.checkbxinput {
color: green;
background-color: green;
}
```

## Обсуждение

Во время написания книги Web-обозреватели Opera, IE, Safari, Chrome и Firefox не поддерживали свойство color для зависимых переключателей и флажков (см. http://tr.im/colorradiobutton и http://tr.im/colorcheckboxes).

IE и Opera поддерживают цвет фона для зависимых переключателей и флажков. Но IE заполняет цветом фона все окно, а Opera устанавливает цвет фона в пределах элементов управления формы (см. http://tr.im/bkgdcolorradiobuttons и http://tr.im/bkgdcolorcheckboxes).



Рис. 8.2. Форма без стилей

🖉 CSS Cookbook - Microsoft Internet Explorer	
File Edit View Favorites Tools Help	
Links @ oreilly.com @ christopher.org	
Simple Quiz	Green
Are you C Male or C Female?	Green
What pizza toppings do you like? 🔲 Pepperoni 🔲 Mushrooms 💻 Pineapple	
Who is buried in Grant's tomb?	
Type answer here	
In what country is the Great Wall of China Located?	
Type answer here	
What is your password?	
Reset Submit	
Done     Done     Docal intranet	

Рис. 8.3. Стили, примененные к полям ввода

#### Применение селекторов атрибута

Вместо применения селекторов класса, как показано в данном решении, можно использовать другой способ стилевого оформления элементов ввода разных типов с помощью селекторов атрибутов. При использовании селекторов атрибутов удалите атрибуты class из HTML-разметки и воспользуйтесь только следующими правилами CSS:

```
input[type="text"] {
  margin-bottom: 1.5em;
  width: 50%;
  color: #666;
  background-color: #ccc;
}
input[type="password"] {
  color: white;
  background-color: white;
}
```

Основной недостаток этого метода состоит в том, что он не действует в IE6 и более ранних версиях, поскольку данные Web-обозреватели вообще не поддерживают селекторы атрибута.

В настоящее время они поддерживаются в Firefox, Chrome, Safari и Opera. Если вы хотите обеспечить стилевое оформление в разных Web-обозревателях, вам следует для определения стилей разных элементов формы применять селекторы класса.

#### Примечание

Для того чтобы узнать, применяет ли Web-обозреватель стили к элементу ввода формы и как он это делает, посетите страницу http://www.flicr.com/photos/teleject/sets/ 72157615099024461.

#### См. также

Динамические псевдоклассы в спецификации CSS 2.1 на странице http://www.w3.org/ TR/CSS21/selector.html#x33; и селекторы атрибутов на http://www.w3.org/TR/ CSS21/selector.html#attribute-selectors.

# 8.4. Изменение стилей элементов формы при щелчке по ним кнопкой мыши

#### Проблема

Вы хотите изменить фоновый цвет элемента формы, когда пользователь щелкает по нему кнопкой мыши.

#### Решение

Для активизации изменения, в котором правила CSS применяются к элементу формы (рис. 8.4), используйте псевдокласс : focus:

```
.textinput:focus {
  background-color: yellow;
}
```

000	CSS Cookbook – Mozilla 🔘	
🛓 🟦 Home	🖹 Bookmarks 🛇 oreilly.com 🛇 christopher.org	
>		
Simple	Quiz	
Are you 🦿	Male or C Female?	
What pizza	toppings do you like? 🗆 Pepperoni 🗖 Mushrooms 🗂 Pineapple	
Who is bur	ied in Grant's tomb?	
This field i	s active!	— Highlighted yellow
In what co	untry is the Great Wall of China Located?	
Type answ	er here	
What is yo	ur password?	
Reset	Submit	
	□ @Z Done	

Рис. 8.4. Применение : focus для высвечивания поля ввода

## Обсуждение

Использование псевдокласса : focus — это легкий способ добавить в форму немного практичности или взаимодействия с пользователем. Подобное изменение стилей позволяет пользователю узнать, в каком поле формы он находится и над чем работает в данный момент, не сбивая его при этом с толку.

#### См. также

Разд. 2.6 с обсуждением других псевдоклассов.

# 8.5. Применение разных стилей к разным элементам ввода в одной форме

## Проблема

Вы хотите задать разное стилевое оформление для нескольких элементов input в одной и той же форме.

#### Решение

Для применения разных стилей воспользуйтесь двумя или несколькими разными селекторами класса. Сначала примените атрибуты class с разными значениями в разметке элементов input:

```
<lpre><label for="fmname">Name</label>
<input type="text" name="fmname" class="fmname" />
<label for="fmemail">Email</label>
<input type="text" name="fmemail" class="fmemail" />
```

Затем задайте стили для каждого атрибута class в элементах input:

```
.fmname {
  text-align: left;
}
.fmemail {
  text-align: center;
}
```

## Обсуждение

Метод использования селекторов класса для применения разных стилей к одинаковым элементам на одной Web-странице работает в большинстве Web-обозревателей.

#### Применение селекторов атрибута

Другой метод назначения разных стилей одинаковым элементам доступен в Webобозревателях, распознающих включенные в спецификацию CSS 3 селекторы атрибута, которые обсуждались в *разд 8.3*. Для данного решения код с использованием селекторов атрибута (после удаления атрибутов class и их соответствующих значений из HTML-разметки) выглядит следующим образом:

```
input[name="fmname"] {
  text-align: left;
}
input[name="fmemail"] {
  text-align: center;
}
```

#### См. также

Разд. 8.9 о стилевом оформлении кнопок в одной и той же форме.

# 8.6. Назначение стилей элементам *textarea*

#### Проблема

Вы хотите задать стили элементов textarea в Web-форме, включая изменение цвета, размера и начертания шрифта и других свойств элемента, как показано на рис. 8.5.

00	CSS Cookbook
Back Fo	Nward Reload Stop Home Print 1Password
CSS	Cookbook 😟 👔 Christopher Schm 😵 🊈 Technology Book 😵 🕇
Ord	er Pizza
Customer	Name
Size O S	Small O Medium O Large
Delivery	
	Pepperoni 🔺
	Sausage Green Peppers
	Pineapple
Coppings	Ham •
11 0	
PI	ease deliver to the second
do	or and knock *loudly* if it's
ra	ining.
Notes	
Reset For	m Order Now

Рис. 8.5. Элемент textarea со стилевым оформлением

#### Решение

Для связи стилей с элементом textarea воспользуйтесь селектором типа:

```
textarea {
  width: 300px;
  height: 100px;
  background-color: yellow;
  font-size: lem;
  font-weight: bold;
  font-family: Verdana, Arial, Helvetica, sans-serif;
  border: lpx solid black;
}
```

## Обсуждение

Связать стили элементов textarea, применяя селектор типа, очень просто:

```
textarea {
  background-color: blue;
}
```

Добавив псевдокласс : focus, вы сможете изменить стиль активного поля textarea:

```
textarea:focus {
  background-color: green;
}
```

Таким образом, когда пользователь заполняет форму, поле textarea, заполняемое в данный момент, изменит цвет.

#### Примечание

Для выяснения, поддерживает ли Web-обозреватель стили элемента textarea, и просмотра результата посетите страницу http://www.flickr.com/photos/teleject/sets/72157615099816279/.

#### См. также

Динамические псевдоклассы в спецификации CSS 2.1 на странице http://www.w3.org/ TR/CSS21/selector.html#x33 и селекторы атрибутов на странице http://www.w3.org/ TR/CSS21/selector.html#attribute-selectors.

# 8.7. Задание стилей элементов select и option

#### Проблема

Вы хотите изменить в форме отображение меню в виде списка (list menu), откорректировав цвет и шрифт, как показано на рис. 8.6.



#### Решение

Для того чтобы связать стили с элементами select, воспользуйтесь селектором типа:

```
select {
  color: white;
  background-color: blue;
  font-size: 0.9em;
}
option {
  padding: 4px;
}
```

## Обсуждение

В формах существует только один тип элемента select, в отличие от элементов ввода, поэтому связать стили с этим элементом просто и можно сделать это с помощью селектора типа. Задать стилевое оформление поля option также легко.

Для применения стилей к чередующимся полям option в списке элемента select сначала вставьте атрибут class в элемент option:

```
<select name="Topping_ID" size="6" multiple>
<option value="1">Pepperoni</option>
<option value="2" class="even">Sausage</option>
<option value="3">Green Peppers</option>
<option value="4" class="even">Pineapple</option>
<option value="5">Chicken</option>
<option value="6" class="even">Ham</option>
<option value="7">Olives</option>
<option value="7">Class="even">Ham</option>
<option value="8" class="even">Onions</option>
<option value="9">Red Peppers</option>
</select>
```

#### Примечание

Если вы используете язык XHTML, необходимо задать атрибут multiple, в котором название также служит его собственным значением:

<select name="Topping ID" size="6" multiple="multiple">

Затем задайте правила CSS для двух наборов полей option, чтобы элементы option со значением even (четный) (задаваемым селектором класса even) внешне отличались от других элементов. Например, у полей option с селектором even фоновый цвет — красный, в то время как у "обычных" полей option он синий (см. рис. 8.6):

```
select {
  font-size: 0.9em;
}
option {
  color: white;
  background-color: blue;
}
```

```
option.even {
  color: blue;
  background-color: red;
}
```

Вместо селектора класса вы можете использовать селектор :nth-child(odd) для стилевого оформления смежных элементов option. Однако во время написания книги некоторые из популярных Web-обозревателей не поддерживали этот тип селектора.

#### Примечание

Чтобы выяснить, применяет ли Web-обозреватель стили к элементу select, и увидеть, как он это делает, посетите страницы http://www.flickr.com/photos/teleject/sets/72157615222275442/ и http://www.flickr.com/photos/teleject/sets/72157615152109523/.

#### См. также

В разд. 8.3 сведения о том, как изменить цвет и размер текстового поля ввода.

# 8.8. Создание поля поиска в стиле Macintosh

## Проблема

Вы хотите создать стилевое оформление поля поиска (search field) для Webобозревателя Safari.

#### Решение

Воспользуйтесь расширениями языка HTML, которые действуют только в Webобозревателе Safari.

Поместите один элемент input в элемент div. Затем задайте значение search в атрибуте type (рис. 8.7):

```
<form method="get" action="/search.php">
<div>
<label for="q">Search</label>
<input type="search" placeholder="keywords"
autosave="com.domain.search" results="7" name="q" />
</div>
</form>
```

## Обсуждение

Разработчики Web-обозревателя Safari из компании Apple создали расширение для HTML-форм, позволяющее формировать более надежный пользовательский интерфейс. Любой Web-обозреватель, отличный от Safari, будет выводить на экран поле ввода как стандартное текстовое поле ввода формы, также подходящее и для использования в качестве типичного поля поиска.

	CSS Cookbook	uments/W C Qr Google	
CSS Cookbook	Christopher Schmitt : De	Technology Books, Tech	<b>+</b>
Search (Q+ keywords			
•			

Рис. 8.7. Поле поиска в Web-обозревателе Safari

#### Атрибуты поля поиска

Атрибут placeholder позволяет Web-разработчикам задавать текст, постоянно хранящийся в поле поиска. Текст выводится таким же образом, как и текст, заданный в атрибуте value текстового поля ввода, как показано на рис. 8.8:

```
<label for="fmwebsite">Web Site:</label>
<input type="text" name="fmwebsite" value="http://" />
```

Разница между атрибутом placeholder и стандартным атрибутом value состоит в том, что пользователи должны вручную удалять текст, помещенный в поле формы.

00	CSS Cookbook		
< > = 0 1P +	Bile:///Users/christopher/Documents/₩ C Q Google		$\square$
CSS Cookbook	Christopher Schmitt : De	Technology Books, Tech	] +
5 (166) - 69.			
Web Site: http://			

Рис. 8.8. Поле ввода в Web-обозревателе Safari

#### Предупреждение

Не рекомендуется использовать оба атрибута, value и placeholder, в одном и том же поле поиска. В Web-обозревателе Safari это приведет к тому, что текст атрибута value запишется поверх текста, заданного в атрибуте placeholder. И пользователям придется вручную удалять текст, выводимый атрибутом value, лишившись таким образом функциональных возможностей, предоставляемых атрибутом placeholder. Атрибут autosave — это маркер, позволяющий сохранять результаты предыдущих поисков на локальной машине пользователя. Пользователь сможет щелкнуть кнопкой мыши по пиктограмме лупы и увидеть предыдущие варианты поиска.

Атрибут results принимает числовое значение. Это значение определяет количество поисков, которые будут храниться на локальном компьютере пользователя (рис. 8.9).



Рис. 8.9. Сохраненные варианты выводятся под полем поиска

Сохраненные варианты поисков могут казаться перемещаемыми с одного сайта на другой. Например, если на одном сайте используется то же значение атрибута autosave, что и на другом сайте, одни и те же сохраненные варианты поисков будут отображаться в поле поиска на обоих сайтах. Этот прием может применяться на ряде разных доменов, принадлежащих общей сети. У пользователя появляется доступ к хронологии поиска, и следовательно, атрибут autosave улучшает условия работы пользователя.

#### Рекомендации

Поле поиска не требует наличия кнопки ввода, поэтому применяйте его в форме только с одним полем ввода. Формы с единственным элементом формы input будут воспринимать нажатие клавиши <Enter> или <Return> как отправку формы на сервер. Вставка дополнительных элементов input означает, что Web-обозревателю потребуется кнопка **Submit** (Отправить), которую следует нажать для обработки формы.

## См. также

В блоге Surfin's Safari о расширении, включающем поле поиска, на странице http://weblogs.mozillazine.org/hyatt/archives/2004\_07.html#005890. *Разд. 6.7* о способе вставки пиктограмм за счет создания отступов элементов с помощью свойства background-image.

# 8.9. Стили кнопок формы

## Проблема

Вы хотите задать для кнопок формы Submit (Отправить) и Reset (Очистить) цвет, отступы, рамки и визуальные эффекты при наведении указателя мыши на кнопку. В верхней части рис. 8.10 показана форма без применения стилей к кнопкам, а в нижней части рис. 8.10 — форма со стилизованными кнопками.



Рис. 8.10. Кнопки формы без применения стилей (вверху) и с применением стилей (внизу)

#### Решение

Сначала воспользуйтесь селектором class для проектирования кнопок:

```
<form action="simplequiz.php" method="post">
<label for="question">Who is president of the U.S.?
</label>
<input type="text" name="question" id="textfield"
value="Type answer here" /><br />
<input name="reset" type="reset" value="Reset"
class="buttonReset" />
<input type="submit" name="Submit" value="Submit"
class="buttonSubmit" />
</form>
```

}

Обсуждение

Затем для стилевого оформления кнопок воспользуйтесь правилами CSS:

```
.buttonReset {
color: #fcc;
background-color: #900;
font-size: 1.5em;
border: 1px solid #660;
padding: 4px;
background-image: url(title-glass.png);
background-repeat: repeat-x;
background-position: 50%;
text-shadow: 0 -1px 0 #666;
-moz-border-radius: 8px;
-webkit-border-radius: 8px;
border-top: 1px solid #900;
}
.buttonSubmit {
color: white;
background-color: #660;
font-size: 1.5em;
border: 1px solid #660;
padding: 4px;
background-image: url(title-glass.png);
```

background-repeat: repeat-x; background-position: 50%; -moz-border-radius: 8px; -webkit-border-radius: 8px; border-top: 1px solid #660;

Задать стилевое оформление кнопок можно также с помощью визуального эффекта при *наведении указателя мыши на кнопку* (rollover state). Для этого воспользуйтесь функцией на языке JavaScript:

```
<script language="JavaScript" type="text/javascript">
function classChange(styleChange,item) {
   item.className = styleChange;
}
</script>
```

Далее добавьте два дополнительных правила CSS, одно для задания визуального эффекта при наведении указателя мыши на кнопку **Reset**, а другое — на кнопку **Submit**:

```
.buttonResetRoll {
  color: white;
  background-color: #c00;
  font-size: 1.5em;
  border: 1px solid #660;
  padding: 4px;
}
```

```
.buttonSubmitRoll {
  color: white;
  background-color: #cc0;
  font-size: 1.5em;
  border: 1px solid #660;
  padding: 4px;
}
```

После того как функция описана и дополнительные правила CSS заданы, вставьте в разметку кнопок события, позволяющие переключаться между состояниями нажатия кнопок формы и их отпускания (рис. 8.11):

```
<form action="simplequiz.php" method="post">
<label for="question">Who is president of the U.S.?</label>
<input type="text" name="question" id="textfield"
value="Type answer here" /><br />
<input name="reset" type="reset" id="reset" value="Reset"
class="buttonReset"
onMouseOver="classChange('buttonResetRoll',this)"
onMouseOut="classChange('buttonReset',this)" />
<input type="submit" name="Submit" value="Submit"
class="buttonSubmit"
onMouseOver="classChange('buttonSubmitRoll',this)"
onMouseOut="classChange('buttonSubmitRoll',this)"
</form>
```



Рис. 8.11. Состояние при наведении на кнопку указателя мыши, созданное с помощью правил CSS и языка JavaScript

#### Примечание

Для пользователей IE6 следует применять селекторы классов для задания стилей кнопок.

Если вы используете селекторы атрибутов при написании правил CSS для кнопок формы, вам не потребуется дополнительная разметка HTML-элемента, как в случае применения селекторов классов. Например, синтаксическая запись селекторов атрибутов для кнопок с применением только правил CSS могла бы выглядеть следующим образом:

```
input[type="reset"] {
  color: #fcc;
  background-color: #900;
  font-size: 1.5em;
  border: 1px solid #660;
  padding: 4px;
  }
  input[type="submit"] {
    color: white;
    background-color: #660;
    font-size: 1.5em;
    border: 1px solid #660;
    padding: 4px;
  }
```

Определить горизонтальный размер кнопки вы также можете с помощью свойства width.

#### Примечание

Для выяснения, применяет ли Web-обозреватель стили к кнопке, и просмотра результата посетите страницу http://www.flickr.com/photos/teleject/sets/72157615221157426/.

#### См. также

В *разд. 8.10* советы по созданию изображения кнопки с помощью CSS; в спецификации CSS 2.1 селекторы атрибутов на странице http://www.w3.org/TR/CSS21 /selector.html#attribute-selectors.

## 8.10. Создание изображения кнопки Submit

#### Проблема

Вы хотите создать пользовательскую кнопку **Submit** (Отправить) с помощью файла изображения, например, такого, как изображение на рис. 8.12.

#### Решение

Для задания изображения используйте элемент input c атрибутом type:

```
<input type="image" name="submit" src="submit.gif" />
```

#### Обсуждение

Несмотря на то, что для вставки изображения, применяемого в качестве кнопки **Submit**, используется язык HTML, после того как изображение помещено на Webстраницу, его можно изменять с помощью свойств CSS, таких как border и margin.

Back Forward Rele	CSS Cookbook	
Epsum factorial non deposit quid pro quo hic escorol. Olypian quarrels et gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum e pluribus unum. Defacto lingo est igpay atiniay. Marquee selectus non provisio incongruous feline nolo contendre. Gratuitous octopus niacin, sodium glutimate.	Li Europan lingues es membres del sam familie. Lor separat existentie es un myth. Por scientie, musica, sport etc., li tot Europa usa li sam vocabularium. Note: Li lingues differe solmen in li grammatica, li pronunciation e li plu commun vocabules. Omnicos directe al desirabilitá de un nov lingua franca: on refusa continuar payar custosi traductores. It solmen va esser necessi far uniform grammatica, pronunciation e plu sommun paroles.	

Рис. 8.12. Визуальный эффект при наведении мыши, созданный с помощью CSS и JavaScript

#### См. также

Элемент input в спецификации HTML на странице http://www.w3.org/TR/html4/ interact/forms.html#h-17.4.

# 8.11. Установка кнопки Submit-Once-Only

#### Проблема

Вы хотите уберечь пользователей от нажатия кнопки **Submit** (Отправить) более одного раза.

#### Решение

Сначала создайте класс, чтобы помешать отображению кнопки:

```
.buttonSubmitHide {
  display: none;
}
```

Затем используйте следующую функцию на языке JavaScript для переключения стилей с помощью селекторов класса:

```
<script language="JavaScript" type="text/javascript">
function classChange(styleChange,item) {
   item.className = styleChange;
}
</script>
```

Теперь для удаления кнопки Submit из Web-документа инициируйте функцию с помощью события onsubmit:

### Обсуждение

В данном решении функция на JavaScript инициирует изменение, применяющее стили к элементу. Для выполнения функции вы должны использовать событие формы onsubmit, сохраняющее активность формы. Если бы функция инициировалась событием onclick для кнопки Submit (Отправить), некоторые Web-обозреватели выполнили бы только функцию, изменяющую класс. После этого, поскольку кнопка исчезла с экрана, пользователь не смог бы отправить форму.

#### См. также

Дополнительные примеры, сочетающие язык JavaScript и правила CSS, в книге Дэнни Гудмана (Danny Goodman) "JavaScript & DHTML Cookbook" (JavaScript & DHTML. Сборник рецептов) (издательство O'Reilly).

# 8.12. Создание кнопки *Submit* в виде HTML-текста

#### Проблема

Вы хотите, чтобы кнопка Submit (Отправить) выглядела как обычный HTML-текст.

#### Решение

Для того чтобы кнопка Submit выглядела как обычный HTML-текст, воспользуйтесь несколькими свойствами форматирования технологии CSS.

Сначала вставьте атрибуты class и value:

<input type="submit" name="submit" value="send &raquo;" class="submit" />

Затем примените свойства CSS, чтобы убрать рамку и фоновый цвет кнопки Submit (рис. 8.13):

```
.submit {
  border: none;
  background-color: #fff;
  padding: 0;
  margin: 0;
  width: 5em;
}
```

Далее добавьте псевдокласс :hover для того, чтобы создать стандартный визуальный эффект при наведении на кнопку указателя мыши, показанный на рис. 8.13:

```
.submit:hover {
 text-decoration: underline;
}
                                                       CSS Cookbook
                                                           -
                                                                     (P
             -
                                                                                                               Q
                                                                                      file:///l 🗘 🔻
            Rack
                   Forward
                                Reload
                                                 Home
                                                          Print
                                                                  1Password
                                          Ston
                                               Christopher Schmitt... 🕄
                   CSS Cookbook
                                      0
                                                                           A Technology Books,
                                                                                                      \odot
                                                                                                            +
                                                                                                                 -
           condominium facile et
                                                                                                                   4
                                      Message
                                                                                                                   ÷
           geranium incognito.
           Epsum factorial non
           deposit quid pro quo hic
           escorol. Marquee
           selectus non provisio
           incongruous feline nolo
                                       send »
           contendre Olypian
           quarrels et gorilla
                                      Note: Due to our busy schedules, we cannot answer every message. If
           congolium sic ad
                                      your message warrants a follow-up with one of our personnel, will be in
           nauseum, Souvlaki
                                      touch with you.
           ignitus carborundum e
           pluribus unum.
            All rights reserved except where noted. If you want something from My Amazing Web Site, you're going to
                                                        have to ask.
           CSS Cookbook
            -
                     .
                                 C
                                                  A
                                                          (P
                                                                                     file:///U 🟠
                                Reload
                                         Stop
                                                 Home
                                                                  1Password
           Back
                   Forward
                                                          Print
                                                                      0
                   CSS Cookbook
                                      0
                                               Christopher Schmitt...
                                                                                Technology Books,
                                                                                                      63
                                      VILLENDER
           condominium facile et
                                                                                                                   4
                                                                                                                   ¥
           geranium incognito.
           Epsum factorial non
           deposit quid pro quo hic
           escorol, Marquee
           selectus non provisio
                                       send »
           incongruous feline nolo
           contendre Olypian
                                     Note: Due to our busy schedules, we cannot answer every message. If
           quarrels et gorilla
                                     your message warrants a follow-up with one of our personnel, will be in
           congolium sic ad
                                     touch with you.
           nauseum. Souvlaki
           ignitus carborundum e
           pluribus unum.
            All rights reserved except where noted. If you want something from My Amazing Web Site, you're going to
                                                        have to ask.
```

Рис. 8.13. Кнопка Submit в виде обычного HTML-текста (send)

### Обсуждение

Кнопка Submit в виде обычного HTML-текста хороша для дизайнеров, чувствующих, что стандартно выглядящая кнопка отправки не соответствует их дизайну и при этом не желающих использовать изображение для формирования кнопки.

Возможны также ситуации, в которых включение в дизайн страницы кнопки Submit осложнит работу пользователей. Обнажение кнопки до голого текста может успокоить страхи пользователей, связанные с пересылкой информации по Интернету.

#### Поддержка Web-обозревателями

Это решение действует в Web-обозревателях, разрешающих изменять кнопки **Submit**. К современным обозревателям, действительно поддерживающим данное решение, относятся Chrome, Safari, Firefox, IE8 и Opera.

### См. также

*Разд. 8.13* с решением, заставляющим настоящий HTML-текст действовать как кнопка **Submit**.

# 8.13. Как заставить текстовую HTML-ссылку действовать как кнопка Submit

## Проблема

Вы хотите заставить текстовую HTML-ссылку отправлять форму.

## Решение

Для включения формы используйте язык JavaScript:

```
<form name="msgform" method="get" action="results.php">
<label for="fmmsg">Message</label>
<textarea name="fmmsg" accesskey="m" id="fmmsg" rows="5" cols="14"></textarea>
<a href="javascript:document.msgform.submit();">Submit</a>
</form>
```

## Обсуждение

В то время как в *разд. 8.12* показано, как изменить внешний облик кнопки **Submit** (Отправить), заставив ее выглядеть как HTML-текст, в данном разделе демонстрируется, как заставить текстовую ссылку играть роль кнопки **Submit**. Основной недостаток этого метода — необходимость применения JavaScript для выполнения поставленной задачи. Web-обозреватели без JavaScript или те, в которых он отключен, не смогут использовать форму.

## См. также

В разд 8.12 создание кнопки Submit в виде текстовой HTML-ссылки.

# 8.14. Разработка Web-формы без таблиц

## Проблема

Вы хотите включить поля формы и пояснительные надписи в строки, не применяя HTML-таблиц, формируя таким образом макет отображения формы только средствами CSS без дополнительной разметки.

## Решение

Сначала включите в разметку пояснительные надписи вместе с полями формы (рис. 8.14 (верх)):

```
<form action="login.php" method="post">
<label for="uname">Username</label>
<input type="text" name="uname" id="uname" value="" /><br />
<label for="pname">Password</label>
<input type="text" name="pname" id="pname" value="" /><br />
<label for="recall">Remember you?</label>
<input type="checkbox" name="recall" id="recall"
class="checkbox" /><br />
<input type="submit" name="Submit" value="Submit"
class="buttonSubmit" />
</form>
```

Затем задайте свойства display со значением block для элементов label, переместите эти элементы влево с помощью свойства float и выровняйте текст по правому краю (рис. 8.14 (низ)):

```
input {
 display: block;
 width: 175px;
 float: left;
 margin-bottom: 10px;
l
label {
 display: block;
 text-align: right;
 float: left;
 width: 75px;
 padding-right: 20px;
.checkbox {
 width: lem;
}
br {
 clear: left;
}
.buttonSubmit {
width: 75px;
 margin-left: 95px;
}
```

CSS Cookbook	$\bigcirc$
	9
Back Forward Reload Stop Home Print 1Password	
CSS Cookbook	
Username	
Password	
Kemember you?	
	0
Back Forward Reload Stop Home Print 1Password	Q
CSS Cookbook 😵 🎦 Christopher Schmitt 🥸 🚑 Technology Books, 😵 🕇	
Username	
Password	
Remember	
Submit	
<b>h</b>	

Рис. 8.14. Макет формы без применения стилей и с применением стилей

#### Обсуждение

Для элементов input и label задано свойство display: block, отображающее их как элементы блочного уровня. Это позволяет задать ширину области вывода текста в элементе label. Теперь элементы label не прижаты к элементам input, а расположены слева на некотором расстоянии. Поскольку у всех элементов label одна и та же ширина, они выглядят одинаково во всей форме.

Ter br создает разрыв строки между парами элементов label и input и очищает float от предыдущих элементов. Это препятствует включению в текущую строку следующей пары элементов (тех, что выводятся после поля input, соответствующего элементу label).

#### См. также

Элемент label в спецификации HTML 4.1 на странице http://www.w3.org/ TR/html401/interact/forms.html#edef-LABEL; свойство float в спецификации CSS 2.1 на http://www.w3.org/TR/CSS21/visuren.html#propdef-float и свойство clear на странице http://www.w3.org/TR/CSS21/visuren.html#propdef-clear.

# 8.15. Разработка формы из двух колонок без применения таблиц

#### Проблема

Вы хотите преобразовать форму, выводящуюся в одну колонку (рис. 8.15), в форму из двух колонок.

00	CSS Cookbook	$\bigcirc$
<ul> <li>. @ ×</li> </ul>	(file:///Users/christopher/Documents/Work/O'Ri 😭      (Google	2
Back Forward Reload Stop	Home Print 1Password	
Quid Pro Quo	Contact Form	4
Epsum factorial non deposit quid pro quo hic escorol. Olypian quarrels et gorilla corgoilum sic ad nauseum. Souvlaki ignitus carborundum e pluribus unum. Defacto lingo est igpay atinlay. Marquee selectus non provisio incongruous felime nolo contendre. Gratuitous octopus niacin, sodium glutimate. Ouote meon an estimate et non interruptus stadium. Sic tempus fugit esperanto hiccup estrogen. Glorious baklava est librus hup hey ad infinitum. Non sequitur condominium facibi et geranium incognito. Epsum factorial non deposit quid pro quo hic escorol. Marquee selectus non provisio incongruous felime nolo contendre Olypian quarrels et gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum e pluribus unum.	Li Europan lingues es membres del sam familie. Lor separat existentie es un myth. Por scientie, musica, sport etc., li tot Europa usa li sam vocabularium. Li lingues differe solmen in li grammatica, li pronunciation e i plu commun vocabules. Omnicos directe al desirabilità de un nov lingua franca: on refusa continuar payar custosis traductores. It solmen va esser necessi far uniform grammatica, pronunciation e plu sommun paroles. Piezse fill out our form below in order to contact me. Register Login	

#### Решение

Сначала разметьте части формы как две различные секции с помощью элементов div:

<form id="regform" name="regform" method="post" action="/regform.php"> <div id="register"> <h4>Register</h4> <label for="fmlogin">Login</label> <input type="text" name="fmlogin" id="fmlogin" /> <label for="fmemail">Email Address</label> <input type="text" name="fmemail" id="fmemail" /> <label for="fmemail2">Confirm Address</label> <input type="text" name="fmemail2" id="fmemail2" /> <label for="fmpswd">Password</label> <input type="password" name="fmpswd" id="fmpswd" /> <label for="fmpswd2">Confirm Password</label> <input type="password" name="fmpswd2" id="fmpswd2" /> </div> <div id="contactinfo"> <h4>Contact Information</h4> <label for="fmfname">First Name</label> <input type="text" name="fmfname" id="fmfname" /> <label for="fmlname">Last Name</label> <input type="text" name="fmlname" id="fmlname" /> <label for="fmaddy1">Address 1</label> <input type="text" name="fmaddy1" id="fmaddy1" /> <label for="fmaddy2">Address 2</label> <input type="text" name="fmaddy2" id="fmaddy2" /> <label for="fmcity">City</label> <input type="text" name="fmcity" id="fmcity" /> <label for="fmstate">State or Province</label> <input type="text" name="fmstate" id="fmstate" /> <label for="fmzip">Zip</label> <input type="text" name="fmzip" id="fmzip" size="5" /> <label for="fmcountry">Country</label> <input type="text" name="fmcountry" id="fmcountry" /> <input type="submit" name="submit" value="send" class="submit" /> </div> </form>

Затем в элементах input и label назначьте свойству display значение block:

```
label {
  margin-top: .33em;
  display: block;
}
input {
  display: block;
  width: 250px;
}
```

Создайте вторую колонку формы, задав для первого элемента div id="register" плавающую модель позиционирования с выравниванием по левому краю, как показано на рис. 8.16:

000		C 11 1
< ▶ . @ 🗵		Cookbook
Back Forward Reload Stop CSS Cookbook	Home Print 1Password	Designer, 😧 🛛 🛵 Technology Books, Tech Confer 😧 📗 🕇
Quid Pro Quo	Contact Form	
Epsum factorial non deposit quid pro quo hic escorol. Olypian quarrels et gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum e pluribus unum. Defacto lingo est igpay atinlay. Marquee selectus non provisio incongruous feline nolo contendre. Gratuitous octopus niacin, sodium glutimate. Quote meon an estimate et non interruptus stadium. Sic tempus fugit esperanto hiccup estrogen. Glorious baklava ex librus hup hey ad infinitum. Non sequitur condominium facile et geranium incognito. Epsum factorial non deposit quid pro quo hic escorol.	Li Europan lingues es membres c musica, sport etc., li tot Europa u pronunciation e li plu commun vo refusa continuar payar custosi tra pronunciation e plu sommun parc Please fill out our form below in o Register Login Email Address Confirm Address Password	lel sam familie. Lor separat existentie es un myth. Por scientie, sa li sam vocabularium. Li lingues differe solmen in li grammatica, li cabules. Ornnicos directe al desirabilità de un nov lingua franca: on ductores. It solmen va esser necessi far uniform grammatica, les. rder to contact me. Contact Information First Name Last Name Address 1 Address 2
Olypian quartels et gorilla congolium sic ad nauseum. Souvlaki ignitus carborundum e pluribus unum.	Confirm Password State or Province	City
	Zip Country	

Рис. 8.16. Элементы формы начинают формировать вторую колонку

Далее задайте достаточный отступ слева для второй колонки формы, чтобы создать некоторое свободное пространство между двумя колонками (рис. 8.17):

```
#register {
 float: left;
}
#contactinfo {
padding-left: 275px;
}
```

#### Обсуждение

Применение свойства float позволяет дизайнерам быстро создать форму из двух колонок. Главная проблема этого метода проявляется в том случае, когда правая колонка длиннее левой. Размещение элементов формы может сбить с толку пользователей. Задавая отступ с учетом ширины первой колонки, разработчики создают форму, которая выглядит цельной.

00	CSS C	ookbook
🔹 🕞 , 🕑 🗙	) 🏚 🖨 😰 🜈	file:///Users/christopher/Documents/Work/ ☆ 💎 🔇
Back Forward Reload Stop	Home Print 1Password	
CSS Cookbook	Christopher Schmitt : D	esigner, 🕲 🖉 Technology Books, Tech Confer 🕲 🕇
Quid Pro Quo	Contact Form	
Epsum factorial non deposit quid	Li Europan lingues es membres del	sam familie. Lor separat existentie es un myth. Por scientie,
pro quo hic escorol. Olypian	musica, sport etc., li tot Europa usa	li sam vocabularium. Li lingues differe solmen in li grammatica, li
juarreis et gorilia congolium sic ad	pronunciation e li plu commun voca	bules. Omnicos directe al desirabilitá de un nov lingua franca: on
arborundum e pluribus unum.	refusa continuar payar custosi tradu	ctores. It solmen va esser necessi far uniform grammatica,
Defacto lingo est igpay atinlay.	pronunciation e plu sommun parole	5
Marquee selectus non provisio		
ncongruous feline nolo contendre.	Please fill out our form below in ord	er to contact me.
Fratuitous octopus niacin, sodium	Register	Contact Information
iuumate.	Login	First Name
Quote meon an estimate et non		
terruptus stadium. Sic tempus	Email Address	Leat Name
lgit esperanto hiccup estrogen.	Email Address	Last Name
d infinitum Non seguitur		
ondominium facile et geranium	Confirm Address	Address 1
ncognito. Epsum factorial non		
eposit quid pro quo hic escorol.	Password	Address 2
Aarquee selectus non provisio		
livoian quarrels et gorilla	Confirm Password	City
ongolium sic ad nauseum.		
ouvlaki ignitus carborundum e		State or Province
luribus unum.		State of Province
		Zip
		Country
		send
		<b>k</b>
	It solmen va esser necessi far unifo	rm grammatica, pronunciation e plu sommun paroles.

Рис. 8.17. Форма выводится в две колонки

## См. также

В главе 11 дополнительные методы компоновки элементов на Web-странице.

# 8.16. Включение в форму информации для пользователей

## Проблема

Вы хотите показать пользователям, заполнение каких частей формы обязательно.

## Решение

Сначала поместите текст предупреждения следом за пояснительным текстом полей формы, которые должны быть заполнены (рис. 8.18).

Примените атрибут class со значением required в элементах с пояснительным текстом (label) и элементах формы, требующих обязательного заполнения, чтобы можно было корректно обработать форму.

```
<form id="msgform" name="msgform" method="post" action="/process.php">
<fieldset>
 <legend>Contact Information</legend>
 <label for="fmtitle" accesskey="i">T<span class="akey">i</span>tle</label>
 <select name="fmtitle" id="fmtitle">
  <option value="ms">Ms.</option>
  <option value="mrs">Mrs.</option>
  <option value="miss">Miss</option>
   <option value="mr">Mr.</option>
 </select>
 <label for="fmname" accesskey="n"><span class="akey">N</span>ame</label>
 <input type="text" name="fmname" id="fmname" />
 <label for="fmemail" accesskey="e" class="required">
<span class="akey">E</span>mail <img src="alert.gif" /> Required</label>
 <input type="text" name="fmemail" id="fmemail" class="required" />
</fieldset>
<fieldset>
 <legend>Your Message</legend>
 <label for="fmstate" accesskey="y">Subject</label>
 <input type="text" name="fmcountry" id="fmcountry" />
  <label for="fmmsg" class="required"><span class="akey">M</span>essage
<img src="alert.gif" /> Required</label>
  <textarea name="fmmsg" accesskey="m" id="fmmsg" rows="5" cols="14"
class="required"></textarea>
</fieldset>
<input type="submit" name="submit" value="send" class="submit" />
</form>
```

CSS Cookbook	SS Cookbook
Back Forward Reload Stop Home Print 1Passwo	CSS Cookbook 😳 🚺 Christopher Sch 😧 👰
Contact Information	Contact Information
Title Ms. 🗘	Title
Name	Name
Email Required	Email Required
Your Message	- Your Message
Message Required	Message Required
(send)	send

Рис. 8.18. Необходимый текст предупреждения слева со стилизованными элементами формы справа

Примените правила CSS для изменения цвета текста и рамки элементов формы (рис. 8.18 (справа)):

```
label {
margin-top: .33em;
display: block;
}
input {
display: block;
width: 250px;
}
textarea {
width: 250px;
height: 75px;
label.required {
color: #c00;
font-weight: bold;
}
textarea.required, input.required {
border: 1px solid red;
background-color: #eee;
}
```

# Обсуждение

Изменение элементов ввода и элементов label с помощью цвета и жирного начертания шрифта позволяет пользователям легко понять, какие части формы вызывают проблемы.

Вставка слова "required" (обязательное) и предупреждающей пиктограммы также помогает информировать пользователей о проблемах, которые могут возникнуть при отправке их формы на сервер. Если Web-обозреватель пользователя не поддерживает технологию CSS, текст и изображение будут единственными отличительными знаками, информирующими пользователей о том, что необходимо исправить для успешной отправки формы на сервер.

## См. также

Руководство о включении в форму информации для пользователя с помощью языка PHP на странице http://www/maketemplate.com/feedback/.

# 8.17. Стилевое оформление клавиш доступа в Web-формах

# Проблема

Вы хотите создать зрительный индикатор, показывающий, какие символы в форме являются клавишами доступа.

#### Решение

Используйте селектор потомка для выделения в теге label символов, представляющих клавиши доступа.

Прежде всего, создайте правило CSS с селектором, задающим подчеркивание текста в теге ем, входящем в форму:

```
form em {
  text-decoration: underline;
  font-style: normal;
}
```

Заключите в элемент ет букву в элементе label, представляющую клавишу доступа:

```
<form id="msgform" name="msgform" method="post" action="/">
<label for="fmtitle" accesskey="i">T<em>i</em>tle</label>
<select name="fmtitle" id="fmtitle">
 <option value="ms">Ms.</option>
 <option value="mrs">Mrs.</option>
 <option value="miss">Miss</option>
 <option value="mr">Mr.</option>
</select>
<label for="fmname" accesskey="n"><em>N</em>ame</label>
<input type="text" name="fmname" id="fmname" />
<label for="fmemail" accesskey="e"><em>E</em>mail</label>
<input type="text" name="fmemail" id="fmemail" />
<label for="fmstate" accesskey="a">St<em>a</em>te/Province</label>
<input type="text" name="fmstate" id="fmstate" />
<label for="fmcountry" accesskey="y">Countr<em>y</em></label>
<input type="text" name="fmcountry" id="fmcountry" />
<label for="fmmsg" accesskey="m"><em>M</em>essage</label>
<textarea name="fmmsg" id="fmmsg" rows="5" cols="14"></textarea>
<input type="submit" name="submit" value="send" class="submit" />
</form>
```

#### Обсуждение

Клавиша доступа позволяет людям с ограниченными физическими возможностями быстро переходить к разделам Web-страницы. Кроме того, клавиши доступа предоставляют возможность и пользователям без подобных ограничений сделать навигацию с помощью таких клавиш своим обычным приемом. С помощью подчеркнутых символов, соответствующих клавишам доступа, посетители могут перемещаться по полям формы без переключения на мышь или другое устройство-указатель.

Клавиши доступа поддерживаются в Web-обозревателях Safari, Chrome, IE, Firefox и Opera.

#### См. также

Дополнительную информацию о стилевом оформлении клавиш доступа на странице http://www.alistapart.com/articles/accesskeys/.

# 8.18. Группировка общих элементов формы

## Проблема

Вы хотите разделить большую форму на группы элементов меньшего объема, как по-казано на рис. 8.19.

000	CSS Cookbook	$\supset 0 0 0$	CSS Cookbook	0
Back Forward R	C X A D IP C C C C C C C C C C C C C C C C C C	Back Forward Rele	oad Stop Home Print 1Password	D 🏠 🛛 🚷
CSS Cookbook	😧 🚺 Christopher Sch 🥹 🖓 Technology Book 🔇	CSS Cookbook	🕲 🚺 Christopher Sch 🕲 🖉 Techn	ology Book 😳
niacin, sodium glutimate. Quote meon an estimate et non interruptus stadium. Sic tempus fugit esperanto hiccup estrogen. Glorious baklava ex librus hup hey ad infinitum. Non sequitur condominium facile et geranium factorial non deposit quid pro quo hic escorol. Marquee selectus non provisio incongruous fellne nolo contendre Olypian quarrels et gorilla congolium sic ad nauseum. Souvlaki Ignitus carborundum e pluribus unum.	Christopher Sch  Contact Information Title  Ms.  Contact Information Title  Email Email  Your Message Subject Message	glutimate. Glutimate. Quote meon an estimate et non interruptus stadium. Sic tempus fugit esperanto hiccup estrogen. Glorious baklava ex librus hup hey ad infinitum. Non sequitur condominium faccie et geraniumi incognito. Epsum factorial non deposit quid pro quo hic escorot. Marquee selectus non provisio incongnuos feline	Contact INFORMATION Title Ms.  Name Email YOUR MESSAGE Subject	iology Book
	Note: Due to our busy schedules, we cannot answer every message. If your message warrants is following with one	incongruous tenine nolo contendre Otypian quarrels et gorilla congolium sic ad nauseum. Souviaki ignitus carborundum e pluribus unum.	Message	

Рис. 8.19. Измененная группировка и пояснительные надписи

## Решение

Воспользуйтесь HTML-элементом fieldset для разделения формы на несколько секций:

```
<form id="msgform" name="msgform" method="post" action="/">
<fieldset>
 <legend>Contact Information</legend>
 <label for="fmtitle">Title</label>
 <select name="fmtitle" id="fmtitle">
  <option value="ms">Ms.</option>
  <option value="mrs">Mrs.</option>
  <option value="miss">Miss</option>
   <option value="mr">Mr.</option>
 </select>
 <label for="fmname">Name</label>
 <input type="text" name="fmname" id="fmname" />
 <label for="fmemail">Email</label>
  <input type="text" name="fmemail" id="fmemail" />
</fieldset>
<fieldset>
 <legend>Your Message</legend>
```
```
<label for="fmstate">Subject</label>
 <input type="text" name="fmcountry" id="fmcountry" />
 <label for="fmmsg">Message</label>
 <textarea name="fmmsg" accesskey="m" id="fmmsg" rows="5"
cols="14"></textarea>
 </fieldset>
<input type="submit" name="submit" value="send" class="submit" />
```

```
</form>
```

#### Обсуждение

HTML-элементы fieldset и legend позволяют легко группировать типовые элементы.

Вы также можете применять к элементам fieldset и legend правила CSS для изменения внешнего вида формы:

```
fieldset {
margin-bottom: lem;
border: 1px solid #888;
border-right: 1px solid #666;
border-bottom: 1px solid #666;
}
legend {
font-weight: bold;
border: 1px solid #888;
border-right: 1px solid #666;
border-bottom: 1px solid #666;
padding: .5em;
background-color: #666;
background-image: url(title-glass.png);
background-repeat: repeat-x;
background-position: 50% 50%;
color: #fff;
text-shadow: 0 -1px 0 #333;
letter-spacing: .lem;
text-transform: uppercase;
}
```

#### См. также

Элементы fieldset спецификации HTML 4.01 И legend В на странице http://www.w3.org/TR/html4/interact/forms.html#h-17.10.

#### 8.19. Ввод данных в форму в виде электронной таблицы

#### Проблема

Вы хотите создать конфигурацию формы, напоминающую электронную таблицу (рис. 8.20).

iscal Year	Worksite Presentations	Passing Grades	Number of Presentators	
1999	24	20	3	
2000	23	19	3	
2001	24	34	4	
2002				
2003				
2004				
Save				

Рис. 8.20. Выделенная строка таблицы

#### Решение

Сначала поместите элементы input в HTML-таблицу:

```
<form action="/process.php" method="get" name="copresentations">
<caption>
 Summary of Financial Data
 </caption>
 Fiscal Year 
 Worksite<br />
  Presentations 
 Passing Grades 
 Number of Presenters 
 1999
 input type="text" name="wkpst1999" />/td>
 input type="text" name="pass1999" />/td>
 input type="text" name="numpst1999" />
```

```
2000
  input type="text" name="wkpst2000" />
  input type="text" name="pass2000" />/td>
  input type="text" name="numpst2000" />
 \langle t.r \rangle
  2001
  input type="text" name="wkpst2001" />/td>
  input type="text" name="pass2001" />/td>
  input type="text" name="numpst2001" />
 \langle t.r \rangle
  2002
  input type="text" name="wkpst2002" />/td>
  input type="text" name="pass2002" />/td>
  input type="text" name="numpst2002" />/td>
 2003
  input type="text" name="wkpst2003" />/td>
  input type="text" name="pass2003" />
  input type="text" name="numpst2003" />//td>
 2004
  input type="text" name="wkpst2004" />
  input type="text" name="pass2004" />
  input type="text" name="numpst2004" />/td>
 <input type="submit" class="save" value="Save" />
</form>
```

Вставьте тонкую рамку вокруг таблицы и задайте для вывода рамки на экран значение collapse (сливающаяся):

```
table {
  border-collapse: collapse;
  border: 1px solid black;
}
```

Задайте определенную ширину ячейки таблицы и вывод тонкой рамки вокруг нее:

```
th {
  border: 1px solid black;
  width: 6em;
}
td {
  width:6em;
  border: 1px solid black;
}
```

Удалите поля и отступы у ячеек таблицы:

```
th {
  border: 1px solid black;
  width: 6em;
}
td {
  width:6em;
  border: 1px solid black;
  padding: 0;
  margin: 0;
}
```

Задайте ширину элементов input равной ширине ячеек таблицы, одновременно удалив все рамки, которыми Web-обозреватели автоматически снабжают элементы форм.

```
input {
  width: 100%;
  border: none;
  margin: 0;
}
```

При установке ширины для элементов input кнопка **Submit** тоже растянется до максимальной ширины ее родительского элемента, поэтому она будет выглядеть слишком большой. Для управления размером кнопки **Submit** напишите отдельное правило CSS:

```
.save {
  margin-top: lem;
  width: 5em;
}
```

Для завершения показанного визуального представления электронной таблицы задайте выравнивание вводимого текста по правому краю:

```
input {
  width: 100%;
  border: none;
  margin: 0;
  text-align: right;
}
```

#### Обсуждение

Электронные таблицы помогают пользователям отслеживать большой объем числовых и финансовых данных. Стандартный макет контактной формы или формы для электронной коммерции может оказаться неудобным при необходимости вводить большое количество чисел. Имитируя вид электронной таблицы, вы предоставляете пользователю возможность быстро ввести данные.

Применяя псевдокласс :hover, можно выделить цветом строку и ячейку таблицы, в которых пользователь работает в данный момент:

```
tr:hover {
  background-color: #ffc;
}
tr:hover input {
  background-color: #ffc;
}
input:focus {
  background-color: #ffc;
}
```

#### См. также

Разд. 8.3 о стилевом оформлении элементов ввода.

# 8.20. Пример дизайна: простая регистрационная форма

Регистрационные формы встречаются везде в Web-пространстве. Например, вам нужно зарегистрироваться и ввести пароль для проверки вашей электронной почты, заказа книг на сайте **Amazon.com** и даже оплаты по Интернету штрафного талона за нарушение правил парковки.

Пользователю видны только несколько компонентов регистрационной формы: кнопка **Submit** для отправки данных из полей ввода, а также пояснительный текст и сами поля для ввода имени и пароля. Далее приведена разметка формы, для которой будет создаваться стилевое оформление; на рис. 8.21 показаны поля ввода без применения стилей:

```
<form action="login.php" method="post">
<label for="uname">Username</label>
<input type="text" name="uname" id="uname" value="" /><br />
<label for="pword">Password</label>
<input type="text" name="pword" id="pword" value="" /> <br />
<input type="submit" name="Submit" value="Submit" />
</form>
```



Рис. 8.21. Форма регистрации входа без стилевого оформления

Сначала добавьте символ, следующий за текстом в элементе label. Воспользуйтесь способностью псевдоэлемента :after автоматически генерировать символ:

```
label:after {
  content: ": ";
}
```

Далее, для того чтобы пояснительный текст отличался от полей ввода формы, измените цвет фона элементов label и начертание шрифта. С помощью правил CSS задайте для этих элементов серый цвет фона и белый шрифт с жирным начертанием (рис. 8.22):

```
label {
  background-color: gray;
  color: #fff;
  font-weight: bold;
}
```



Рис. 8.22. Стили, задающие цвета элементов label

Back Forward	CSS Cookbook	assword
USERNAME: PASSWORD: Submit		

Рис. 8.23. Текст, выведенный заглавными буквами, и другие стилевые установки

Теперь задайте отступ вокруг текста и измените буквы на заглавные. Также добавьте фоновое изображение с тенью, отбрасываемой текстом, для создания впечатления глубины (рис. 8.23):

```
label {
  background-color: gray;
  color: #fff;
  font-weight: bold;
  padding: 4px;
  text-transform: uppercase;
  background-image: url(title-glass.png);
  background-repeat: repeat-x;
  background-position: 50%;
  text-shadow: 0 -1px 0 #000;
}
```

Как видите, стиль пояснительного текста надо смягчить, поскольку он отвлекает внимание от полей ввода. Для ослабления визуального эффекта сделайте меньше размер шрифта, оставив жирное начертание. Далее для скругления краев в Web-обозревателях Firefox и Safari примените свойства border-radius. Кроме того, для пояснительного текста задайте гарнитуру Verdana, которая остается четкой даже при малых размерах шрифта (рис. 8.24):

```
label {
background-color: gray;
color: #fff;
font-weight: bold;
padding: 4px;
background-image: url(title-glass.png);
background-repeat: repeat-x;
background-position: 50%;
text-shadow: 0 -1px 0 #000;
-moz-border-radius-topleft: 5px;
-webkit-border-top-left-radius: 5px;
-moz-border-radius-topright: 5px;
-webkit-border-top-right-radius: 5px;
text-transform: uppercase;
font-family: Verdana, Arial, Helvetica, sans-serif;
font-size: xx-small;
}
```

00	CSS Cookbook	$\bigcirc$
Back Forward Relo	ad Stop Home Print 1P.	ⓐassword □ ☆ (
CSS Cookbook	Christopher Sch 🕄	👸 Technology Book 🕄
USERNAME: PASSWORD: Submit		

Теперь пришло время задать стили элементов ввода. Поскольку у формы два типа полей ввода, разделите их, поместив атрибут class в кнопку **Submit** (Отправить). Этот прием позволит задать разные стили для полей ввода и кнопки **Submit**. Если не сделать этого, стили, предназначенные для полей ввода, будут применяться и к кнопке **Submit**. С помощью селектора класса вы сможете переопределить или изменить свойства, назначенные для одного элемента так, что они не будут применяться ко всем элементам:

```
<input type="submit" name="Submit" value="Submit" class="buttonSubmit" />
```

Для того чтобы добавить немного свободного пространства вокруг элементов формы, задайте отображение полей ввода как блочных элементов и укажите величину поля снизу (рис. 8.25):

```
input {
  display: block;
  margin-bottom: 1.25em;
}
```



Рис. 8.25. Элементы input сместились под элементы label

Далее увеличьте ширину полей ввода до 150 пикселов и поместите рамку шириной 1 пиксел вокруг поля так, чтобы исчез стандартный визуальный эффект снятия фаски (bevel), создаваемый в большинстве Web-обозревателей. Придайте легкую глубину странице, добавив двухпиксельную границу справа и снизу от поля ввода (рис. 8.26):

```
input {
  display: block;
  margin-bottom: 1.25em;
  width: 150px;
  border: solid black;
  border-width: 1px 2px 2px 1px;
}
```

Далее задайте стили градиента только для полей ввода. Для этого используйте селекторы атрибута и свойства из спецификации CSS3 (рис. 8.27):

```
input[type="text"] {
  background-image: -moz-linear-gradient(left top, left bottom, from(#999),
  to(#fff), color-stop(0.2, #fff));
  background-image: -webkit-gradient(linear,left top, left bottom,
  from(#999), to(#fff), color-stop(0.2, #fff));
}
```

Back	Forward	C (	CSS Cookl	book	(Password		□ ☆▼) (
	CSS Cookboo	ok 🛛 🚺	Christophe	er Sch	🕲 🦾 Te	chnology	Book 🕲
USER	NAME:						
-		2					
PASS	WORD:						
	Submit						
							//

Рис. 8.26. Измененные поля ввода

00	CSS Cookbook	$\Box$
Back Forward Reload	Stop Home Print 1Pa	assword
CSS Cookbook	Christopher Sch 🕄	👸 Technology Book 🕄
USERNAME:		
PASSWORD:		
Submit		

Рис. 8.27. Небольшие градиенты в фоновом изображении текстовых полей

Закончив с основными полями ввода, можно применить стили к кнопке **Submit**. Поскольку вы не хотите, чтобы кнопка **Submit** выглядела как обычное текстовое поле ввода, используйте селектор класса.

Начните с изменения размера и позиции кнопки **Submit**. Прежде всего, уменьшите ширину кнопки до 75 пикселов (что равно половине ширины полей ввода). Затем сместите кнопку вправо, задав левое поле равным 75 пикселам (рис. 8.28): }

buttonSubm	it {				
width: 75pz	x;				
margin-left	t: 75px;				
	00	(	SS Cookbook		$\bigcirc$
		. C 🗵		(P)	
	CSS Cookbo	Reload Stop	Home Print	Password	chnology Book
				a ling is	
	USERNAME:				
	2				
	PASSWORD:				
	SI	hmit			
		billie			
					11.

Рис. 8.28. Отформатированная кнопка Submit

00	CSS Cookbook	$\Box$
Back Forward Reload	Stop Home Print	IPassword
CSS Cookbook	Christopher Sch C	Can Technology Book 🕄
USERNAME:		
PASSWORD:		
SUBMIT		2

Рис. 8.29. Зеленая кнопка Submit с дополнительным стилевым оформлением

Далее измените цвет кнопки **Submit** на зеленый с зеленой рамкой и преобразуйте буквы текста в заглавные с помощью свойства text-transform. Кроме того, скруглите нижние углы и добавьте градиент и тень от текста, чтобы добиться соответствия стилю пояснительных надписей (рис. 8.29):

```
.buttonSubmit {
  width: 75px;
  margin-left: 75px;
  color: green;
  text-transform: uppercase;
  border: 1px solid green;
```

```
-moz-border-radius-bottomleft: 5px;
-webkit-border-bottom-radius: 5px;
-moz-border-radius-bottomright: 5px;
-webkit-border-bottom-right-radius: 5px;
text-shadow: 0 -1px 0 #000;
background-image: -moz-linear-gradient(left top, left bottom, from(#ccc),
to(#999), color-stop(0.2, #999));
background-image: -webkit-gradient(linear,left top, left bottom, from(#ccc),
to(#999), color-stop(0.2, #999));
}
```

Добавьте последний штрих, запретите вывод элемента br, поскольку он создает дополнительную пробельную зону в форме. Результат показан на рис. 8.30.

```
br {
  display: none;
}
```

Back Forward Reload S	Stop Home Print 1P	assword
	entistopilei seitti 🦉	Con recimology booking @
PASSWORD:		
SUBMIT		

Рис. 8.30. Окончательный вид формы регистрации входа со стилевым оформлением

#### 8.21. Пример дизайна: бланк регистрации

В некоторых формах может понадобиться размещение элементов формы в таблице из двух столбцов с пояснительным текстом в одном столбце и полями ввода в другом. Код такой формы представлен в листинге 8.1. На рис. 8.31 показана форма с таблицами без применения стилей.

```
Листинг 8.1. Стилизованная длинная регистрационная форма
<form action="registration.cfm" method="post">
Account Information
```

```
Login Name*
   <input name="uname" type="text" size="12"
maxlength="12" />
  Password*
   <input name="pword" type="text" size="12"
maxlength="12" />
  Confirm Password* 
   <input name="pword2" type="text" size="12"
maxlength="12" />
  Email Address*
   <input name="email" type="text" />
  Confirm Email*
   <input type="text" name="email2" />
  Contact Information
  First Name* 
   <input name="fname" type="text" size="11" />
  Last Name* 
   <input name="lname" type="text" size="11" />
  Address 1*
   <input name="address1" type="text" size="11" />
  <t.r>
   Address 2 
   <input type="text" name="address2" />
  City* 
   <input type="text" name="city" />
  State or Province*
   <select name="state">
     <option selected="selected"</pre>
```

```
disabled="disabled">Select...</option>
       <option value="alabama">Alabama</option>
      </select>
   Zip*
    <input name="zipcode" type="text" id="zipcode"
size="5" maxlength="5" />
   Country*
    <input type="text" name="country" />
   Gender*
     <input type="radio" name="sex" value="female" />
      Female
      <input type="radio" name="sex" value="male" />
      Male 
   Misc. Information
   <t.r>
     Annual Household Income 
    \langle t.d \rangle
     <select name="income" size="1" >
       <option selected="selected" disabled="disabled">
Select ... </ option>
       <option value="notsay">I'd rather not say</option>
      </select> 
   <t.r>
    Interests
    <input name="interests" type="checkbox"
value="shopping-fashion" />
      Shopping/fashion
      <input name="interests" type="checkbox"
value="sports" />
      Sports
      <input name="interests" type="checkbox"
value="travel" />
      Travel
   </t.r>
   Eye Color
    <input name="eye" type="checkbox" value="red" />
      Red
      <input name="eye" type="checkbox" value="green" />
      Green
      <input name="eye" type="checkbox" value="brown" />
```

```
Brown

<input name="eye" type="checkbox" value="blue" />

Blue Gold

<input type="submit" name="Submit" value="Submit"

id="buttonSubmit" />

<input type="reset" name="Submit2" value="Reset"

id="buttonReset" />

</form>
```

00	CSS Cookbook
<	
Back Forward Reload	Stop Home Print 1Password
CSS Cookbook	3 📔 Christopher Schmitt : D 😵 🖉 Technology Books, Tec 😵 📘 🕂 💻
Ac	count Information
Login Name*	
Password*	
Confirm Password*	
Email Address*	
Confirm Email*	
C	ontact Information
First Name*	
Last Name*	
Address 1*	
Address 2	
City*	
State or Province*	Select 🗘
Zip*	
Country*	
Gender*	○ Female ○ Male
Ν	Aisc. Information
Annual Household Incom	e Select 🗘
Interests	📄 Shopping/fashion 📄 Sports 📄 Travel
Eye Color	📄 Red 📄 Green 📄 Brown 📄 Blue Gold
(Submit) (Reset)	

Рис. 8.31. Форма и таблица без применения стилей

Первый элемент, подлежащий стилевому оформлению, — элемент table. Задайте способ отображения рамки, ее цвет и вид (рис. 8.32):

```
table {
  border-collapse: collapse;
  color: black;
  border: 1px solid black;
}
```

Далее принимайтесь за ячейки заголовка таблицы, расположенные в левом столбце (рис. 8.33). Для них задайте ширину 200 пикселов, текст внутри ячеек выровняйте по правому краю, выберите шрифт Verdana размером 0.7 ет.

00	CSS Cookbook
Back Forward Reload	Stop Home Print 1Password
CSS Cookbook	🛿 📔 Christopher Schmitt : D 😵 🖉  Technology Books, Tec 😒 🕇 🕇
A	ccount Information
Login Name*	
Password*	
Confirm Password*	
Email Address*	
Confirm Email*	
C	ontact Information
First Name*	
Last Name*	
Address 1*	
Address 2	
City*	
State or Province*	Select 🗘
Zip*	
Country*	
Gender*	🔘 Female 🔘 Male
1	Misc. Information
Annual Household Incom	e Select 🗘
Interests	Shopping/fashion Sports Travel
Eye Color	🔲 Red 📄 Green 📄 Brown 📄 Blue Gold

Рис. 8.32. Рамка вокруг таблицы

00	CSS Cookbook
Back Forward Reload Stop	Home Print 1Password
CSS Cookbook	📔 Christopher Schmitt : D 😵 🛛 👰 Technology Books, Tec 😵 📗 🕇 💻
	Account Information
Login Name*	
Password*	
Confirm Password*	
Email Address*	
Confirm Email*	
	Contact Information
First Name*	
Last Name*	
Address 1*	
Address 2	
City*	
State or Province*	Select
Zip*	
Country*	
Gender*	🔘 Female 🔘 Male
	Misc. Information
Annual Household Income	Select 🗘
Interests	🔄 Shopping/fashion 📄 Sports 📄 Travel
Eye Color	📄 Red 📄 Green 📄 Brown 📄 Blue Gold

Рис. 8.33. Отформатированные ячейки заголовка таблицы

```
th {
  width: 200px;
  text-align: right;
  vertical-align: top;
  border-top: 1px solid black;
  font-family: Verdana;
  font-size: 0.7em;
}
```

Back Forward Reload Stop Home Print Password   CSS Cookbook Christopher Schmitt : Account Information     Login Name*   Password*   Password*   Confirm Password*   Email Address*   Confirm Email*   Confirm Email*   Address 1*   Address 2   City*   State or Province*   Select \$   Country*   Gender*   Female   Misc. Information   Annual Household Income   Select \$   Interests   Shopping/fashion   Sports   Travel	● ○ ○ . @ ×	CSS Cookbook	☐ file:///Users/cl☆▼ (▲ 오
Account Information         Login Name*         Password*         Confirm Password*         Email Address*         Confirm Email*         Address 1*         Address 1*         City*         State or Province*         Select \$         State or Province*         Select \$         Gender*       Female Male         Misc. Information         Annual Household Income       Select \$         Interests       Shopping/fashion Sports Travel         Eye Color       Red Green Brown Blue Gold	Back Forward Reload Stop	Home Print 1Password Christopher Schmitt :	Technology Books, Te 🔇 🕇 🖣
Login Name*   Password*   Confirm Password*   Email Address*   Confirm Email*   Confirm Email*   Confirm Email*   Contact Information   First Name*   Last Name*   Address 1*   Address 2   City*   State or Province*   Select *   Country*   Gender*   Female   Misc. Information   Annual Household Income   Select *   Interests   Shopping/fashion   Sports   Travel		Acco	unt Information
Password*   Confirm Password*   Email Address*   Confirm Email*   Confirm Email*   Confirm Email*   Contact Information   First Name*   Last Name*   Address 1*   Address 2   City*   State or Province*   Select	Login Name*		
Confirm Password*   Email Address*   Confirm Email*   Address 1*   Address 2   City*   State or Province*   Select *   Country*   Country*   Gender*   Female   Misc. Information   Annual Household Income   Select *   Interests   Shopping/fashion   Sports   Travel   Eye Color	Password*		
Email Address*   Confirm Email*   Contact Information   First Name*   Last Name*   Last Name*   Address 1*   Address 2   City*   State or Province*   Select**   Country*   Gender*   Female   Misc. Information   Annual Household Income   Select**   Interests   Shopping/fashion   Sports   Travel   Eye Color	Confirm Password*		
Confirm Email*   Contact Information   First Name*   Last Name*   Address 1*   Address 2   City*   State or Province*   Select*   Country*   Country*   Gender*   Female   Misc. Information   Annual Household Income   Select*   Interests   Shopping/fashion   Sports   Travel   Eye Color	Email Address*		
Contact Information   First Name*   Last Name*   Address 1*   Address 2   City*   State or Province*   Select*   State or Province*   Select*   Country*   Gender*   Female © Male   Misc. Information   Annual Household Income   Select*   Interests   Shopping/fashion   Sports   Travel   Eye Color	Confirm Email*		
First Name*   Last Name*   Address 1*   Address 2   City*   State or Province*   Select;   Country*   Country*   Gender*   Female   Misc. Information   Annual Household Income   Select;   Interests   Shopping/fashion   Sports   Travel   Eye Color		Cont	act Information
Last Name*   Address 1*   Address 2   City*   State or Province*   Select*   Country*   Country*   Gender*   Female () Male   Misc. Information  Annual Household Income Select*  Interests Shopping/fashion () Sports () Travel Eye Color Red () Green () Brown () Blue Gold	First Name*		
Address 1*   Address 2   City*   State or Province*   Select*   Zip*   Country*   Gender*   Female   Misc. Information   Annual Household Income   Select*   Interests   Shopping/fashion   Sports   Travel   Eye Color	Last Name*		
Address 2   City*   State or Province*   Select*   Zip*   Country*   Gender*   Female () Male   Misc. Information  Annual Household Income Select*  Interests Shopping/fashion () Sports () Travel Eye Color Red () Green () Brown () Blue Gold	Address 1*		
City*         State or Province*         Select**         Zip*         Country*         Gender*       Female         Misc. Information         Annual Household Income       Select**         Interests       Shopping/fashion         Eye Color       Red	Address 2		
State or Province*       Select *         Zip*	City*		
Zip*         Country*         Gender*       Female         Misc. Information         Annual Household Income         Select         Interests         Shopping/fashion         Sports         Travel         Eye Color         Red         Green         Brown         Blue Gold	State or Province*	Select 🗘	
Country*         Gender*       Female         Misc. Information         Annual Household Income       Select         Interests       Shopping/fashion         Eye Color       Red         Green       Brown         Blue Gold	Zip*		
Gender*       Female       Misc. Information         Misc. Information       Select       Information         Interests       Shopping/fashion       Sports       Travel         Eye Color       Red       Green       Brown       Blue Gold	Country*	O Famala O Mala	
Annual Household Income Select	Gender*	O Female O Male	ing Treformation
Interests     Shopping/fashion     Sports     Travel       Eye Color     Red     Green     Brown     Blue Gold	Annual Household Income		
Eye Color Red Green Brown Blue Gold	Interests	Shopping/fashion $\square$ S	ports 🖂 Travel
	Eye Color	Red Green Brow	wn 📃 Blue Gold
Submit) Reset	Submit Reset		

Рис. 8.34. Ячейки заголовка таблицы с заданным отступом

Откорректируйте отступы в ячейках заголовка (рис. 8.34):

```
th {
  width: 200px;
  text-align: right;
  vertical-align: top;
```

```
border-top: 1px solid black;
font-family: Verdana;
font-size: 0.7em;
padding-right: 12px;
padding-top: 0.75em;
padding-bottom: 0.75em;
```

00	CSS Cookbook	$\bigcirc$
		sers/cl 😭 🔻 🔇
CSS Cookbook	Christopher Schmitt : 😵 🔗 Technolog	y Books, Te 🛞 🕂 🛒
	Account Information	
Login Name*		
Password*		
Confirm Password*		
Email Address*		
Confirm Email*		
	Contact Information	
First Name*		
Last Name*		
Address 1*		
Address 2		
City*		
State or Province*	Select 🛟	
Zip*		
Country*		
Gender*	🔵 Female 🌑 Male	
	Misc. Information	
Annual Household Income	Select	
Interests	Shopping/fashion Sports Travel	
Eye Color	🔳 Red 🔳 Green 🔳 Brown 🔳 Blue Gold	
(Submit) Reset		
		1.

Рис. 8.35. Стилевое оформление ячеек правого столбца таблицы

Затем примените стили к правым ячейкам таблицы. Для того чтобы подчеркнуть отличие между левым и правым столбцами таблицы, задайте черный цвет фона для правых ячеек. Кроме того, установите серый цвет границы слева для смягчения перехода при чтении слева направо (рис. 8.35):

```
td {
  vertical-align: middle;
```

}

```
background-color: black;
border-bottom: 1px solid white;
color: white;
border-left: 4px solid gray;
padding: 4px;
font-family: Verdana;
font-size: .7em;
}
```

	CSS Cookbook	
Back Forward Reload Stop	Home Print 1Password	sers/cl 😭 🔻 🔇
CSS Cookbook 🛛 🕄 🏹	Christopher Schmitt : 😧 🖉 🖉 Technolog	gy Books, Te 🛞 🕂 葉
	Account Information	
Login Name*		
Password*		
Confirm Password*		
Email Address*		
Confirm Email*		
	Contact Information	
First Name*		
Last Name*		
Address 1*		
Address 2		
City*		
State or Province*	Select 🗘	
Zip*		
Country*		
Gender*	单 Female 🌘 Male	
	Misc. Information	
Annual Household Income	Select	
Interests	Shopping/fashion Sports Travel	
Eye Color	🛢 Red 🛢 Green 🛢 Brown 🛢 Blue Gold	
(Submit) (Reset)		2

Рис. 8.36. Скос в фоновых изображениях

Далее добавьте в фоновое изображение эффект снятия фаски или скос с легким блестящим отливом, как показано на рис. 8.36:

```
td {
  vertical-align: middle;
  background-color: black;
```

```
border-bottom: 1px solid white;
color: white;
border-left: 4px solid gray;
padding: 4px;
font-family: Verdana;
font-size: .7em;
background-image: url(title-glass.png);
background-repeat: repeat-x;
background-position: 50%;
}
```

0 0 CSS Cookbook  $\bigcirc$ . (C) (X)(IP) 🗋 🤇 file:///Users/cl 🏠 🔻 R-Q Reload Print Back Forward Stop Home 1Password 🐣 Technology Books, Te... 🕲 CSS Cookbook  $\odot$ Christopher Schmitt : ...  $\pm$ Account Information Login Name\* Password\* **Confirm Password\*** Email Address\* Confirm Email\* Contact Information First Name\* Last Name\* Address 1\* Address 2 City\* Select... 💠 State or Province\* Zip\* Country\* Gender\* Female 🕘 Male Misc. Information ÷ Annual Household Income Select... Interests Shopping/fashion Sports Travel Eye Color 🗖 Red 📕 Green 📕 Brown 📕 Blue Gold Submit Reset

Рис. 8.37. Пояснительный текст красного цвета к обязательным полям, отмеченным звездочкой

Некоторые поля требуют обязательного заполнения, измените у них цвет шрифта пояснительного текста. Эта модификация позволит определить даже при беглом взгляде на форму, какие поля являются обязательными (рис. 8.37):

```
.required {
  color: red;
}
```

#### Примечание

Несмотря на то, что в правиле CSS задан красный цвет, при печати он преобразуется в оттенок серого.

Откорректируйте заголовки формы, обозначающие разные секции, задав вывод текста заглавными буквами немного большего размера по сравнению с размером шрифта остальных элементов формы (рис. 8.38):

```
.header th {
  text-align: left;
  text-transform: uppercase;
  font-size: .9em;
}
```



Рис. 8.38. Стилевое оформление заголовков секций

Передвиньте слегка заголовки секций, чтобы они располагались над вторым столбцом. Для определения размещения заголовков сложите ширину левого столбца (200 пикселов), отступ правого столбца (4 пиксела), ширину рамки слева у правого столбца (4 пиксела) и отступ справа для левого столбца (12 пикселов):

```
.header th {
  text-align: left;
  text-transform: uppercase;
  font-size: .9em;
  padding-left: 220px;
}
```

00	CSS Cookbook	$\bigcirc$
<ul> <li>. C ×</li> </ul>		
Back Forward Reload Stop	Home Print 1Password	
	Christopher Schmitt : 😸 🖉 🖓 Technology Books, Te	
	ACCOUNT INFORMATION	-
Login Name*		
Password*		
Confirm Password*		
Email Address*		
Confirm Email*		
	CONTACT INFORMATION	
First Name*		
Last Name*		
Address 1*		
Address 2		
City*		
State or Province*	Select	
Zip*		
Country*		
Gender*	● Female ● Male	
	MISC. INFORMATION	
Annual Household Income	Select	-
Interests	Shopping/fashion Sports Travel	
Eye Color	🗖 Red 🛢 Green 🛢 Brown 🛢 Blue Gold	
(Submit) (Reset)		Y

Рис. 8.39. Увеличенный отступ заголовков секций

Затем добавьте немного визуальной привлекательности, увеличив толщину рамок заголовков сверху и снизу (рис. 8.39).

```
.header th {
  text-align: left;
```

Формы

}

```
text-transform: uppercase;
font-size: .9em;
padding-left: 220px;
border-bottom: 2px solid gray;
border-top: 2px solid black;
}
```

В качестве последнего штриха переместите кнопки Submit и Reset так, чтобы они располагались под полями формы с горизонтальным выравниванием таким же, как у заголовков секций, назначив поле слева шириной 220 пикселов (рис. 8.40):

```
#buttonSubmit {
margin-left: 220px;
margin-top: 4px;
              00
                                                  CSS Cookbook
                                                                                                 \bigcirc
               .
                     ( >
                               (C)
                                     (X)
                                            ( 🛖 ) ( 🚔 )
                                                           (1P)
                                                                   📋 🚺 file:///Users/c 😭 🔻 😵
                     Forward
               Back
                              Reload
                                     Stop
                                            Home
                                                  Print
                                                        1Password
               Π
                     CSS Cookbook
                                     \odot
                                            Christopher Schmitt : ... 😮 🖉 Technology Books, Te... 😮 🕇 💻
                                            ACCOUNT INFORMATION
                              Login Name*
                                Password*
                         Confirm Password*
                            Email Address*
                            Confirm Email*
                                            CONTACT INFORMATION
                               First Name*
                               Last Name*
                                Address 1*
                                 Address 2
                                     City*
                                            Select... 🗘
                          State or Province*
                                     Zip*
                                 Country*
                                  Gender*
                                             🕘 Female 🏾 🔵 Male
                                            MISC. INFORMATION
                                                          ÷
                   Annual Household Income
                                             Select...
                                 Interests
                                             🗏 Shopping/fashion 📕 Sports 📕 Travel
                                 Eye Color
                                             Red Green Brown Blue Gold
                                            (Submit) (Reset)
                                                             k
```

453

### глава 9 Таблицы

#### 9.0. Введение

Благодаря технологии CSS Web-дизайнеры поняли, что они могут отказаться от практики манипулирования HTML-таблицами для сохранения целостности их проектов. Такие приемы, как разрезание изображения для вставки "кусков" в отдельные ячейки таблицы или вложение таблиц в макеты Web-страниц, теперь выходят из моды. Тем не менее таблицы все еще занимают свое место в дизайнерских проектах. Webразработчики применяют таблицы для представления табличных данных, таких как календарь или научные данные, и следовательно, могут использовать технологию CSS для стилевого оформления этих таблиц.

В данной главе показано, как улучшить внешний вид таблиц, применяя стили к заголовкам, задавая рамки таблицы и ее ячеек и уменьшая зазоры между изображениями, хранящимися в ячейках таблицы. Пример дизайнерского проекта в конце главы проведет вас через все стадии, необходимые для стилевого оформления календаря.

#### 9.1. Задание границ и отступов в ячейках и таблицах

#### Проблема

Вы хотите задать границы и объем свободного пространства в ячейках таблицы для формирования более привлекательного внешнего вида (рис. 9.1) таблицы.

#### Решение

Используйте свойство padding для задания объема свободного пространства между содержимым ячейки и ее краями. Примените свойство border для вывода границ, как ячеек, так и всей таблицы:

```
table {
  border-collapse: collapse;
  border: 5px solid #444;
}
td {
  padding: 10px;
}
```

#### Таблицы

```
th {
   padding: 10px;
   color: white;
   background-color: black;
}
td, th+th {
   border: 5px solid #666;
}
td+td {
   border: 5px solid #ccc;
   text-align: center;
}
td#winner {
   border: 7px dotted #999;
}
```



Рис. 9.1. Рамки и отступы для таблицы и ее ячеек

#### Обсуждение

Для отображения рамок или границ HTML-таблиц есть две модели: collapse и separate. В модели collapse у смежных ячеек таблицы общие границы, как у первой таблицы на рис. 9.1.

Вторая, нижняя таблица на рис. 9.1 демонстрирует модель separate, в которой ячейки таблицы отделены друг от друга с помощью собственных отдельных рамок.

#### Проблемы Web-обозревателей

Во время написания книги модель collapse была шире распространена в Webобозревателях, и потому чаще использовалась дизайнерами, чем модель separate.

Во всех современных обозревателях по умолчанию задается модель collapse, в Firefox по умолчанию применяется модель separate. Поскольку в спецификации CSS не описано такое поведение, в ваших таблицах стилей следует явно задавать модель collapse на случай, если в последующих версиях Web-обозревателей будут установлены другие значения по умолчанию. Модель отображения границ задается с помощью свойства border-collapse со значением collapse:

```
table {
  border-collapse: collapse;
}
```

#### Задание границ ячеек

Атрибут border элемента table определяет границы таблицы и входящих в нее ячеек. Вы можете задать свойство border с разной толщиной границ для таблицы и отдельных ячеек.

Если задается рамка ячейки, противоречащая предшествующему правилу CSS, для разрешения конфликтной ситуации применяются следующие четыре правила из спецификации CSS.

- Если задано свойство border-style со значением hidden, все другие стили границ отменяются.
- Если задано свойство border-style со значением none, побеждает любой другой стиль границ.
- ◆ Если у ячейки свойство border-style не равно hidden или none, более толстые границы переопределяют более тонкие. Если у смежных ячеек одна и та же толщина, стиль границ будет выбираться в следующем порядке: double, solid, dashed, dotted, ridge, outset, groove, inset.
- Если у смежных ячеек разный цвет границ при одинаковых стилях и толщине, цвет будет определяться в следующем порядке: ячейка, строка, группа строк, столбец, группа столбцов и затем таблица.

В модели separate у каждой ячейки собственные границы и они могут иметь стилевое оформление, не зависящее от границ других ячеек. В этой модели для задания горизонтальной и вертикальной пробельных зон между ячейками применяется свойство border-spacing:

```
table#runoffdata {
  border-collapse: separate;
  border-spacing: 4px 4px;
}
```

Если значение свойства border-collapse равно separate, любые стили, заданные для строк, столбцов или группы ячеек таблицы, не применяются. Кроме того, ячейки таблицы, не содержащие контента, могут отображаться или скрываться с помощью свойства empty-cells со значением show или hide соответственно.

Несмотря на то, что модель separate предоставляет Web-дизайнерам больше возможностей управления, во время написания книги ее поддерживали только Webобозреватели Firefox, Safari и Chrome, но не IE. Таким образом, большинство Webразработчиков привязаны к модели collapse.

#### См. также

Модели границ в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/ tables.html#propdef-border-collapse; дополнительное обсуждение таблиц в главе 11 книги Эрика Мейера (Eric A. Meyer) "Cascading Style Sheets: the Definitive Guide" (Каскадные таблицы стилей: подробное руководство) (издательство O'Reilly).

# 9.2. Задание объема свободного пространства вокруг ячейки

#### Проблема

Вы хотите откорректировать объем свободного пространства между рамкой таблицы и рамками ячеек.

#### Решение

Воспользуйтесь атрибутом таблицы cellspacing:

```
General Demographic Characteristics of Tallahassee, FL
```

```
Total population
272,091
```

#### Обсуждение

В спецификации CSS 2.1 описан стандартный метод манипулирования атрибутом таблицы cellspacing с помощью свойства border-spacing при одновременном задании значения separate в свойстве border-collapse:

```
border-collapse: separate;
border-spacing: 15px;
```

Однако в Internet Explorer 6 для Windows эта часть спецификации не реализована.

В настоящее время применение HTML-атрибута cellspacing — рекомендуемый вариант, работающий в современных обозревателях и обладающий обратной совместимостью.

#### См. также

*Разд. 9.1* о задании рамок и отступов ячеек; в спецификации CSS 2.1 свойство bordercollapse на странице http://www.w3.org/TR/CSS21/tables.html#propdef-bordercollapse и свойство border-spacing на странице http://www.w3.org/TR/CSS21/ tables.html#propdef-border-spacing; *pa3d. 8.2* и *8.8*.

#### 9.3. Задание стиля заголовка

#### Проблема

Вы хотите задать стиль для заголовка таблицы.

#### Решение

Для стилевого оформления заголовка воспользуйтесь селектором элемента caption:

```
table caption {
   font: 1.5em Georgia, "Times New Roman", Times, serif;
   padding: 1em;
}
```

#### Обсуждение

Заголовки применяются для описания содержимого таблицы и должны располагаться после открытия элемента table и перед другими элементами таблицы, такими как thead или tr:

```
in your shopping cart.">
<caption>Shopping Cart Listing - <strong>Subtotal: $45.16</strong>;
changed quantities? <input type="submit" value="Update price(s)" /></caption>
...
```

Web-обозреватели могут по-разному визуализировать элемент caption. Тем не менее caption всегда будет выводиться на экран любым Web-обозревателем, к нему можно применять стили с помощью технологии CSS, и таким образом, это самый доступный способ отображения заголовка таблицы.

#### См. также

Элемент caption в спецификации HTML 4.01 на странице http://www.w3.org/TR/ html4/struct/tables.html#h-11.2.2.

### 9.4. Задание стилей в ячейках таблицы

#### Проблема

Вы хотите применить стили к ссылкам в ячейке таблицы, чтобы сделать их визуально отличающимися от остальной информации на странице.

#### Решение

Для изменения стиля содержимого табличной ячейки используйте селектор потомка (descendant selector) (иногда называемый контекстным селектором (contextual selector)):

```
td a {
  display: block;
  background-color: #333;
  color: white;
  text-decoration: none;
  padding: 4px;
}
```

#### Обсуждение

Используя для применения стилей селекторы типа и потомка в правиле CSS td a, вы уменьшаете объем разметки, необходимой для улучшения ваших проектов, и размер файла документа. Стиль влияет только на элементы а в ячейках таблицы (помеченных тегами td).

Если вам нужен больший контроль над стилевым оформлением содержимого ячейки таблицы, применяйте селектор класса:

```
<a href="/">Home</a>
```

Затем вы можете применить правила CSS к содержимому ячейки с помощью комбинации селекторов класса и потомка:

```
td.navText a {
  font-size: x-small;
}
```

Если необходимо задать стили данных табличной ячейки, содержащей не только ссылки, а дополнительное содержимое и разметку, заключите этот контент в элемент div для того, чтобы воспользоваться селектором класса.

В приведенном далее примере маркированный список заключен в элемент div, в котором задан атрибут class:

```
<div class="tblcontent">
To-do list on your day off.
<a href="http://www.imdb.com/title/tt0120737">Watch <cite>Fellowship of
the Rings</cite>, Extended Version</a>
<a href="http://www.imdb.com/title/tt0167261/">Watch
<cite>Two Towers</cite>, Extended Version</a>
<a href="http://www.imdb.com/title/tt0167260/">Watch <cite>Return of the
King</cite>, Extended Version</a>
Start or join local Elvish society.
```

Правила CSS для стилевого оформления содержимого ячейки таблицы могли бы выглядеть следующим образом:

```
.tblcontent p {
margin: 0;
padding: 0;
font-weight: bold;
}
.tblcontent ul {
margin: 0;
padding: 0;
}
.tblcontent li {
margin: 0;
padding: 0;
line-height: 1.5;
}
.tblcontent li a {
padding-left: 15px;
background-image: url(bullet.gif);
background-repeat: no-repeat;
}
```

#### См. также

В спецификации CSS 2.1 информацию, касающуюся селекторов типа, на странице http://www.w3.org/TR/CSS21/selector.html#type-selectors; сведения о селекторах потомка на http://www.w3.org/TR/CSS21/selector.html#descendant-selectors.

### 9.5. Задание стилей для элементов шапки таблицы

#### Проблема

Вы хотите создать стилевое оформление шапки таблицы, отличающееся от стиля оформления содержимого обычных ячеек таблицы; на рис. 9.2 показана стандартная шапка таблицы, а на рис. 9.3 — стилизованная версия той же таблицы.

00		(	SS Cookbook				$\bigcirc$
Back Forward	Reload	Stop Hom	e Print 1Pass	P	□ · ☆ ·	• G	٩
CSS Cookbo	ook 😳	Christop	oher Schmi 🕲	两	Technology Books, 🕲	+	=
Table 1. Ger	neral Den	nographic C	haracteristics				
	Estimate	e Lower Bou	and Upper Bou	ind			
Total population	272,091	****	****				
Sex							
Male	129,179	127,670	131,908				
Female	142,302	140,183	144,421				
							11

Рис. 9.2. Вид таблицы до применения стилей к элементам шапки таблицы

Back Forwa	ard Reload St	CSS Cook	Dook 1P 1Password	ې ۲۰۰۹ - ۲۰۰۹ م
CSS Co	ookbook 😳 其	Christopher Schm	i 😧 🛛 🛵 Technology B	ooks, 🕲   +
Table 1. Ge	eneral Demograp	hic Characteristics		
	Estimate	Lower Bound	Upper Bound	
Total popula	ation 272,091	***	***	
Sex				
Male	129,179	127,670	131,908	
Female	142,302	140,183	144,421	

#### Решение

Для стилевого оформления шапки таблицы воспользуйтесь селектором элемента:

```
th {
  text-align: left;
  padding: 1em 1.5em 1em 0.5em;
  color: white;
  border-right: 1px solid rgba(0, 204, 255, .8);
  border-bottom: 1px solid rgba(0, 204, 255, .8);;
  text-shadow: 0 1px 0 rgba(0, 0, 0, .8);
  background: blue url(title-glass.png) repeat-x 50%;
  font: .9em Arial, Helvetica, Verdana, sans-serif;
}
```

Если у элементов th в таблицах несколько строк, требующих применения разных стилей, для различения строк используйте селектор класса:

```
.secondrow th {
/* Использование в качестве фона более светлого оттенка синего */
background-color: #009;
}
```

Поместите в этот класс соответствующие строки таблицы:

```
Table 1. General Demographic Characteristics
>
Estimate
</t.h>
>
Lower Bound
>
Upper Bound
```

#### Обсуждение

Элемент th определяет содержимое ячейки как заголовок, или элемент шапки таблицы. Задавая его оформление, применяйте стили, отличающие внешний вид такой ячейки от отображения содержимого ячейки таблицы td. Для таблицы это разновидность заголовка, поэтому он должен выглядеть как заголовок к статье.

Вы можете создать контрастирующие стили, просто откорректировав значения любого из следующих свойств: font-family, background-color, font-size, font-weight и textalign (см. *разд. 3.1* о задании шрифтов и *разд. 3.6* о выборе единиц измерения и размеров). Независимо от того, какое из этих свойств изменяется, есть шанс улучшить внешний вид шапки таблицы.

#### Примечание

Имейте в виду, что заголовки в шапке таблицы с неразрывными пробелами между символами интерпретируются Web-обозревателем как одно слово и, следовательно, не разбиваются принудительно на две строки.

#### См. также

Селекторы типа в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/ selector.html#type-selectors.

# 9.6. Удаление зазоров у изображений, помещенных в ячейки таблицы

#### Проблема

Вы хотите избавиться от пробельной зоны в ячейке таблицы, содержащей только изображение. Другими словами, перейти от рис. 9.4 к рис. 9.5.

#### Решение

Задайте вывод на экран изображения как элемента блочного уровня:

```
td img {
  display: block;
}
```

#### Обсуждение

Web-обозреватель помещает изображение на базовую линию, применяемую для текстового содержимого, т. к. изображение вставляется как внутристрочный элемент. Следовательно, чтобы заставить Web-обозреватель визуализировать его иначе, превратите изображение в элемент блочного уровня. Упомянутая базовая линия текста располагается не на нижней стороне ячейки, потому что у некоторых букв (например, g, p, q и y) есть нижние выносные элементы, которые расположены под базовой линией (рис. 9.6).

Поскольку положение базовой линии текста определяется как процент от размера шрифта, вы не можете просто удалить пробельную зону под ней. Заставляя Webобозреватель обрабатывать изображение иначе, можно вообще избежать автоматического создания пробельной зоны для нижних выносных элементов букв. Таким образом, задайте для изображения значение block в свойстве display, как показано в решении.



Рис. 9.4. Зазор, появляющийся в ячейке таблицы под изображением



Рис. 9.5. Вывод на экран изображения, помещенного в ячейку таблицы как элемент блочного уровня

#### Вертикальное выравнивание

Иногда установка для изображения значения block в свойстве display — не лучшее решение для удаления зазора вокруг изображения, помещенного в ячейку таблицы. В этом случае есть другой метод удаления зазора, если высота изображения больше высоты строчного блока, — установить для изображения свойство vertical-align со значением bottom.

#### См. также

Свойство display в спецификации CSS 2.1 на странице http://www.w3.org/ TR/CSS21/visuren.html#propdef-display; информацию о режиме обратной совместимости и почти стандартном режиме на сайте http://developer.mozilla.org/en/docs/ Mozilla's\_DOCTYPE\_sniffing.



Рис. 9.6. Нижние выносные элементы строчных букв g, p, q и y, указывающие на пробельную зону под изображением

### 9.7. Устранение зазоров между ячейками таблицы

#### Проблема

Вы хотите устранить промежутки между соседними ячейками таблицы.

#### Решение

Примените к таблице модель collapse:

```
#shoppingcartTable {
    border-collapse: collapse;
    width: 100%;
    border: 1px solid #666;
    #shoppingcartTable th {
        background: #888 url(th_bkgd.jpg) repeat-x;
        font: italic 1.5em Georgia, "Times New Roman", Times, serif;
        padding: .5em 0 .5em 7px;
        text-align: left;
        border-top: 1px solid #666;
        border-bottom: 1px solid #666;
        text-shadow: #ccc -2px 2px -2px;
    }
```

#### Обсуждение

При установке значения collapse в свойстве border-collapse Web-обозреватель удаляет промежутки между ячейками таблицы. Таким образом, когда вы примените свойство border к ячейкам таблицы, в результате вдоль строки или столбца будет проведена четкая неразрывная линия.

#### См. также

Разд. 9.1 с более подробным обсуждением модели collapse.

#### 9.8. Создание чередующихся цветов фона в строках таблицы

#### Проблема

Вы хотите разнообразить таблицу (рис. 9.7) чередующимися цветами фона строк (рис. 9.8).





Рис. 9.8. Чередующиеся цвета в строках таблицы

#### Решение

Создайте селектор класса, специально для нечетных строк таблицы:

```
tr {
  background-color: #eee;
}
tr.odd {
  background-color: #ccc;
}
```

Затем добавьте чересстрочно в таблицу строки с атрибутом class, имеющим значение odd (см. рис. 9.8):

```
class="dltprod">
Item added on March 22, 2006.
<a href="" title="Delete this product"><img src="x.gif" alt="delete"
class="dltitem" /></a>
```
```
<img class="prod" alt="product image" src="u2-dismantle.jpg" />
 <div class="prodtitle"><a href="/product.php?id=B0006399FS">How
to Dismantle an Atomic Bomb</a></div>
  ~ <strong>U2</strong>
<input type="text" value="1" name="qty" size="2" />
$9.66
Item added on March 22, 2006.
 <a href="" title="Delete this product"><img src="x.gif" alt="delete"
class="dltitem" /></a>
</t.d>
<img class="prod" alt="product image" src="apple-whenthepawn.jpg" />
 <div class="prodtitle"><a href="/product.php?id=B00002MZ4W">When The Pawn
Hits...</a></div>
 ~ <strong>Fiona Apple</strong>
<input type="text" value="1" name="qty" size="2" />
$7.97
```

## Обсуждение

Предлагаемая в данном решении разметка каждого чересстрочного элемента tr, несмотря на большую трудоемкость в случае ручного кодирования длинных таблиц, гарантирует соответствие отображений в разных Web-обозревателях.

Второе предлагаемое решение устраняет необходимость дополнительной разметки HTML-таблицы. Решение упрощается за счет использования селектора nth-child, включенного в спецификацию CSS3:

```
tr {
   background-color: #eee;
}
tr:nth-child(odd) {
   background-color: #ccc;
}
```

К сожалению, поддержка спецификации CSS3 ограничена Web-обозревателями Safari 3, Firefox 3.5 и Opera 9.5 и их более поздними версиями.

#### Альтернативные решения

Другие решения не ограничиваются использованием только технологии CSS. В одном из них применяется язык JavaScript, взаимодействующий с объектной моделью документа (Document Object Model, DOM) и автоматически применяющий чересстрочно стили в таблице. Такое решение можно найти в *разд. 14.6*. Недостаток данного решения состоит в том, что оно не сработает, если пользователь отключит поддержку JavaScript в своем Web-обозревателе.

Другим программным решением может быть использование серверного языка программирования, например PHP или ColdFusion, для написания простых скриптов, которые автоматизируют создание таблицы. (Этот метод особенно полезен, если для формирования и сопровождения табличных данных применяется серверная база данных (backend database).)

## См. также

Селектор псевдокласса nth-child в спецификации CSS 3 на странице htpp://www.w3.org/TR/css3-selectors/#nth-child-pseudo.

## 9.9. Создание эффекта выделения цветом строки таблицы

## Проблема

Вы хотите выделить цветом всю строку таблицы, когда курсор перемещается по ячейкам в этой строке.

## Решение

Используйте в элементе tr псевдокласс :hover:

```
tr:hover {
  background: yellow;
}
```

## Обсуждение

Псевдокласс :hover часто встречается в ссылках, с описанием создания визуальных эффектов наведения указателя мыши на ссылку (rollover). Но спецификация CSS не ограничивает его применение только ссылками. Он также может использоваться и в других элементах, таких как р или div.

Современные Web-обозреватели поддерживают это решение. Но IE7 и более ранние версии не создают визуального эффекта при наведении указателя мыши на другие элементы, не ссылки.

#### См. также

Динамические псевдоклассы в спецификации CSS 2.1 на странице htpp://www.w3.org/ TR/CSS21/selector.html#dynamic-pseudo-classes.

## 9.10. Пример проекта: элегантный календарь

Незаменимые для организации информации календари позволяют назначать деловые завтраки, помнить дни рождения и планировать медовые месяцы. Мы, дизайнеры, можем воспринимать все эти месяцы, даты и назначенные встречи как табличные данные.

Если отобразить ваш календарь как обычную HTML-таблицу, скорее всего, она будет выглядеть довольно незамысловатой, а если в нее включено множество событий, то, возможно, и запутанной. В данном проекте мы используем технологию CSS для создания календаря, который будет привлекательней созданного с помощью простого стандартного языка HTML.

Прежде всего, взгляните на рис. 9.9, демонстрирующий отображение разметки для календаря без применения стилей.

000		CSS Coo	kbook				$\bigcirc$
Rack	Econyard Peload Storn Home Print 1Passwo	file:///	Jsers/christoph	er/Documents/Work/	<b>☆</b> ▼)	- Google	٩
	CSS Cookbook 😵 🚺 Christopher Sc	hmitt : Designer,	C 🚑 Techno	logy Books, Tech Conf	er 🕲	+	=
		<u>≤ 2010 ≥</u>					
		≤ October ≥				<b>6</b>	
Sunday	Monday	Tuesday We	dnesday	Thursday	Friday	7 Saturday	
<u>1 ±</u>	$\frac{2 \pm}{\text{Regular City Commission meeting agenda}} \pm$	<u>4</u> ±	Dad'	s birthday	<u>6</u> ±	FSU at UM	
<u>8</u> ±	$\frac{9 \pm}{\text{Regular City Commission meeting agenda}} \frac{10 +}{10 + 10}$	11	<u>+ 12 +</u>		<u>13 +</u>	<u>14 ±</u> FSU at Duke	
<u>15 ±</u>	$\frac{16 \pm}{\text{Regular City Commission meeting agenda}} \frac{17 \pm}{17 \pm}$	<u>18</u>	± <u>19</u> ± Food	Safety & Awarenes	<u>ss</u> 20 ±	<u>21 ±</u> FSU at Virginia	
<u>22 ±</u>	$\frac{23 \pm}{\text{Regular City Commission meeting agenda}} \frac{24 \pm}{24 \pm}$	25	± 26±		<u>27 ±</u>	<u>29 ±</u> FSU at UM	
<u>29 ±</u>	$\frac{30 \pm}{\text{Regular City Commission meeting agenda Hallo}}$	oween Flu shot					
							1

Рис. 9.9. Календарь без стилей

Далее посмотрите на саму разметку, чтобы увидеть, как она устроена. Как вы видели в *разд. 9.2*, атрибут cellspacing необходимо задать в элементе table:

Теперь задайте первые три строки шапки таблицы (th), содержащие год, месяц и дни в строках, относящихся к соответствующим заголовкам:

```
<a href="year.html?previous">&lt;</a> 2010 <a
href="year.html?next">&gt;</a>
```

```
<a href="month.html?previous">&lt;</a> October <a
href="month.html?next">&gt;</a>
```

Первая дата в этом календаре — первое октября (October 1) — выпадает на воскресенье (Sunday). Для обозначения воскресений и суббот (Saturday) как выходных дней используйте селектор class в элементе td.

У каждой даты месяца есть ссылка на саму дату (которая теоретически будет перенаправлять пользователя к подробному списку дел на день), а также ссылка для добавления событий, относящихся к этому дню. Заключите эти две ссылки в элемент div, чтобы после вставки новых событий существовало четкое разделение между двумя секциями в ячейке таблицы:

```
<div>
<div>
<a href="1.html" class="date">1</a>
<a href="add.html" class="addevent">+</a>
</div>
```

У следующей даты — второго октября (October 2) — есть записанное событие. Оно размечено как ссылка и помещено следом за элементом div, содержащим дату и ссылки со значением класса addevent (поскольку второе октября — это рабочий день, атрибут class со значением weekend к элементу td не применяется):

```
</div>
</div>
</a href="2.html" class="date">2</a>
</a href="add.html" class="addevent">+</a>
</div>
</a href="16.html?id=1" class="event">Regular City
Commission meeting agenda</a>
```

Остальная разметка организована аналогичным образом:

```
<div>
<a href="3.html" class="date">3</a>
```

```
<a href="add.html" class="addevent">+</a>
  </div>
 <div>
    <a href="4.html" class="date">4</a>
    <a href="add.html" class="addevent">+</a>
   </div>
 \langle t.d \rangle
  <div>
   <a href="5.html" class="date">5</a>
   <a href="add.html" class="addevent">+</a>
  </div>
  <a href="5.html?id=1" class="event">Dad's birthday</a>
 <div>
    <a href="6.html" class="date">6</a>
    <a href="add.html" class="addevent">+</a>
   </div>
 <div>
    <a href="7.html" class="date">7</a>
    <a href="add.html" class="addevent">+</a>
   </div>
   <a href="7.html?id=1" class="event">FSU at UM</a>
 </t.d>
[...]
<div>
    <a href="29.html" class="date">29</a>
    <a href="add.html" class="addevent">+</a>
   </div>
   <div class="event">Buy candy</div>
 \langle t.d \rangle
   <div>
    <a href="30.html" class="date">30</a>
    <a href="add.html" class="addevent">+</a>
   </div>
   <a href="16.html?id=1" class="event">Regular City
Commission meeting agenda</a>
 <div>
    <a href="31.html" class="date">31</a>
```

```
<a href="add.html" class="addevent">+</a>
   </div>
   <a href="31.html?id=1" class="event">Halloween</a>
   <a href="31.html?id=2" class="event">Flu shot</a>
 <div class="emptydate">&nbsp;</div>
 \langle t.d \rangle
  <div class="emptydate">&nbsp;</div>
 </t.d>
 <div class="emptydate">&nbsp;</div>
 <div class="emptydate">&nbsp;</div>
```

Выполнив разметку календаря, можно приступать к установке стилей. Сначала задайте font-size, равным 62.5 %, как обсуждалось в *разд. 3.6*.

Затем примените стили к таблице и к ссылкам. Ширина таблицы задана равной 100 %, выбрана модель рамки — collapse (см. *разд. 9.1*), распространенная модель, которой обычно пользуются дизайнеры и которую большинство Web-обозревателей отображают верно благодаря реализации в них технологии CSS, декоративное оформление с применением подчеркивания отключено (рис. 9.10):

```
body {
  font-size: 62.5%;
}
table {
  width: 100%;
  border-collapse: collapse;
}
td a:link, td a:visited {
  text-decoration: none;
  font-family: "Gill Sans", Calibri, Trebuchet, sans-serif;
}
```

Далее задайте стили первых трех строк таблицы. Строки размечены с помощью селекторов ID, поскольку вы хотите отображать их в документе один раз. Задайте простое стилевое оформление для этих строк, установив моноширинный шрифт (см. *разд. 3.2*) в заголовках, а затем уменьшив размеры шрифта и сохранив самый большой размер шрифта для месяца (рис. 9.11):

```
#year {
  font-family: Consolas, "Lucida Console", Monaco, monospace;
  font-size: 3em;
  padding: 0;
  margin: 0;
}
```



Рис. 9.10. Подчеркивание ссылок удалено

0	0	CSS	Cookbook			$\bigcirc$
Back	Forward Reload Stop Home Print	1P 1Password	///Users/christor	oher/Documents/Work/ 🏠 💙	- 🚷 🕻 Goo	gle Q
	CSS Cookbook 🛛 🕄 Chri	stopher Schmitt : Designer,	🕄 🖉 Tech	nology Books, Tech Confer 🛞	+	
		< 2	010 >			
		< 0c	tober >			
Sund	ay Monday	Tuesday	Wedne	sday Thursday	Friday	Saturday
1+	2 + Regular City Commission meeting agenda	3 +	4+	5 + Dad's birthday	6+	7 + FSU at UM
8+	9 + Regular City Commission meeting agenda	10 +	11 +	12 +	13 +	14 + FSU at Duke
15 +	16 + Regular City Commission meeting agenda	17 +	18+	19 + Food Safety & Awareness	20 +	21 + FSU at Virginia
22 +	23 + Regular City Commission meeting agenda	24 +	25 +	26 +	27 +	29 + FSU at UM
29 +	30 + Regular City Commission meeting agenda	31 + Halloween Flu shot				

#### Таблицы

```
#month {
font-family: Consolas, "Lucida Console", Monaco, monospace;
font-size: 2em;
padding: 0;
margin: 0;
}
#days {
background: black url(title-glass.png) repeat-x 50%;
color: white;
font-family: "Gill Sans", Calibri, Trebuchet, sans-serif;
width: 75px;
text-shadow: 0px -1px 0px rgba(0, 0, 0, .8);
border: lpx solid rgba(0,0,0,.5);
border-top: none;
}
#days th {
padding: 4px;
}
```

Теперь пришло время задать стиль дат и добавить ссылки на события в каждую ячейку. Для отображения даты в рамке, как в большинстве календарей, поместите границу справа и снизу от текста и переместите содержимое влево.

Вы хотите вставить ссылки на события рядом с датами. Включение ссылки в плавающую модель со смещением вправо означает, что ссылка будет располагаться рядом с датой следующего дня. Сместив влево ссылку на добавляемое событие, вы сообщите пользователю, что знак + означает вставку события в данный конкретный день (рис. 9.12):

```
.date {
border-right: 1px solid black;
border-bottom: 1px solid black;
font-family: Consolas, "Lucida Console", Monaco, monospace;
text-decoration: none;
float: left;
width: 1.5em;
height: 1.5em;
background-color: white;
text-align: center;
}
.addevent {
display: block;
float: left;
width: 1em;
height: lem;
text-align: center;
background-color: #666;
color: white;
font-weight: bold;
text-decoration: none;
}
```



Рис. 9.12. Стили, введенные для даты и ссылок на вставляемые события

Теперь пора посмотреть, какое стилевое оформление можно задать для списков событий дня. Поскольку предыдущие ссылки включены в поток плавающей модели, необходимо создать видимый разрыв строки и переместить события под дату.

Разрыв достигается установкой свойства clear со значением both. Это свойство применяется для обозначения сторон элемента, которые не следует размещать рядом с элементом, включенным в плавающую модель.

В данном случае вы не хотите, чтобы левая сторона располагалась рядом с датой и ссылками для вставки событий. Однако на случай изменения дизайна в будущем и размещения дат на противоположной стороне воспользуйтесь в свойстве clear значением both вместо значения left.

Далее для ссылки на событие измените в свойстве display значение на block и задайте отступ снизу (рис. 9.13). Эти изменения вносятся для того, чтобы помешать множественным событиям, размещаемым в ячейке таблицы, накладываться друг на друга. Отступ также действует как отличный визуальный буфер, позволяющий глазу легко различать события:

```
.event {
  clear: both;
  padding-left: 1em;
  padding-bottom: .75em;
  display: block;
}
```

Для каждой ячейки таблицы задайте ширину 14 %. Это значение применяется потому, что 7 (соответствующее 7 разделам календаря или количеству дней в неделе) уклады-

	° • . C × A 🖶		SS Cookbook	ristopher/Documents/Work/		
Back	Forward Reload Stop Home Print CSS Cookbook 😗 🏹 Chri	1Password stopher Schmitt : Desig	mer 😳 🍂	Technology Books, Tech Confer.	3 +	
		<mark>۲</mark> ۲	2010	2		
Sund	ay Monday	Tuesda	y Wednes	day Thursday	Friday	Saturday
1 41	Regular City Commission meeting agenda	3 🖽	4 5	Dad's birthday	6 🖽	FSU at UM
8	9 Regular City Commission meeting agenda	10	11 🖸	12 4	13 🖶	14 + FSU at Duke
15 4	16 Regular City Commission meeting agenda	17	18 -	19 Tood Safety & Awareness	20 🗭	21 FSU at Virginia
22 +	23 Regular City Commission meeting agenda	24 =	25 +	26 +	27 🖽	29 FSU at UM
29	30 🔁 Regular City Commission meeting agenda	31 Halloween Flu shot				

Рис. 9.13. Ссылки на события, трактуемые как элементы блочного уровня

вается примерно 14 раз в 100 (число соответствующее 100 % окна просмотра). Кроме того, поместите белую рамку вокруг ячейки и разместите весь контент в верхней части ячейки с помощью свойства vertical-align (рис. 9.14):

```
td {
  width: 14%;
  background-color: #ccc;
  border: 1px solid white;
  vertical-align: top;
  font-size: 1.2em;
  padding: 1px;
  background: url(content-bkgd.png) repeat-x;
  border: 1px solid rgba(0,0,0,.5);
  border-top: none;
}
```

Задайте для выходных дней более темный фоновый цвет по сравнению с цветом фона рабочих дней недели (рис. 9.15):

```
.weekend {
  background-color: #999;
}
```

Закрасьте светло-серым дни, оставшиеся в календаре без дат (рис. 9.16):

```
.emptydate {
  border-right: 1px solid #666;
  border-bottom: 1px solid #666;
```

00			CSS Cook	book		C
Back Forward	, C X A Reload Stop Home SS Cookbook &	Print 1Password Christopher Schr	j file:///Us	ers/christopher/Documents	s/Work/ 🏠 🔻 -	Google Q
			≤ 201 ≤ Octob	0 <u>&gt;</u> er <u>&gt;</u>		
Sunday	2 - Regular City Commission meeting agenda	Tuesday	Wednesd	day Thursday	Friday 6 +	7 + FSU at UM
8	9 Regular City Commission meeting agenda	10 =	<u>  </u> +	12 +	13	I4 ≠ FSU at Duke
15 +	16 Regular City Commission meeting agenda	17 🖬	18 +	19 + Food Safety & Awareness	20	21 FSU at Virginia
22 +	23 + Regular City Commission meeting agenda	24 -	25 +	26 +	27 +	29 FSU at UM
29 +	30 th Regular City Commission meeting agenda	31 + Halloween Flu shot				

Рис. 9.14. Контент в каждой ячейке смещен к верхнему краю

00			CSS Cook	book		$\bigcirc$
Back Forward	Reload Stop Home	Print 1Password	(file:///Us	ers/christopher/Documents	/Work/ 🏠 🔻 -	Google Q
	COORDOOK CI	Christopher Schn	<u>≤ 201</u> <u>≤ 0ctob</u>	0 <u>&gt;</u> er <u>&gt;</u>		÷
Sunday	And ay 2 - Regular City Commission meeting agenda	Tuesday	Wednesd	ay Thursday	Friday	Saturday 7 ∓ FSU at UM
8	9 Regular City Commission meeting agenda	10		12	13 🖶	I4 = FSU at Duke
15	16 Regular City Commission meeting agenda	17 8	18 +	19 + Food Safety & Awareness	20	21 - FSU at Virginia
22 +	23 Regular City Commission meeting agenda	24 +	25 +	26	27 +	29 🖪 FSU at UM
29 +	30 Regular City Commission meeting agenda	31 - Halloween Flu shot				

```
font-family: monospace;
text-decoration: none;
float: left;
width: 1.5em;
height: 1.5em;
background-color: #ccc;
text-align: center;
}
```

00			CSS Cookbook	2		$\Box$
	. C 🗙 🍙		file:///Users/c	hristopher/Documents	/Work/ 🏫 🔻 - 猪	Google Q
Back Forward	i Reload Stop Home SS Cookbook 🛛 🕄 🎵	Christopher Schmi	tt : Designer, 🕲 🏾 🖉	Technology Books, Te	ch Confer 🕄 🕇 🕇	च
			≤ 2010 ≤ October	<u>&gt;</u>		
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
<u> </u>	Regular City Commission meeting agenda	3 [23	4	Dad's birthday	6 12	FSU at UM
8	9 + Regular City Commission meeting agenda	10 +	<u>11</u> 0	12	13	14 + FSU at Duke
15	16 Regular City Commission meeting agenda	17	18	19 Tood Safety & Awareness	20	21 + FSU at Virginia
22 +	23 + Regular City Commission meeting agenda	24	25	26	27	29 + FSU at UM
29 +	30 + Regular City Commission meeting agenda	31 Halloween Flu shot				
				,		

Рис. 9.16. Стилевое оформление незаполненных дат из следующего месяца

Вокруг текущего дня (в данном примере текущий день 27-е) поместите двухпиксельную черную рамку:

```
#today {
   border: 2px solid black;
}
```

И после этого календарь завершен.

# Проектирование Web-страниц, предназначенных для печати

## 10.0. Введение

Для создания версии Web-страницы, предназначенной для печати, традиционно Webразработчики вынуждены вручную преобразовывать ее контент, создавая отдельную упрощенную версию макета страницы, или использовать скрипт для динамического формирования отдельного макета.

Благодаря технологии CSS вы можете автоматически применять новую таблицу стилей к документам, которые нужно напечатать, таким образом экономя время и ресурсы сервера, необходимые для создания версии страницы, предназначенной для печати.

В наши дни поддержка таблиц CSS для печатающих устройств довольно широко распространена. Ее обеспечивают основные современные Web-обозреватели, включая Firefox, Internet Explorer для Windows, Safari, Chrome и Opera.

В этой главе приводятся основные сведения о том, как сообщить Web-обозревателю о таблице стилей, которую нужно использовать при отправке документа на печать. В ней также обсуждаются способы перехода от экранного стилевого оформления графики к предназначенному для печати и методы разработки документа для вывода на принтер.

#### Примечание

Поскольку книга сосредоточена на практической природе технологии CSS, пригодной для разных Web-обозревателей, примеры данной главы ориентированы на стилевое оформление страницы, а не подробное описание свойств CSS, предназначенных для вывода страницы на принтер. Дополнительную информацию об этих свойствах можно найти в книге Эрика Мейера (Eric A. Meyer) "Cascading Style Sheets: The Definitive Guide" (Каскадные таблицы стилей: подробное руководство) (издательство O'Reilly).

## 10.1. Применение к Web-странице таблицы стилей для печати

## Проблема

Вы хотите создать версию страницы, предназначенную для печати, без ручного формирования отдельного HTML-файла.

## Решение

Сначала создайте отдельную таблицу стилей, содержащую правила CSS, которые определяют желаемый внешний вид страницы при выводе на печать. В данном примере таблица стилей с правилами CSS, предназначенными только для вывода на печать, названа print.css.

Затем свяжите эту таблицу с документом и задайте значение print в атрибуте media:

```
<link rel="stylesheet" type="text/css" href="adv.css"
media="screen" />
<link rel="stylesheet" type="text/css" href="print.css"
media="print" />
```

## Обсуждение

Для создания таблицы стилей для печати закомментируйте экранную таблицу стилей и затем создайте отдельную дополнительную таблицу. В этой второй таблице стилей сформируйте правила, определяющие внешний вид страницы при выводе на печать. После формирования таблицы стилей свяжите ее с элементом link, как показано в данном решении.

#### Типы носителей

Вы можете применять таблицы стилей для представления документов на разнообразных устройствах вывода. По умолчанию у атрибута media установлено значение all. Если в элементе link не задан атрибут, агент пользователя применит правила CSS в таблице стилей ко всем носителям.

Несмотря на то, что самое часто встречающееся значение атрибута — screen, применяемое главным образом для отображения документов на экранах цветных мониторов, в спецификации CSS 2.1 определено десять типов носителей информации, перечисленных в табл. 2.5.

При определении стилей для вашей Web-страницы можно использовать одну таблицу для всех устройств вывода:

```
<link rel="stylesheet" type="text/css" href="uber.css" media="all" />
```

Или можно применить одну таблицу стилей для нескольких, но не всех устройств вывода.

Например, при использовании одной таблицы стилей для проектора и печатающего устройства отделите в атрибуте названия типов устройств вывода запятой:

```
<link rel="stylesheet" type="text/css" href="print.css"
media="print,projection" />
```

В приведенном коде для отображения документа с помощью проектора и принтера применяется таблица стилей из файла print.css. (Возможно это не идеальное решение, т. к. макет для печати может оказаться неподходящим для проектора.)

#### Использование правила @import при назначении устройств вывода

Для назначения устройств вывода помимо элемента link можно использовать другие методы. Один из них — приведенное далее правило @import, позволяющее назначить таблицу стилей для вывода документа с помощью проектора и принтера:

Правило @import необходимо поместить в элемент style или во внешнюю таблицу стилей. Однако поскольку Internet Explorer не отображает таблицы стилей для печати с помощью правила @import, лучше избегать его применения.

#### Применение правила @media при назначении устройств вывода

Еще один возможный метод связывания и определения таблиц стилей и типов устройств вывода — правило @media, позволяющее писать блоки правил CSS, предназначенные для разных устройств вывода и хранящиеся в *одной* таблице стилей:

```
<style type="text/css">
@media print {
   body {
    font-size: 10pt;
   background-color: white;
   color: black;
   }
}
@media screen {
   font-size: medium;
   background-color: black;
   color: white;
   }
}
</style>
```

#### См. также

Section 7 (секция 7) "Media Types" (типы носителей информации) в CSS 2.1 Working Draft (рабочий проект CSS 2.1) на странице http://www.w3.org/TR/CSS21/media.html.

## 10.2. Замена цветного логотипа черно-белым при печати Web-страниц

## Проблема

Вы хотите заменить цветной логотип на более подходящий для печати, не вставляя двух изображений логотипа в HTML-документ и не создавая отдельной версии Webстраницы, предназначенной для печати.

## Решение

Вставьте в Web-документ следующий HTML-код, добавляющий черно-белый логотип:

```
<div id="header">
<h1><a href="/"><img src="e4h_logo-print.gif" /></a></h1>
</div><!-- /#header -->
```

Затем исключите вывод черно-белого логотипа в Web-обозревателе:

```
<style type="text/css" rel="stylesheet" media="screen">
#header h1 img {
    display: none;
    }
</style>
```

Далее вставьте цветной логотип как фон элемента h1 (рис. 10.1):

```
<style type="text/css" rel="stylesheet" media="screen">
#header h1 img {
    display: none;
    }
#header h1 a {
    display: block;
    background-image: url(e4h_logo.gif);
    background-repeat: no-repeat;
    width: 494px;
    height: 85px;
    }
</style>
```



Рис. 10.1. Цветной логотип, вставленный с помощью свойства background-image

Поскольку эта таблица стилей предназначена для вывода на экран, Web-обозреватель игнорирует экранные правила CSS и в режиме предварительного просмотра перед печатью отображает черно-белый логотип, как показано на рис. 10.2.

## Обсуждение

В этом приеме используется основной метод замещения изображения, описанный в *разд. 4.20*. Вместо удаления HTML-текста изображение, предназначенное для печати, вставляется вместо изображения более подходящего для полноцветного отображения.

Такая замена изображений действует благодаря заданию специальных правил, основанных на используемых типах устройств вывода. Если в исходном фрагменте таблицы



Рис. 10.2. Черно-белый логотип, который будет напечатан

CSS вы зададите тип устройства screen, Web-обозреватель, когда начнет подготовку документа к печати, проигнорирует правила CSS, скрывающие черно-белое изображение.

#### Предупреждение

Если в вашей основной таблице стилей вы не разделили правила CSS в соответствии с типом устройства вывода, Web-обозреватель считает, что тип устройства равен значению all. В этом случае любые дополнительные правила CSS, предназначенные для вывода на принтер, смешиваются с другими правилами CSS, что может привести к нежелательным результатам при печати документа, т. к. каскадная таблица попытается определить внешний вид страницы во время вывода на принтер. Таким образом, создавая таблицу стилей, предназначенную только для печати, убедитесь в том, что вы задали ваши стили для корректного типа устройства вывода.

#### См. также

Статью "CSS Logo replacement" (Замещение логотипа средствами CSS) в блоге по адреcy http://www.ibloomstudios.com/articles/css\_logo\_replacement/.

## 10.3. Создание готовой к печати Web-формы

#### Проблема

Вам нужна форма, подобная приведенной на рис. 10.3, которую пользователи смогут заполнить в интерактивном режиме или напечатать и заполнить после отключения от сети.

#### Решение

Сначала создайте таблицу стилей для вывода на принтер и включите в нее селектор класса, позволяющий трансформировать элементы формы таким образом, чтобы выводился текст черного цвета и рисовалась внизу граница толщиной 1 пиксел.

Например, рассмотрим следующий HTML-код текстового элемента input:

```
<label for="fname">First Name<label>
<input class="fillout" name="fname" type="text" id="fname" />
```

000	CSS Cookbook – Mozilla	$\bigcirc$
🔺 🖆 Home 📑 Boo	kmarks 🛇 oreilly.com 🛇 christopher.org	
Order For	m	
First Name:		
Last Name:		
Email:		
Address:		
City:		
State/Province:	Select 💌	
Zip Code:		
Daytime Phone:		
Product(s):	□ EZWeb (\$19.95) □ iNtroduction Ping (\$29.95)	
Type of Credit Card:	← Mastercard ← Visa ← Discover	
Name on Credit Card:		
Card Number:		
Card Expiration Date:		
Submit		
	Done	II- f /

Рис. 10.3. Интерактивная форма

Для стилевого оформления элемента формы требуется следующее правило CSS:

```
<style type="text/css" media="print ">
.fillout {
  color: black;
  border-width: 0;
  border-bottom: 1px solid #000;
  width: 300pt;
}
</style>
```

В раскрывающихся меню скройте полностью элемент select и вставьте дополнительную разметку для создания нижней границы:

```
<lpre><label for="bitem">Breakfast Item</label>
<select name="bitem" size="1">
<option selected="selected">Select</option>
<option>Milk</option>
<option>Eggs</option>
<option>Orange Juice</option>
<option>Newspaper</option>
</select><span class="postselect"> </span>
```

Затем в правилах CSS превратите внутристрочный элемент span в блочный элемент. Это позволит задать ширину элемента span и поместить снизу границу, такую же, как у элементов input в предыдущем правиле CSS:

```
<style type="text/css" media="print">
select {
display: none;
}
.postselect {
display: block;
width: 300pt;
height: lem;
border: none;
border-bottom: 1px solid #000;
}
</style>
```

Для элементов, таких как кнопка **Submit**, которые нельзя использовать на напечатанной странице, задайте свойство display со значением none. Окончательный вариант формы показан на рис. 10.4.

000	CSS Cookbook – Mozilla	$\Box$
🔺 🛍 Home 🛅 Bookmarks	🛇 oreilly.com 🛇 christopher.org	
Order Form		
First Name:		
Last Name:		
Email:		
Address:		
City:		
State/Province:		
Zip Code:		
Daytime Phone:		
Product(s):	□ EZWeb (\$19.95) □ iNtroduction Ping (\$29.95)	
Type of Credit Card:	C Mastercard C Visa C Discover	
Name on Credit Card:		
Card Number:		
Card Expiration Date:		
Done 🛛 🖓 🗍 🖾 Done		- <b></b>

Рис. 10.4. Та же форма, подготовленная для печати

## Обсуждение

В форме заказа линии, формируемые таблицей стилей для печати, подсказывают пользователям, что они могут заполнить форму. С помощью свойства border такие линии легко создать в любом Web-обозревателе, сделав Web-формы полезными, как в интерактивном, так и в автономном режиме. К элементам select применяется искусственный прием, который противоречит идеальной осмысленной разметке, но на самом деле он действует и представляет собой образец корректного HTML-кода. Поместите элемент span после элемента select:

```
<select name="bitem" size="1">
  <option selected="selected">Select</option>
  <option>Milk</option>
  <option>Eggs</option>
  <option>Orange Juice</option>
  <option>Newspaper</option>
  </select>
  <span class="postselect"> </span>
```

Затем отмените отображение элемента select:

```
select {
  display: none;
}
```

Далее задайте вывод элемента span как блочного, сделав таким образом доступными свойства width и height. Вместе со свойствами width и height можно задать нижнюю границу в полном соответствии с нижними границами других элементов формы:

```
.postselect {
  display: block;
  width: 300pt;
  height: 1em;
  border: none;
  border-bottom: 1px solid #000;
}
```

#### Применение селекторов атрибута для разделения элементов формы

Селекторы атрибутов из спецификации CSS облегчают стилевое оформление форм, предназначенных для печати. При использовании селекторов атрибутов указать, к каким элементам формы применить стили легче, чем вставлять в разметку атрибуты class и их соответствующие значения.

В приведенном далее коде первое правило CSS применяется только к текстовым элементам input, в то время как второе правило скрывает кнопку **Submit** и раскрывающееся поле **Select**:

```
input[type="text"] {
  color: black;
  border-width: 0;
  border: 1px solid #000;
}
input[type="submit"], select {
  display: none;
}
```

#### Примечание

К счастью, в настоящее время большинство современных Web-обозревателей поддерживают селекторы атрибута; не делает этого Internet Explorer 6.

#### Будьте внимательны к пользователю

Поскольку теперь форма печатается, посетители сайта не могут воспользоваться кнопкой **Submit** для передачи своей информации. Обязательно укажите, какие шаги они должны предпринять после заполнения формы.

Например, если вы хотите, чтобы форму отправили по почте, на страницу с напечатанной формой вставьте нужный почтовый адрес, как показано далее:

```
<div class="print">
```

Обратите внимание на то, что операторы заключены в элемент div с селектором class, у которого установлено значение print. В таблице стилей для вывода на экран задайте в свойстве display для этого конкретного класса значение none:

```
<style type="text/css" media="screen">
.print {
    display: none;
    }
</style>
```

При наличии отдельной таблицы стилей для печати разрешите отображение этих операторов, установив в свойстве display значение block:

```
<style type="text/css" media="print">
.print {
    display: block;
    }
</style>
```

#### См. также

Документацию о селекторах атрибутов в спецификации консорциума W3C на странице http://www.w3.org/TR/CSS21/selector.html#attribute-selectors; тег label в спецификации языка HTML 4.01 на странице http://www.w3.org/TR/html401/interact/ forms.html#edef-LABEL.

## 10.4. Отображение URI после ссылок

## Проблема

Вам необходимо при печати страницы выводить URI (Uniform Resource Identifier, универсальный идентификатор ресурса) ссылок в статье.

## Решение

Для того чтобы отдать распоряжение Web-обозревателю печатать URI ссылок, встречающихся в абзаце, используйте псевдоэлемент :after:

```
p a:after {
    content: " <" attr(href) "> " ;
}
```

## Обсуждение

Селекторные структуры, такие как :after, называют псевдоэлементами. Web-обозреватель трактует такие селекторы как дополнительные элементы, применяемые для разметки Web-документа.

Например, с помощью следующего правила CSS вы можете сделать первую букву абзаца высотой 2 em:

```
p:first-letter {
  font-size: 2em;
}
```

Селектор :after (или :before) применяется для вставки генерируемого содержимого после (или до) элемента. В данном примере значение атрибута href, содержащее сведения об URI, помещается после каждого элемента anchor, включенного в элемент р.

Для вставки угловых скобок, охватывающих URI, задайте знаки угловых скобок в кавычках. Для добавления пробелов между элементом anchor и следующим элементом в строке вставьте один пробел перед левой угловой скобкой и еще один после правой угловой скобки. Затем вставьте URI с помощью функции attr(x). Если х заменить именем любого атрибута, CSS найдет этот атрибут в элементе и вернет его значение как строку.

Другой пример, демонстрирующий возможности описываемого псевдоэлемента, включает возврат значений аббревиатур и акронимов в документ, изобилующий специальными терминами.

The W3C makes wonderful things like CSS!

Для этого первым делом поместите полную форму слова или поясняющую фразу в атрибут title элементов abbr или acronym:

```
The <acronym title="World Wide Web Consortium">W3C</acronym>
makes wonderful things like <abbr title="Cascading Style
Sheets">CSS</abbr>!
```

Затем в правилах CSS сообщите Web-обозревателю о необходимости вернуть значение атрибута title:

```
abbr:after, acronym:after {
  content: " (" attr(title) ") ";
}
```

#### Вставка имени домена перед абсолютными ссылками

Во внутренних абсолютных ссылках (absolute link) (указывающих на корневой каталог сайта) на напечатанной странице будут указаны только прямой слэш (/) и сведения о папке и файле. Для выхода из этого затруднительного положения спецификация CSS3 предлагает решение с использованием селектора подстроки:

```
p a:after {
  content: " <" attr(href) "> ";
}
p a[href^="/"]:after {
  content: " <http://www.csscookbook.com" attr(href) "> " ;
}
```

Знак вставки ^ свидетельствует о том, что селектор выбирает все ссылки, начинающиеся с прямого слэша, который в свою очередь обозначает абсолютную ссылку.

#### Известные проблемы Web-обозревателей

В настоящее время генерацию содержимого с помощью псевдоэлементов поддерживают только Web-обозреватели Firefox, Chrome и Safari. Корпорация Microsoft ввела поддержку псевдоэлементов :after и :before в IE8.

#### См. также

*Разд. 3.6* о задании шрифта в Web-документе; генерируемое содержимое в спецификации CSS 2.1 на странице http://www.w3.org/tr/rec-css2/generate.html#content.

## 10.5. Вставка специальных символов перед ссылками

#### Проблема

Вы хотите вставить перед ссылкой специальные символы, например знак », в таблицу стилей, предназначенную для печати страницы.

#### Решение

Убедитесь, что ваша таблица стилей предназначена для вывода на печатающее устройство, используйте псевдоэлементы :after (или :before) для указания URI после ссылки в Web-документе:

```
p a:after {
   content: attr(href);
}
```

Далее поместите перед ссылкой шестнадцатеричный код специального символа:

```
p a:after {
  text-decoration: underline;
  content: " \00BB " attr(href);
}
```

Когда страница будет напечатана, текст после ссылки может выглядеть следующим образом

» http://www.csscookbook.com/

## Обсуждение

Не забудьте вставить обратный слэш при включении шестнадцатеричного кода как Escape-последовательности, чтобы Web-обозреватель не выводил шестнадцатеричное значение как обычный текст. В данном случае, если не пометить шестнадцатеричное значение символа открывающих кавычек-"елочек" как Escape-последовательность, вместо них будет выведен текст "00BB":

00BB http://www.csscookbook.com/

В соответствии с синтаксическими правилами технологии CSS нельзя использовать в свойстве content числа и специальные имена HTML для обозначения специальных символов. Такие символы следует задавать как Escape-последовательности: с помощью знака обратного слэша и их шестнадцатеричного кода.

Специальные символы в свойстве content можно задавать не только при выводе страницы на печать. Попробуйте применить их в экранной презентации вашего дизайнпроекта. Для просмотра полученной страницы обязательно включите в вашу таблицу стилей объявление CSS с правилом media, в котором установлено значение all или screen.

#### Известные проблемы Web-обозревателей

В настоящее время генерацию содержимого с помощью псевдоэлементов поддерживают только Web-обозреватели Firefox, Mozilla, Chrome и Safari. Генерация содержимого работает в Internet Explorer 8 для Windows.

#### См. также

Список специальных символов и их шестнадцатеричных кодов на странице http://www.ascii.cl/htmlcodes.htm; обзор специальных символов в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/syndata.html#escaped-characters.

## 10.6. Вставка в документ для печати разрывов страниц

## Проблема

Вы хотите при выводе на печать длинного Web-документа вставить в него разрывы страниц, как показано на рис. 10.5.



Рис. 10.5. Стандартное отображение страницы при печати

## Решение

Используйте свойство page-break-before для обозначения перехода документа на следующую страницу при печати (рис. 10.6):

```
h3 ~ h3 {
  page-break-before: always;
}
```

## Обсуждение

Если в вашем документе применена семантическая разметка, вставить в Web-документ разрывы страниц очень просто.

В коде решения используется комбинированный селектор соседнего элемента. Правило утверждает, что каждый раз, когда элементу h3 предшествует другой элемент h3, следует выполнить разрыв страницы. Другими словами, практически каждый элемент h3 будет находиться вверху печатной страницы, начинающейся со второго элемента h3 в строке.

#### Применение селекторов класса

Предложенное решение работает, поскольку используется структурированный документ с семантической разметкой и Web-обозреватель, распознающий селекторы CSS3. Но при работе с документом без семантической разметки определить местоположение разрывов страниц в HTML-коде все равно довольно легко.



Рис. 10.6. Разрывы страниц, включенные в печатаемый документ

Сначала создайте селектор класса, содержащий свойство page-break-before:

```
.pageBreak {
  page-break-before: always;
}
```

Затем вставьте правило в те места контента, перед которыми хотите создать разрыв страницы:

```
 ...
```

#### См. также

Описание "Page Breaks" (Разрывы страниц) в Working Draft CSS3 (рабочий проект CSS3) на странице http://kent.w3.org/TR/css3-page/#page-breaks.

## 10.7. Пример дизайна: версия страницы для печати, созданная средствами CSS

В данном примере вид существующего Web-документа (рис. 10.7) будет преобразован в более подходящий для печати.

Помимо того, что технология CSS изменила способ разработки дизайна для Web, она также позволила разработчикам по-другому формировать версии документов, предна-

значенные для печати. Вместо того чтобы создавать отдельные страницы или писать скрипты, вы можете применить CSS для создания документа, предназначенного для печати, как только пользователь щелкнет мышью кнопку **Print** (Печать). В этой книге не рассматривается HTML-код для создания страницы, поскольку чудесные свойства технологии CSS позволят нам изменить представление страницы без каких-либо изменений HTML-разметки.



Рис. 10.7. Web-страница со стилевым оформлением, предназначенным для вывода на экран

Создавая таблицу стилей для печати, вы фактически используете Web-обозреватель. Это позволяет быстро увидеть воздействие правил CSS на отображение документа (такое же, как при выводе на печать), но гораздо более легким способом и с экономией денег, т. к. не тратятся чернила (или порошок) для вывода на принтер. Итак, заключите в знаки комментариев таблицу стилей, используемую для вывода на экран, для того чтобы создать новые правила CSS:

```
<!-- Скройте экранную таблицу стилей, работая над таблицей CSS
для печати -->
<!-- link href="adv/css" type="text/css" rel="stylesheet"
media="screen -->
<style type="text/css">
/* Здесь находятся правила CSS для вывода на печать*/
</style>
```

#### Задание печати черно-белой страницы

Первое правило CSS примените к элементу body. В этом правиле установите фоновый цвет белый, а цвет шрифта черный:

```
body {
  background-color: white;
  color: black;
}
```

Затем задайте гарнитуру шрифта для печати страницы, выбрав шрифт с засечками (serif). Чтение с экрана текста, набранного шрифтом без засечек (sans-serif), легче для глаз, но для печатных страниц основным все еще остается шрифт с засечками. Во избежание неприятностей в дальнейшем вам, возможно, подойдет для печати документов гарнитура Times, поскольку она установлена на большинстве компьютеров (если не на всех). Если же у ваших пользователей нет установленной гарнитуры Times, обеспечьте альтернативу:

```
body {
  background-color: white;
  color: black;
  font-family: Times, "Times New Roman", Garamond, serif;
}
```

Теперь вы должны избавиться от навигационных ссылок и других страничных элементов, которые не хотите видеть в окончательном варианте для печати. К ним относятся навигационная полоса под главным заголовком, а также внутренние ссылки на самой странице. Если на вашей странице есть рекламные баннеры, хорошо бы и их скрыть (рис. 10.8):

```
#navigation, hr, body>div>a, #blipvert {
    display: none;
}
```

#### Дизайн главного заголовка

Поскольку вы имеете дело с черным и серым шрифтом на белой странице, у вас есть несколько вариантов для создания стилевого оформления внешнего вида главного заголовка страницы. Но даже имеющимися в распоряжении средствами очень легко создать заголовки, которые привлекут к себе внимание.

Прежде всего, задайте черный фон и белый шрифт:

```
#header h1 {
  color: white;
  background-color: black;
}
```

000	CSS Cookbook – Mozilla	$\bigcirc$
🔺 🟦 Home 🖹 Bookma	ks 🛇 oreilly.com 🛇 christopher.org	
<u>Christoph</u>	er Schmitt	
Color Your V	Veb	
by Christopher S	hmitt	
Your color scheme of theory and schemes	an make or break your web site. In this article, we examine the basics of color to make color work in web development.	
Color is the first item	on which I judge a site, before either reading its offerings or moving on.	=
Yes, it's not the site's not how tightly you c Shockwave, ActiveX	content that gets you past the front door. It's not how you broke down the architecture. It impressed the graphics. It's not what new technology you've thrown at me - be it Java, ChromeEffects or <u>Kai's</u> latest Photoshop filter. It's your use of color.	ťs
Better use of colors n the exit tunnel, you'll and forcing <u>Web-safe</u> out there today. With	eans a better Web page or site. If you can carry a color scheme from the splash page to uutomatically create a cohesive look and feel for your sites. By mixing hexadecimal tags <u>colors</u> to do your bidding, you'll earn one more little edge over a majority of Web pages Web page production always exploding on the Internet, isn't that enough of a reason?	;, ;
So, lest we simply sla bit more about Color	o down colors or steal schemes from other Web sites for the rest of our days, let's learn a Theory.	ι
Before you can learn vocabulary words is i	now to "mix and match" colors and know more about color theory, a foundation of a fer a order.	~
	ne -III-	

Рис. 10.8. Исключение полосы навигации и других элементов

Поскольку вы хотите, чтобы люди прочли заголовок, текст нужно сделать белым для обеспечения достаточной контрастности. В данном примере главный заголовок действует еще и как устройство самонаведения — он служит ссылкой на начальную страницу. Следовательно, цвет заголовка диктуется стилевыми правилами, заданными для ссылок. Для исправления ситуации добавьте отдельное правило:

```
#header h1 {
  background-color: black;
}
#header h1 a {
  color: white;
}
```

Теперь, когда текст виден, добавьте немного декоративного оформления для того, чтобы текст заголовка выделялся. Ваша задача — центрировать текст, увеличить размер шрифта и сделать все буквы прописными:

```
#header h1 {
  background-color: black;
  font-size: 24pt;
  text-align: center;
  text-transform: uppercase;
}
```

Несмотря на то, что теперь заголовок выглядит хорошо, его внешний вид можно еще улучшить, если изменить гарнитуру на шрифт без засечек (для того чтобы он отличался

от остального текста документа) и добавить отступы сверху и снизу от заголовка (рис. 10.9):

```
#header h1 {
   background-color: black;
   font-size: 24pt;
   text-align: center;
   font-family: Helvetica, Verdana, Arial, sans-serif;
   padding: 7pt;
   text-transform: uppercase;
}
```



Рис. 10.9. Стилевое оформление главного заголовка

## Стилевое оформление заголовка статьи и строки с именем автора

Для заголовка статьи и строки с именем автора выберите более эффектный дизайн, обнулив поля и отступы элементов h2 и h3:

```
#content h2 {
   padding: 0;
   margin: 0;
}
```

```
#content h3 {
   padding: 0;
   margin: 0;
}
```

Затем увеличьте размер шрифта в заголовке статьи и создайте тонкую линию под ним. Далее выровняйте текст с именем автора по правому краю и задайте курсивное начертание шрифта (рис. 10.10):

```
#content h2 {
  padding: 0;
  margin: 0;
  font-size: 20pt;
  border-bottom: 1px solid black;
}
#content h3 {
  padding: 0;
  margin: 0;
  text-align: right;
  font-style: italic;
}
```



Рис. 10.10. Дизайн заголовка статьи и строки с именем автора

### Привлечение внимания с помощью броской аннотации

Под строкой с именем автора расположен элемент h4. Поскольку он служит для статьи рекламной приманкой (teaser), его следует сделать отличающимся визуально от текста статьи. Для этого задайте для цвета фона примерно 30 % черного, измените гарнитуру шрифта на шрифт без засечек и вставьте дополнительный отступ (рис. 10.11):





Рис. 10.11. Стилевое оформление рекламной аннотации к статье

Что касается содержимого статьи, оставьте текст практически без изменений за исключением двух параметров: межстрочного интервала, о котором речь пойдет в этом разделе, и ссылок, о которых написано в следующем разделе.

Напоминаю, что в элементе body для всей страницы задана гарнитура шрифта с засечками и благодаря наследованию этот же тип шрифта выбран и для элементов-абзацев. Но возможно вам захочется раздвинуть строки или увеличить *межстрочный интервал* (leading) текста в абзаце. Для этого измените значение свойства line-height:

```
#content p {
   line-height: 18pt;
}
```

500

### Отображение URI после ссылок

При печати страницы любые ссылки в статье становятся бессмысленными. Для того чтобы сделать их полезными читателю, когда страница напечатана, добейтесь вывода всех URI ссылок. Для этого задайте правило CSS, отображающее URI после каждой ссылки в разделах документа. Также для достижения визуального эффекта удалите стандартное подчеркивание ссылок, задайте жирное начертание и серый цвет шрифта для них (рис. 10.12):

```
#content a:after {
  content: " <" attr(href) "> ";
  font-family: courier, monospace;
  font-weight: normal;
  }
  a {
   text-decoration: none;
   font-weight: bold;
   color: #626466;
  }
```

000 CSS Cookbook - Mozilla  $\bigcirc$ 🐔 Home 🛛 Bookmarks 🛇 oreilly.com 🛇 christopher.org **CHRISTOPHER SCHMITT** Color Your Web by Christopher Schmitt Your color scheme can make or break your web site. In this article, we examine the basics of color theory and schemes to make color work in web development. Color is the first item on which I judge a site, before either reading its offerings or moving on. Yes, it's not the site's content that gets you past the front door. It's not how you broke down the architecture. It's not how tightly you compressed the graphics. It's not what new technology you've thrown at me - be it Java, Shockwave, ActiveX, ChromeEffects or Kai's <http://www.edge.org/3rd\_culture/bios/krause.html> latest Photoshop filter. It's your use of color. Better use of colors means a better Web page or site. If you can carry a color scheme from the splash page to the exit tunnel, you'll automatically create a cohesive look and feel for your sites. By mixing hexadecimal tags, and forcing Web-safe colors <http://www.lynda.com/hex.html> to do your bidding, you'll earn one more little edge over a maiority of Web nages out there today. With Web nage production always

Рис. 10.12. Корректировка стиля ссылок и межстрочного интервала в основном тексте статьи

#### Нижняя часть страницы — "подвал"

Теперь вы готовы заняться "подвалом", или нижним колонтитулом страницы (footer), содержащим сведения об авторских правах. Поскольку для главного заголовка выбрана гарнитура шрифта без засечек, сбалансируйте дизайн страницы, отцентрировав сведения об авторских правах, проведя горизонтальную линию с помощью свойства border-top и задав гарнитуру шрифта без засечек (рис. 10.13):

<pre>#footer {</pre>				
border-top: 1	lpx solid #(	;000		
text-align: d	center;			
font-family:	Helvetica,	Verdana,	Arial,	<pre>sans-serif;</pre>
}				



Copyright © 2003 Christopher Schmitt. All rights reserved.

Done 🖾 🖓 🗍 🖓 Done

Рис. 10.13. Стилизованный нижний колонтитул

Завершив формирование правил CSS для печати, скопируйте их и поместите во внешнюю таблицу стилей с именем файла print.css. Затем удалите комментарии из таблицы

CSS для экрана и свяжите с документом таблицу CSS для печати с помощью элемента link:

```
<link href="adv.css" type="text/css" rel="stylesheet"
media="screen" />
<link href="print.css" type="text/css" rel="stylesheet"
```

media="print" />

Теперь вы умеете создавать таблицу стилей, предназначенную для печати документа. Если у посетителей вашего сайта есть Web-обозреватель, способный отображать таблицы стилей для печати, они автоматически получат подходящий для их принтеров макет, когда решат напечатать страницу.

## глава 11 Макеты страниц

## 11.0. Введение

Один из последних рубежей дизайна с применением технологии CSS — создание макета страницы.

Долгое время Web-дизайнеры пользовались для формирования многоколоночных и многоуровневых макетов HTML-таблицами, часто вложенными.

Новый подход, возлагающий выполнение тяжелой работы на CSS, дает множество преимуществ. Информация, когда-то заключавшаяся в огромное множество вложенных таблиц и изображений, теперь хранится в смысловых тегах заголовков и абзацев, что улучшает критерии ранжирования поисковыми машинами.

Кроме того, заметно уменьшаются размеры файлов, как и проблемы сопровождения и технической поддержки. Полное перепроектирование Web-сайта, занимавшее обычно часы, а иногда и дни при использовании HTML-таблиц, становится легкой прибыльной работой благодаря технологии CSS.

В этой главе обсуждаются многочисленные возможные способы создания макетов, состоящих из колонок, включая простые одноколоночные макеты, макеты из четырех колонок и все промежуточные варианты.

## 11.1. Построение одноколоночного макета

## Проблема

Вы хотите создать макет, состоящий из одной основной колонки, как показано на рис. 11.1.

## Решение

Задайте процентное значение левого и правого полей элемента body в Web-документе:

```
body {
  margin-left: 15%;
  margin-right: 15%;
}
```
00	CSS Cookbook	$\Box$
Back Forward R	C X A I IP	- 🚱 GooglQ
CSS Cool	kbook 🛛 😧 👔 Christopher Schmitt : Designer, 🕄 👰 Technology Books, Tech Confer 🕄	+ =
	One Column Layout	-
	Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maccenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.	
	Quisque facilisis erat a dui. Nam malesuada ornare dolor. Cras gravida, diam sit amet rhoncus ornare, erat elit consectetuer erat, id egestas pede nibh eget odio. Proin tincidunt, velit vel porta elementum, magna diam molestie sapien, non aliquet massa pede eu diam. Aliquam iaculis. Fusce et ipsum et nulla tristique facilisis. Donec eget sem sit amet ligula viverra gravida. Etiam vehicula urna vel turpis. Suspendisse sagittis ante a urna. Morbi a est quis orci consequat rutrum. Nullam egestas feugiat felis. Integer adipiscing semper ligula. Nunc molestie, nisl sit amet cursus convallis, sapien lectus pretium metus, vitae pretium enim wisi di lectus. Donec vestibulum. Etiam vel nibh. Nulla facilisi. Mauris pharetra. Donec augue. Fusce ultrices, neque id dignissim ultrices, tellus mauris dictum elit, vel lacinia enim metus eu nunc.	
	Proin at eros non eros adipiscing mollis. Donec semper turpis sed diam. Sed consequat ligula nec tortor. Integer eget sem. Ut vitae enim eu est vehicula gravida. Morbi ipsum ipsum, porta nec, tempor id, auctor vitae, purus. Pellentesque neque. Nulla luctus erat vitae libero. Integer nec enim. Phasellus aliquam enim et tortor. Quisque aliquet, quam	•

Рис. 11.1. Страница с одной колонкой, усиленная полем большого размера

# Обсуждение

Когда вы задаете значение в процентах для левого и правого полей элемента body, ширина колонки становится настраиваемой. Это позволяет содержимому занимать всю ширину окна Web-обозревателя пользователя.

#### Создание колонки фиксированной ширины

Этот прием выравнивает колонку по левому краю окна Web-обозревателя пользователя. Если вы хотите центрировать колонку с фиксированной шириной, заключите все содержимое Web-документа в элемент div с особым уникальным атрибутом id, таким как easel:

```
<div id="easel">
[...]
</div>
```

Задайте свойство width для селектора id:

```
#easel {
  width: 600px;
}
```

Затем примените свойство с сокращенной формой записи margin со значениями auto для левой и правой сторон элемента div:

```
#easel {
  width: 600px;
  margin: 0 auto;
}
```

### См. также

*Разд. 5.4* о центрировании элементов в Web-документе; *разд. 7.12* о горизонтальном навигационном меню из закладок.

# 11.2. Создание двухколоночного макета

# Проблема

Вы хотите создать двухколоночный макет с колонками, меняющими ширину в соответствии с шириной окна Web-обозревателя, как показано на рис. 11.2.



Рис. 11.2. Двухколоночный макет, полученный с помощью CSS

# Решение

Сначала разметьте содержимое с помощью элементов div, используя атрибуты id с соответствующими значениями. Осмысленные значения, например mainContent или sidebar, предпочтительнее значений, указывающих на местоположение информации на странице.

Применяемые значения атрибутов id указывают на местоположение отображаемого контента *только для наглядности*, чтобы помочь вам лучше понять из разметки, как предполагается разместить контент:

```
<div id="columnLeft">
[...]
</div>
<div id="columnRight">
[...]
</div>
<div id="footer">
[...]
</div>
```

Затем в правилах CSS воспользуйтесь свойством float для перемещения содержимого левой колонки влево и задания ширины, равной двум третям ширины Web-документа:

```
#columnLeft {
  float: left;
  width: 67%;
  background: #fff;
  margin-top: 0;
  margin-right: 1.67em;
  border-right: 1px solid black;
  padding-top: 0;
  padding-right: 1em;
  padding-bottom: 20px;
}
```

Правая колонка обтекает содержимое левой колонки. Установка нулевого значения для ее поля и отступа сверху позволяет правой колонке и первому элементу в ней располагаться на одном уровне с левой колонкой:

```
#columnRight {
   padding-left: 2em;
   margin-top: 0;
   padding-top: 0;
}
h1 {
   margin-top: 0;
   padding-top: 0;
}
```

Для отображения колонтитула страницы (footer) в нижней части Web-документа задайте в свойстве clear значение both:

```
#footer {
    clear: both;
    padding-bottom: 1em;
    border-top: 1px solid #333;
    text-align: center;
}
```

## Обсуждение

Свойство float (см. *paзd*. 2.21) подобно применяемому в языке HTML атрибуту align, который задает обтекание текста или других элементов вокруг изображения:

```
<img src="this.jpg" width="250" height="150" hspace="7" vspace="7" alt="example" align="right" />
```

После того как задано выравнивание изображения по левому или правому краю, содержимое обтекает его со стороны, противоположной направлению выравнивания изображения.

Например, выравнивание изображения по правому краю заставляет контент обтекать его с левой стороны. В правилах CSS свойство float выполняет ту же функцию, за исключением того, что оно предлагает средства более тонкого управления отображением благодаря использованию рамок, полей, отступов и других свойств.

Для того чтобы гарантировать размещение служебной информации под колонками, задайте свойство clear со значением both. Такая установка информирует Webобозреватель о том, что содержимое заключительной или нижней части страницы не обтекает включенную в плавающую модель левую колонку и его следует поместить ниже или (за пределами) любого элемента из плавающей модели.

Единственное условие применения этого метода создания двухколоночного макета — наличие в левой колонке контента большего объема по сравнению с правой. Поскольку содержимое левой колонки выводится в документе первым, контент правой колонки обтекает левую. Контент слишком большого объема в правой колонке перестает быть свободно перемещаемым и приводит к аномалии, отображенной на рис. 11.3.

Устранить эту проблему можно корректировкой значения левого поля или отступа правого элемента таким образом, чтобы по крайней мере сохранялась ширина колонки после того, как контент начнет обтекать плавающий элемент снизу:

```
#mainColumn {
  width: 400px;
  /* Отступ, достаточный для учета ширины левой колонки */
  padding-left: 200px;
  }
  #navigation {
  float: left;
  width: 175px;
  }
```

#### Примечание

Как альтернативу можно задать свойство overflow со значением hidden (см. разд. 2.22):

```
#mainColumn {
   overflow: hidden;
}
```

#### Зеркальное отражение макета

Если вы хотите поменять колонки местами, измените порядок отображения колонок с помощью следующей разметки:

```
<div id="columnRight">
[...]
</div>
```

```
<div id="columnLeft">
[...]
</div>
<div id="footer">
[...]
</div>
```

#### Примечание

Если бы я назвал атрибуты в соответствии с семантической разметкой, а не указывал бы в их названиях на предполагаемое местоположение контента в окне Web-обозревателя, я мог бы просто изменить правила CSS, сместив плавающие элементы в противоположных направлениях.



Footer: Proin at eros non eros adipiscing mollis.

Затем примените к колонкам следующие правила CSS:

```
#columnRight {
  float: right;
  width: 67%;
  padding-bottom: 20px;
  padding-top: 0;
 }
#columnLeft {
  width: 29%;
  padding-right: 1em;
  border-right: 1px solid black;
  padding-top: 0;
}
```

### См. также

*Разд. 11.3* о построении двухколоночного макета с фиксированной шириной колонок; предпосылки для этого решения в статье Джеффри Зельдмана (Jeffrey Zeldman) "From Table Hacks to CSS Layout: A Web Designer's Journal" (От хитростей с использованием таблиц к верстке с помощью CSS: журнал Web-дизайнера) на странице http://www.alistapart.com/articles/journey/.

# 11.3. Построение двухколоночного макета с колонками фиксированной ширины

# Проблема

Вы хотите создать макет страницы с двумя колонками фиксированной ширины.

# Решение

Сначала разметьте контент с помощью элементов div, используя атрибуты id, содержащие соответствующие значения, представляющие расположение содержимого на странице:

```
<div id="header">
[...]
</div>
<div id="columnLeft">
[...]
</div>
<div id="columnRight">
[...]
</div>
<div id="footer">
[...]
</div>
```



Рис. 11.4. Двухколоночный макет, реализованный средствами CSS

С помощью свойства float задайте ширину левой колонки в единицах длины, а не в процентах. Установите также ширину всего документа в единицах длины (рис. 11.4):

```
body {
  margin: 0;
  padding: 0;
  font-family: Georgia, Times, "Times New Roman", serif;
  color: black;
  width: 600px;
  border-right: 1px solid black;
  }
  #header {
   background-color: #666;
   border-bottom: 1px solid #333;
  }
  #columnLeft {
   float: left;
  }
}
```

```
width: 160px;
margin-left: 10px;
padding-top: 1em;
}
#columnRight {
  padding-top: 1em;
margin: 0 2em 0 200px;
}
#footer {
  clear: both;
  background-color: #ccc;
  padding-bottom: 1em;
  border-top: 1px solid #333;
  padding-left: 200px;
}
```

# Обсуждение

По умолчанию элементы блочного уровня растягиваются до ширины их контейнера. Если окно Web-обозревателя мало, элементы блочного уровня сжимаются — иными словами, текст из контента помещается в узкие рамки.

Если же вы используете *пикселы* вместо процентов, ширина колонок становится фиксированной. Даже если окно Web-обозревателя сжимается или расширяется, ширина колонок остается неизменной.

Для того чтобы сохранить ширину левой колонки фиксированной, а ширине основной колонки дать возможность увеличиваться, просто удалите свойство width, назначенное элементу body.

#### Зеркальное отражение макета

Если вы хотите отобразить колонки в обратном порядке, реорганизуйте документ с помощью следующей разметки:

```
<div id="header">
[...]
</div>
<div id="columnRight">
[...]
</div>
<div id="columnLeft">
[...]
</div>
<div id="footer">
[...]
</div>
```

#### Затем используйте следующие модифицированные правила CSS:

#columnLeft {
 width: 340px;

```
margin-left: 10px;
padding-top: 1em;
}
#columnRight {
float: right;
width: 200px;
}
#footer {
clear: both;
background-color: #ccc;
padding-bottom: 1em;
border-top: 1px solid #333;
padding-left: 10px;
}
```

# См. также

В разд. 11.2 создание макета, состоящего из двух колонок с меняющейся шириной.

# 11.4. Создание "резинового" многоколоночного макета с помощью плавающей модели

# Проблема

Вы хотите создать макет страницы с тремя колонками, меняющими ширину в соответствии с шириной окна Web-обозревателя (рис. 11.5).

# Решение

Прежде всего, разметьте содержимое с помощью элементов div, использующих атрибуты id, которые содержат соответствующие значения, представляющие расположение содержимого на странице:

```
<div id="header">
[...]
</div>
<div id="columnLeft">
[...]
</div>
<div id="columnMain">
[...]
</div>
<div id="columnRight">
[...]
</div>
<div id="footer">
[...]
</div>
```

CSS Cookbook          Image: Cookbook       Image: Cook				
<image/> <section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header>	CSS Cookbook			
<section-header><section-header>And the state of the state</section-header></section-header>				
<section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header><section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header></section-header>	Back Forward Reload	Stop Home Print 1Password		
<section-header><section-header><section-header><section-header><section-header><section-header><text><list-item><list-item><text></text></list-item></list-item></text></section-header></section-header></section-header></section-header></section-header></section-header>	CSS Cookbook	🛛 📗 🚺 Christopher Schmitt : Designer, 🕄 🗏 🚑 Techn	ology Books, Tech Confer 😒 🕂 📃 🧮	
<ul> <li>Additional of the second of the</li></ul>	Multi Colu	nn with Floots Hoodon		
<ul> <li>Navigation</li> <li>Quisque facilisis</li> <li>Nam malesuada ornare dolor</li> <li>Cras gravida</li> <li>Diam sit amet</li> <li>Rhoncus ornare</li> <li>Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.</li> <li>In on velit non ligula lacreet ultrices. Praesent ultrices facilisis nisl. Vivamus planet en demonstrate dolor is transplanet activate en dipsicing mollis.</li> </ul>	Multi-Colui	ini witii Floats Heauer		
<ul> <li>Ouisque facilisis</li> <li>Oummodo, ipsum sed pharetra gravida, orci magna thoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat.</li> <li>Cras gravida</li> <li>Diam sit amet</li> <li>Rhoncus ornare</li> <li>Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquan, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum.</li> <li>Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.</li> <li>In non velit non ligula laoreet ultrices. Praesent ultricies facilisis nisl. Vivamus plelentesque, erat eget elementum volutpat, dolor nisl porta neque, vitae sodales ipsum nibh in ligula. Maecenas mattis pulvinar diam.</li> </ul>	Navigation	Lorem ipsum dolor sit amet, consectetuer adipiscing elit.	Integer eget sem. Ut vitae enim eu est	
<ul> <li>Quisque facilisis</li> <li>Nam malesuada ornare dolor</li> <li>Cras gravida</li> <li>Diam sit amet</li> <li>Rhoncus ornare</li> <li>Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum.</li> <li>Donec portitior ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.</li> <li>In non velit non ligula laoreet ultrices. Praesent ultricies facilisis nisl. Vivamus luctus elit sit amet mi. Phasellus pellentesque neque, ita gestas.</li> <li>In non velit non ligula laoreet ultrices. Praesent ultricies facilisis nisl. Vivamus luctus elit sit amet mi. Phasellus pellentesque, vitae sodales ipsum nibh in ligula. Maecenas mattis pulvinar diam.</li> </ul>	0	Morbi commodo, ipsum sed pharetra gravida, orci magna	vehicula gravida. Morbi ipsum ipsum,	
<ul> <li>Nam malesuada ornare dolor</li> <li>Cras gravida</li> <li>Diam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat.</li> <li>Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum.</li> <li>Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.</li> <li>In non velit non ligula laoreet ultrices. Praesent ultricies facilisis nisl. Vivamus luctus elt sit amet mi. Phasellus pellentesque, erat eget elementum volutpat, dolor nisl porta neque, vitae sodales ipsum nibh in ligula. Maecenas mattis pulvinar diam.</li> </ul>	<ul> <li>Quisque facilisis</li> </ul>	rhoncus neque, id pulvinar odio lorem non turpis. Nullam	porta nec, tempor id, auctor vitae, purus.	
dolor       commentum. Anquan era volupat.         • Cras gravida       Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum.       Quisque aliquet, quam elementum condimentum elementum condimentum elementum accumsan eu, blandit sed, blandit a, eros.       Quisque aliquet, quam elementum condimentum elementum condimentum feugiat, tellus odio consectetuer wisi, vel nonumy sem neque in elit. Curabitur eleifend wisi iaculis ipsum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.         In non velit non ligula laoreet ultrices. Praesent ultricies facilisis nisl. Vivamus luctus elit sit amet mi. Phasellus pellentesque, erat eget elementum volutpat, dolor nisl porta neque, vitae sodales ipsum nibh in ligula. Maecenas mattis pulvinar diam.         Proin at eros non eros adipiscing mollis.	<ul> <li><u>Nam malesuada ornare</u></li> </ul>	sit amet enim. Suspendisse id velit vitae ligula volutpat	Pellentesque neque. Nulla luctus erat	
<ul> <li>Cras gravida</li> <li>Diam sit amet</li> <li>Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum.</li> <li>Donec portitior ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.</li> <li>In non velit non ligula laoreet ultrices. Praesent ultricies facilisis nisl. Vivamus luctus elit sit amet mi. Phasellus pellentesque, erat eget elementum volutpat, dolor nisl porta neque, vitae sodales ipsum nibh in ligula. Maecenas mattis pulvinar diam.</li> </ul>	dolor	condimentum. Anquam erat volutpat.	aliquam enim et tortor	
<ul> <li>Diami strainer</li> <li>Diami strainer</li> <li>Diami strainer</li> <li>Bed quis verta vent, valua factula, routa fa</li></ul>	<ul> <li>Cras gravida</li> <li>Diam ait amot</li> </ul>	Sed quis velit. Nulla facilisi. Nulla libero. Vivamus	anquan enni et tortor.	
<ul> <li>nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum.</li> <li>Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.</li> <li>In non velit non ligula laoreet ultrices. Praesent ultricies facilisis nisl. Vivamus luctus elit sit amet mi. Phasellus pellentesque, erat eget elementum volupat, dolor nisl porta neque, vitae sodales jpuvinar diam.</li> <li>Proin at eros non eros adipiscing mollis.</li> </ul>	Rhoncus ornare	pharetra posuere sapien. Nam consectetuer. Sed aliguam.	Ouisque aliquet, quam elementum	
orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros. In non velit non ligula laoreet ultrices. Praesent ultricies facilisis nisl. Vivamus luctus elit sit amet mi. Phasellus pellentesque, erat eget elementum volutpat, dolor nisl porta neque, vitae sodales ipsum nibh in ligula. Maecenas mattis pulvinar diam.	- Innoneus ornare	nunc eget euismod ullamcorper, lectus nunc ullamcorper	condimentum feugiat, tellus odio	
Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros. In non velit non ligula laoreet ultrices. Praesent ultricies facilisis nisl. Vivamus luctus elit sit amet mi. Phasellus pellentesque, erat eget elementum volutpat, dolor nisl porta neque, vitae sodales ipsum nibh in ligula. Maecenas mattis pulvinar diam.		orci, fermentum bibendum enim nibh eget ipsum.	consectetuer wisi, vel nonummy sem	
Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros. In non velit non ligula laoreet ultrices. Praesent ultricies facilisis nisl. Vivamus luctus elit sit amet mi. Phaellus pellentesque, erat eget elementum volutpat, dolor nisl porta neque, vitae sodales ipsum nibh in ligula. Maecenas mattis pulvinar diam.			neque in elit. Curabitur eleifend wisi	
consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros. In non velit non ligula laoreet ultrices. Praesent ultricies facilisis nisl. Vivamus luctus elit sit amet mi. Phaellus pellentesque, erat eget elementum volutpat, dolor nisl porta neque, vitae sodales ipsum nibh in ligula. Maecenas mattis pulvinar diam.		Donec porttitor ligula eu dolor. Maecenas vitae nulla	iaculis ipsum. Pellentesque habitant	
accumsan eu, blandit sed, blandit a, eros. In non velit non ligula laoreet ultrices. Praesent ultricies facilisis nisl. Vivamus luctus elit sit amet mi. Phasellus pellentesque, erat eget elementum volutpat, dolor nisl porta neque, vitae sodales ipsum nibh in ligula. Maecenas mattis pulvinar diam.		consequat libero cursus venenatis. Nam magna enim,	morbi tristique senectus et netus et	
In non velit non ligula laoreet ultrices. Praesent ultricies facilisis nisl. Vivamus luctus elit sit amet mi. Phasellus pellentesque, erat eget elementum volutpat, dolor nisl porta neque, vitae sodales ipsum nibh in ligula. Maecenas mattis pulvinar diam.		accumsan eu, blandit sed, blandit a, eros.	malesuada fames ac turpis egestas.	
Praesent ultricies facilisis nisl. Vivamus luctus elit sit amet mi. Phasellus pellentesque, erat eget elementum volutpat, dolor nisl porta neque, vitae sodales ipsum nibh in ligula. Maecenas mattis pulvinar diam.			In non velit non ligula laoreet ultrices.	
luctus elit sit amet mi. Phasellus pellentesque, erat eget elementum volutpat, dolor nisl porta neque, vitae sodales ipsum nibh in ligula. Maecenas mattis pulvinar diam.			Praesent ultricies facilisis nisl. Vivamus	
pellentesque, erat eget elementum volutpat, dolor nisl porta neque, vitae sodales ipsum nibh in ligula. Maecenas mattis pulvinar diam.			luctus elit sit amet mi. Phasellus	
Volutpat, dolor nisi porta neque, vitae sodales ipsum nibh in ligula. Maecenas mattis pulvinar diam.			pellentesque, erat eget elementum	
Proin at eros non eros adipiscing mollis.			volutpat, dolor nisi porta neque, vitae	
Proin at eros non eros adipiscing mollis.			mattis pulvinar diam.	
Proin at eros non eros adipiscing mollis.				
Proin at eros non eros adipiscing mollis.				
	Proin at eros non eros adipiscing mollis.			

Рис. 11.5. Макет с тремя колонками, созданный с помощью технологии CSS

Далее задайте для каждой колонки свойство float с выравниванием по левому краю, убедившись, что ширина задана в процентах. В сумме ширина трех колонок должна быть равна 100 % (рис. 11.6):

```
#columnRight {
width: 33%;
float: left;
background: white;
padding-bottom: 1em;
}
#columnLeft {
width: 20%;
float:left;
background: white;
padding-bottom: 1em;
text-align: justify;
}
#columnMain {
width:47%;
float:left;
background: white;
padding-bottom: 1em;
}
```

```
#footer {
   clear: both;
   padding-bottom: lem;
   border-top: lpx solid #333;
   text-align: center;
}
```



Рис. 11.6. Увеличенная ширина основной колонки, заставившая правую колонку спуститься ниже

# Обсуждение

Этот метод работает, потому что все колонки включены в плавающую модель с выравниванием по левому краю и их суммарная ширина не больше 100 %. Задание выравни-

вания по правому краю может создать зеркальное отображение колонок, но результат будет тот же.

Задавайте поля и отступы элементов, не выходя за пределы ширины колонок (если вы не учитываете их величины при определении размеров колонок). Если это условие нарушено, суммарная ширина колонок будет более 100 %, и одна или несколько колонок расположатся друг под другом.

# См. также

В *разд. 11.5* создание макета с тремя колонками фиксированной ширины; на странице **http://www.realworldstyle.com/nn4\_3col\_header.html** сведения о создании макета страницы с тремя колонками, у одной из которых меняющаяся ширина, а у двух других — фиксированная.

# 11.5. Создание с помощью плавающей модели макета с тремя колонками фиксированной ширины

# Проблема

Вы хотите создать макет с тремя колонками фиксированной ширины.

# Решение

Сначала разметьте содержимое с помощью элементов div, использующих атрибуты id с соответствующими значениями, отражающими размещение информации на странице:

```
<div id="header">
[...]
</div>
<div id="columnMain">
[...]
</div>
<div id="columnLeft">
[...]
</div>
<div id="columnRight">
[...]
</div>
<div id="footer">
[...]
</div>
```

Далее заключите в еще один элемент div элементы div, содержащие основную и левую колонку, и задайте для атрибута id этого элемента значение enclose. Заключите также

в дополнительный элемент div весь набор существующих элементов div, установив в нем для атрибута id значение frame:

```
<div id="frame">
 <div id="header">
 [...]
 </div>
 <div id="enclose">
  <div id="columnMain">
  [...]
  </div>
  <div id="columnLeft">
  [...]
  </div>
 </div>
 <div id="columnRight">
 [...]
 </div>
 <div id="footer">
 [...]
 </div>
</div>
```

Задайте ширину страницы, используя селектор ID для элемента div со значением frame:

```
#frame {
  margin-left: 20px;
  width: 710px;
}
```

Затем включите элементы div колонок и элемент div со значением enclose в плавающую модель (рис. 11.7):

```
#columnMain {
float: right;
width: 380px;
}
#columnLeft {
float: left;
width: 150px;
}
#columnRight {
 float: right;
width: 120px;
}
#enclose {
float:left;
width:560px;
}
#footer {
 clear: both;
padding-top: 1em;
text-align: center;
}
```



Рис. 11.7. Макет с тремя колонками фиксированной ширины

# Обсуждение

Поскольку ширина колонок задана в пикселах, она фиксирована. Для отображения колонок вам необходим дополнительный элемент div, включающий основную и левую колонки. С помощью дополнительного элемента div со значением enclose у атрибута id основная и левая колонки как единое целое вставляются в поток плавающей модели с выравниванием по левому краю. Внутри этого элемента div основная колонка выровнена по правому краю, а левая колонка — по левому.

# См. также

Разд. 11.4 о создании "резинового" макета с тремя колонками.

# 11.6. Создание с помощью позиционирования "резинового" многоколоночного макета

# Проблема

Вы хотите создать макет с четырьмя колонками, которые изменяют ширину в соответствии с шириной окна Web-обозревателя (рис. 11.8).



Рис. 11.8. Макет с четырьмя колонками, ширина которых задается в процентах

# Решение

Сначала разметьте содержимое с помощью элементов div, используя атрибуты id с соответствующими значениями, отражающими размещение информации на странице:

```
<div id="header">
[...]
</div>
<div id="columnLeft">
[...]
</div>
<div id="columnInnerLeft">
[...]
</div>
```

```
<div id="columnInnerRight">
[...]
</div>
<div id="columnRight">
[...]
</div>
```

Далее используйте в каждой колонке свойство position со значением absolute, одновременно задавая положение колонок с помощью свойств left и top:

```
#columnLeft {
position: absolute;
 left:1%;
width:20%;
top: 4em;
background:#fff;
}
#columnInnerLeft {
position: absolute;
 left: 22%;
 width: 28%;
 top: 4em;
background: #fff;
 text-align: justify;
border-width: 0;
}
#columnInnerRight {
position: absolute;
 left: 51%;
 width: 28%;
 top: 4em;
background: #fff;
}
#columnRight {
position: absolute;
 left: 80%;
 width: 19%;
top: 4em;
background: #fff;
}
```

# Обсуждение

Установив в свойстве position значение absolute, вы полностью удаляете элемент из нормального потока документа. Когда элемент включается в поток плавающей модели (float), другие элементы на странице могут обтекать этот "плавающий" элемент. Если же для элемента задано абсолютное позиционирование, остальными элементами на странице, включенными в нормальный поток, он трактуется как фиктивный.

По умолчанию элемент с абсолютным позиционированием отображается в левом верхнем углу его ближайшего позиционированного предка или начального блока-

контейнера. (Другими словами, для размещения дочернего элемента с абсолютным позиционированием внутри родительского элемента прежде всего задайте для родительского элемента свойство position с конкретным значением.) Если на странице есть другие элементы, при этом возникает наложение содержимого, как показано на рис. 11.9.



Рис. 11.9. Наложение текста в Web-документе на изображение и другой текст

Для устранения этой проблемы воспользуйтесь четырьмя дополнительными свойствами CSS: top, left, bottom и right, позволяющими свободно перемещать элемент в любое место. Убедитесь в том, что эти свойства *сдвига* колонок заданы в процентах, что позволит поддерживать переменную ширину колонок при изменении размеров окна Web-обозревателя пользователя.

Задайте также в процентах значение свойства left для обозначения отступа от левого края окна просмотра Web-обозревателя. А для указания значения свойства top, компенсирующего высоту заголовка, в качестве единиц измерения используйте кегельные (em). Если в качестве заголовка вы хотите применить изображение, измените единицы измерения значения свойства top на пикселы, обеспечив достаточно места для отображения графического заголовка.

#### Известные проблемы

Несмотря на то, что этот метод дает возможность свободно располагать элементы на странице, у абсолютного позиционирования есть и недостатки. Например, если размеры шрифта в Web-обозревателе изменяются, текст может наложиться на другие элементы или текст, что затруднит чтение и зрительное восприятие Web-страницы.

Кроме того, с помощью предложенного метода легко расположить колонки бок о бок, однако трудно определить местоположение заключительной части страницы (колонтитула) под колонками, если не знать, где именно в нижней части страницы они заканчиваются.

# См. также

Свойство position в спецификации CSS 2.1 на странице http://www.w3.org/TR/CSS21/ visuren.html#propdef-position; информацию в CSS 2.1 об абсолютном позиционировании элементов на странице http://www.w3.org/TR/CSS21/visuren.html#position-props; дополнительные сведения о блоках-контейнерах на http://www.w3.org/TR/2003/WD-CSS21-20030915/visudet.html#containing-block-details.

# 11.7. Создание с помощью позиционирования многоколоночного макета с колонками фиксированной ширины

# Проблема

Вы хотите создать макет, состоящий из четырех колонок фиксированной ширины.

# Решение

Сначала разметьте содержимое с помощью элементов div, используя атрибуты id с соответствующими значениями, отражающими размещение элементов на странице:

```
<div id="header">
[...]
</div>
<div id="columnLeft">
[...]
</div>
<div id="columnInnerLeft">
[...]
</div>
<div id="columnInnerRight">
[...]
</div>
<div id="columnRight">
[...]
</div>
```

Далее в каждой колонке воспользуйтесь свойством position со значением absolute, одновременно с помощью свойств left и top (в качестве единиц измерения используйте пикселы) определяя местоположение колонок на странице:

```
#columnLeft {
   position: absolute;
   left:5px;
   width:190px;
   top: 44px;
   background:#fff;
  }
#columnInnerLeft {
   position: absolute;
```

```
left: 205px;
 width: 190px;
 top: 44px;
background: #fff;
 text-align: justify;
border-width: 0;
1
#columnInnerRight {
position: absolute;
 left: 405px;
width: 190px;
 top: 44px;
background: #fff;
}
#columnRight {
position: absolute;
left: 605px;
 width: 190px;
 top: 44px;
background: #fff;
}
```

# Обсуждение

Применение единиц длины для задания ширины, а также свойств left и top, позволяют сформировать колонки фиксированной ширины. Данное решение также легко применить для верстки страницы с двумя или тремя колонками. Но помните, что макеты с числом колонок, большим четырех или пяти, практически не используются.

#### Примечание

Несмотря на то, что допустимо применение абсолютного позиционирования, для создания макетов гораздо лучше использовать плавающую модель, которая позволяет создавать гибкий Web-дизайн, уберегающий вашу страницу от разрушения.

#### См. также

*Разд. 11.3* о создании макета с двумя колонками фиксированной ширины; *разд. 11.5* о создании макета с несколькими колонками фиксированной ширины с помощью плавающей модели.

# 11.8. Использование плавающей модели для отображения колонок в произвольном порядке

# Проблема

Вы хотите разработать систему отображения содержимого в колонках, которые выводятся на экран в произвольном порядке.

## Решение

#### Есть следующая разметка:

```
<div id="container-outer">
<div id="container">
 <div id="content" class="column">
  <div class="wrap">
    [...]
  </div>
 </div><!-- /END #content -->
 <div id="navigation" class="column">
  <div class="wrap">
    [...]
  </div>
 </div><!-- /END #navigation -->
 <div id="related-info" class="column">
  <div class="wrap">
    [...]
   </div>
 </div><!-- /END #related-info -->
</div><!-- /END #container -->
</div><!-- /END #container-outer -->
```

#### Примените следующие правила CSS:

```
.column {
 float: left;
}
#content {
margin-left: 20%;
width: 60%;
}
#navigation {
margin-left: -80%;
width: 20%;
}
#related-info {
width: 19%;
}
/* IEx patches \*/
* html .column {
display: inline;
}
* html #navigation li {
height: 1%;
}
/**/
```

В результате получится показанный на рис. 11.10 макет страницы с двумя узкими боковыми полосами переменной ширины, которые обрамляют центральную колонку с такой же регулируемой шириной.

Рис. 11.10. Базовое форматирование макета страницы

На эту довольно слабую основу можно наслаивать дополнительные правила CSS. Включение следующего кода в вашу таблицу стилей приведет к дизайну, показанному на рис. 11.11:

```
body {
  font: normal 62.5%/1.7 Verdana, Geneva, Helvetica, Arial, sans-serif;
  margin: 0;
  padding: 0;
}
#container:after {
  clear: both;
  content: ".";
  display: block;
  height: 0;
  visibility: hidden;
}
#container {
  display: inline-block;
}
```

```
#container-outer {
background: url("bg-left.gif") repeat-y 20% 0;
}
#container {
background: url("bg-right.gif") repeat-y 80% 0;
}
.column .wrap {
padding: 20px;
#content .wrap {
padding: 20px 30px;
}
#content p {
margin-top: 0;
1
#content p:first-child {
font: normal 1.4em/1.6 Georgia, Times, "Times New Roman", serif;
}
#content p:first-child:first-line {
text-transform: uppercase;
}
#navigation ul, #navigation ul li {
list-style: none;
margin: 0;
padding: 0;
#navigation ul li {
margin-bottom: .4em;
#navigation li a {
background: #36C;
color: #FFF;
border-left: 7px solid #09F;
display: block;
padding: .4em .4em .4em 20px;
text-decoration: none;
}
#navigation li a:hover {
border-left: none;
border-right: 7px solid #09F;
padding-left: 27px;
}
#related-info {
color: #555;
font-style: italic;
#copyright {
border: 1px solid #B2B2B2;
border-width: 1px 0;
clear: both;
padding: 10px 20px;
text-align: center;
}
```

```
#copyright p {
  margin: 0;
}
```

00	http://frylock.build – Liquid Columns	0
Link 1 Link 2 Link 3 Link 4	I LIKE BIG BLOCKS, AND I CANNOT LIE. YOU OTHER CODERS CAN'T deny: when a lorem ipsum walks in, with a sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nune eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitori ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit s, eros. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nuc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.	Proin at eros non eros adipiscing mollis. Donec semper turpis sed diam. Sed consequat liguia nec tortor. Integer eget sem. Ut vitae enim eu est vehicula gravida. Morbi ipsum ipsum, porta nec, tempor id, auctor vitae, purus. Pellentesque neque. Nulla luctus erat vitae libero. Integer nec enim. Phasellus aliquam enim et tortor. Quisque aliquet, quam elementum condimentum feugiat, tellus odio consectetuer wisi, vel nonummy sem neque in elit.
	Copyright © 2006 FictitiousCo, LLC.	
Done		<b>a</b> 7

Рис. 11.11. Улучшенный дизайн многоколоночного макета страницы

# Обсуждение

Авторы спецификации CSS никогда не предназначали плавающую модель для управления макетом страницы: скорее это было средство, которое управляло обтеканием контекста вокруг объекта почти так же, как align="left" или align="right" создает обтекание текста вокруг элемента img. Но вопреки первоначальной направленности, принятой в спецификации, плавающая модель на самом деле предложила мощную альтернативу традиционной верстке с помощью таблиц.

Дизайнер Алекс Робинсон (Alex Robinson) опубликовал оказавшую решающее влияние статью о выводе колонок в произвольном порядке ("Any Order Columns") с помощью технологии CSS (http://www.positioniseverything.net/articles/onetruelayout/). Метод Робинсона позволил разработчикам легко создавать многоколоночные макеты, используя плавающую модель для отображения колонок в произвольном порядке, независимо от порядка появления в разметке блоков описания этих колонок.

#### Разметка

Для применения этого метода сначала необходимо сформировать колонки в разметке страницы следующим образом:

```
<div id="container">
  <div id="container">
  <div id="content" class="column">
  [...]
  </div><!-- /END #content -->
  <div id="navigation" class="column">
  [...]
  </div><!-- /END #navigation -->
  <div id="related-info" class="column">
  [...]
  </div><!-- /END #related-info -->
  </div><!-- /END #container -->
  <div id="copyright">
  Copyright notice goes here.
  </div>
```

Внутри каждого элемента div поместите разметку, которая вам нравится. На рис. 11.12 показано, как выглядит документ без стилевого оформления, содержащий несколько абзацев и маркированный список, вставленный для полноты картины.

E C Attp://frylock.build - Liquid Columns
I like big blocks, and I cannot lie. You other coders can't deny: when a lorem ipsum walks in, with a sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliguam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donce portitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.
• Link 1 • Link 2 • Link 3 • Link 4
Proin at eros non eros adipiscing mollis. Donec semper turpis sed diam. Sed consequat ligula nec tortor. Integer eget sem. Ut vitae enim eu est vehicula gravida. Morbi ipsum ipsum, porta nec, tempor id, auctor vitae, purus. Pellentesque neque. Nulla luctus erat vitae libero. Integ nec enim. Phasellus aliguam enim et tortor. Quisque aliquet, quam elementum condimentum feugiat, tellus odio consectetuer wisi, vel nonummy sem neque in elit.
Copyright © 2006 FictitiousCo, LLC.
Done

К настоящему моменту в примере для каждой из трех колонок введен элемент div, которому присвоен идентификатор, описывающий вид информации, которая будет помещаться в элемент. Значения атрибута id, используемые в данном решении, content, navigation и related-info.

#### Примечание

Легче было бы воспользоваться идентификаторами center, left и right, но это крайне недальновидно — что произойдет, когда вы решите изменить файл CSS для вашего сайта и новый дизайн потребует вывода "левого" элемента div справа?

#### Определение колонок

Имея описанную простую структуру разметки, вы можете применить общее правило для свойства float ко всем трем элементам div колонок:

```
.column {
  float: left;
}
```

Как видно из рис. 11.13, макет не выглядит принципиально по-иному. Текст об авторских правах выбивается из общего выравнивания, но основная часть вашей страницы отображается, как и раньше, с размещением всех элементов div колонок один под другим. Но все быстро изменится, как только этим блокам будут присвоены размеры.

```
http://frylock.build - Liquid Columns
                                                                                                                                                                 I like big blocks, and I cannot lie. You other coders can't deny: when a lorem ipsum walks in, with a sit amet, consectetuer adipiscing elit.
Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim.
Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra
posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim
nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu,
blandit sed, blandit a, eros.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id
pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed
quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper,
lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat
libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.
Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id
pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed
quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper,
lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat
libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.
               Copyright © 2006 FictitiousCo, LLC.

    Link 1

    Link 2

       Link
Proin at eros non eros adipiscing mollis. Donec semper turpis sed diam. Sed consequat ligula nec tortor. Integer eget sem. Ut vitae enim eu est vehicula gravida. Morbi ipsum ipsum, porta nec, tempor id, auctor vitae, purus. Pellentesque neque. Nulla luctus erat vitae libero. Integer
nec enim. Phasellus aliquam enim et tortor. Quisque aliquet, quam elementum condimentum feugiat, tellus odio consectetuer wisi, vel
nonummy sem neque in elit.
Done
```

Начните с блока content. Создайте следующее правило, устанавливающее ширину блока, равной 60 % ширины окна, а ширину левой боковой полосы, равной 20 % ширины экрана:

```
#content {
  margin-left: 20%;
  width: 60%;
}
```

На рис. 11.14 показано, что макет выглядит немного странно, но начинает обретать форму.



Рис. 11.14. Применение стилей к фрагменту макета content

Установив левое поле равным ширине левой боковой полосы, вы по существу "зарезервировали" для нее некоторое свободное пространство. Следующий шаг — использование отрицательных значений полей для "переброски" элемента div с идентификатором navigation через элемент div с идентификатором content к левому краю страницы:

```
#navigation {
  margin-left: -80%;
  width: 20%;
}
```

Заданное значение margin-left равно сумме ширины центральной колонки (60 %) и поля с левой стороны (20 %). Эта установка перемещает колонку navigation на нужное место (рис. 11.15).

530



Рис. 11.15. Содержимое блока navigation перемещается в левую колонку

Теперь завершим создание макета с тремя колонками, показанного на рис. 11.16, определив ширину блока related-info:

```
#related-info {
  width: 20%;
}
```

Выглядит отлично, хотя элемент div с идентификатором copyright все еще не на месте. Это легко исправить с помощью свойства clear (рис. 11.17):

```
#copyright {
   clear: both;
}
```

Несмотря на то, что макет кажется завершенным, рис. 11.18 иллюстрирует необходимость некоторого дополнительного внимания в случае применения Web-обозревателя Internet Explorer для Windows.

К счастью, это документированная ошибка IE, известная как "Doubled Float-Margin Bug" (ошибка удвоения поля "плавающего" элемента) (http://positioniseverything.net/ explorer/doubled-margin.html): по существу, когда в "плавающем" блоке применяется поле со стороны, совпадающей с направлением выравнивания блока, величина поля удваивается.

Поскольку поле с левой стороны задается для "плавающего" элемента, выровненного по левому краю, Internet Explorer для Windows берет значение поля (20 %) и удваивает его до 40 %.

● ● ●	http://frylock.build - Liquid Columns
• Link 1 • Link 2 • Link 3 • Link 4	<ul> <li>I like big blocks, and I cannot lie. You other coders can't deny: when a lorem ipsum Proin at eros non eros walks in, with a sit arnet, consectetuer adipiscing elit. Morbi commodo, ipsum se merer turgisse di dans. Ses diam. Suspendisse di velit. Nulla facilisi. Nulla libero. Vivamus pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turgis. Nam magna enim. Suspendisse di velit. Nulla facilisi. Nulla libero. Ses diaugum entar consectetuer. Sed aliquam, nunc eget euismod ullam. Corper, lectus nunc ullam. Suspendisse di velit. Nulla facilisi. Nulla libero. Vivamus pharetra ses diam. Marconsecteture. Sed aliquam, nunc eget euismod ullam. Corper, lectus nunc ullam. Suspendisse di velit. Nulla facilisi. Nulla libero. Vivamus pharetra ses diam. Ascumsan eu, blandit sed, blandit a, eros.</li> <li>Cory diam. Suspendisse di velit. Nulla facilisi. Nulla libero. Vivamus pharetra se se ses diam. Suspendisse di velit. Nulla facilisi. Nulla libero. Vivamus pharetra set se ses diam. Suspendisse di velit. Nulla facilisi. Nulla libero. Ses diam. Suspendisse di velit. Nulla facilisi. Nulla libero. Vivamus pharetra sus nunc ullam. Suspendisse di velit. Nulla facilisi.</li></ul>
Done	(A)

#### Рис. 11.16. Перемещение на нужное место содержимого правой колонки

00	http://frylock.build - Liquid Columns	0
<ul> <li>Link 1</li> <li>Link 2</li> <li>Link 3</li> <li>Link 4</li> </ul>	I like big blocks, and I cannot lie. You other coders can't deny: when a lorem ipsun walks in, with a sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivanus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitor ligula eu dolor. Maccenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivanus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitor ligula eu dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivanus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunce eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitor ligula eu dolor. Maccenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros. useCo, L	Proin at eros non eros adipiscing mollis. Donec semper turpis sed diam. Sed consequat ligula nec tortor. Integer eget sem. Ut vitae enim eu est vehicula gravida. Morbi ipsum ipsum, porta nec, tempor id, auctor vitae, purus. Pellentesque neque. Nulla luctus erat vitae libero. Integer nec enim. Phasellus aliquam enim et tortor. Quisque aliquet, quam elementum condimentum feugiat, tellus odio consecteture wisi, vel nonummy sem neque in elit.
Done		<u> </u>



Рис. 11.18. Иллюстрация проблем отображения макета в Internet Explorer для Windows

Исправить поведение Internet Explorer, к счастью, легко, благодаря применению в проблемном элементе свойства display: inline. Для этого вставьте следующие строки в свою таблицу CSS:

```
/* Исправления для IE \*/
* html .column {
display: inline;
}
/**/
```

Необычная структура комментариев и префикс \* html гарантируют, что приведенный код увидят более ранние версии IE. Теперь, как показано на рис. 11.19, IE ведет себя должным образом.

В результате создан шаблон "резинового" макета с тремя колонками. Как можно его еще усовершенствовать?

#### Создание свободного пространства

Свободное пространство между колонками называется *средником* (gutter), или промежутком между колонками. Увеличить размеры этих промежутков можно было бы за счет задания определенных полей вокруг колонок. Есть несколько способов получения желаемого эффекта, но сначала вставьте для каждой из ваших колонок дополнительные элементы div следующим образом:

```
<div id="container">
  <div id="container">
  <div id="content" class="column">
  <div class="wrap">
  [...]
  </div>
  </div><!-- /END #content -->
  <div id="navigation" class="column">
  <div class="wrap">
  [...]
  </div>
  </div><!-- /END #navigation -->
  <div id="related-info" class="column">
  <div class="wrap">
</div class="wrap">
</div</pre>
```

```
[...]
```

</div>

</div><!-- /END #related-info --> </div><!-- /END #container -->



**Рис. 11.19.** Внесено исправление, позволяющее корректно отображать макет в Internet Explorer для Windows

После этого примените к добавленным элементам div (к которым мы еще вернемся в обсуждении) правила CSS, описывающие отступы, для создания дополнительного свободного пространства (рис. 11.20):

```
.column .wrap {
  padding: 20px;
}
#content .wrap {
  padding: 20px 30px;
}
```

000	http://frylock.build - Liquid Columns	0
• Link 1 • Link 2 • Link 3 • Link 4	http://frylock.build – Liquid Columns I like big blocks, and I cannot lie. You other coders can't deny: when a lorem ipsum walks in, with a sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donce portitor ligula eu dolor. Maccenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc consecuter. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc consecuter. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc consecuter. Sed aliquam, nunc eget autimod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donee portitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuree sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum biendum enim hole eget ipsum. Do	Proin at eros non eros adipiscing mollis. Donec semper turpis sed diam. Sed consequat ligula nec tortor. Integer eget sem. Ut vitae enim eu est vehicula gravida. Morbi ipsum ipsum, porta nec, tempor id, auctor vitae, purus. Pellentesque neque. Nulla luctus erat vitae libero. Integer nec enim. Phasellus aliquam enim et tortor. Quisque aliquet, quam elementum feugiat, tellus odio consectetuer wisi, vel nonummy sem neque in elit.
	Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.	
Copyright © 2006 FictitiousCo	,LLC.	
Done		۵ (

Рис. 11.20. Увеличение размеров промежутков между колонками

#### Управление порядком отображения колонок

Как вы уже могли заметить, метод отображения колонок в произвольном порядке основан на настройке значений полей: *положительные* значения полей применяются для резервирования свободного пространства, а *отрицательные* — для изменения естественного местоположения колонок.

Упростим в одно мгновение таблицу CSS, удалив все свойства margin в колонках:

```
#content {
  width: 60%;
}
#navigation {
  width: 20%;
}
#related-info {
  width: 19%;
}
```

Теперь ваш макет выглядит, как показано на рис. 11.21, с отображением каждой колонки в соответствии с ее естественным местоположением, определенным в потоке плавающей модели.

😝 🖯 🖨 http://frylock.build – Liquid Colur	mns	
<ul> <li>http://fylock.build - Liquid Colur</li> <li>Nike big blocks, and I cannot lie. You other coders can't deny: when a lorem ipsum walks in, with a sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, ord magna rhonous neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ulamcorper, lectus nunc ulamcorper orci, fermentum bibendum enim nibh eyenenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.</li> <li>Lorem ipsum dolor sit amet, consecteture adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, ord magna rhoneus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla il suo. Vivamus pharetra posuere sapien. Nam consecteur. Sed aliquam, nunc eget euismo dulamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitori ligula eu dolor. Maecenas vitae nulla consequa libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.</li> <li>Lorem ipsum dolor sit amet, consecteure adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, ord magna rhoncus neque, id pulvinar dola lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consecuteur. Sed aliquam, nunc eget euismod ulamcorper, lectus nunc ulamcorper orci, fermentum bibendum enim nibh egit josum. Donec portitori ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.</li> <li>Copyright © 2006 FictitiousCo, LLC.</li> </ul>	mns • Link 1 • Link 2 • Link 3 • Link 4	Proin at eros non eros adipiscing mollis. Donec semper turpis sed alam. Sed consequat ligula nec tortor. Integer eget sem. Ut vitae enim eu est vehicula gravida. Morbi ipsum ipsum, porta nec, tempor id, auctor vitae, purus. Pelientesque neque. Nulla luctus erat vitae libero. Integer nec enim. Phaselius aliquam enim et tortor. Quisque aliquet, quam elementum condimentum fegigiat, tellus odio consectetuer wisi, vel nonummy sem neque in elit.
Done		<b>a</b> 7

Рис. 11.21. Размещение блока navigation между колонками

Вставив левое поле в элемент div для блока navigation и применив отрицательное значение левого поля для перемещения элемента div блока related-info, вы можете на самом деле изменить на обратный порядок отображения второй и третьей колонок. Благодаря следующим правилам CSS получится макет, показанный на рис. 11.22:

```
#content {
  width: 60%;
}
#navigation {
  margin-left: 20%;
  width: 20%;
}
#related-info {
  margin-left: -39%;
  width: 19%;
}
```

И в завершение поместите колонку с идентификатором content на правую сторону страницы, как показано на рис. 11.23.

#### Глава 11

 $\bigcirc$ 

• Link 1

Link 2

Link 3

Link 4

#### 00

#### http://frylock.build - Liquid Columns

I like big blocks, and I cannot lie. You other coders can't deny: when a lorem ipsum walks in, with a sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum see pharetra gravida, orci magna fhoncus neque, ig duvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed guis velit. Nulla facilis. Nulla libero. Vitamus pharetra posure sapien. Nam consectutere. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitor ligula eu doir. Maecenas vitae nulla consequat libero cursus venenais. Nam magna enim, accumsan eu, biandi sed, biandi ta, eros.

Lorem ipsum dolor sit amet, consecteture adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitior ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velik vitae liguia volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitor liguia eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

Copyright © 2006 FictitiousCo, LLC.

Done

adipiscing mollis. Donec semper turpis sed diam. Sed consequat liquia nec tortor. Integer eget sem. Ut vitae enim eu est vehicula gravida. Morbi ipsum ipsum, porta nec, tempor id, auctor vitae, purus. Pellentesque neque. Nulla luctus erat vitae libero. Integer nec enim. Phasellus aliquam enim et tortor. Quisque aliquet, quam elementum condimentum feugiat, tellus odio consectetuer wisi, vei nonummy sem neque in elit.

Proin at eros non eros

#### Рис. 11.22. Изменение порядка отображения колонок

$\bigcirc \bigcirc \bigcirc$	\varTheta 🕤 🕘 http://frylock.build – Liquid Columns 🖸		
<ul> <li>Link 1</li> <li>Link 2</li> <li>Link 3</li> <li>Link 4</li> </ul>	Proin at eros non eros adipiscing mollis. Donec semper turpis sed diam. Sed consequat ligula nec tortor. Integer eget sem. Ut vitae enim eu est vehicula gravida. Morbi Ipsum ipsum, porta nec, tempor id, auctor vitae, purus. Pellentesque neque. Nulla luctus erat vitae libero. Integer nec enim. Phasellus aliquam enim et tortor: Quisque aliquet, quam elementum condimentum feugiat, tellus ocio consecteure wisi, vel nonummy sem neque in elit.	1 like big blocks, and I cannot lie. You other coders can't deny: when a lorem journ walks in, with a sit amet, consectetuer adipiscing elit. Morbi commodo, josum sed pharetra gravida, orci magna rhoncus neque, id puivinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilis. Nulla libero. Vivamus pharetra possure sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget josum. Donec portitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, biandit sed, blandit a, eros. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id puivinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nuc ullamcorper orci, fermentum bibendum enim nibh eget josum. Donec portitor ligula e udolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, or dimagna rhoncus neque, id puivinar dol lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consecteuer. Sed aliquam, nunc eget euismod ullamcorper, icetus nuc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.	
Done		😐 🖉	

#### Для этого используйте следующий код:

```
#content {
  margin-left: 40%;
  width: 60%;
}
#navigation {
  margin-left: -100%;
  width: 20%;
}
#related-info {
  margin-left: -80%;
  width: 19%;
}
```

Как и в первом макете, вы использовали поле в колонке content для "резервирования" некоторого объема свободного пространства с левого края вашей страницы. Затем вы применяли отрицательные значения левосторонних полей для перемещения в нужное место элементов div блоков navigation и related-info.

#### Псевдоколонки

Теперь вернемся к нашему первому макету (рис. 11.24) и посмотрим, как улучшить внешний вид колонок, сделать их более выразительными. Первый шаг: фоновые изображения.

000	http://frylock.build - Liquid Columns	$\bigcirc$
• Link 1 • Link 2 • Link 3 • Link 4	http://frylock.build - Liquid Columns         I like big blocks, and I cannot lie. You other coders can't deny: when a lorem ipsum walks in, with a sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar otio lorem non turpis. Nullam sit amet enim. Suspendisse id velit viae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla libero. Vivamus pharetra posure sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitor ligula eu dolor. Maccenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.       Proin at eros non eros adipiscing molls. Donec oscupati biero cursus venenatis. Nulla miser enim. Suspendisse id velit vitae ligula rodio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula colofirentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitor ligula eu dolor. Maecenas vitae nulla consequet libero cursus venenatis. Nam magna enim, accumsan eu, blandit a, eros.       Proin at eros non eros adipiscing elit. Morbi cursus duametor cursus venenatis. Nam magna enim, accumsan eu cursus venenatis. Nam magna enim, accumsan eu, blandit a, eros.	,
	Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donce portitior ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.	
Copyright © 2006 FictitiousCo	,LLC.	
Done		<u>)</u>

Рис. 11.24. Первоначальный макет в ожидании графического оформления колонок

"Faux columns" (псевдоколонки или ложные колонки) — метод, разработанный Webдизайнером Дэном Седерхольмом (Dan Cederholm) (http://www.alistapart.com/articles/ fauxcolumns/) и использующий повторяющееся по горизонтали фоновое изображение.

Применяя одно повторяющееся изображение, метод Седерхольма работает очень хорошо в макетах с фиксированной шириной и благодаря своей универсальности нуждается лишь в незначительной модификации для функционирования в "резиновой" верстке.

Прежде всего, вам понадобятся два изображения, по одному для каждой боковой стороны колонки content. На рис. 11.25 показано левостороннее изображение, а на рис. 11.26 — правостороннее.

Рис. 11.25. Рисунок для графического оформления левой стороны колонки

Рис. 11.26. Рисунок для графического оформления правой стороны колонки

Затем вы должны заключить блок container в дополнительный элемент div, как показано далее:

```
<div id="container-outer">
<div id="container">
[Остальной шаблон расположен здесь]
</div>
</div>
```

И наконец, необходимо включить следующие правила в вашу таблицу стилей:

```
#container:after {
 clear: both;
 content: ".";
display: block;
height: 0;
visibility: hidden;
}
#container {
display: inline-block;
}
/*\*/
#container {
display: block;
}
/**/
/*\*//*/
#container {
display: inline-block;
1
/**/
#container-outer {
background: url("bg-left.gif") repeat-y 20% 0;
}
```

```
#container {
  background: url("bg-right.gif") repeat-y 80% 0;
}
```

#### Примечание

Вы также могли бы использовать свойство overflow со значением hidden, как обсуждалось в разд. 2.22.

Благодаря вставке этого кода появятся вертикальные линии вдоль всей длины колонок, как показано на рис. 11.27. Теперь вы можете добавить любые типографические стили, которые вам нравятся; стили, примененные в *разд. "Решение"* данного примера, подойдут как нельзя лучше и в результате сформируют окончательный макет, показанный на рис. 11.28.

00	http://frylock.build - Liquid Columns	$\Box$
Link 1     Link 2     Link 3     Link 4  Copyright © 2006 FictitiousC	I like big blocks, and I cannot lie. You other coders can't deny: when a lorem ipsum walks in, with a sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliguam, nunc eget euismod ullamcorper, lectus nunc ulamcorper orei, fermentum bibendum enim nibh eget ipsum. Donec portitor ligula eu dolor. Maccenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orei magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer, feed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orei, fermentum bibendum enim nibh eget ipsum. Donec portitor ligula eu dolor. Maccenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orei magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orei, fermentum bibendum enim nibh eget ipsum. Donec portitor ligula eu dolor. Maccenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.	Proin at eros non eros adipiscing mollis. Donec semper turpis sed diam. Sed consequat ligula nec tortor. Integer eget sem. Ut vitae enim eu est vehicula gravida. Morbi ipsum ipsum, porta nec, tempor id, auctor vitae, purus. Pellentesque neque. Nulla luctus erat vitae libero. Integer nec enim. Phasellus aliquam enim et tortor. Quisque aliquet, quam elementum feugiat, tellus odio consectetuer wisi, vel nonummy sem neque in elit.
		8 ///

Рис. 11.27. Графическое оформление колонок, включенное в макет

#### Альтернативное решение

Плавающая модель очень эффективна при верстке страниц, но она может потребовать тщательного изучения. В результате многие Web-дизайнеры считают абсолютное позиционирование отличной альтернативой, позволяющей им точно позиционировать различные компоненты их дизайн-проекта с помощью координат X и Y.

К сожалению, позиционируемые элементы удаляются "из нормального потока документа", что сильно сжимает содержащий их элемент. В результате проекты с использо-
540

	http://fo/ock.build - Liquid Columns	
	http://inviock.build = Elquid Columns	0
Link 1 Link 2 Link 3 Link 4	I LIKE BIG BLOCKS, AND I CANNOT LIE. YOU OTHER CODERS CAN'T deny: when a lorem ipsum walks in, with a sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nune eget euismod ullamcorper, lectus nune ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitior ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nuc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitor ligula eudoir. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitat ligula volutpat condimentum. Aliquam eru volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec portitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.	Proin at eros non eros adipiscing mollis. Donec semper turpis sed diam. Sed consequat ligula nec tortor. Integer eget sem. Ut viae enim eu est vehicula gravida. Morbi ipsum ipsum, porta nec, tempor id, auctor vitae, purus. Pellentesque neque. Nulla luctus erat vitae libero. Integer nec enim. Phasellus aliquam enim et tortor. Quisque aliquet, quam elementum condimentum reugiat, tellus doio consecteture visi, vel nonummy sem neque in elit.
	Copyright © 2006 FictitiousCo, LLC.	
Done		9 //

Рис. 11.28. Окончательный макет страницы

ванием позиционирования не обладают мощными функциональными возможностями свойства clear, применяемого в плавающей модели и позволяющего разным частям вашего макета быть "контекстно-зависимыми": т. е. элемент div блока footer (например, в решении блок со сведениями об авторских правах, описанный ранее) можно с помощью свойства clear расположить ниже блоков, включенных в поток плавающей модели, но никак не элементов, точно позиционированных на странице.

Шон Инман (Shaun Inman), талантливый Web-дизайнер и разработчик, написал простенькую функцию на языке JavaScript для решения этой проблемы (http:// shauninman.com/plete/2006/05/clearance-position-inline-absolute.php). Вставленный в ваши Web-страницы скрипт Инмана автоматически "очистит" элементы от любых других блоков, позиционированных на странице (рис. 11.29).

Единственный возможный недостаток этого метода — расчет на активность JavaScript в Web-обозревателе пользователя. Но если содержимое вашей страницы доступно в ваших тестовых Web-обозревателях при отключении JavaScript во время тестирования, значит, все будет хорошо.

### См. также

*Разд. 11.9* о проектировании асимметричного макета с помощью абсолютного позиционирования.

```
00
                                            http://shauninman.com - Clearance (Fluid) // ShaunInman.com
                                                                                                                                                              \bigcirc
                  position: relative: Header
                  position: absolute; This absolutely positioned <div> can flow as long as it
                                                                                                  position: absolute; This absolutely
                  needs to and the footer will always remain below it. Lorem ipsum dolor sit
                                                                                                  positioned <div> can flow as long as it needs
                  amet, consectetuer adipiscing elit. Praesent sed leg sed lacus luctus
                                                                                                  to and the footer will always remain below
                  egestas. Suspendisse potenti. Morbi enim. Morbi id diam. Praesent tempor
                                                                                                  it. Lorem ipsum dolor sit amet, consectetuer
                  tellus ac diam. Nullam cursus. Vestibulum volutpat hendrerit ipsum. Integer
                                                                                                  adipiscing elit. Praesent sed leo sed lacus
                  bibendum libero et lectus. Etiam conque, turpis sed laoreet auctor, mi nisl
                                                                                                  luctus egestas. Suspendisse potenti.
                  mattis pede, nec sodales magna felis porta mauris.
                                                                                                  Morbi enim, Morbi id diam, Praesent tempor
                  Aliguam guis mi, Nulla facilisi, Aliguam erat volutpat, Suspendisse
                                                                                                  tellus ac diam. Nullam cursus. Vestibulum
                  bibendum. Fusce ac sapien. Donec ac turpis. Nulla magna velit, sagittis at,
                                                                                                  volutpat hendrerit ipsum. Integer bibendum
                  vestibulum sit amet, lobortis at, purus. Quisque mattis pretium justo. Cras
                                                                                                  libero et lectus.
                  sagittis, quam ut imperdiet congue, nisl ante auctor odio, vitae gravida nibh
                                                                                                  Etiam congue, turpis sed laoreet auctor, mi
                  enim quis massa. Nullam urna wisi, blandit quis, porttitor pharetra,
                                                                                                  nisl mattis pede, nec sodales magna felis
                  dignissim nec, mauris. Curabitur posuere mattis ipsum. Nunc in metus
                                                                                                  porta mauris. Aliguam guis mi, Nulla facilisi,
                  aliquam felis bibendum tincidunt.
                                                                                                  Aliquam erat volutpat. Suspendisse
                  Nullam tincidunt volutpat orci. Morbi scelerisque. Aliquam vitae neque eget
                                                                                                  bibendum. Fusce ac sapien.
                  urna consectetuer suscipit. Nullam luctus augue guis est.
                  Donec mollis, sapien non tempus semper, nulla diam sagittis magna, sed
                  sodales ligula dui eu wisi. Cras at erat nec quam pharetra sagittis
                  Pellentesque lacinia nonummy mi. Morbi a ante in odio luctus vulputate
                  Sed tincidunt. Nullam tincidunt volutpat orci, Morbi scelerisque, Aliquam
                  vitae neque eget urna consectetuer suscipit. Nullam luctus augue quis est.
                  Donec mollis, sapien non tempus semper, nulla diam sagittis magna, sed
                  sodales ligula dui eu wisi. Cras at erat nec guam pharetra sagittis
                  Pellentesque lacinia nonummy mi, Morbi a ante in odio luctus vulputate.
                  Sed tincidunt.
                  position: relative; Footer
Done
```

Рис. 11.29. Применение абсолютного позиционирования в макете страницы

# 11.9. Проектирование асимметричного макета

# Проблема

Вы хотите создать "резиновый" асимметричный или с естественным расположением элементов макет, похожий на показанный на рис. 11.30.

# Решение

Сначала разметьте содержимое с помощью элементов div, используя атрибуты id с соответствующими значениями, отражающими их размещение на странице:

```
<div id="header">
[...]
</div>
<div id="columnSmall">
[...]
</div>
<div id="columnMain">
[...]
</div>
<div id="columnMedium">
[...]
</div>
```



Рис. 11.30. Асимметричное размещение содержимого

Далее используйте свойство position со значением absolute для каждой колонки, одновременно определяя позицию колонок с помощью значений свойств left и top, заданных в процентах. Для размещения фонового изображения также задайте значения в процентах:

```
body {
margin:5px 0 0 5px;
background-image: url(flower5.jpg);
background-position: 50% 35%;
background-repeat: no-repeat;
 }
#header {
position: absolute;
 left: 65%;
 top: 50%;
 width: 125px;
 font-size: small;
}
#columnSmall {
 position: absolute;
 left: 35%;
 width: 15%;
 top: 1%;
background: #fff;
font-size: small;
}
#columnMain {
 position: absolute;
```

```
left: 5%;
width: 45%;
top: 40%;
background: #fff;
text-align: justify;
border-width: 0;
font-size: large;
}
#columnMedium {
position: absolute;
left: 80%;
width: 20%;
top: 10%;
background: #fff;
}
```

Несмотря на представление о том, что традиционно в Web-сайтах применяются колоночные макеты, технология CSS позволяет Web-разработчикам использовать новые подходы для отображения своих документов. С помощью свойств position, top и left вы можете разделить содержимое на порции, задать стили для каждой из них и поместить их в точно заданных местах страницы.

Фоновое изображение будет перемещаться вместе с контентом при изменении размеров окна Web-обозревателя, поскольку для определения местоположения изображения использованы значения, заданные в процентах.

Вместо изменения значений свойств position, top и left вручную вы можете легко разместить элементы div с помощью приложения, применяющего технологию WYSIWYG (What You See Is What You Get — что видишь, то и получаешь), например, такого как Adobe Dreamweaver.

Если вы хотите создать асимметричный макет или макет с естественным расположением элементов и колонками фиксированной ширины вместо применения "резиновой" верстки, используйте единицы длины для указания Web-обозревателю точной позиции как информационного содержимого страницы, так и фонового изображения:

```
body {
  margin:5px 0 0 5px;
  background-image: url(flower5.jpg);
  background-position: -400px -200px;
  background-repeat: no-repeat;
}
#header {
  position: absolute;
  left: 500px;
  top: 200px;
  width: 125px;
  font-size: small;
}
```

```
#columnLeft {
position: absolute;
left: 200px;
width: 125px;
top: 10px;
background:#fff;
font-size: small;
}
#columnInnerLeft {
position: absolute;
left: 50px;
width: 375px;
top: 175px;
background: #fff;
text-align: justify;
border-width: 0;
font-size: large;
}
#columnInnerRight {
position: absolute;
left: 600px;
width: 150px;
top: 50px;
background: #fff;
}
```

# См. также

*Разд. 4.5* и *4.6* о размещении фоновых изображений на Web-странице; дополнительную информацию о программе Adobe Dreamweaver на сайте http://www.dremweaver.com.

# 11.10. Проектирование макетов, не зависящих от разрешения

# Проблема

Вы хотите создать Web-страницу, которая адаптируется к размеру окна Web-обозревателя.

# Решение

Примените предложенное Web-дизайнером Кэмероном Адамсом (Cameron Adams) решение Resolution JavaScript со страницы http://www.themaninblue.com/writing/ perspective/2006/01/19/ (рис. 11.31).

Предлагаемое решение основано на двух таблицах стилей: main.css, действующей как основа с базовым форматированием страницы, и features\_1024.css, предлагающей стилевое оформление, показанное на рис. 11.32.



Рис. 11.31. Страница Кэмерона Адамса о его скрипте

Если окно Web-обозревателя уже 1024 пикселов, применяется только базовая таблица стилей. Когда ширина окна превышает 1024 пиксела, используются дополнительные параметры.

Скрипт Resolution, хотя для этого и потребуются некоторые базовые знания языка JavaScript, можно усовершенствовать, включив разные точки останова для поддержки разного разрешения. Например, можно добавлять дополнительные стили при ширине окна Web-обозревателя, равной 800, 1024, 1440 и даже 1660 пикселам.

### Независимость от разрешения без применения JavaScript

Часть спецификации CSS3, именуемая media queries, содержит свойство, позволяющее выбирать применяемые правила CSS, основываясь на размерах окна Web-обозревателя.



Рис. 11.32. Разный внешний вид страницы с одинаковым контентом, зависящий от размера окна Web-обозревателя

Вставляя дополнительные условия в атрибут media элемента link, вы можете применять разные таблицы стилей в зависимости от ширины окна просмотра обозревателя, как показано на рис. 11.33:

```
<link media="screen and (max-width: 300px)" rel="stylesheet" href="ssr.css"
type="text/css" />
<link media="screen and (min-width: 300px) and (max-width: 750px)"
rel="stylesheet" href="msr.css" type="text/css" />
<link media="screen and (min-width: 750px)" rel="stylesheet" href="lsr.css"
type="text/css" />
```



Рис. 11.33. Применение разных стилей без использования JavaScript

Во время написания книги эти новые свойства поддерживались Web-обозревателями Firefox 3.5 и Opera 10.

# См. также

Свойства media queries в спецификации CSS3 на странице http://www.w3.org/TR/css3mediaqueries/; информацию центра разработчиков Mozilla Developer Center на странице https://developer.mozilla.org/En/CSS/Media\_queries; *pa3d. 14.3* о формировании разных таблиц стилей для мобильных устройств.

# Программистские трюки, искусственные приемы и выявление неисправностей

# 12.0. Введение

Исторически сложилось так, что разработчики, создавая дизайн-проекты для Webпространства, вынуждены применять программистские трюки и искусственные приемы из-за ограничений, накладываемых Web-обозревателями.

В середине 90-х годов XX века наблюдалось распространение таких искусственных приемов, к которым можно отнести однопиксельные GIF-изображения, теги font, вложенные таблицы — это лишь несколько трюков, первыми пришедших на ум. Несмотря на то, что спецификация CSS 2.1 стала рекомендованным стандартом еще в мае 1998 г., производители Web-обозревателей лишь недавно полностью включили этот стандарт в свои программные продукты. Временной разрыв между Web-обозревателями без под-держки технологии CSS и обозревателями с полной или почти идеальной реализацией CSS означает, что у некоторого количества Web-обозревателей, которыми пользуются большинство пользователей, — слабая поддержка технологии CSS.

Для преодоления дефектов в популярных Web-обозревателях со слабой поддержкой CSS Web-разработчики вынуждены были снова прибегнуть к применению программистских трюков и обходных путей для создания удачных дизайн-проектов Web-страниц.

Даже если проблемы можно решить, применяя более новые версии Web-обозревателей, у Web-разработчиков есть много причин, заставляющих их использовать программистские трюки и обходные пути для обеспечения подходящего представления страниц аудитории.

В отличие от Web-разработчиков большинство людей не обновляют автоматически свои Web-обозреватели при каждом появлении новой версии. Они склонны оставаться верными своему Web-обозревателю, установленному на их компьютере, поскольку он хорошо работает, и переходят на новый Web-обозреватель только при покупке нового компьютера.

Кроме того, отделы информационных технологий многих компаний блокируют системы, запрещая сотрудникам обновлять программные приложения самостоятельно.

В помощь Web-разработчикам, прилагающим массу усилий для отделки своих проектов, в этой главе описываются методы работы с Web-обозревателями, обладающими неполной поддержкой технологии CSS. В нее включены разные технические приемы, начиная с поиска и выявления ошибок в правилах CSS и заканчивая тестированием разных Web-обозревателей на одной машине.

# 12.1. Переопределение встроенных стилей

# Проблема

Вы хотите заменить встроенные стили стилями из внешней таблицы CSS:

```
<div style="font-size: 100px;">
Epsum factorial non deposit quid pro quo hic escorol.
</div>
```

# Решение

Во внешней таблице стилей используйте селектор атрибута (см. *разд. 2.5*), чтобы точно указать, у какой части Web-страницы есть встроенные стили:

div[style] {

```
font-size: 1.2em;
}
```

Затем примените объявление !important (см. *разд. 2.14*), чтобы придать правилу CSS больше веса:

```
div[style] {
  font-size: 1.2em !important;
}
```

# Обсуждение

Общее практическое правило, касающееся разных способов связывания правил CSS с Web-страницей (см. *разд. 2.11*), "побеждает правило, которое ближе всего к HTML-коду".

В большинстве случаев это верно. Однако специфичность (см. *разд. 2.15*) и источник правил CSS (см. *разд. 2.12*) могут использоваться для управления выбором правила, побеждающего в конфликтной ситуации. Это позволяет переопределять встроенные стили.

# См. также

Главу 2 с дополнительной информацией об основах стилевого оформления средствами CSS.

# 12.2. Диагностика ошибок CSS и проблем Web-обозревателей

# Проблема

Вы хотите найти ошибку в вашем коде или в способе отображения правил CSS вашим Web-обозревателем.

## Решение

Для того чтобы локализовать проблемы проектов с применением CSS, выполните следующие действия.

- 1. Проверьте корректность HTML-кода с помощью валидатора на сайте http:// validator.w3.org/.
- 2. Проверьте ваши правила CSS с помощью валидатора на сайте http://jigsaw.w3.org/ css-validator/.
- 3. Тщательно подберите значения свойств. Добавляйте новое правило CSS *в конец* таблиц (ы) стилей, применяя универсальный селектор и задавая свойства для всех элементов:

```
* {
  margin: 0;
  padding: 0;
}
```

4. Создайте рамки для всех блочных элементов:

```
* {
  margin: 0;
  padding: 0;
  border: 1px solid red;
}
```

- 5. Опробуйте разные значения свойств.
- 6. Заключите в комментарии правила CSS и/или свойства, вызывающие проблему. Извлекайте из комментариев свойства CSS поочередно, одно за другим до тех пор, пока проблема не возникнет снова. Информацию о том, как добавить комментарии в CSS, см. в *разд. 2.17*.
- 7. Поищите описание аналогичных проблем в Google и на сайте http://www. positioniseverything.net с хорошо документированной коллекцией ошибок CSS.

# Обсуждение

Ошибки при написании текста — обычная вещь. То же самое справедливо для процесса кодирования.

Исходя из личного опыта, 90 % проблем в проектах с CSS возникает из-за ошибок в синтаксисе CSS. Остальные проблемы вызваны тонкой настройкой параметров элементов или значений свойств CSS.

Если после выполнения действий, описанных в предыдущем разделе, проблема сохраняется, поищите в Google или на Web-сайте, посвященном ошибкам CSS, не написал ли кто-нибудь о похожей проблеме.

### Примечание

Вместо задания нулевых полей и отступов для некоторых элементов в начале проекта, применяйте специальную таблицу сброса стандартных стилей (CSS Reset).

### См. также

Дополнительную информацию о спецификации CSS в книге Эрика Мейера (Eric A. Meyer) "Cascading Style Sheets: the Definitive Guide" (Каскадные таблицы стилей: подробное руководство) (издательство O'Reilly).

# 12.3. Применение букмарклетов для поиска ошибок

### Проблема

Вы хотите для поиска и устранения проблем на ваших Web-страницах использовать сторонние Web-сайты.

### Решение

Как обсуждалось в *разд. 1.15* и 2.27, закладки Web-обозревателя с применением JavaScript позволяют легко проверять в валидаторе страницы, загруженные в Web-обозреватель в данный момент (рис. 12.1).



Рис. 12.1. Букмарклет для проверки корректности HTML

### Далее приведен код для HTML-букмарклета:

```
javascript:void(document.location='http://validator.w3.org/check?charset=
%28detect+automatically%29&doctype=Inline&ss=1&group=0&verbose=1&uri='+escape
(document.location))
```

### Далее приведен код букмарклета для валидатора CSS:

javascript:void(document.location='http://jigsaw.w3.org/cssvalidator/validator?profile=css21&usermedium=all&warning=1&lang=en&uri= '+escape(document.location))

Помимо проверки корректности HTML-кода и правил CSS есть букмарклеты для проверки других аспектов Web-сайта.

Например, консорциум W3C предоставляет сервис для поиска потерянных гиперссылок (link rot):

```
javascript:void(document.location=
'http://validator.w3.org/checklink?url='+escape(document.location))
```

Потерянная ссылка — это ссылка на вашем сайте, указывающая на другой, исчезнувший из Web-пространства ресурс, и возвращающая ошибку File Not Found (файл не найден).

# См. также

Букмарклеты Тантека Челика (Tantek Çelik) на странице http://tantek.com/favelets.

# 12.4. Применение расширений Web-обозревателя для поиска ошибок

# Проблема

Вы хотите использовать Web-обозреватель для поиска и устранения проблем в HTML-коде и правилах CSS.

# Решение

Для поиска ошибок HTML, CSS и JavaScript на Web-странице, загруженной в данный момент в окно просмотра, используйте свободно распространяемое расширение Firebug (рис. 12.2) для Web-обозревателя Firefox (см. http://getfirebug.com/).

Доступны также следующие средства разработки, созданные производителями Webобозревателей:

- ♦ Internet Explorer Developer Toolbar (http://www.microsoft.com/downloadS/ details.aspx?familyid=E59C3964-672D-4511-BB3E-2D5E1DB91038&displaylang=en);
- Opera Dragonfly (в Web-обозревателе выберите последовательность команд Tools | Advanced | Developer Tools (Инструменты | Дополнительно | Средства разработки);
- ◆ Inspect Element (Инспектировать элемент) в Web-обозревателе Safari (находясь в Web-обозревателе, щелкните любой элемент на Web-странице правой кнопкой мыши или левой кнопкой, но с нажатой клавишей <Ctrl> и выберите команду Inspect Element).

### Примечание

У корпорации Microsoft есть Web-страница, посвященная средствам разработки IE8 Developer Tools по адресу http://msdn.microsoft.com/en-us/library/dd565628(VS.85).aspx.



Рис. 12.2. Расширение Firebug, помогающее протестировать Web-страницы и найти в них ошибки

### См. также

Список рассылки рабочей группы Firebug Working Group на странице http://groups. google.com/group/firebug-working-group.

# 12.5. Исправление Internet Explorer 6

### Проблема

Вы хотите исправить некоторые общие проблемы в IE6.

### Решение

Для устранения проблем переполнения при очистке плавающих элементов в IE6 примените следующее объявление CSS в соответствующем селекторе: Для корректировки в IE6 положительных значений полей элементов примените следующее объявление CSS в соответствующем селекторе:

display: inline;

Для исправления проблемы многослойного расположения элементов, вызванной механизмом относительного позиционирования в IE6, примените следующее объявление CSS в соответствующем селекторе:

position: static;

Для добавления макета элемента, не имеющего макета в IE6, примените следующее объявление CSS в соответствующем селекторе:

zoom: 1;

### Обсуждение

Судя по недавнему опросу, проведенному Web-сайтом Digg.com среди своих пользователей (см. http://blog.digg.com/?p=878), большинство людей, использующих IE6, не ощущают потребности в его замене новой версией или же привязаны к этой версии обозревателя из-за действующих технических ограничений. Таким образом предложение пользователям IE6 контента, лишенного стилевого оформления, это всего лишь плохой сервис, а напоминание о необходимости обновления версии при невозможности сделать это, гораздо более неприятная вещь.

Однако это не означает, что работа для пользователей в IE6 должна выглядеть точно так же, как поиск информации в Интернете для пользователей Web-обозревателей IE8, Firefox 3.5, Safari 4 и их более свежих версий.

Определяя уровень поддержки, предоставляемой пользователям IE6, воспользуйтесь следующими критериями.

- Определите, какого рода поддержку вы хотите предложить пользователям IE6 (или других Web-обозревателей, составляющих меньшинство в ваших файлах оперативных журналов).
- Решите, что заставит вас прекратить поддержку IE6.
- Создайте отдельные фрагменты кода, обеспечивающие поддержку IE6 (т. е. отделите Web-страницу от искусственных приемов CSS, предназначенных для IE6).
- Оповестите пользователей IE6 о предпринимаемых вами шагах. Вы ведь хотите после внесения изменений сохранить их как посетителей сайта.

### См. также

Дополнительную информацию об ошибках Internet Explorer и методах их устранения на странице http://positioniseverything.net/explorer.html.

# 12.6. Исправление Internet Explorer 6 с помощью JavaScript

# Проблема

Вы хотите применить JavaScript, чтобы заставить IE6 для Windows лучше отображать страницы.

# Решение

Для исправления IE6 используйте скрипт для IE7 программиста Дина Эдвардса (Dean Edwards).

Поместите в элемент head вашей Web-страницы следующий код:

```
<!--[if lt IE 7]>
<script src="http://ie7-js.googlecode.com/svn/version/2.0(beta3)/IE7.js"
type="text/javascript"></script>
<![endif]-->
```

# Обсуждение

До выпуска корпорацией Microsoft версии IE7 Дин Эдвардс работал над скриптом, исправляющим изъяны CSS в Web-обозревателе IE6. Скрипт для IE7 исправляет поддержку селекторов, проблемы блочной модели, проблемы переполнения, проблемы альфа-прозрачности PNG-изображений и многое другое.

С помощью условных комментариев (см. *разд. 12.7*) вы обеспечите получение кода только теми Web-обозревателями, которые нуждаются в корректировке.

В предложенном решении код на языке JavaScript поставляется серверами Google. Если другие Web-сайты применяют такое же решение, код сохраняется в кэше на машине пользователя IE6, повышая производительность Web-обозревателя.

### Применение jQuery

Другой вариант решения на JavaScript, корректирующий реализацию CSS в IE6, — использование в каждом отдельном случае библиотеки jQuery для применения правил CSS. Дополнительную информацию см. в *разд. 14.4*.

### См. также

Формулу Натали Даун (Natalie Downe) и Джереми Кейта (Jeremy Keith), позволяющую определить, стоит ли использовать скрипт Дина Эдвардса (Dean Edwards) для поддержки пользователей IE6 на странице http://24ways.org/2008/the-ie6-equation.

# 12.7. Использование условных комментариев для передачи стилей в разные версии Internet Explorer

## Проблема

Вы хотите передать определенный код в разные версии Web-обозревателя Internet Explorer для Windows.

### Решение

Используйте специальные условные комментарии (conditional comments) Webобозревателя Internet Explorer для Windows корпорации Microsoft:

```
<!--[if IE]>
Вы видите это предложение, потому что применяете Internet Explorer.
<![endif]-->
```

Для передачи кода разным версиям Internet Explorer для Windows укажите версию Web-обозревателя:

```
<!--[if IE 5]>
Sbi Budute это предложение, потому что применяете Internet Explorer.5 
<![endif]-->
<!--[if IE 5.0]>
 Bbi Budute это предложение, потому что применяете Internet Explorer.5.0 
<![endif]-->
<!--[if IE 5.5]>
 Bbi Budute это предложение, потому что применяете Internet Explorer.5.5 
<![endif]-->
<!--[if IE 6]>
 Bbi Budute это предложение, потому что применяете Internet Explorer.6 
<![endif]-->
<![endif]-->
```

Для передачи кода в Internet Explorer для Windows версии 5 и более поздних используйте следующий код:

```
<!--[if gte IE 5]>
 Вы видите это предложение, потому что применяете Internet Explorer.5 и более
поздние версии 
<![endif]-->
```

Для передачи кода в Internet Explorer для Windows версии 5.5 и более ранних используйте следующий код:

```
<!--[if lte IE 5.5]>
 Вы видите это предложение, потому что применяете версию Internet Explorer
с номером, меньшим или равным 5.5
<![endif]-->
```

Для передачи кода в Internet Explorer для Windows версии ниже IE6 используйте следующий код:

```
<!--[if lt IE 6]>
 Вы видите это предложение, потому что применяете версию Internet Explorer
с номером ниже 6
<![endif]-->
```

Корпорация Microsoft разработала собственную оригинальную систему комментариев для передачи особого HTML-кода в разные версии своего Web-обозревателя Internet Explorer для Windows.

Этот особый код можно применять только для размещения обычного HTML-кода между условными операторами. Это означает, что с помощью условных комментариев можно особым образом передавать нужные правила CSS.

Например, для передачи таблицы стилей, предназначенной для Internet Explorer 5.*x*, поместите тег link с указанием таблицы стилей между двумя условными комментариями:

```
<link rel="stylesheet" type="text/css" media="screen, presentation"
href="/_assets/css/screen/screen.css" />
<link rel="stylesheet" type="text/css" media="aural"
href="/_assets/css/aural.css" />
<!--[if lt IE 6]>
<link rel="stylesheet" type="text/css" media="screen, presentation"
href="/_assets/css/screen/ie.css" />
<![endif]-->
```

Встроенные стили тоже можно поместить между условными комментариями:

```
<!--[if lt IE 6]>
<style type="text/css">
    h1 {
    font-family: Verdana, Arial, Helvetica, sans-serif;
    font-size: 36px;
}
</style>
<![endif]-->
```

Условные комментарии могут применяться в сочетании с интеллектуальной системой программистских трюков и искусственных приемов (intelligent hacking system). См. подробную информацию об установке такой системы в *разд. 12.9.* 

#### Примечание

В случае использования условных комментариев существует разница между версиями IE 5 и IE 5.0.

Для передачи кода только в версию Web-обозревателя Internet Explorer 5.0 используйте обозначение IE 5.0. Для предоставления кода во все версии Internet Explorer с номером основного выпуска 5 используйте IE 5.

Сокращения lt и gt означают "меньше чем" и "больше чем" соответственно. В то время как lte и gte означают "не больше" и "не меньше".

### См. также

Статью в MSDN (Microsoft Developer Network — собрание документов корпорации Microsoft, содержащее сведения обо всех ее разработках) об условных комментариях на странице http://msdn.microsoft.com/en-us/library/ms537512(VS.85) aspx.

# 12.8. Применение фильтров CSS для передачи правил в практически любой Web-обозреватель

# Проблема

Вы хотите для устранения ошибок визуализации связать правила CSS с Web-обозревателями, отличными от Internet Explorer.

			G		(	cer	itrie	cle	: c	ss	filte	rs (	css	had	ks)	1														
■ Eack Forward Reload Stop Home Print 1Password @ http://centricle.com/ref/css/f 🗟☆▼ - 🔥 Google Q											٩																			
centricle : css filters (css hacks)	Θ	1	) c	hris	toph	er S	ichr	mit	t : C	Desi	igne	r,	٢	ŀ		Гec	hnol	ogy	Во	oks	, Te	ch	Cor	fer.	. 6	3	+			-
Will the browser a	ap	oly	y t	h	e r	ul	e	(s	)?	,																				
turn on interactive highlight			-																											
	Windows														Mad	o OS	x						Ma	Other						
	IE Mz Ns Op						•	iC	iC IE Mz Ns Om					Op Sf			f	IE Mz					Ор	Ко						
	7	65	.5	54	1	7	6	4	8	7	6 5	2	5	1	7	6	4	8	7	6	5 2	2 5	4	1	7	6	4	6 5	3	
voice-family:"\"}\"; voice-family:inherit; property:value;	Y	Y	N	NN			Y	N	Y		YY		Y	Y	Y	Y		Y	Y		h	r   Y	N			Y	N	۲		
p\roperty:value;	Y	Y	N	N N	Y	Y	Y	N	Y	Y	YN	N	Y	Y	Y	Y	N	Y	Y	Y	4 N	( Y	N	Y	Y	Y	N	YN	I N	
/*/*/property:value;/* */	Y	Y	Y	ΥY	Ŷ	Y	Y	N	Y	Ŷ	ΥY	Ŷ	Y	Y	Y	Y	Y	Y	Y	Ϋ́	( )	Y	Y	Y	Y	Y	Ν	Ŷ١	Y	
/*/*//*/property:value;/* */	N	N	N	NN	N	N	N	Y	N	N	N Y	N	N	N	N	N	N	N	N	N		N	N	N	N	N	Y	N	N	
div#test	Y		Y I	Y	Ŷ	Y	Y	N	Y	Ŷ	YY	Ŷ	Ŷ	Ŷ	Y	Y	Y	Y	Ŷ				Ŷ	Ŷ	Y	Y	N	Y N	Ŷ	
nead:first-child+body div	v	Ν	N		Y	Y	Y	N.	Y	Y	Y Y	N	Y	Y	Y	Y.	Y	Y	Y	Y			N	Y	Y	Y	N	Y A	Y Y	
btml(xmlns) div	Ŷ	N	N	NN	Y	Y	Y	N	Y	Y	YY	N	N	Y	Y	Y	N	Y	Y	Y			I N	Y	Y	Y	N	Y	Y	
@import 'styles.css'	Y	Y	Y	YN	Y	Y	Y	N	Y	Y	YY	Y	N	Y	Y	Y	Y	Y	Y	Y	( )	/ N	I N	Y	Y	Y	N	Y		
@import "styles.css"	Y	Y	Y	Y N	Y		Y	N	Y	Y	YY	Y	Y	Y	Y	Y	Y	Y	Y	Y	( )	r Y	N	Y		Y	N	Υ۱		
@import url(styles.css)	Y	Y	Y	YY	Y		Y	N	Y	Y	YY	Y	Y	Y	Y	Y	Y	Y	Y	Y	( )	r Y	Y	Y		Y	N	Υ۱		
<pre>@import url('styles.css')</pre>	Y	Y	Y	YY	Y	Y	Y	N	Y	Y	YY	Y	N	Y	Y	Y	Y	Y	Y	Y١	( )	r N	N	Y	Y	Y	N	Υ۱	Ý	
<pre>@import url("styles.css")</pre>	Y	Y	Y	YY	Y	Y	Y	N	Y	Y	YY	Y	Y	Y	Y	Y	Y	Y	Y	۲ľ	( )	r Y	Y	Y	Y	Y	N	۲	Y Y	
<pre>@import "null?\"\{"; @import "styles.css";</pre>	Y	Y	Y	N N	Y	Y	Y	N	Y	Y	YY	Y	Y	Y	Y	Y	Y	Y	Y	Y	()	r Y	N	۲	Y	Y	N	۲	Y	
<pre>@media all{/* rules */}</pre>	Y	Y   '	Y I	YN	Y	Y	Y	N	Y	Y	YY	N	N	Y	Y	Y	N	Y	Y	Y   1	()	r N	N	Y	Y	Y	N	۲	Y Y	
<li>k media="all"&gt;</li>	Y	Y	Y	ΥY	Y	Y	Y	N	Y	Y	YY	Y	Y	Y	Y	Y		Y	Y	Y	()	r Y	Y	Y	Y	Y	N	Υ١	Y Y	
<li>k media="All"&gt;</li>	Y	Y	Y	YY	Ŷ	Y	Y	N	Y	Y	YY	Y	Y	Y	Y	Y	N	Y	Y	Y	()	( Y	Y	Y	Y	Y	N	Υ١	Y Y	
* html div	Ν	Y	Y	ΥY	N	N	N	N	N	N	NN	N	Y	N	N	N	N	N	N	NI	1	Y	Y	N	N	N	N	NN	I N	
*+html div	Ŷ	N	N	r N	N	N	N	N	N	N	NN	N	N	N	N	N	N	N	N		• •		N	N	N	N	N		N	
div{property:value}	Y	Y	N	N N	Y		Y	N	Y	Y	YY	Y	Y	Y	Y	Y		Y	Y		()	Y	N			Y	N	۲N	N	
div{property:value}	Y	Y		Y Y			Y	Y	Y		YY		N	Y	Y	Y		Y	Y		h	r N	Y			Y		۲		
html*#test	Y	Y	Y	N N	Y		Y	N	N	N	N N	N	Y	Y	Y	Y	N	N	Ν	N	( )	r Y	N	Y		Y	N	? 1	N	
_property:value	Ν	Y	Y	YY	Ν	Ν	N	N	N	N	N N	N	N	N	Ν	Ν	N	N	Ν	N	4 1	I Y	Y	Ν	N	Ν	Ν	N	I N	
*property:value	Y	Y	Y	YN	N	N	N	Ν	Ν	N	N N	N	N	Ν	Ν	Ν	N	Ν	Ν	NI	1	I N	N	Ν	N	Ν	Ν	N	I N	
<pre></pre>	N	N		Y N	N	N	N	N	N	N	N N	N	N	N	N	N	N	N	N	N	ŀ	4 N	N	N	N	N	N	N	N	
<pre>@media tty {     i{content:"\";/*" "*/}};     @import 'styles.cs';         (;)/*";)         //* */</pre>	N	N	N	Y N	N	N	N	N	N	N	NN	N	N	N	N	N	N	N	N	N 1	• •	4 N	N	N	N	N	N	N 1	I N	
<pre>@media tty {     i{content:"\";/*" "*/}}@m;     @import 'styles.css'; /*";}         //* */</pre>	N	N	Y	N N	N	N	N	N	N	N	N N	N	N	N	N	N	N	N	N	N	• •	4 N	N	N	N	N	N	N 1	N	
/*\*//*/ @import "styles.css"; /**/	N	N	N	N N	N	N	N	N	N	N	N N	N	Y	N	N	N	N	N	N	N	•	4 Y	N	N	N	N	N	N	N	
	7	6 5	.5	5 4	1	7	6	4	8	7	6 5	2	5	1	7	6	4	8	7	6	5 2	2 5	4	1	7	6	4	6 5	3	
			IE		Mz	<u> </u>	Ns			Op	•	iC	IE	IE Mz Ns Om Op								f	IE	Mz	Ļ	Ns		Ор	Ko	
					Win	dow	/s	_								Mad	C OS	x						Ma	icin	tos	h		Other	1
		_	_	_	_	_	_	-				_	_		_	_		_	_		_			_	_			_		•

Рис. 12.3. Матрица фильтров CSS

### Решение

Используйте *фильтры CSS* для того, чтобы заставить правила CSS применяться при отображении страницы в конкретном Web-обозревателе.

После определения, к какому или каким Web-обозревателям применять "CSS-исправления", сошлитесь на ссылочную таблицу фильтров CSS на странице http:// centricle.com/ref/css/filters/, показанную на рис. 12.3.

Щелкните кнопкой мыши селектор CSS, чтобы указать, как применять фильтр к вашим таблицам стилей и прочитать сведения о фильтре.

# Обсуждение

Фильтры — это искусственные приемы, использующие "дыры" в поддержке CSS Webобозревателем. В зависимости от типа фильтра эти приемы могут указывать на один или два Web-обозревателя или на целый ряд Web-обозревателей.

#### Примечание

Не путайте фильтры CSS со свойством filter, специализированным свойством CSS, pasработанным корпорацией Microsoft.

### См. также

Статью о фильтрах CSS в Википедии на странице http://en/wikipedia.org/wiki/ Css\_filter.

# 12.9. Установка интеллектуальной системы передачи CSS для современных Web-обозревателей

### Проблема

Вы хотите разработать систему, отделяющую настоящие правила CSS от правил, применяемых для реализации программистских трюков и искусственных приемов.

### Решение

Свяжите таблицу стилей с Web-страницей:

```
<!--[if gte IE 7]><!-->
<link rel="stylesheet" type="text/css" media="screen, projection"
href="screen.css" />
<!--<![endif]-->
```

В таблице стилей, помещенной в файл screen.css, импортируйте три отдельные таблицы стилей: таблицу сброса стандартных стилей CSS Reset, основную таблицу стилей для современных Web-обозревателей и таблицу стилей, содержащую искусственные и обходные приемы для более старых версий IE:

```
/* Import style sheets, hiding from IE/Mac \*/
@import url("reset.css");
@import url("csscookbook.css");
@import url("ie.css");
/* End of import hide */
```

# Обсуждение

Хранение таблиц стилей, разделенных в соответствии с поддержкой того или иного Web-обозревателя, обладает несколькими преимуществами. Во-первых, основная таблица освобождается от любых трюков и хитростей.

Во-вторых, хранение в отдельных файлах трюков и искусственных приемов, специфичных для каждого Web-обозревателя, позволяет легко удалить правила CSS, когда придет время прекратить поддержку конкретного Web-обозревателя.

Метод, предлагаемый в решении, пользуется слабой реализацией CSS в Internet Explorer для Macintosh, скрывая от этого Web-обозревателя таблицы стилей. Несмотря на то, что Microsoft больше не поддерживает этот обозреватель, проверьте оперативные журналы (logfile) вашего Web-сайта, чтобы увидеть, есть ли у вас трафик к дополнительным правилам CSS, оправданно скрываемым от пользователей IE Mac. (Вероятнее всего, вы можете удалить трюк с комментариями, но лучше соблюдать осторожность, чем сожалеть о случившемся.)

Перечень фильтров CSS для конкретных Web-обозревателей см. в разд. 12.7.

### Серверное решение

Методика, обсуждавшаяся в предложенном решении, использует для передачи таблиц стилей искусственные приемы на базе CSS. Другой подход заключается в применении серверного решения. Web-разработчик Марк Пильгрим (Mark Pilgrim) придумал решение на базе модуля mod\_rewrite сервера Apache.

Благодаря определению пользовательского HTTP-агента, каждый обозреватель получает собственную таблицу стилей в дополнение к основной таблице. Более подробную информацию об этом методе см. на сайте http://diveintomark.org/archives/2003/01/16/the\_one\_ive\_never\_tried.

### См. также

Статью Молли Хольцшлаг (Molly Holzschlag) об управлении искусственными программистскими приемами на странице http://www.peachpit.com/articles/article.aspx? p=170511.

# 12.10. Тестирование дизайна сайта на нескольких платформах с помощью единственного компьютера

# Проблема

Вы хотите проверить ваш Web-сайт в нескольких Web-обозревателях, предназначенных для разных OC, но у вас есть доступ только к одному компьютеру.

## Решение

Запуск на компьютере эмуляторов может помочь снизить затраты, поскольку вам не понадобится несколько рабочих станций.

Существуют следующие эмуляторы для пользователей РС.

- ◆ *Knoppix* это операционная система Linux, находящаяся на загрузочном CD, что избавляет от необходимости ее установки. Дополнительную информацию см. на сайте http://www.knoppix.org/.
- ◆ Виртуальная машина *VMWare Workstation* позволяет устанавливать виртуальные операционные системы для запуска поверх операционной системы Microsoft. Несмотря на то, что программное обеспечение нуждается в установке, оно не требует перезапуска компьютера при каждом тестировании Web-сайта. Дополнительную информацию см. на сайте http://www.vmware.com/products/ws/.
- OC Debian Linux можно установить на компьютере в отдельный раздел диска, что позволит загружать либо OC Linux, либо OC Windows. Дополнительную информацию см. на странице http://www.aboutdebian.com/dualboot.htm.

Для пользователей Macintosh.

- ♦ Программа Parallels Desktop позволяет устанавливать виртуальные операционные системы, включая Windows XP Professional, Windows XP Home и Windows 2000 Professional и следовательно, разные версии Web-обозревателя Internet Explorer. Дополнительную информацию см. на странице http://www.microsoft.com/mac/products/virtualpc.aspx.
- ◆ Утилита *Boot Camp* приложение компании Apple, позволяющее устанавливать операционную систему Windows в раздел диска на компьютерах Mac на базе Intel. Дополнительную информацию см. на сайте http://www.apple.com/support/bootcamp/.

Для пользователей рабочих станций Linux.

◆ *Wine* — это программное обеспечение с открытым кодом, представляющее собой реализацию API (Application Programming Interface, прикладной программный интерфейс) Windows, который выполняется поверх операционных систем Mac OS X и Linux. Дополнительную информацию см. на сайте http://www.winehq.com/.

Для достижения совместимости дизайн-проектов на базе CSS с разными платформами и Web-обозревателями необходимы тестирование и проверка Web-сайтов на максимально возможном количестве ПК. Для того чтобы выполнить их, не выходя за рамки бюджета, нужно установить несколько операционных систем на ваш компьютер.

После этого установите на новую систему Web-обозреватель. Сделать это быстро поможет архив Web-обозревателей на сайте http://browsers.evolt.org/.

### Применение удаленного сервиса захвата экрана

BrowserCam — Web-сервис, создающий моментальный снимок экрана. Заполните форму, указав ссылку на Web-страницу и несколько конфигураций Web-обозревателей и операционных систем, которые вы хотите увидеть. Затем сервис предоставит вам для проверки снимки экранов Web-сайта в этих системах. Дополнительную информацию см. на сайте http://www.browsercam.com/.

Сервис Adobe BrowserLab отображает Web-страницы во Flash-ориентированной среде. Во время написания книги Adobe BrowserLab показывал отображения страниц в Webобозревателях Firefox 2 и его более поздних версиях, IE6, IE7 и Safari 3 для Macintosh. Несмотря на то, что сервис бесплатный, для его использования у вас должна быть учетная запись Adobe ID. См. https://browserlab.adobe.com/.

Сервис BrowserShots (см. http://browsershots.org/) обеспечивает свободную, но медленную обработку моментальных снимков экранов (скриншотов) в разных Webобозревателях. Для ускорения обработки требуется символическая плата.

# См. также

Дополнительную информацию об установке нескольких Web-обозревателей на один компьютер на странице http://www.thesitewizard.com/webdesign/multiplebrowsers.shtml.

# 12.11. Проверка Web-сайта с помощью текстового Web-обозревателя

# Проблема

Вы хотите протестировать Web-сайт с помощью текстового Web-обозревателя.

# Решение

Воспользуйтесь интерактивным Web-средством, таким как Lynx Viewer (http:// www.delorie.com/web/lynxview.html), эмулирующим текстовый Web-обозреватель.

Поместите в корневую папку вашего Web-сайта файл с именем delorie.html (файл может быть пустым). Файл извещает Web-приложение о том, что есть сайт для просмотра и визуализации в Web-обозревателе с помощью интерактивного сервиса. Затем введите URL вашего сайта и проверьте результат в интерактивном режиме (рис. 12.4).

```
LynxView: http://csscookbook.com/
       ( >
                 (C)
                             ) 🍙 🖨 (1P)
 .
                         (X)
                                                        (d httj 😭 🔻 ) - 🚷 (Google
                                                                                          Q
                                Home Print
                                             1Password
 Back
      Forward
                 Reload
                        Stop
d LynxView: http://cssc... 🛞
                                Christopher Schmitt : ... 😧 🛛 🖓 Technology Books, Te...
                                                                                  ⊕ +
                   Services provided by Delorie Software and its sponsors (the ad above)
                                        Lynx Viewer:
     http://csscookbook.com/
              This service is intended to be used only by content developers, on their own pages.
     The CSS Cookbook is filled with applied, practical solutions to overcome everyday hurdles of CSS-enabled design. Take time and
     frustration out of the CSS learning process with these pre-packaged
     solutions.
     Dave Shea, author of The Zen of CSS Design
Links & Downloads
     * View the table of contents at orielly.com
     * View the book's errata at orielly.com
     * Download free, expanded book chapter PDF, ~6.6MB
* Download the code and examples from the book ZIP, ~2.5MB
Buy the Book
     * Buy the book from United States flag Amazon.com
     * Buy the book from United Kingdom flag Amazon.co.uk
        Buy the book from Canada flag Amazon.ca
        Buy the book from German flag Amazon.de
        Buy the book from Japan flag Amazon.co.jp
   Already own the CSS Cookbook? Please write a review at Amazon.com
   letting others know how the book was useful for you.
About Christopher Schmitt
   author mugshot
   Christopher Schmitt is the founder of Heatvision.com, Inc., a small
   new media publishing and design firm.
   An award-winning web designer who has been working with the Web since
   1993, Christopher interned for both David Siegel and Lynda Weinman in
   the mid 90s while he was an undergraduate at Florida State University
   working on a Fine Arts degree with an emphasis on Graphic Design.
   Afterwards, he earned a Masters in Communication for Interactive and
New Communication Technologies while obtaining a graduate certificate
   in Project Management from FSU's College of Communication.
   At conferences such as Web Visions and SXSW, Christopher has given
   talks demonstrating the use and benefits of practical standards-based
   designs. He is the list moderator for Babble, a mailing list community
   devoted to advanced web design and development topics.
```

Рис. 12.4. Вид Web-сайта в Lynx Viewer

### Обсуждение

Lynx — настоящий текстовый Web-обозреватель. Вместо запуска Web-эмулятора можно загрузить программу Lynx и установить ее на вашем персональном компьютере. Дополнительную информацию см. на странице http://lynx.isc.org/.

### См. также

Краткую историю программы Lynx на Web-странице http://people.cc.ku.edu/~grobe/ early-lynx.html.

# Проектирование с помощью CSS

# 13.0. Введение

Несмотря на то, что Web-разработчики часто тратят много времени на исправление ошибок Web-обозревателя и чтение сообщений о последних программистских трюках, созданных гуру, не следует забывать о том, что мы, прежде всего и главным образом, — дизайнеры, а технология CSS — всего лишь способ воплощения дизайнерских идей в жизнь.

CSS — превосходная технология, позволяющая завладеть вниманием посетителей Web-сайта. С помощью CSS вместо трюков с HTML-таблицами и разрезания на части изображений для создания привлекающих взгляд проектов вы можете пойти дальше, имея корректную разметку и экономя на размерах файлов за счет удаления из них избыточного HTML-кода и изображений. В общем вы можете делать то, что и следует делать профессиональному дизайнеру: добиваться максимального эффекта с минимальными затратами.

На базовом уровне разработчик может получить все существующие сведения о синтаксисе CSS и технических ограничениях технологии. Но никогда не стоит забывать о том, что код всего лишь воплощает дизайн. В основе CSS — изобразительный язык (visual language), поэтому возникает необходимость понять, по крайней мере, хотя бы отчасти, как реализовать дизайнерские представления средствами CSS.

С учетом вышесказанного в этой главе объясняется, как проектировать с помощью CSS. Особое внимание уделяется нескольким средствам привлечения внимания с помощью методов, ставших возможными благодаря CSS, например, как остановить взгляд посредством контраста, применять шрифт очень большого размера, создавать из цитат выноски с текстом и использовать разные графические форматы для создания связных презентаций.

# 13.1. Существенное увеличение размера шрифта

### Проблема

Вы хотите привлечь внимание к Web-странице, увеличив часть текста (рис. 13.1).

### Решение

Увеличьте размер заголовка так, чтобы он был непропорционален остальному тексту на странице. Сначала используйте следующий HTML-код:



Рис. 13.1. Пример значительного увеличения размера шрифта

Затем воспользуйтесь следующим кодом CSS:

```
h1 {
  font-size: 17em;
  margin: 0;
  padding: 0;
  text-align: center;
  font-family: Arial, Verdana, Helvetica, sans-serif;
}
```

# Обсуждение

Очевидно, что бросается в глаза любой элемент, который значительно больше окружающих его. Этот подход делает представление страницы более динамичным по сравнению с макетом, в котором применяются все элементы одинакового размера.

Следовательно, если вы хотите привлечь внимание к какой-либо области Webстраницы, один из возможных способов — применение шрифта чрезвычайно большого размера.

В данном примере для слова "Hi." задан размер шрифта 17 em. В свойстве font-size единица измерения кегельная (em) равна тому размеру шрифта, который задан в контейнере.

Таким образом, 17 ет равно размеру шрифта по умолчанию, увеличенному в 17 раз. Теоретически нет ограничений на степень увеличения размера шрифта, но на практике разные Web-обозреватели в какой-то момент исчерпывают ресурс. Не у всех есть достаточно большой монитор, чтобы увидеть шрифт высотой в 1 милю (или 63 360 дюймов):

```
h3 {
  font-size: 63360in;
}
```

### См. также

В *разд. 3.6* специальные единицы измерения и размеры шрифта; статью "The Elements of Text and Message Design and Their Impact on Message Legibility: Literature Review" (Текстовые элементы и информационный дизайн и их влияние на удобочитаемость представляемой информации: литературное обозрение) в журнале для дизайнеров Journal of Design Communication на странице http://scholar.lib.vt.edu/ejournals/JDC/Spring-2002/bix.html; единицы длины (включая единицы еm) в спецификации CSS 2 на странице http://www.w3.org/TR/REC-CSS2/syndata.html#length-units.

# 13.2. Создание неожиданного сочетания

### Проблема

Вам необходимо привлечь внимание читателей с помощью двух элементов, которые кажутся не подходящими друг другу.

### Решение

Визуально поместите один элемент внутрь другого. На Web-странице, показанной на рис. 13.2, изображена Земля, находящаяся на волосок от астероида, при этом вид Земли из космоса помещен на изображение бильярдного стола с шаром и кием.

HTML-код для этой страницы прост:

```
<h2><span class="no">Earth News</span></h2>
Earth escapes potential impact with killer asteroid;
will we escape the next one in 2014? <a href="more.html">Read
more</a></p
```

В коде CSS вставьте фотографию принадлежностей для игры в бильярд в элемент body и поместите ее в верхний левый угол страницы. Затем используйте метод замещения



Рис. 13.2. Изображение Земли совмещено с изображением бильярда

изображением, обсуждавшийся в *разд. 4.20*, для вставки фотографии Земли в элемент h2:

```
<style type="text/css">
body {
background-color: #009E69;
margin: 0;
background-image: url(billiard.jpg);
background-repeat: no-repeat;
}
h2 {
background-image: url(earth.gif);
position:absolute;
width:126px;
height:126px;
 z-index:1;
 left: 166px;
 top: 69px;
}
```

```
.no {
  display: none;
}
p {
  width: 120px;
  margin: 260px 100px 0 170px;
  font-family: Verdana, sans-serif;
  font-size: small;
  font-weight: bold;
}
</style>
```

Отличный способ завладеть вниманием — показать что-то неожиданное. Остроумное комбинирование двух различных элементов на одном изображении может заставить зрителей обратить на него внимание (рис. 13.3) или просто подчеркнуть назначение содержимого.



Рис. 13.3. Соединение фотографий ребенка и мужчины

В приведенном ранее примере используются два изображения — на одном показаны бильярдные кий и шар, а на другом — Земля. Первое изображение поместили как фоновое изображение элемента body. Изображение Земли вставили как фон в элемент h2 и переместили с помощью абсолютного позиционирования. Затем оно было скомпоновано с изображением бильярдных принадлежностей.

### См. также

*Разд. 4.22* о комбинировании изображений разных форматов; *разд. 13.3* о комбинировании разнородных элементов.

# 13.3. Комбинирование разнородных элементов для создания контраста

# Проблема

Вы хотите создать контраст на Web-странице, соединяя два разных элемента, например шрифты с засечками и без них (рис. 13.4).



Рис. 13.4. Разные шрифты, соседствующие в одном заголовке

### Решение

Используйте разные гарнитуры шрифтов в одном заголовке. Сначала подготовьте разметку, чтобы можно было менять свойства font:

```
<h2>Cross<span>i</span>ng <span>Over</span></h2>
<h4>Sen. Jane Gordon (<span>I</span>-Utah) bolts GOP;
<br />changes part<span>i</span>es to be
<span>I</span>ndependent</h4>
```

Затем откорректируйте правила CSS для элемента span, чтобы создать комбинацию гарнитур:

```
body {
  margin: 25% 10% 0 10%;
}
h2 {
  font-size: 2em;
  font-weight: bold;
```

```
font-family: Arial, Verdana, Helvetica, sans-serif;
 text-transform: uppercase;
 text-align: center;
padding: 0;
margin: 0;
}
h2 span {
 font-family: Times, "Times New Roman", Georgia, serif;
 font-size: 1.1em;
 font-weight: normal;
 }
h4 {
margin: 0;
padding: 0;
 font-size: 1.25em;
 font-weight: bold;
 font-family: Arial, Verdana, Helvetica, sans-serif;
 text-transform: uppercase;
 text-align: center;
}
h4 span {
 font-family: Times, "Times New Roman", Georgia, serif;
 font-size: 1.1em;
font-weight: normal;
}
```

570

Комбинация непохожих элементов создает визуальный контраст. В этом примере формируют контраст использованные в заголовке разные характеристики гарнитур шрифтов с засечками и без них. Но создать контраст можно и с помощью изображений. Например, в данном примере вы могли бы сгруппировать символы Демократической и Республиканской политических партий и поместить их рядом друг с другом. Или можно пойти дальше для достижения большего символического контраста, поместив фотографии двух видов вечеринок рядом: на одной изображено многолюдное сборище в клубе, а на другой — девушка, дующая в свисток над маленьким кексом с торчащей в нем горящей свечкой.

# См. также

Разд. 4.22 о комбинировании изображений разных форматов.

# 13.4. Привлечение взгляда с помощью контраста

# Проблема

Вы хотите создать ощущение глубины и движения в тексте. На странице с четырьмя почти одинаковыми абзацами трудно понять, какой из них просматривать первым.

Если вы измените в определенном направлении размер шрифта в колонках (уменьшая его справа налево), то будете вести взгляд читателя (рис. 13.5).



Рис. 13.5. Четыре почти идентичных абзаца, которые затем изменены для усиления контраста

# Решение

Для управления взглядом читателя измените размер шрифта, добавив следующие правила CSS:

```
/* PasMep mpu@ra */
#layer4 {
  font-size: .7em;
  line-height: 20px;
}
#layer3 {
  font-size: 1em;
  line-height: 20px;
}
#layer2 {
  font-size: 2em;
  line-height: 10px;
}
```

```
#layer1 {
  font-size: 3em;
  line-height: 10px;
}
```

Контраст возникает при наличии очевидной разницы между двумя элементами. Если на странице нет контрастных участков, читатель не знает, что важно на ней. Управляя визуальными свойствами элемента, можно создать резкое отличие в отображении двух похожих элементов. К таким визуальным характеристикам можно отнести следующие:

- размер;
- ♦ цвет;
- форму;
- положение на странице;
- направление;
- плотность.

У правильно размеченного содержимого есть присущий ему стиль, т. к. если нет другой таблицы, Web-обозреватель использует собственную таблицу стилей для визуализации содержимого. Заголовки, такие как элемент h1, выводятся более крупным шрифтом с жирным начертанием и отделяются от абзацев. Это шрифтовое различие создает контраст и помогает читателю понять смысл документа.

Без акцентов, обеспечиваемых таблицами стилей, глаз читателя бесцельно блуждает по документу.

# См. также

Создание других эффектных вариантов контраста на интернациональном Web-сайте Lighthouse (http://www.lighthouse.org/color\_contrast.htm).

# 13.5. Проверка достаточности цветового контраста

# Проблема

Вы хотите убедиться в том, что два цвета достаточно контрастны.

# Решение

Используйте Luminosity Contrast Ratio Analyser (анализатор уровня цветового контраста) от JuicyStudio.com на странице http://juicystudio.com/services/luminositycontrastratio.php.

Введите значения цвета в программу проверки и нажмите кнопку Calculate Luminosity Contrast Ratio (Вычислить уровень цветового контраста).

### 572

Вместе с образцом двух цветов выводится резюме о том, превысили вы уровень 2 или 3 цветового различия или нет. На рис. 13.6 приведен пример с сообщением о том, что заданная комбинация превысила уровень 3.



Рис. 13.6. Результаты проверки уровня цветового контраста

# Обсуждение

Рекомендации консорциума W3C, касающиеся легкости просмотра Web-содержимого, утверждают, что для разборчивости текста дизайнеры должны создавать контент, легко воспринимаемый на цветном фоне.

Когда оттенок цвета шрифта близок к цвету фона, текст становится неразборчивым. Для создания легкочитаемого текста у комбинации цветов должна быть достаточно большая контрастность за счет значительного удаления их друг от друга в частотном спектре или, другими словами, текст должен быть значительно темнее или светлее фона.

### Уровни цветового различия

Для того чтобы цвета преодолели второй уровень цветового различия, отношение цветового контраста должно быть как минимум 5:1. Это означает, что один цвет должен быть, по крайней мере, в 5 раз темнее или светлее другого цвета.

Для цветов, превышающих третий уровень цветового различия, отношение цветового контраста должно быть как минимум 10:1.

### См. также

Объяснение JuicyStudio.com предлагаемого алгоритма вычисления уровня цветового контраста на странице http://juicystudio.com/article/luminositycontrastratioalgorithm.php.

# 13.6. Выделение цитаты с помощью эффектных кавычек

### Проблема

Вы хотите привлечь особое внимание к цитате, используя увеличенные, с жирным начертанием кавычки (рис. 13.7).



Рис. 13.7. Стилевое оформление цитаты

### Решение

Сначала создайте код разметки для цитаты (рис. 13.8):

```
<blockquote>
There is a tendency for things to right themselves.
<cite>Ralph Waldo Emerson</cite>
</blockquote>
```

Затем используйте правила CSS для стилевого оформления цитаты:

```
blockquote {
  padding: 0;
  margin: 0;
  text-align: center;
}
p {
  font-size: 1em;
```

```
padding-bottom: 3em;
text-transform: lowercase;
font-family: Georgia, Times, "Times New Roman", serif;
margin: 0;
padding: 0;
}
cite {
display: block;
text-align: center;
}
```



Рис. 13.8. Обычное отображение цитаты

И наконец, в цитате используйте псевдоэлементы :before и :after для стилевого оформления кавычек и вставки тире длиной 1 ет (горизонтального штриха, равного текущему размеру шрифта) перед именем автора цитируемого источника:

```
blockquote p:before {
 content: "\201C";
 font-size: 1.2em;
 font-weight: bold;
 font-family: Georgia, Times, "Times New Roman", serif;
}
blockquote p:after {
 content: "\201D";
 font-size: 1.2em;
 font-weight: bold;
 font-family: Georgia, Times, "Times New Roman", serif;
}
cite:before {
 content: "\2014 ";
}
cite {
display: block;
text-align: center;
}
```

### Обсуждение

Псевдоэлементы — это конструкции селекторов, которые Web-обозреватели используют для выбора и последующего стилевого оформления фрагментов Web-страницы,
которые нельзя пометить стандартными средствами разметки языка HTML. Например, псевдоэлементы можно применить для стилизации первой строки абзаца или, как в данном примере, для вставки генерируемого содержимого перед и после действительного элемента.

В предложенном решении вы заключаете в эффектные кавычки имеющуюся цитату. Для открывающих кавычек применяется следующее объявление:

content: "\201C";

Любой текст, который вы хотите поместить после элемента, следует заключать в кавычки. Поскольку в разметке вы используете кавычки для обозначения того, что нужно выводить на экран, вы не можете вставить другой набор кавычек внутрь первого набора. Для заключения в кавычки цитаты следует применять шестнадцатеричный код знака кавычек, для открывающих кавычек — 201С.

Поскольку все, находящееся в HTML-коде между знаками кавычек, автоматически отображается без каких-либо изменений, необходимо избавиться от шестнадцатеричного значения, т. е. сообщить Web-обозревателю о необходимости визуализации знака кавычек, поместив перед кодом кавычек обратный слэш.

В спецификацию CSS 2.1 включено свойство content, содержащее стандартные значения для упрощения вставки знаков кавычек. Например, для воссоздания открывающих кавычек используйте следующее объявление:

content: open-quote;

Учтите, что значение ключевого слова open-quote, определенное в спецификации, отображается только в Web-обозревателях Mozilla и Opera. И еще имейте в виду, что псевдоэлементы :before и :after не работают в Web-обозревателях Internet Explorer 5 для Windows и его более поздних версиях и Internet Explorer для Macintosh.

## См. также

Информацию о цитировании и генерируемом содержимом в спецификации CSS2 на странице http://www.w3.org/TR/REC-CSS2/generate.html#quotes.

# 13.7. Задание перемещающейся фоновой сцены при изменении пользователем размеров окна

## Проблема

Вы хотите, чтобы при изменении пользователем размеров окна Web-обозревателя сдвигались накладывающиеся фоновые изображения (рис. 13.9).

## Решение

Задайте фоновое изображение в элементе body с помощью отрицательного значения в процентах:

```
body {
  font-size: 62.5%;
  background-color: #fff;
  background-image: url(birds-flock.png);
  background-repeat: repeat-x;
  background-position: -80% 0;
}
```



Рис. 13.9. Изображения, изменяющие местоположение в зависимости от размеров окна Web-обозревателя

Затем поместите контент Web-страницы в два дополнительных элемента div:

```
<body>
<div id="birds2">
<div id="birds1">
<div id="filler">
<div id="filler">
</div>
</div>
</div>
</div>
</div>
</div>
</div>
```

Задайте в соответствующих элементах div фоновые изображения с разными значениями позиционирования:

```
#birds2 {
  background-image: url(birds-two.png);
  background-position: 90% 0;
  background-color: transparent;
```

```
background-repeat: no-repeat;
width: 100%;
}
#birds1 {
background-image: url(birds-one.png);
background-position: -20% 0;
background-color: transparent;
background-repeat: no-repeat;
width: 100%;
}
```

## Обсуждение

Применяя разные расстояния, заданные в процентах, вы создаете у пользователя впечатление просмотра объекта с разных точек наблюдения, когда фоновые изображения сдвигаются в разных направлениях. Этот эффект называют параллаксом, от греческого слова parallaxis ("смена", "чередование").

Для того чтобы этот эффект отображался, фоновые изображения должны быть PNGизображениями с альфа-прозрачностью. Небольшие изменения прозрачности при смещении фоновых изображений относительно друг друга помогают создать визуальный эффект.

Применяя JavaScript, можно активизировать этот эффект при движении мыши. Получить дополнительную информацию и загрузить код можно с Web-страницы http://webdev.stephband.info/parallax.html.

## См. также

Более подробное описание эффекта на странице http://carsonified.com/blog/ hoe-to=recreate-silverbacks-parallax-effect/.

# 13.8. Вставка анимации в элементы на странице

## Проблема

Вы хотите "оживить" элемент Web-страницы, как показано на примере изображения облаков на рис. 13.10.

## Решение

Применяя специализированные свойства анимации Safari, задайте элемент, движущийся назад и вперед.

Сначала добавьте HTML-элемент после открытия элемента body:

```
<body>
<div id="clouds1"></div>
[...]
</body>
```



Рис. 13.10. Фон с движущимися облаками

Вставьте стилевое оформление фонового изображения с заданной шириной, высотой и абсолютным позиционированием:

```
#clouds1 {
  width: 627px;
  height: 410px;
  position: absolute;
  right: -300px;
  top: 0;
  background-image: url(clouds.png);
}
```

Далее задайте ключевые кадры анимации (keyframes) с именами облаков (clouds):

```
@-webkit-keyframes "clouds" {
  from {
    right: 0px;
  }
  to {
    right: 100px;
  }
}
```

Начните анимацию, связав keyframes с элементом div "clouds1" и инструкциями, определяющими продолжительность анимации, частоту повторения и направление движения:

```
#clouds1 {
  width: 627px;
  height: 410px;
  position: absolute;
  right: -300px;
  top: 0;
```

```
background-image: url(clouds.png);
-webkit-transform: translate(300px, 0px);
-webkit-animation-name: "clouds";
-webkit-animation-duration: 10s;
-webkit-animation-iteration-count: 10;
-webkit-animation-direction: alternate;
}
```

## Обсуждение

Несмотря на то, что во время написания книги анимация средствами CSS поддерживалась только Web-обозревателем Safari, консорциум W3C в настоящее время работает над спецификацией. Другие поставщики Web-обозревателей могут разработать аналогичные расширения для своих обозревателей до завершения разработки спецификации анимации.

#### Keyframes анимации

Первый шаг при задании анимации — определение так называемого правила keyframes:

```
@-webkit-keyframes "clouds" {
  from {
    right: 0px;
  }
  to {
    right: 100px;
  }
}
```

Ключевые кадры (keyframes) задают начальную и конечную точки анимации, а также позволяют более тонко управлять отображением анимации. Вместо указания с помощью селекторов ключевых кадров from и to начальной и конечной точек, можно применять значения, задаваемые в процентах:

```
@-webkit-keyframes "clouds" {
    0% {
    right: 0px;
    }
    33% {
    right: 10px;
    }
    68% {
    right: 90px;
    }
    100% {
    right: 100px;
    }
}
```

В правиле keyframes можно задавать несколько свойств, например для создания диагональной анимации:

```
@-webkit-keyframes "clouds" {
 0% {
  right: Opx;
  top: 0px;
 }
 33% {
 right: 10px;
  top: 10px;
 }
 68% {
  right: 90px;
  top: 90px;
 }
 100% {
  right: 100px;
  top: 100px;
 }
}
```

#### Примечание

Для создания интересных анимационных эффектов попробуйте применить в селекторах ключевых кадров свойство opacity.

#### Свойства анимации

В элементе с анимацией свойство animation-name применяется для связывания с используемым правилом ключевых кадров:

```
@-webkit-keyframes "clouds" {
from {
 right: Opx;
}
to {
 right: 100px;
 }
}
#clouds1 {
width: 627px;
height: 410px;
position: absolute;
right: -300px;
top: 0;
background-image: url(clouds.png);
-webkit-transform: translate(300px, 0px);
-webkit-animation-name: "clouds";
}
```

Свойство animation-duration задает продолжительность отображения анимации:

#clouds1 {
 width: 627px;

```
height: 410px;
position: absolute;
right: -300px;
top: 0;
background-image: url(clouds.png);
-webkit-transform: translate(300px, 0px);
-webkit-animation-name: "clouds";
-webkit-animation-duration: 10s;
}
```

#### Примечание

Отрицательное значение свойства animation-duration трактуется как нулевое.

Для ограничения анимации 10 циклами повторения задайте значение 10 в свойстве animation-iteration-count:

```
#clouds1 {
  width: 627px;
  height: 410px;
  position: absolute;
  right: -300px;
  top: 0;
  background-image: url(clouds.png);
  -webkit-transform: translate(300px, 0px);
  -webkit-animation-name: "clouds";
  -webkit-animation-duration: 10s;
  -webkit-animation-iteration-count: 10;
}
```

Для непрерывной анимации задайте в свойстве animation-iteration-count значение infinite:

```
-webkit-animation-iteration-count: infinite;
```

Когда анимация достигает конечного ключевого кадра при заданном повторении циклов, можно задать анимацию в обратном порядке, определив значение alternate в свойстве animation-direction:

```
#clouds1 {
  width: 627px;
  height: 410px;
  position: absolute;
  right: -300px;
  top: 0;
  background-image: url(clouds.png);
  -webkit-transform: translate(300px, 0px);
  -webkit-animation-name: "clouds";
  -webkit-animation-duration: 10s;
  -webkit-animation-iteration-count: 10;
  -webkit-animation-direction: alternate;
```

}

Значение по умолчанию в свойстве animation-direction — normal, что означает циклическое повторение анимации. После достижения конечной точки элемент скрывается и появляется снова в начальное точке, чтобы повторить все сначала.

## См. также

Описание анимации в спецификации CSS3 на странице http://www.w3.org/TR/css3-animations/.

# 13.9. Создание отображения фейерверков при прокрутке страницы пользователем

## Проблема

Вы хотите отображать как фон страницы фейерверки с меняющимся цветом при прокрутке пользователем окна Web-обозревателя (рис. 13.11).



Рис. 13.11. Фейерверки с меняющимся цветом при прокрутке Web-страницы

## Решение

Сначала задайте в элементе body окрашенное в цвета радуги фоновое изображение, при этом обязательно указав свойство background-attachment со значением fixed:

```
body {
  font-size: 62.5%;
  background-color: #fff;
  background-image: url(bkgd.jpg);
  background-position: center;
  background-color: white;
  background-attachment: fixed;
```

Далее поместите контент Web-страницы в элемент div:

```
<body>
<div id="easel">
[...]
</div>
</body>
```

Далее поместите изображение, содержащее обратные контурные изображения фейерверков:

```
#easel {
  background-image: url(fireworks.png);
  width: 100%;
  height: 100%;
}
```

## Обсуждение

Предлагаемое решение, как и решение в *разд. 13.7*, основано на применении слоев с изображениями, накладывающихся друг на друга. Ограничив разноцветное фоновое изображение подготовленным трафаретом, вы заставляете цвета радуги формировать отображение фейерверков.

Когда пользователь прокручивает страницу, разноцветное изображение остается на месте, а изображение фейерверков располагается как мозаика и отображается при прокрутке. Подобное поведение пользователя создает эффект простой радужной анимации.

Помимо анимации при прокрутке, если пользователь изменяет размер окна Webобозревателя, цвета фейерверков тоже меняются. Такой эффект создается благодаря центрированию радужного изображения в элементе body. Когда размер окна Webобозревателя изменяется, обозреватель заново позиционирует фоновое изображение, добиваясь его центрирования.

## См. также

*Разд. 13.7* о задании перемещающегося фонового изображения при изменении пользователем размера окна Web-обозревателя.

# 13.10. Настройка таблицы стилей для просмотра исходного кода в Firefox

## Проблема

Вы хотите изменить дизайн разметки, которая отображается при просмотре исходного кода Web-страниц в Firefox.

## Решение

Если вы используете Macintosh, можно выполнить следующие действия.

- 1. Найдите файл приложения Firefox, Firefox.app, в папке Applications (Программы).
- 2. Щелкните приложение кнопкой мыши с нажатой клавишей <Ctrl>, чтобы вывести на экран контекстное меню, и выберите строку Show Package Contents (Показать содержимое пакета).
- 3. Перейдите в папку Contents/MacOS/res и откройте файл viewsource.css. После этого отредактируйте файл по вашему усмотрению.

Если вы — пользователь OC Windows, таблицу стилей можно найти в папке C:\Program Files\Mozilla Firefox\res\viewsource.css.

# Обсуждение

Для использования собственных изображений поместите их в ту же папку, что и файл viewsource.css.

## Установка внешнего редактора

Вместо настройки в Firefox внешнего вида исходного кода средствами CSS можно применить внешнюю стороннюю программу.

Введите в поле адреса Web-обозревателя **about:config**. Затем найдите строку view\_source.editor.external. Щелкните параметр правой кнопкой мыши или левой кнопкой, но с нажатой клавишей <Ctrl>, чтобы изменить значение false на true. Затем щелкните правой кнопкой мыши или левой кнопкой, но с нажатой клавишей <Ctrl> параметр view\_source.editor.path и задайте путь к редактору кода.

В версии Firefox для PC используйте приложение Notepad++:

view\_source.editor.path: C:\Program Files\Notepad++\notepad++.exe

В версии Firefox для Mac — приложение TextMate:

view\_source.editor.path: /Applications/TextEdit.app/Contents/MacOS/TextMate

## См. также

*Разд. 2.1* с информацией о том, как можно самому создавать собственные таблицы стилей.

# 13.11. Проектирование с помощью сетки (наборы CSS)

## Проблема

Вы хотите разрабатывать дизайн Web-страницы средствами CSS, используя заранее подготовленную сетку.

## Решение

Прежде чем формировать HTML- или CSS-код, выберите созданный в программах Illustrator или Photoshop шаблон из набора Blueprint CSS, загрузив файл(ы) с Webстраницы http://urlgreyhot.com/personal/weblog/photoshop\_and\_illustrator\_templates\_ blueprint css framework.

Затем, используя шаблон как основу, разработайте макет вашего сайта.

### Примечание

По умолчанию система столбцов из набора Blueprint CSS состоит из 24 столбцов шириной 30 пикселов с межстолбцовыми промежутками или средниками, равными 10 пикселам.

Загрузите файлы набора Blueprint CSS с Web-сайта http://www.blueprintcss.org и включите эти файлы в ваши файлы разработки Web-сайта. Затем установите связь с файлами CSS в элементе head:

```
<link rel="stylesheet" href="css/blueprint/screen.css" type="text/css"
media="screen, projection">
<link rel="stylesheet" href="css/blueprint/print.css" type="text/css"
media="print">
<!--[if lt IE 8]>
<link rel="stylesheet" href="css/blueprint/ie.css" type="text/css"
media="screen, projection">
<![endif]-->
```

Для настройки макета страницы вы можете создать любые дополнительные правила CSS в отдельной таблице стилей и связать их со страницей с помощью элемента link так, чтобы они переопределили правила CSS (см. *разд. 2.15*) из готового набора:

```
<link rel="stylesheet" href="css/blueprint/screen.css" type="text/css"
media="screen, projection">
<link rel="stylesheet" href="css/blueprint/print.css" type="text/css"
media="print">
<!--[if lt IE 8]>
<link rel="stylesheet" href="css/blueprint/ie.css" type="text/css"
media="screen, projection">
<![endif]-->
<link rel="stylesheet" href="css/example.css" type="text/css"
media="screen, projection">
```

В элементе body заключите контент в элемент div со значением container у атрибута class:

```
<div class="container">
...
</div>
```

Для создания колонок примените атрибут class со значением, равным "span", в конец которого добавлено количество столбцов, которое должен занять контент:

```
<div class="container">
  <div class="span-24">
```

```
Header

</div>

<div class="span-5">

Side column

</div>

<div class="span-14">

Main column

</div>

<div class="span-5">

Side column

</div>

</div>

</div>
```

Поскольку поля или средники задаются справа от столбцов, последняя колонка должна быть урезана на ширину промежутка, в противном случае ширина страницы превысит 960 пикселов.

Вы можете удалить правостороннее поле с помощью селектора класса с другим значением атрибута class:

```
<div class="container">
<div class="span-24 last">
Header
</div>
<div class="span-5">
Side column
</div>
<div class="span-14">
Main column
</div>
<div class="span-5 last">
Side column
</div>
<div class="span-5 last">
Side column
</div>
```

Для включения нескольких колонок в еще одну колонку примените тот же прием:

```
<div class="container">
<div class="span-24 last">
Header
</div>
<div class="span-5">
Side column
</div>
<div class="span-14">
<div class="span-5">
Sidebar
</div>
<div class="span-9 last">
Main column
</div>
```

```
<div class="span-14 last">
Footer for main column
</div>
</div>
<div class="span-5 last">
Side column
</div>
</div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></ti>
```

## Обсуждение

Сетчатые макеты применялись в печатной верстке в течение столетий и часто считались отличительным признаком искусного дизайна.

Использование подготовленных файлов из наборов (библиотек) — быстрый способ построения макетов с сеткой, корректно отображаемых в разных Web-обозревателях. Обычно наборы шаблонов CSS включают файл для сброса стилей, CSS Reset (см. *разд. 5.2*), простую таблицу стилей для печати (см. *главу 10*), базовое форматирование текста (см. *главу 3*) и, что особенно важно, метод для легкого создания колоночной структуры.

Критики подготовленных библиотек или наборов в качестве причин для отказа от применения систем, подобных Blueprint, упоминают их раздутый исходный код, негибкую сеточную структуру и долгие паузы с нарастающим количеством HTTP-запросов на загрузку файлов.

#### Примечание

Для изменения ширины, количества столбцов сетки или размера межстолбцовых промежутков в наборе Blueprint CSS используйте интерактивное приложение по адресу http://kematzy.com/blueprint-generator/.

## См. также

Учебные пособия по применению Blueprint CSS на странице http://wiki.github.com/ joshuaclayton/blueprint-css/tutorials.

# 13.12. Пример дизайна: связный Web-дизайн

В этом примере дизайна вы, применяя некоторые из решений, обсуждавшихся в книге, сведете дизайн Web-страницы к разработке HTML-кода и правил CSS.

## Разметка страницы

Первый шаг — разметка документа с помощью элементов div:

```
<div id="header" class="">
<hl><a href="/" title="Home Page">Kirkland Composition &amp;
Copyediting</a></hl>
</div><!-- /#header -->
```

```
<hr />
<div id="nav" class="">
<div id="nav-site">
 Content Navigation:
 id="linkservices"><a href="/services/">Services</a>
  id="linkrates"><a href="/rates/">Rates</a>
  id="linkclients"><a href="/clients/">Clients</a>
  <a href="/samples">Samples</a>
 </div>
<div id="nav-info">
 Information Navigation:
 <a href="/about/">About</a>
  <a href="/guarentee/">Guarantee</a>
  <a href="/contact/">Contact</a>
 </11>
</div><!-- /#nav-info -->
</div><!-- /#nav -->
<hr />
<div id="content" class="">
<div id="article">
 [...]
</div><!-- /#article -->
<div id="aside" class="">
 [...]
</div><!-- /#aside -->
</div><!-- /#content -->
<hr />
<div id="footer" class="">
 [...]
</div>
```

#### Примечание

Везде, где возможно, я использовал в качестве значений атрибутов id названия элементов HTML5. Когда HTML5 получит более широкую поддержку в Web-обозревателях, легче будет полностью перейти к HTML5 DOCTYPE.

Следующий шаг — применение сброса стилей CSS Reset (см. *разд. 5.2*) для удаления встроенных установок Web-обозревателя (рис. 13.12):

```
html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6, p,
blockquote, a, abbr, acronym, big, font, img, small, center, dl, dt, dd, ol,
ul, li, fieldset, form, label, legend, table, tbody, tfoot, thead, tr, th, td {
margin: 0;
padding: 0;
border: 0;
outline: 0;
```

```
font-size: 100%;
 vertical-align: baseline;
 background: transparent;
}
body {
 line-height:1;
}
ol, ul {
 list-style:none;
}
blockquote, q {
 quotes:none;
}
:focus {
 outline:0;
ļ
ins {
 text-decoration:none;
}
del {
 text-decoration:line-through;
}
table {
 border-collapse:collapse; border-spacing:0;
}
```



Задайте свойства элемента body, включая фоновые цвета и изображения:

```
body {
  background-color: #000;
  color: rgba(255,255,255,.8);
  font-size: 62.5%;
  font-family: Times, "Times New Roman", Georgia, serif;
}
```

### Сжатие контента

Следующий шаг — создание многократно используемого селектора класса для уменьшения ширины контента в разных фрагментах Web-документа, разрешая при этом фоновым изображениям для заголовка и "подвала" страницы превышать ширину контента:

.eight5x11 {
 width: 805px;
 margin: 0 auto;
}

Затем поместите эти ограничители между основными разделами страницы:

```
<div id="header" class="">
<div class="eight5x11">
 <h1><a href="/" title="Home Page">Kirkland Composition & amp;
Copyediting</a></h1>
</div><!-- /.eight5x11 -->
</div><!-- /#header -->
<hr />
<div id="nav" class="">
<div class="eight5x11">
 <div id="nav-site">
  Content Navigation:
  <111>
   id="linkservices"><a href="/services/">Services</a>
   id="linkrates"><a href="/rates/">Rates</a>
   id="linkclients"><a href="/clients/">Clients</a>
   id="linksamples"><a href="/samples">Samples</a>
  </div>
 <div id="nav-info">
  Information Navigation:
  <111>
   <a href="/about/">About</a>
   <a href="/guarantee/">Guarantee</a>
   <a href="/contact/">Contact</a>
```

```
</div><!-- /#nav-info -->
</div><!-- /.eight5x11 -->
</div><!-- /#nav -->
<hr />
<div id="content" class="">
<div class="eight5x11">
 <div id="article">
    [...]
 </div><!-- /#article -->
 <div id="aside" class="">
  [...]
 </div><!-- /#aside -->
</div><!-- /.eight5x11 -->
</div><!-- /#content -->
<hr />
<div id="footer" class="">
<div class="eight5x11">
  [...]
</div><!-- /.eight5x11 -->
```

</div><!-- /#footer -->

В первом разделе документа, основном заголовке, примените фоновое изображение:

```
/* Logo */
#header {
  background-image: url(img/bkgd-header.gif);
  background-repeat: no-repeat;
  background-position: top center;
  margin-top: 30px;
}
```

Поскольку класс-ограничитель применен в элементе div основного заголовка и контента, можно использовать изображение, которое больше ширины контента, не вызывая появления на экране горизонтальной прокрутки при уменьшении пользователем размеров окна.

## Введение логотипа

Вы можете вставить изображение логотипа через ссылку основного заголовка. Значение block в свойстве display (см. *разд. 7.10*) позволит задать ширину и высоту элемента, чтобы вставить логотип как фоновое изображение.

Затем с помощью относительного позиционирования поместите логотип примерно в левый верхний угол, как показано на рис. 13.13:

```
/* Logo */
#header {
   background-image: url(img/bkgd-header.gif);
   background-repeat: no-repeat;
```

#### Проектирование с помощью CSS

```
background-position: top center;
margin-top: 30px;
}
#header h1 a {
background-image: url(img/logo.png);
background-repeat: no-repeat;
background-position: top center;
display: block;
width: 456px;
height: 270px;
text-indent: -9999em;
position: relative;
left: -78px;
top: -30px;
border: none;
}
```





#### Далее удалите элементы hr и подчеркивание текстовых ссылок:

```
/* Common Elements */
.hide, hr {
  display: none;
}
```

```
a {
  text-decoration: none;
}
```

## Навигационное меню

Расположите в нужном месте главное навигационное меню:

```
#nav-site {
  width: 433px;
  margin-left: 380px;
  position: relative;
  top: -95px;
  }
 #nav-site ul, #nav-site ul li {
  display: inline;
  }
```

#### Примечание

Другой способ размещения логотипа и навигационного меню — сделать оба элемента плавающими с выравниванием по левому краю и затем применить относительное позиционирование, чтобы сдвинуть их в нужное место на странице.

После того как маркированный список помещен на свое место, самое время создать горизонтальное меню (см. *разд. 7.12*). Для этого снова задайте отображение ссылок как блочных элементов, скройте текст и убедитесь в том, что фоновое изображение не повторяется:

```
#nav-site ul li a {
  display: block;
  float: left;
  text-indent: -9999em;
  background-repeat: no-repeat;
}
```

Имея основные блоки построения навигационного меню, создайте для него изображение-спрайт, включающее обычное состояние, состояние при наведении указателя мыши и активное состояние пункта меню, показанные на рис. 13.14.

SERVICES	RATES	CLIENTS	SAMPLES
SERVICES	RATES	CLIENTS	SAMPLES
SERVICES	RATES	CLIENTS	SAMPLES

Рис. 13.14. Изображение-спрайт для навигационного меню

Далее задайте для каждой ссылки свое изображение:

```
#nav-site ul li#linkservices a {
  width: 118px;
```

```
height: 39px;
background-image: url(img/nav.gif);
}
#nav-site ul li#linkrates a {
width: 88px;
height: 39px;
background-image: url(img/nav.gif);
background-position: -118px 0;
l
#nav-site ul li#linkclients a {
width: 107px;
height: 39px;
background-image: url(img/nav.gif);
background-position: -206px 0;
}
#nav-site ul li#linksamples a {
width: 120px;
height: 39px;
background-image: url(img/nav.gif);
background-position: -313px 0;
}
```

Для отображения эффекта наведения указателя мыши на пункт меню сдвиньте позицию фонового изображения:

```
#nav-site ul li#linkservices a:link:hover,
#nav-site ul li#linkservices a:visited:hover {
background-position: 0 -39px;
}
#nav-site ul li#linkrates a:link:hover,
#nav-site ul li#linkrates a:visited:hover {
background-position: -118px -39px;
}
#nav-site ul li#linkclients a:link:hover,
#nav-site ul li#linkclients a:visited:hover
                                              {
background-position: -206px -39px;
}
#nav-site ul li#linksamples a:link:hover,
#nav-site ul li#linksamples a:visited:hover {
background-position: -313px -39px;
}
```

Теперь задайте контекстное меню, отображающее адрес страницы, просматриваемой в данный момент (рис. 13.15):

```
#pageservices #nav-site ul li#linkservices a {
  background-position: 0 -78px;
}
#pagerates #nav-site ul li#linkrates a {
  background-position: -118px -78px;
}
```

```
#pageclients #nav-site ul li#linkclients a {
   background-position: -206px -78px;
}
#pagesamples #nav-site ul li#linksamples a {
   background-position: -313px -78px;
}
```



Рис. 13.15. Законченное навигационное меню

#### Примечание

Для работы контекстного меню в элемент body добавлен атрибут id со значением pagerates:

```
<body id="pagerates">
[...]
</body>
```

#### Вспомогательное меню ссылок

Расположив основное навигационное меню, сместите текстовые ссылки в верхнюю часть страницы с помощью абсолютного позиционирования и задайте отдельные элементы списка, формирующие горизонтальный ряд:

```
/* Top Navigation Links */
#nav-info {
position: absolute;
width: 100%;
top: 0;
left: 0;
font-size: 1.2em;
height: 30px;
l
#nav-info ul {
width: 805px;
margin: 0 auto;
text-align: right;
}
#nav-info ul li {
display: inline;
padding-left: 1.2em;
line-height: 1.9;
text-transform: lowercase;
}
```

## Стилевое оформление колонки контента

Когда заданы заголовок и навигационные элементы, можно перейти к контенту страницы. Сначала очистите обе стороны плавающих элементов, чтобы контент не обтекал навигационное меню, и задайте цвет шрифта:

```
/* Content */
#content {
   clear: both;
   color: rgba(255,255,255,.8);
}
```

Теперь с помощью свойства float сдвиньте основную колонку вправо, одновременно оставив достаточно свободного пространства под колонкой. Эта пробельная зона действует как буфер между основным контентом и "подвалом" страницы:

```
/* Content */
#content {
  clear: both;
  color: rgba(255,255,255,.8);
}
#content #article {
  float: right;
  width: 585px;
  margin-bottom: 70px;
}
```

Поместив основную колонку в нужное место, отрегулируйте высоты заголовков и абзацев так, чтобы у них был одинаковый межстрочный интервал:

```
#content {
clear: both;
color: rgba(255,255,255,.8);
}
#content #article {
float: right;
width: 585px;
margin-bottom: 70px;
#content #article h2 {
font-size: 2.4em;
font-weight: normal;
line-height: .9998;
}
#content #article h3 {
font-size: 1.6em;
font-weight: bold;
line-height: 1.4975;
margin-top: 1.6em;
3
#content #article p {
font-size: 1.4em;
line-height: 1.714;
font-family: Verdana, Geneva, Tahoma, sans-serif;
}
```

В первом слове основного заголовка за заглавной буквой w следует строчная о. Между этими буквами большой пробел или кернинг. Для корректировки расстояния между ними (см. *разд. 3.33*) примените псевдоэлемент :first-letter (см. *разд. 3.13*), позволяющий уменьшить пробел только для первой буквы (рис. 13.16):

```
/* Content */
#content {
clear: both;
color: rgba(255,255,255,.8);
}
#content #article {
float: right;
width: 585px;
margin-bottom: 70px;
}
#content #article h2 {
font-size: 2.4em;
font-weight: normal;
line-height: .9998;
}
#content #article h3 {
font-size: 1.6em;
font-weight: bold;
```

/\* Content \*/

```
line-height: 1.4975;
margin-top: 1.6em;
}
#content #article p {
font-size: 1.4em;
line-height: 1.714;
font-family: Verdana, Geneva, Tahoma, sans-serif;
}
#content #article h2:first-letter {
letter-spacing: -.1em;
```

```
}
```



Рис. 13.16. Колонка основного контента со стилями

# Работа над узкой колонкой

После того как заданы стили для колонки основного контента, пришло время заняться левой колонкой. С помощью поля вы по существу задаете ширину боковой колонки:

```
#content #aside {
  margin-right: 650px;
  color: #999;
}
```

#### Примечание

Вместо применения полей, увеличивающих ширину колонки с контентом, можно задать двухколоночный макет, сместив боковую колонку вправо с помощью плавающей модели и затем задать отступ с правой стороны, чтобы создать пробельную зону или средник между двумя колонками.

Затем откорректируйте стилевое оформление текста, сохраняя межстрочный интервал таким же, как и в колонке с основным контентом. При одинаковом межстрочном интервале обе колонки текста используют одну и ту же базовую линию:

```
#content #aside {
margin-right: 650px;
color: #999;
l
#content #aside h4 {
font-size: 1.4em;
text-transform: uppercase;
font-style: italic;
letter-spacing: .3em;
line-height: 2.3996;
margin-bottom: -.7em;
}
#content #aside p {
font-size: 1.4em;
line-height: 1.714;
}
```

С помощью селектора смежного элемента (см. *разд. 2.4*) добейтесь курсивного начертания второго абзаца:

```
#content #aside {
margin-right: 650px;
color: #999;
#content #aside h4 {
font-size: 1.4em;
text-transform: uppercase;
font-style: italic;
letter-spacing: .3em;
line-height: 2.3996;
margin-bottom: -.7em;
}
#content #aside p {
font-size: 1.4em;
line-height: 1.714;
#content #aside p+p {
font-size: 1.4em;
font-style: italic;
line-height: 1.714;
}
```

Завершив стилевое оформление текстового контента, примените только к этой порции документа набор стилей для ссылок (рис. 13.17):

```
#content a:link, #content a:visited {
  color: rgba(51,153,204,.8);
  border-bottom: lpx solid #3399cc;
  border-bottom: lpx solid rgba(51,153,204,.8);
}
#content a:link:hover, #content a:visited:hover {
  color: rgba(51,153,204,.8);
  border-bottom: none;
}
```



Рис. 13.17. Стилевое оформление ссылок

## Заключительный раздел

Завершающая часть Web-страницы — "подвал". Поместите фоновое изображение внизу страницы и отодвиньте весь контент на 25 пикселов от верхнего края "подвала", чтобы дать изображению немного жизненного пространства:

```
#footer {
   background-image: url(img/bkgd-footer.gif);
   background-repeat: no-repeat;
   background-position: top center;
   font-size: 1.4em;
   line-height: 1.4;
   clear: both;
   padding-top: 25px;
   color: #666;
}
```

Далее задайте стиль с помощью микроформата hCard, позволяющего загрузить пиктограмму, которая появится в левой части страницы:

```
#footer .vcard {
  padding-left: 40px;
}
#footer .org {
  color: #999;
}
#footer img {
  float: left;
}
```

Задайте девиз компании справа на странице, применив абсолютное позиционирование (рис. 13.18):

```
#footer .tagline {
   position: absolute;
   right: 0;
   top: 0;
   font-family: Times, "Times New Roman", Georgia, serif;
   letter-spacing: .4em;
   font-style: italic;
   font-weight: bold;
}
```

## Последние штрихи

В завершение задайте отображение белым цветом ссылок на другие области страницы (рис. 13.19):

```
a:link, a:visited {
  color: rgba(255,255,255,.8);
  border-bottom: 1px solid #fff;
  border-bottom: 1px solid rgba(255,255,255,.8);
}
a:link:hover, a:visited:hover {
  color: rgba(255,255,255,.8);
  border-bottom: none;
}
```



Рис. 13.18. Помещенный в нужное место девиз

С последним приведенным правилом Web-страница полностью подготовлена для современных Web-обозревателей, но как быть с более ранними версиями IE? В них дизайн страницы наверняка будет испорчен. Для передачи в эти Web-обозреватели исправлений CSS используйте условные комментарии (см. *разд. 12.7*):

```
<!--[if lt IE 8]>
<style type="text/css">
#content {
  color: #fff;
  }
  a:link, a:visited {
    color: #fff;
    border-bottom: 1px solid #fff;
  }
  a:link:hover, a:visited:hover {
    color: #fff;
    border-bottom: none;
  }
```

```
#content a:link, #content a:visited {
  color: #3399cc;
  border-bottom: 1px solid #3399cc;
}
#content a:link:hover, #content a:visited:hover {
  color: #3399cc;
  border-bottom: none;
}
</style>
<![endif]-->
```



Рис. 13.19. Стилевое оформление остальных ссылок

Дальнейшими шагами могли бы быть создание таблицы стилей для печати (см. *разд. 10.1*), а также таблицы, предназначенной для печати версии логотипа (см. *разд. 10.2*), но я оставляю их вам в качестве домашнего задания.

# 13.13. Пример дизайна: флаг США

Задача этого упражнения — создание с помощью CSS и семантической разметки флага США, который не только внешне привлекателен, но и полезен, потому что щелчок кнопкой мыши по любой его звездочке перенесет вас на главную страницу администрации того или иного штата, а щелчок мыши по любой из полос — на правительственные сайты одной из тринадцати первоначальных колоний.

## Основная разметка

Сначала создайте основу для флага, подготовив разметку. Включите в нее заголовок страницы и 50 штатов, представленных на флаге США.

Поместите название в тег header и вставьте 50 штатов в список:

```
<hl>
<hl>
<a href="http://www.usa.gov/">United States of America</a>
</hl>

<a href="http://www.alabama.gov/">
<strong>State of Alabama</strong>
<i><i></i>
</a>
<a href="http://www.state.ak.us/">
<strong>State of Alaska</strong><i><i></i>
</a>
</o>>
```

У заголовка и каждого отдельного штата есть ссылка на соответствующий административный сайт. Кроме того, в каждую ссылку включены дополнительные теги задания курсива, которые вы будете использовать в дальнейшем как селекторы для стилевого оформления.

Вам нужно отделить 13 колоний от остальных штатов, поскольку они образуют 13 полос на флаге. Сделайте это, заключив их в тег ет:

```
<a href="http://www.ct.gov/">
<em><strong>State of Connecticut</strong></em>
<i></i>
</a>
```

Пока результаты мало похожи на флаг (рис. 13.20). Вот где не обойтись без CSS. Теперь, когда у вас есть базовая разметка, начнем создавать стилевое оформление вашей страницы.

## Создание мольберта

Сначала создадим мольберт, на котором вы нарисуете ваш флаг. Начнем с дополнительного структурирования HTML-кода. Заключите весь контент страницы в тег div и задайте в нем атрибут id со значением easel:

```
<div id="easel">
<hl>
<a href="http://www.usa.gov/">United States of America</a>
</hl>
<a href="http://www.alabama.gov/">
<strong>State of Alabama</strong><i></i>
</a>
[...]
</div>
```



Рис. 13.20. Стандартное отображение контента

Далее необходимо задать стилевое оформление div, чтобы создать холст. Как это сделать? Взгляните на следующие правила CSS:

```
body {
  margin: 10px 0 0 0;
  padding: 0;
}
```

```
#easel {
  width: 955px;
  margin: 0 auto;
  position: relative;
}
```

Прежде всего задайте ширину холста, а затем отцентрируйте его.

Для центрирования на странице холста для флага, как показано на рис. 13.21, задайте поля со значениями auto с обеих сторон элемента div с идентификатором "easel". Webобозреватель автоматически разделит поле пополам и присвоит полученное значение справа и слева.



Теперь уберем текст из easel (т. е. создадим чистый холст, на котором будем рисовать флаг), добавив следующие правила CSS:

```
#easel ol {
   list-style: none;
   margin: 0;
   padding: 0;
}
#easel strong {
   display: none;
}
```

В приведенном коде вы сначала избавились от маркеров в пунктах списка, а также от полей и отступов, задаваемых по умолчанию. Затем вы скрываете весь текст из тега strong, который включает в себя все пункты списка (рис. 13.22).



## Создание полос

Пора рисовать флаг. Начнем с полос.

Напоминаю о том, что 13 колоний, которые представлены на флаге полосами, выделены в перечне штатов с помощью тега ет.

Поэтому вы можете создать полосы, задавая стили для этого селектора:

```
#easel em {
  width: 955px;
  height: 50px;
  display: block;
  background: #BF0A30;
  position: absolute;
  top: 0;
  left: 0;
  z-index: 1;
}
```

На этом этапе вы задали ширину и высоту флага (рис. 13.23). И после задания цвета фона вы превратили селектор в блочный элемент, чтобы можно было определить его местоположение на странице. Для полос задается абсолютное позиционирование; это означает сдвиг полос относительно краев их родительского элемента.

При выборе относительного позиционирования полосы располагаются относительно заданного по умолчанию положения элемента на странице. В данном случае вы помещаете полосу в левый верхний угол родительского элемента, элемента div с идентификатором easel: полоса отстоит на ноль пикселов от верхнего и левого краев содержащего ее элемента.

Свойство z-index определяет "третье измерение" элементов на Web-странице. Другими словами, оно задает порядок размещения элементов. Чем больше значение свойства z-index, тем выше элемент в наборе размещения.

Например, элемент со значением z-index, равным 5, располагается поверх элемента, у которого значение z-index равно 1. Вы хотите, чтобы полосы находились под синим полем со звездами, поэтому задайте малое значение z-index, равное 1.

Пока это выглядит так, как будто у вас только одна полоса (см. рис. 13.23). На самом деле на странице есть все 13 полос, но все они расположены в одном и том же месте — и все они красные. Для разделения 13 полос необходимо позиционировать и окрасить каждую полосу отдельно.

Для выбора каждой конкретной полосы, требующей стилевого оформления, сначала отметим, какие атрибуты в разметке выделяют каждую из полос:

```
<a href="http://www.ct.gov/">
<em>State of Connecticut</em>
<i></i>
</a>
```

Каждая полоса помечается собственным адресом ссылки. Вы можете использовать атрибуты href для задания селекторов конкретных колоний (см. *разд. 2.5*):

```
#easel a[href="http://delaware.gov/"] em {
   background: white;
   top: 50px;
   left: 0;
  }
#easel a[href="http://www.georgia.gov/"] em {
   top: 100px;
   left: 0;
  }
[...]
```



Рис. 13.23. Первая красная полоса

Поскольку первоначально вы задали для селектора ет красный цвет фона, необходимо изменить его для всех белых полос.

Затем вам нужно сдвинуть каждую полосу из верхней части страницы в назначенную точку. Высота каждой полосы — 50 пикселов, поэтому вторая полоса, в нашем случае для колонии Delaware (Делавэр), должна располагаться на 50 пикселов ниже. Следующая полоса опускается на 100 пикселов и т. д.

После того как вы закончите окрашивание и размещение всех 13 полос, ваш флаг должен выглядеть, как приведенный на рис. 13.24.

00	The Flag of the United States of America			
	http://ion.christopherschmitt.com/cssflag/step4.html	<b>े र</b>	• 🚷 Google	Q
Contacts * 📄 Events * 💽 Locations *	🔚 Tagspaces 🐑 🔜 Bookmarks 🐑 🔜 Resources 🕤			0ptions
🗋 The Flag of the United States of 😒 📗	🚺 Christopher Schmitt : Designer, 😵 🛛 📇 Technology Books, Tech Confer	⊗ +		Ξ.
	American Flag in CSS by Christopher Schmitt			

Рис. 13.24. Все 13 полос

## Создание звезд

Теперь, когда все полосы на месте, пора переходить к звездам. Сначала необходимо создать синее поле, на котором будут находиться звезды. Для этого придется преобразовать заголовок. Для помещения заголовка на синее поле необходимо скрыть текст заголовка, а затем задать размер, положение и цвет селектора-якоря в элементе header:

```
#easel h1 a {
  position: absolute;
  width: 215px;
  height: 175px;
  background: #002868;
  text-indent: -9999em;
  margin: 0;
  padding: 0;
  z-index: 20;
}
```

Абсолютное позиционирование элемента без каких-либо свойств, задающих смещение, помещает элемент в левый верхний угол элемента-контейнера div со значением id, равным easel.
После задания ширины, высоты и цвета фона вы получаете заголовок со "скрытым" текстом за счет отступа, сдвигающего его за пределы страницы.

В заключение, указав значение 20 для свойства z-index, вы накладываете заголовок на полосы, как показано на рис. 13.25.



Рис. 13.25. Синее поле

Для создания звезд (рис. 13.26) воспользуйтесь приемом, аналогичным способу формирования полос. В вашей таблице стилей выберите каждый штат с помощью уникального атрибута href и затем задайте соответствующее стилевое оформление.

Вы не хотите повторно задавать стили для тех элементов, с помощью которых размечены ваши 13 полос, поэтому следует использовать другой селектор, не якорь. Воспользуйтесь тегом italic как селектором конкретного штата (рис. 13.27):

```
#easel ol li a[href="http://www.alabama.gov/"] i {
    background-image: url(stars.gif);
    display: block;
    position: absolute;
    top: 13px;
    left: 13px;
    z-index: 50;
    width: 24px;
```







```
height: 23px;
}
#easel ol li a[href="http://www.state.ak.us/"] i {
  background-image: url(stars.gif);
  display: block;
  position: absolute;
  top: 13px;
  left: 90px;
  z-index: 50;
  width: 24px;
  height: 23px;
}
[...]
```

Каждая звезда — это маленькое изображение в формате GIF, показанное на рис. 13.28.

Вы ссылаетесь на это изображение в свойстве background-image. Затем вы позиционируете звезду, сдвигая элемент от верхнего левого угла элемента-контейнера div. Звезда

## Рис. 13.28. Изображение звезды



Рис. 13.29. Стандартный флаг США

каждого штата помещается в нужное место синего поля. Вы также задаете свойство z-index, равное 50, чтобы поместить ее поверх полос и синего поля.

После того как кропотливая задача формирования правил CSS для всех 50 штатов закончена, у вас появляется флаг США, показанный на рис. 13.29.

#### Добавление текстуры

Почему бы не улучшить ваше творение, придав поверхности определенную фактуру? Это сделать очень легко, поместив в ваши элементы прозрачные PNG-изображения как фон.

Формат PNG поддерживает альфа-прозрачность, позволяющую некоторым фрагментам изображения быть частично прозрачными (см. *разд. 4.17* о применении PNGизображений в Web-обозревателях). Это свойство можно применять для создания интересных эффектов, например создания впечатления "рельефной" поверхности.



Рис. 13.30. Фоновая текстура

Поместите под элементы, формирующие ваш флаг, прозрачное изображение, такое как показанное на рис. 13.30. Сначала расположите его под вашими красными полосами:

```
#easel em {
  width: 955px;
  height: 50px;
  display: block;
  background: #BF0A30;
  position: absolute;
  top: 0;
  left: 0;
```

```
z-index: 1;
background-image: url(flag_bkgd.png);
background-position: 50%;
background-repeat: no-repeat;
background-attachment: fixed;
}
```

Изображение центрируется относительно изображения полосы. В свойстве backgroundrepeat указано, что оно не повторяется ни вертикально, ни горизонтально.

Сделайте то же самое для белых полос:

```
#easel a[href="http://delaware.gov/"] em {
  background: white;
  top: 50px;
  left: 0;
  background-image: url(flag_bkgd.png);
  background-position: 50%;
  background-repeat: no-repeat;
  background-attachment: fixed;
}
```

И в завершение повторите процесс для синего поля:

```
#easel h1 a {
  position: absolute;
  width: 215px;
  height: 175px;
  background: #002868;
  text-indent: -9999em;
  margin: 0;
  padding: 0;
  z-index: 20;
  background-image: url(flag_bkgd.png);
  background-position: 50%;
  background-repeat: no-repeat;
  background-attachment: fixed;
}
```

Эти заключительные правила CSS для отображения флага США, показанного на рис. 13.31, действуют в Web-обозревателях IE7, Firefox 2, Opera 9.5 и их более поздних версиях, а также в Safari.

#### Примечание

Попробуйте создать собственную версию флага США или любой другой страны. Это замечательное упражнение для приобретения и совершенствования разнообразных навыков работы с CSS.



Рис. 13.31. Окончательный вариант флага США

## Взаимодействие с JavaScript

ГЛАВА 14

## 14.0. Введение

JavaScript — один из трех китов, на которых держится конструкция удачного Webсайта.

Web-разработчик Стивен Чемпион (Steven Champeon) сравнил различные Webтехнологии со структурными элементами английского языка: язык JavaScript — глагол, язык HTML — существительное, CSS — прилагательное.

Благодаря HTML как основы, описывающей контент средствами семантической разметки, вы можете эффективно применять таблицы стилей для отображения внешне привлекательного Web-сайта.

Язык JavaScript позволяет совершать на Web-странице действия. Под действием я подразумеваю не только перемещение различных элементов по Web-странице, как в плохой Flash-анимации. JavaScript позволяет управлять Web-страницей или изменять ее уже после загрузки.

В этой главе мы посмотрим, как изменять таблицу стилей для Web-страницы в зависимости от времени суток или разрешения экрана. Кроме того, мы увидим, как применять популярную библиотеку на JavaScript для создания обратно совместимых Web-сайтов и добавить немного оригинальности в ваши варианты отображений ссылок при наведении указателя мыши.

## 14.1. Проверка, включен ли JavaScript в Web-обозревателе

#### Проблема

Вы хотите определить, может ли Web-обозреватель обрабатывать JavaScript.

#### Решение

Для того чтобы пользователи могли узнать, включен ли в их Web-обозревателе JavaScript, в элементе body примените элемент noscript (puc. 14.1):

```
<body>
<noscript>
JavaScript не включен. Некоторые функции этого Web-сайта
не будут работать.
```

</noscript>

```
</body>
        CSS Cookbook
                         C
                                             X
                                      f
                                                     (1P)
         4
                                                               🗋 (file:///Us 🏠 🔻 ) - 🚱 🗸
                                                                                           Q
        Back
              Forward
                        Reload
                                Stop
                                      Home
                                             Print
                                                    1Password
              CSS Cookbook
                                      Christopher Schmitt :... 🕄
                                                              🖉 Technology Books, T... 🕄
       JavaScript is not enable. Some features of this web site will not work.
```

Рис. 14.1. Сообщение о том, что JavaScript не включен

#### Обсуждение

Применение элемента noscript — это способ предоставления Web-обозревателю с неустановленным JavaScript отличающегося, но доступного контента.

Несмотря на то, что JavaScript поддерживается всеми Web-обозревателями настольных компьютеров по умолчанию, некоторые пользователи отключают JavaScript (редко, но случается), или используют устройства без поддержки JavaScript.

#### Примечание

Если вы обнаружили, что Web-страница с элементом noscript (например, в элементе p) не прошла проверку на допустимость HTML-кода, перемещайте элемент за пределы его родительского элемента до тех пор, пока корректность кода не будет подтверждена.

#### См. также

Элемент noscript в спецификации HTML4 на странице http://www.w3.org/TR/html4/ interact/scripts.html#h-18.3.1.

## 14.2. Применение разных таблиц стилей в зависимости от времени суток

#### Проблема

Вы хотите в течение дня применять разные таблицы стилей.

#### Решение

Определите время суток на часах компьютера пользователя и предоставьте подходящую таблицу стилей:

```
<script type="text/javascript">
function setTimedStylesheet() {
  var theTime = new Date().getHours();
   if (8 <= theTime&&theTime < 20) {
      document.write("<link rel='stylesheet' href='daytime.css'
type='text/css'>");
    }
   if (20 <= theTime&&theTime < 8) {
      document.write("<link rel='stylesheet' href='nighttime.css'
type='text/css'>");
   }
   setTimedStylesheet();
   </script>
```

Обязательно включите элемент noscript, содержащий ссылку на таблицу стилей по умолчанию, на случай если у Web-обозревателя нет поддержки JavaScript:

```
<script type="text/javascript">
 function setTimedStylesheet() {
  var theTime = new Date().getHours();
   if (8 <= theTime&&theTime < 20) {
    document.write("<link rel='stylesheet' href='daytime.css'</pre>
type='text/css'>");
   }
  if (20 <= theTime&&theTime < 8) {
   document.write("<link rel='stylesheet' href='nighttime.css'</pre>
type='text/css'>");
  }
 }
setTimedStylesheet();
</script>
<noscript>
<link href="default.css" rel="stylesheet" type="text/css">
</noscript>
```

#### Обсуждение

Создание настраиваемого внешнего вида, зависящего от времени суток, — совсем не надуманная идея. Радио и телеканалы в Соединенных Штатах делят свои программы, исходя из времени суток — например, вручение Дневной премии Эмми (Daytime Emmy Awards), радиопередачи для водителей в пути, вечерние телевизионные шоу в праймтайм и т. д.

Основная проблема применения этого метода состоит в том, что вы предполагаете, что часы на компьютерах пользователей показывают точное время.

Другой вариант решения — получать время суток с вашего сервера средствами языка программирования промежуточного ПО, такого как PHP, и передавать его как переменную в JavaScript.

#### См. также

Справочную информацию об объекте Date на странице http://www.w3schools.com/ sref/jsref\_obj\_date.asp.

## 14.3. Перенаправление на сайт для мобильных устройств в зависимости от ширины экрана Web-обозревателя

### Проблема

Вы хотите изменять таблицу стилей, исходя из ширины экрана Web-обозревателя.

#### Решение

Определите ширину экрана текущего Web-обозревателя и перенаправьте Webобозреватель к версии дизайна сайта, предназначенной для мобильных устройств:

```
<script type= "text/javascript">
if (screen.width <= 481) {
   document.location = "http://mobi.example.com/"
}
</script>
```

### Обсуждение

Предлагаемое решение основано на способности JavaScript определять ширину окна (в пикселах) Web-обозревателя. Если она не более 481 пиксела, предполагается, что это Web-обозреватель для мобильного устройства.

#### Примечание

Не у всех мобильных устройств есть JavaScript.

#### Дизайн для экрана с высоким разрешением

Вы также можете создать зеркальное отражение сценария, заставив его определять, не больше ли ширина открытого окна Web-обозревателя пользователя средней ширины окна Web-обозревателя:

```
<script type= "text/javascript">
if (screen.width <= 481) {
  document.location = "http://mobi.example.com/"
} else if (screen.width >= 1280) {
  document.location = "http://high-def.example.com/";
}
</script>
```

#### См. также

Более надежное решение на JavaScript для предоставления макета, зависящего от разрешения на странице http://www.themaninblue.com/writing/perspective/2004/09/21/.

## 14.4. Вставка в Web-страницу библиотеки JavaScript

## Проблема

Вы хотите вставить в Web-страницу готовую библиотеку на JavaScript (рис. 14.2).



Рис. 14.2. Начальная страница готовой библиотеки jQuery

#### Решение

Воспользуйтесь службой хостинга компании Google для подключения библиотеки jQuery (рис. 14.3, который будет приведен далее) к Web-странице:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script>
```

#### Далее после ссылки на библиотеку jQuery добавьте пользовательский скрипт:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script>
<script type="text/javascript">
```

```
// Ваш код располагается эдесь...
$(document).ready(function(){
  window.alert("Hello, world! You have JavaScript!")
});
</script>
```

#### Обсуждение

Пользуясь услугами Google по размещению кода библиотеки jQuery, вы получаете сразу три преимущества.

Первое состоит в кэшировании. Если другие Web-сайты используют службы Google для связи с jQuery, код кэшируется в Web-обозревателе. Когда посетитель сайта переходит к другому сайту, страница отображается быстрее, потому что библиотека jQuery уже сохранена в кэше. Даже в случае сжатой версии размером 19 Кбайт это превращается в экономию для ваших пользователей.

Второе преимущество связано с количеством подключений к Web-серверу, которые может выполнить Web-обозреватель. Для того чтобы не перегружать сервер, Webобозреватель ограничивает количество подключений к серверу, когда загружает HTML, изображения, скрипты и т. д. Перенос библиотеки jQuery на другой сервер ускоряет отображение страницы.

Третий выигрыш заключается в том, что серверы Google, вероятно, быстрее доставляют на машины посетителей сайта файлы, такие как библиотека jQuery, если только ваш сервер не находится в двух шагах от посетителей вашего сайта.

#### Примечание

Onepatop alert включен просто для обозначения места вставки JavaScript. Если бы был нужен только этот оператор, скрипт состоял бы из простой строки кода и не нуждался бы в готовой библиотеке на JavaScript.

```
<script type="text/javascript">
window.alert("Hello, world! You have JavaScript!")
</script>
```

#### См. также

Coctaв jQuery и других библиотек Ajax, размещенных на серверах Google, на странице http://code.google.com/apis/ajaxlibs/documentation/.

## 14.5. Применение селекторов CSS3 в IE6 и IE7

#### Проблема

Вы хотите использовать в более ранних версиях Internet Explorer селекторы из спецификации CSS3.

#### Решение

Сначала откорректируйте в соответствии с синтаксисом CSS3 таблицу стилей для Webобозревателей, поддерживающих спецификацию CSS3:

```
#content {
  border: 4px solid black;
}
#content p {
  margin: 1em 0;
}
/* удаляет нижнее поле для последнего абзаца */
#content p:last-child {
  margin: 1em 0 0 0;
}
```

Затем воспользуйтесь способностью jQuery ссылаться с помощью стандартного синтаксиса CSS3 на фрагменты документа:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js"></script>
<script type="text/javascript">
// Your code goes here...
$(document).ready(function(){
    $("#content p:last-child").css("margin","1em 0 0 0");
});
</script>
```

#### Обсуждение

Одно из преимуществ применения библиотеки на JavaScript, такой как jQuery, состоит в использовании селекторов CSS. Вместо стилевого оформления целой страницы селекторы связывают функции и процедуры с отдельными частями страницы.

Для применения селектора CSS сначала используйте то, что называют объектом jQuery:

```
$(css-selector);
```

Затем в объекте jQuery задайте селектор CSS:

\$(**"p+p"**);

Далее добавьте объявление CSS:

```
$("p+p").css("font-weight","normal");
```

#### Примечание

jQuery может не понимать некоторые свойства CSS с сокращенной формой записи. Если jQuery с помощью свойства CSS с сокращенной формой записи не меняет внешний вид страницы, как ожидалось, примените вместо него свойства CSS, входящие в свойство с со-кращенной формой записи. Например, свойство border-right вместо просто border.

#### Как скрыть лишний код на JavaScript от современных Web-обозревателей?

Применение jQuery предназначено для исправления ошибок в более старых версиях Internet Explorer, а современные Web-обозреватели уже поддерживают это правило CSS. Следовательно, повторное задание данного визуального эффекта избыточно.

Для уменьшения нагрузки на Web-обозреватель используйте условные комментарии (см. *разд. 12.7*), чтобы JavaScript увидели только более ранние версии Internet Explorer:

```
<!--[if lt IE 8]>
<script type="text/javascript">
    // Your code goes here...
    $(document).ready(function(){
        $("#content p:last-child").css("margin","1em 0 0 0");
    });
</script>
<![endif]-->
```

#### Скрипт для ІЕ7 Дина Эдвардса

Скрипт для IE7 Дина Эдвардса (Dean Edwards) (http://code.google.com/p/ie7-js/) пытается с помощью JavaScript устранить множество проблем в ранних версиях IE.

Присоединив к вашей Web-странице соответствующий код на JavaScript, вы сможете исправить большое количество ошибок, одолевающих эти Web-обозреватели. Но исправление для IE7 касается проблем, возникающих только в этих Web-обозревателях, да и размер файла исправления не малый. Имея в виду размер файла 71.1 Кбайт, следует взвесить важность применения большого файла для исправления более ранних версий Web-обозревателей.

Кроме того, скрипт предлагается как бета-версия и его последнее обновление проводилось в феврале 2008 г. Несмотря на то, что скрипт Дина Эдвардса совершает много полезной работы по устранению проблем в IE6, если вы будете стараться "раздвинуть границы", чтобы заставить IE6 вести себя как современный Web-обозреватель, могут появиться новые проблемы.

#### Примечание

Ряд современных библиотек на JavaScript во многом состоит из кода Дина Эдвардса.

#### См. также

Страницу с функцией css по адресу http://docs.jquery.com/API/1.3/CSS.

## 14.6. Создание с помощью JavaScript HTML-таблицы с окраской зебры

## Проблема

Вы хотите задать чередующиеся цвета фона для строк HTML-таблицы, не добавляя вручную селекторы класса.

#### Решение

Воспользуйтесь способностью jQuery добавлять и удалять атрибуты HTML-элементов.

Сначала создайте правила CSS для чередующихся цветов:

```
table.striping tr.diff td {
  background-color: #cbclbe;
}
```

Затем напишите код для jQuery, в котором выбираются строки через одну:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script>
<script type="text/javascript">
$(document).ready(function(){
$(".striping tr:even").addClass("diff");
});
</script>
```

После того как строка выбрана, для каждой четной строки добавьте атрибут class со значением diff, чтобы применить чередующиеся цвета фона, как показано на рис. 14.3:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$(".striping tr:even").addClass("diff");
});
</script>
```

Test Availability & Cost Types of certification tests along with their prices and test day.				
Type of Test	Cost	Date		
Buying Microwave Popcorn	\$100.00	October 9th		
Bad Popcorn Protocols	\$299.00	October 12th		
Microwave Popcorn Certification	\$599.00 October 13th			
Bad Popcorn Anger management	\$799.00	October 14th		

Рис. 14.3. Чередующиеся полосы из строк таблицы

#### Обсуждение

В отличие от *разд. 14.3*, решение которого полагается на правило CSS, жестко записанное в коде на JavaScript, здесь правило CSS пишется заранее. Затем код на JavaScript автоматически добавляет атрибут class к каждой четной строке таблицы.

#### Применение решения, основанного только на CSS

Для данного раздела решение на основе только правил CSS состоит в использовании псевдоэлемента :nth-child (см. *разд. 9.8*)

```
tr:nth-child(even) td {
  background-color: #cbclbe;
}
```

Для скрытия кода на JavaScript вы можете применить наряду с решением на основе jQuery условные комментарии (как показано в *обсуждении*, в *разд. 12.3*).

#### См. также

Страницу с описанием атрибута jQuery addClass по адресу http://docs.jquery.com/ addclass.

## 14.7. Выделение цветом строки таблицы при наведении указателя мыши

#### Проблема

Вы хотите даже в Internet Explorer 6 предоставить способ выделения цветом строки таблицы.

#### Решение

Создайте правило CSS с селектором класса для указания цвета фона выделенной строки таблицы:

```
table.striping tr.over td {
  background-color: #fbc057;
}
```

Затем воспользуйтесь объектом jQuery для указания места применения селектора класса:

```
$(".striping tr");
```

Заставьте jQuery активизироваться, только когда пользователь наведет указатель мыши на ссылку:

```
$(".striping tr").mouseover();
```

Далее запустите функцию:

```
$(".striping tr").mouseover(function() {
});
```

Сообщите jQuery о том, что функция применяется только к тому, что выделено в настоящий момент (а это строки таблицы):

```
$(".striping tr").mouseover(function() {
  $(this);
});
```

Примените функцию addClass() для вставки в строку таблицы атрибута класса со значением over:

```
$(".striping tr").mouseover(function() {
    $(this).addClass("over");
});
```

Теперь, когда пользователь перемещает указатель мыши по строкам таблицы, они выделяются цветом. Но пока в решении атрибут class только вставляется и не удаляется, когда указатель мыши пользователя переходит на другую строку или за пределы таблицы (рис. 14.4).

Test Availability & Cost Types of certification tests along with their prices and test day.				
Type of Test	Cost	Date		
Buying Microwave Popcorn	\$100.00	October 9th		
Bad Popcorn Protocols	\$299.00	October 12th		
Microwave Popcorn Certification	\$599.00 🕭	October 13th		
Bad Popcorn Anger management	\$799.00	October 14th		

Рис. 14.4. Строки таблицы с измененным цветом фона

Для удаления атрибута class используйте функцию removeClass () (рис. 14.5):

```
$(".striping tr").mouseover(function() {
   $(this).addClass("over");
});
$(".striping tr").mouseout(function() {
   $(this).removeClass("over");
});
```

Test Availability & Cost Types of certification tests along with their prices and test day.				
Type of Test	Cost	Date		
Buying Microwave Popcorn	\$100.00	October 9th		
Bad Popcorn Protocols	\$299.00	October 12th		
Microwave Popcorn Certification \$599.00		October 13th		
Bad Popcorn Anger management	\$799.00	October 14th		

Рис. 14.5. Возвращение прежнего цвета фона при перемещении указателя мыши за пределы строки таблицы

#### Обсуждение

В предложенном решении для создания интересных визуальных эффектов на Webстранице вводятся два полезных события: mouseover() и mouseout(). Оба представляют собой обычные часто используемые функции на JavaScript, которые применялись для отображения графики при наведении указателя мыши до того, как популярность приобрели решения, использующие только средства CSS.

#### Цепочки вызовов функций

Для событий jQuery, связанных с одними и теми же элементами Web-страницы, строками таблицы, можно сократить код с помощью создания *цепочек вызовов функций* (chaining). Этот прием удаляет дублирующийся объект jQuery следующим образом:

```
$(".striping tr").mouseover(function() {
  $(this).addClass("over");
}).mouseout(function() {
  $(this).removeClass("over");
});
```

#### См. также

Страницу с описанием функции jQuery removeClass по адресу http://docs.jquery.com/ Attributes/removeClass.

## 14.8. Визуальное оформление простого наведения указателя мыши

#### Проблема

Вы хотите добавить на Web-страницу угасание или обесцвечивание отображения элементов при наведении указателя мыши.

### Решение

Задайте объект jQuery с функцией mouseover (рис. 14.6):

```
$("#site-nav a").mouseover(function () {
});
```

Затем примените функцию fadeTo() с непрозрачностью, равной 50 %:

```
$("#site-nav a").mouseover(function() {
  $(this).fadeTo("slow", .50);
});
```

CERTIFICATIONS	DOES IT REALLY SMELL THAT BAD?	ABOUT US	CONTACT
Salted and Buttered Versions	The One 🆓 Smelled it Has Not Always Dealt it	We Are For Real	Pop on Over

Рис. 14.6. Перемещение указателя мыши на изображение для создания эффекта обесцвечивания

Теперь вставьте дополнительную функцию mouseout, чтобы вернуть 100 % непрозрачности, когда пользователь перемещает указатель мыши в другое место (рис. 14.7):

```
$("#site-nav a").mouseover(function () {
  $(this).fadeTo("slow", .50);
}).mouseout(function () {
  $(this).fadeTo("slow", 1);
});

CERTIFICATIONS
DOES IT REALLY SMELL THAT BAD?
ABOUT US
CONTACT
The One Who Smelled It Has Not Aways Dealt It
We Are For Real
Pop on Over
```

Рис. 14.7. Отображение возвращается к 100-процентной непрозрачности при перемещении указателя за пределы изображения

#### Обсуждение

Функция fadeTo() принимает одно из трех ключевых слов, определяющих скорость: slow (медленно), normal (обычно) или fast (быстро). Вместо одного из этих слов можно использовать число, задающее миллисекунды:

```
$("#site-nav a").mouseover(function() {
$(this).fadeTo(2000, .50);
});
```

Обесцвечивание элементов на Web-странице — это лишь один из множества встроенных в jQuery визуальных эффектов. К ним относятся настраиваемые анимации, скольжение элемента, отображение и скрытие. Полный перечень можно найти на странице http://docs.jquery.com/Effects.

#### См. также

Страницу с описанием функции jQuery fadeTo() по адресу http://docs.jquery.com/ Effects/fadeTo.

## 14.9. Формирование строки одной высоты из элементов с разным объемом контента

#### Проблема

Вы хотите, чтобы высота строки из нескольких элементов была равна высоте самого высокого из них.

#### Решение

Сначала инициализируйте переменную с нулевым значением:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script>
<script type="text/javascript">
```

```
$ (document).ready(function() {
    var topHeight = 0;
    });
</script>
```

Заставьте jQuery циклически обработать все элементы, определенные в объекте jQuery. В данном случае jQuery обрабатывает в цикле каждый элемент р в родительском элементе div со значением id, равным content:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script>
<script>
<script type="text/javascript">
$(document).ready(function(){
var topHeight = 0;
$("#content p").each();
});
</script>
```

По мере циклической обработки элементов р jQuery определяет, больше ли высота элемента первоначально заданного значения. Поскольку начальное значение переменной topHeight равно нулю, очевидно, что будет выполняться оператор if:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js"></script>
<script type="text/javascript">
$ (document).ready(function() {
    var topHeight = 0;
    $("#content p").each(function() {
        if ($(this).height() > topHeight) {
        }
     });
    });
</script>
```

Поскольку значение topHeight изменилось, сохраните последнее значение самой большой высоты элемента:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script>
<script>
<script type="text/javascript">
$(document).ready(function() {
   var topHeight = 0;
   $("#content p").each(function() {
      if ($(this).height() > topHeight) {
        topHeight = $(this).height();
      }
   });
});
</script>
```

Когда jQuery завершит циклическую обработку элементов р и определит максимальную высоту, присвойте полученное значение всем остальным элементам в строке (рис. 14.8):

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script>
<script type="text/javascript">
$(document).ready(function(){
var topHeight = 0;
$("#content p").each(function(){
if ($(this).height() > topHeight) { topHeight = $(this).height(); }
});
$("#content p").height(topHeight);
});
```

```
</script>
```



Рис. 14.8. Одинаковая высота у всех элементов в строке

#### Обсуждение

Это очень простой, но полезный прием. Для определения высоты ряда элементов на странице применяйте для их циклической обработки функцию each(). После того как значение найдено и сохранено в переменной, его можно применять ко всем этим элементам.

#### См. также

Страницу с описанием функции jQuery each() по адресу http://docs.jquery.com/ Utilities/jQuery/each.

## 14.10. Задание ссылки, открывающей новое окно

### Проблема

Вы хотите раскрыть новое окно при щелчке кнопкой мыши по ссылке.

#### Решение

Сначала воспользуйтесь атрибутом rel и задайте ему значение external:

<a href="http://csscookbook.com/" rel="external">Click here</a> to check it out!

Задайте объект jQuery, чтобы пометить с помощью селектора атрибута все ссылки на Web-странице с атрибутом rel, равным значению external:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script>
<script>
<script type="text/javascript">
$(document).ready(function(){
$('a[rel="external"]');
});
</script>
```

Примените функцию click():

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script>
<script type="text/javascript">
$(document).ready(function(){
$('a[rel="external"]').click();
});
</script>
```

Вставьте функцию, которая открывает новое окно с адресом ссылки уже вписанным в элемент а:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script>
```

```
<script type="text/javascript">
  $(document).ready(function() {
   $('a[rel="external"]').click(function() {
    window.open($(this).attr('href')););
  });
});
</script>
```

Благодаря этим действиям Web-обозреватель будет загружать адрес ссылки и в новое окно, и в родительское. Для того чтобы этого не произошло, запретите переход по ссылке в родительском окне Web-обозревателя:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script>
<script type="text/javascript">
$(document).ready(function() {
$('a[rel="external"]').click(function() {
window.open($(this).attr('href'));
return false;
});
});
</script>
```

#### Обсуждение

В случае strict DOCTYPE применение атрибута target не разрешено и превращает разметку в некорректную с точки зрения валидатора. Использование JavaScript позволяет устранить это затруднение.

#### Примечание

Лучше вовсе отказаться от открытия нового окна, если возможно. Не рассчитывайте на то, что у ваших пользователей есть настольная система с Web-обозревателем для просмотра вашего контента.

#### См. также

Страницу с описанием функции jQuery click() по адресу http://docs.jquery.com/ Events/click.

## 14.11. Превращение элемента *div* целиком в элемент, активируемый щелчком мыши

#### Проблема

Вы хотите сформировать блочный элемент, активируемый щелчком кнопки мыши.

#### Решение

Задайте в элементе div атрибут class со значением link:

```
<div class="link" id="blipvert">
<h2>Amazing Sale</h2>
<a href="http://csscookbook.com/">Click here</a> to check it out!
</div>
```

Примените объект jQuery для выбора всех элементов div со значением селектора класса, равным link:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script>
<script>
<script type="text/javascript">
$(document).ready(function() {
    $("div.link").click(function() {
    });
});
</script>
```

Используйте функцию find() для поиска ссылки в элементе div и применения этой ссылки как пункта назначения при щелчке кнопкой мыши этого элемента div:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script>
<script type="text/javascript">
$(document).ready(function() {
    $("div.link").click(function() {
    window.location=$(this).find("a").attr("href");
    return false;
    });
});
</script>
```

### Обсуждение

Версия HTML5, поддерживаемая не во всех современных Web-обозревателях, позволяет активировать блочные элементы в ссылке щелчком кнопки мыши:

```
<div id="blipvert">
<a href="http://csscookbook.com/">
<h2>Amazing Sale</h2>
Click here to check it out!
</a>
</div>
```

#### См. также

Страницу с описанием функции jQuery find() по adpecy http://docs.jquery.com/ Traversing/find.

## 14.12. Поддержка средствами JavaScript формата PNG в IE6

## Проблема

Вы хотите применять альфа-прозрачные PNG-изображения в Internet Explorer 6.

### Решение

Примените дополнительный модуль (plug-in), специально разработанный для поддержки формата PNG в более старых версиях Internet Explorer 6.

Сначала загрузите файл jquery.pngFix.js c Web-страницы http://jquery.andreaseberhard.de/ pngFix (рис. 14.9).

Вставьте файл jquery.pngFix.js после библиотеки jQuery:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script>
```

<script type="text/javascript" src="/\_assets/js/jquery.pngFix.js"></script>



Рис. 14.9. Начальная страница с дополнительным модулем jQuery

#### Затем активируйте дополнительный модуль:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script></script>
```

```
<script type="text/javascript" src="/_assets/js/jquery.pngFix.js"></script>
  <script type="text/javascript">
    $(document).ready(function() {
    $(document).pngFix();
    });
</script>
```

## Обсуждение

Поскольку это решение относится к старым версиям Internet Explorer, условные комментарии помогут изолировать файлы от современных Web-обозревателей, естественным образом поддерживающих альфа-прозрачные PNG-изображения:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script>
</script>
<script type="text/javascript" src="/_assets/js/jquery.pngFix.js"></script>
<script type="text/javascript">
    $(document).ready(function() {
        $(document).ready(function() {
        $(document).pngFix();
      });
</script>
<![endif]-->
```

#### Проблемы PNG-изображений в IE6

Несмотря на то, что решение на JavaScript требует большого объема ручного кодирования во всем документе, Web-разработчикам следует иметь в виду несколько проблем, касающихся способа обработки альфа-прозрачных PNG-изображений в Internet Explorer.

Во-первых, в решении применяется собственное свойство Microsoft filter, способное обрабатывать альфа-прозрачные PNG-изображения. Это позволяет включить альфа-прозрачность в набор инструментов Web-дизайнера, но иметь в IE внутристрочные PNG-изображения все равно невозможно, их можно использовать только как фон элементов.

Несмотря на то, что изображение помещается в фон элемента, оно растягивается в соответствии с размерами этого элемента. Это вторая проблема, т. к. этот способ поведения противоречит общепринятой практике, на которую рассчитывают Web-дизайнеры: изображение сохраняет свои размеры и просто отображается как мозаика.

Таким образом, применяя PNG-изображения в IE6, убедитесь в том, что размеры PNGизображения точно совпадают с размерами элемента; в противном случае может возникнуть нежелательное растяжение изображения.

#### Поиск дополнительных модулей jQuery

Одно из преимуществ применения jQuery — огромное множество потенциальных разработчиков. Если у вас появилась какая-то проблема, вполне вероятно, что профессиональный программист разработал дополнительный модуль на JavaScript для ее решения. Просто выполните поиск в Google вашей проблемы с ключевым словом jQuery, и возможно вы будете приятно удивлены.

#### Примечание

Дополнением к этой книге может служить книга Коди Линдли (Cody Lindley) jQuery Cookbook (jQuery. Рецепты программирования) (изд. O'Reilly).

#### См. также

*Разд. 4.17* о способе создания изображений с альфа-прозрачностью в формате PNG8, который работает в IE6.

## 14.13. Предоставление кода на HTML5 и свойств CSS3 Web-обозревателям, способным их обрабатывать

#### Проблема

Вы хотите задать стили, использующие преимущества свойств спецификации CSS3, и включить альтернативный вариант для Web-обозревателей, не способных их обрабатывать.

#### Решение

Загрузите библиотеку Modernizr на JavaScript с Web-сайта http://www.modernizr.com.

В элемент head включите ссылку на библиотеку Modernizr:

<script src="modernizr-0.9.min.js"></script>

Затем примените селекторы класса, чтобы использовать свойства CSS3 в тех Webобозревателях, которые могут их отображать:

```
h1 {
   background-color: #333;
   color: #fff;
}
.rgba h1 {
   background-color: rgba(0,0,0, .8);
}
```

Для применения кода на HTML5 используйте аналогичный подход. Сначала выполните разметку с помощью элементов на HTML5, например, элемента audio:

```
<audio>
<source src="example.ogg" />
<source src="example.mp3" />
</audio>
```

Затем примените правила CSS, скрывающие элемент audio от Web-обозревателей, не поддерживающих его:

```
.no-audio audio {
  display: none;
}
```

## Обсуждение

Несмотря на то, что разработка новых свойств и функциональных возможностей Webобозревателей всячески приветствуется, трудности все еще остаются. В то время как быстрыми темпами в Web-обозревателях началась реализация стандартов HTML5 и CSS3, проблемой становится разный уровень поддержки этих стандартов в популярных современных обозревателях.

Web-дизайнеры Фарук Атеш (Faruk Ateş) и Пол Айриш (Paul Irish) создали простую библиотеку на JavaScript, обеспечивающую базовые разработки, не зависящие от конкретного Web-обозревателя.

Во время написания книги библиотека Modernizr проверяла наличие следующих свойств и функций HTML5 и CSS3:

- ♦ СВОЙСТВО орасіту:
- ♦ анимацию CSS
- ♦ колонки CSS
- ♦ градиенты CSS
- ♦ отражения CSS
- ◆ 2D-преобразования CSS
- ◆ 3D-преобразования CSS
- ♦ переходы CSS
- Geolocation API
- ♦ @font-face

#### См. также

Документацию к библиотеке Modernizr на странице http://www.modernizr.com/docs/.

- ♦ Canvas
- Canvas text
- ♦ HTML5 audio
- ♦ HTML5 video
- rgba()
- ♦ hsla()
- ♦ border-image:
- ♦ border-radius:
- ♦ box-shadow:
- множественные фоновые изображения

## приложение 1 Источники

Работая с каскадными таблицами стилей, помните о двух правилах: упрощай и проверяй.

После того как вы создали ваши правила CSS, упростите их, применяя только те селекторы и свойства, в необходимости которых вы уверены, любые излишества впоследствии могут вызвать затруднения. Затем проверьте код HTML, XHTML и CSS с помощью валидаторов (validator). Эти два этапа разрешают большую часть проблем, с которыми сталкиваются разработчики, применяющие технологию CSS.

Но если вы все же столкнетесь с трудностями, в данном приложении перечислены некоторые основные ссылки, дискуссионные группы и средства в сети Интернет, которые могут помочь в разработке таблиц CSS.

## Основные обучающие материалы, относящиеся к языку HTML и технологии CSS

 "A Roadmap to Standards" (Верная дорога к стандартам) (http://www.mezzoblue.com/archives/2004/04/30/a\_roadmap\_to/index.php).

Это эссе Дэвида Ши (David Shea) — хорошее введение и подбадривающая информация для Web-дизайнеров, решивших узнать о разработке на основе Webстандартов.

 "CSS from the Ground Up" (CSS, начиная с азов) (http://www.wpdfd.com/editorial/basics/index.html).

Web-разработчики, впервые знакомящиеся с технологией CSS, найдут полезную информацию в этом подробном пошаговом учебнике с Web-сайта Web Page Design for Designers.

 "Basics of CSS Positioning" (Основы позиционирования в CSS) (http://www.communitymx.com/content/article.cfm?cid=3B56F&print=true).
 Ининте в этом унебном пособии компании Community MX пополнители имо и

Ищите в этом учебном пособии компании Community MX дополнительную информацию о позиционировании средствами CSS.

4. Floatutorial (Учебное пособие, описывающее плавающую модель) (http://css.maxdesign.com.au/floatutorial/index.htm).

Этот учебник знакомит с элементами плавающей модели CSS на разнообразных практических примерах кода, предоставленных Max Design.

 Selectutorial (Учебное пособие по селекторам CSS) (http://css.maxdesign.com.au/selectutorial/index.htm).

Это учебное пособие также от Max Design помогает лучше понять селекторы CSS. В него включена демонстрация возможного применения селекторов в структуре трехколоночного макета.

## Дизайнерские ресурсы

1. A List Apart: "CSS Topics" (http://alistapart.com/topics/code/css/).

На этом Web-сайте большинство статей, публикуемых в рубрике CSS, приходят от Web-дизайнеров, которые делятся своими идеями и открытиями, связанными с дизайном средствами CSS.

2. Хранилище макетов (http://www.buerobot.com/web/layouts/).

Маленький, но ценный ресурс на сайте BlueRobot.com, содержащий двух- и трехколоночные макеты страниц.

3. CSS/Edge (http://www.meyerweb.com/eric/css/edge/).

Мастерская Эрика Мейера (Eric Meyer), отображающая некоторые из его наиболее передовых экспериментов в области CSS.

4. CSS Zen Garden (http://www.csszengarden.com/).

На сайте CSS Zen Garden выставляются образцы стилевого оформления одного и того же контента, выполненные Web-разработчиками со всего света. Просмотр нескольких примеров дизайна дает не только огромный заряд вдохновения, но и фантастическую возможность осознания идеи разделения визуального представления и информационного содержимого страницы.

5. "CSS Layout Techniques" (Технические приемы верстки средствами CSS) (http://www.glish.com/css/).

На этой Web-странице представлена одна из первых коллекций многоколоночных макетов, созданных средствами CSS без применения HTML-таблиц.

6. Блог Microformats (http://www.microformats.org/).

Этот блог описывает и продвигает стандарты для кодирования специфических фрагментов контента. Найдите в перечне микроформатов методы, которые можно использовать для кодирования общераспространенных данных, таких как календарные события, контактная информация или элемент abbr.

7. Архивы SimpleQuiz (http://www.simplebits.com/bits/simplequiz/).

Web-дизайнер и автор Дэн Седерхольм (Dan Cederholm) проводит опросы, стараясь определить наилучшие методы разметки и стилевого оформления распространенных сценариев Web-разработки. Помимо прилагаемых к каждому опросу выводов вы можете прочесть в любом опросе комментарии Web-разработчиков, чтобы узнать больше о практических приемах кодирования.

#### Статьи о CSS журнала Smashing Magazine (http://www.smashingmagazine.com/category/css/).

Этот интерактивный журнал собирает сообщения из блогов и статьи Web-разработчиков и дизайнеров в Интернете и публикует краткие рефераты о своих находках.

9. Typetester (http://typetester.maratz,com/).

Гибкое средство, позволяющее Web-разработчикам настроить три набора шрифтов и затем генерирующее базовые правила CSS, которые легко скопировать и вставить. Доступные параметры настройки включают установку шрифтов, их размеров, трекинга, межстрочного интервала, межбуквенного расстояния, выравниваний и т. д.

## Дискуссионные группы

#### 1. Babble List (http://www.babblelist.com/).

Координируемый Кристофером Шмиттом (Christopher Schmitt) список рассылки, посвященный Web-дизайну и разработке, ориентирован на сложные задачи Webдизайна. На сайте предлагается живой обмен информацией, источниками, теоретическими и практическими знаниями дизайнеров и разработчиков.

2. css-discuss.org (http://www.css-discuss.org/).

Этот список рассылки, координируемый экспертом по CSS, Эриком Мейером (Eric A. Meyer), автором книги "Cascading Style Sheets: The Definitive Guide " (Каскадные таблицы стилей: подробное руководство) издательства O'Reilly, стремится поддерживать практическую дискуссию о применении CSS.

3. WebDesign-L.com (http://www.webdesign-l.com/).

Этот список рассылки действует с 1997 г. и касается почти всех аспектов создания Web-сайтов, включая (но не ограничиваясь) CSS. Встречаются серьезные вопросы с мудрыми советами.

4. Usenet Stylesheets Newsgroup

#### (http://news:comp.infosystems.www.authoring.stylesheets).

Эта группа новостей без модератора, основанная в 1997 г., посвящена вопросам теории и практики применения CSS. Материалы группы могут включать практические приложения, вопросы, касающиеся спецификации, преимущества технологии CSS, дефекты реализации Web-обозревателей и т. д.

5. www-style@w3.org Mail Archives (архив списков рассылки W3C).

#### http://lists.w3.org/Archives/Public/www-style/

Поддерживаемый консорциумом World Wide Web Consortium (W3C), этот список рассылки служит для обсуждения теоретических вопросов и будущего технологии CSS. Приветствуются вопросы, касающиеся спецификации и предложений CSS, однако обсуждения, вращающиеся вокруг практического применения технологии, не одобряются.

## Справочные материалы

1. Css Browser Support charts

#### (http://www.westciv.com/wiki/CSS\_Compartibility\_Guide/).

Если вы столкнулись с проблемами при разработке средствами CSS, проверьте таблицы поддержки CSS, чтобы определить, нет ли проблемы у Web-обозревателя (Web-обозревателей), которым вы пользуетесь.

#### 2. CSS filters (http://centricle.com/ref/css/filters/).

Обратите несогласованность Web-обозревателей в свою пользу. Если вы хотите ориентировать правила CSS на конкретный Web-обозреватель или набор Web-обозревателей, обратитесь к этому исчерпывающему перечню программистских приемов и фильтров. Он подскажет, какие правила и объявления CSS действуют в конкретных Web-обозревателях или не действуют, и такое случается.

#### 3. Рекомендуемые W3C DTD (http://www.w3.org/QA/2002/04/valid-dtd-list.html).

Назначение Web-странице правильного **DOCTYPE** поможет обеспечить корректный способ визуализации Web-обозревателем вашей страницы и метод проверки вашего кода валидаторами. На этой странице представлен список наиболее часто используемых типов **DOCTYPE**.

4. Начальная страница консорциума W3C, посвященная CSS, (http://www.w3.org/Style/CSS/).

Это официальный сайт, посвященный технологии CSS. Здесь вы можете узнать об истории CSS, ресурсах и авторизованных средствах и прочитать текущие новости CSS.

#### 5. Спецификация CSS 2.1 (http://www.w3.org/TR/CSS21/).

Реализация в Web-обозревателях спецификации CSS порой представляет собой полную неразбериху. Когда вы пытаетесь выяснить, как добиться определенного визуального представления или устранить ошибку визуализации, обращайтесь сюда для проверки спецификации (и таблиц поддержки CSS).

#### 6. Спецификация CSS3 (http://www.w3.org/Style/CSS/current-work/).

Грядущая спецификация CSS3 все еще пишется. Поскольку она сложна и объемна, рабочий вариант проекта разбит на отдельные модули; идея заключается в том, что работа, продолжающаяся в одном модуле, может идти независимо от других модулей, не вызывая замедления работы в них. В результате во время написания книги существовали разные разделы CSS3 с различной степенью завершенности, но большинство в стадии разработки (Working Draft).

#### 7. Спецификация HTML 4.01 (http://www.w3.org/TR/html4/).

Для оптимального применения CSS в Web-дизайне необходимо создавать Webдокументы со структурированной разметкой, а не пользоваться искусственными приемами и программистскими трюками. Более того, следует размечать документы с помощью элементов, заключающих в себе репрезентативный смысл. Например, выделять важные слова следует с помощью элемента em, а не элемента b. Если вам необходимо изменить методы в вашем продукте, прочесывайте HTML-спецификацию на этом сайте и снова и снова получайте сведения об элементах.

#### 8. Спецификация HTML5 (http://dev.w3.org/ html5/spec/Overview.html).

В связи с потребностями развития Web-приложений и несовершенством версии HTML4 работа над HTML5 продолжается, и пока нет назначенного срока в ближайшем будущем для утверждения ее в качестве приемлемой рекомендации. Но даже имея незавершенную спецификацию, Web-разработчики могут применять ту ее часть, которая реализована в современных Web-обозревателях.

#### 9. Спецификация XHTML 1.0 (http://www.w3.org/TR/xhtml1/).

Extensible HyperText Markup Language (расширяемый язык гипертекстовой разметки) (XHTML) — это преобразование языка HTML4 в язык XML 1.0 (Extensible Markup Language, расширяемый язык разметки). Несмотря на то, что разметка XHTML строже, чем разметка HTML4, преимущества его очевидны: более логичная разметка, возросшая функциональная совместимость и большая доступность.

## Программные средства

1. BrowserCam (http://www.browsercam.com/).

BrowserCam — доступный Web-сервис, который тестирует Web-дизайн в разных Web-обозревателях под управлением нескольких операционных систем. Во время написания книги был доступен 24-часовой период ознакомления для Web-разработчиков, зарегистрировавшихся на сайте.

2. CleanCSS (http://www.cleancss.com/).

Интерактивное средство форматирования и сжатия длинных и сложных таблиц стилей. Это свободно распространяемое программное средство позволяет экспортировать сжатые файлы CSS, что устраняет потенциальные проблемы с наборами символов при копировании из Web-обозревателей и вставке в редакторы кода.

#### 3. Firebug (https://addons.mozilla.org/en-US/firefox/addon/1843).

Это свободно распространяемое средство для Web-разработчиков позволяет быстро редактировать, кодировать и находить ошибки в коде HTML, CSS и JavaScript, включенном в Web-страницу, просматриваемую в текущий момент. Кроме того, это чудесное средство для поиска ошибок в Web-приложениях на базе Ajax. Вы также можете установить дополнительный модуль CodeBurner (см. http://tools.sitepoint.com/codeburner/), предоставляющий справочные материалы по HTML и CSS в среде разработки Firebug.

#### 4. IE NetRenderer (http://ipinfo.info/netrenderer/index.php).

Это свободно распространяемое средство позволяет Web-разработчикам выполнять предварительный просмотр Web-страниц, как если бы они отображались в Internet Explorer. Кроме того, это очень полезный сайт для пользователей Macintosh, не имеющих компьютеров с OC Windows, но желающих протестировать или использовать программное обеспечение для виртуальной машины, чтобы запускать Windows вместе с OS X.

#### 5. SelectORacle (http://gallery.theopalgroup.com/selectoracle/).

Бесплатный сервис, разработанный для того, чтобы помочь людям узнать больше о сложных селекторах CSS благодаря переводу их значений на простой английский. Селекторы CSS можно задать двумя способами. Первый метод — скопировать и вставить селектор CSS в форму. Второй метод — ввести URL Web-страницы со встроенной таблицей стилей или URL внешней таблицы стилей. Затем сервис переведет селекторы на понятный английский язык.

6. W3C CSS Validator (http://jigsaw.w3.org/css-validator/).

Бесплатный сервис, представленный на сервере консорциума W3C, проверяет точность структуры правил CSS. Вы можете протестировать свои стили, загрузив файлы на сервер, введя Web-адрес в форму и затем копируя и вставляя правила CSS в поле формы. Можно загрузить из Интернета и установить программу проверки (валидатор) на вашем сервере.

7. W3C HTML Validator (http://validator.w3.org/).

W3C HTML Validator — еще один бесплатный сервис консорциума W3C. Аналогично валидатору CSS он проверяет, соответствует ли ваша разметка Webстандартам.

8. Web Developer browser extension (Расширение Web-обозревателя для Web-разработчика) (https://addons.mozilla.org/en-US/firefox/addon/60).

Крис Педрик (Chris Pedrick) создал незаменимое расширение для популярных Webобозревателей Firefox и Mozilla. Его функциональные возможности включают редактирование в Web-обозревателе таблицы стилей Web-страницы, отправку кода Web-страницы прямо в валидатор консорциума W3C, вывод контура вокруг элементов блочного уровня и много-много других функций, выполняемых одним щелчком кнопки мыши.

#### 9. xScope (http://iconfactory.com/software/xscope).

хScope — это набор средств Мас, разработанных для экономии времени при тонкой настройке Web-проекта. В набор входит мое любимое средство Dimensions, немедленно определяющее расстояние между объектами; прошли те дни, когда вам нужно было сделать моментальный снимок экрана и затем открыть Photoshop для того, чтобы измерить расстояние между колонками или межстрочный интервал в тексте. Для пользователей Windows есть несколько отдельно разработанных сторонними организациями расширений Firefox, которые дублируют некоторые функции программных средств, включенных в данный пакет. Например, Measurelt — расширение для вывода линеек в Web-обозревателе (см. https://addons.mozilla.org/ en-US/firefox/addon/539).

### ПРИЛОЖЕНИЕ 2

# Свойства CSS 2.1 и специализированные дополнения

В это приложение включено несколько таблиц. В табл. П2.1 перечислены свойства CSS из спецификации CSS 2.1 консорциума W3C (http://www.w3.org/TR/CSS21). В ней приведен перечень названий свойств, начальных значений, типов элементов, к которым применяется свойство, характеристика наследуемости значений свойства, возможность использования процентов как единицы измерения и типов вывода (media group).

В табл. П2.2 включен перечень дополнений к спецификации, разработанных корпорацией Microsoft. Корректность этих свойств не будет подтверждена программой проверки (валидатором), и они будут успешно действовать только в Web-обозревателе, разработанном фирмой Microsoft. Польза от применения этих свойств может быть разной.

К моменту написания книги частные дополнения для программных продуктов Mozilla, доступные в Web-обозревателях Mozilla, Firefox и Netscape Navigator 6 и более поздних версиях, были не полностью документированы. В табл. П2.3 приведена часть дополнений, описанная на Web-сайте разработчиков Mozilla. Полный перечень этих дополнений можно найти на странице http://developer.mozilla.org/en/docs/CSS\_Reference: Mozilla Extensions.

Таблица П2.1 принадлежит консорциуму W3C (Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University). Все права защищены. http://www.w3.org/Consortium/Legal/2002/copyright-documents-20021231.
Название	Значения	Начальная величина	Применяется к (по умолчанию: аll)	Насле- дуется?	Проценты (по умолчанию: N/A)	Типы вывода
'background-attachment'	scroll   fixed   inherit	scroll		Нет		Визуальный
'background-color'	<color>   transparent   inherit</color>	transparent		Нет		Визуальный
'background-image'	<ur><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li></ur>	none		Нет		Визуальный
'background-position'	<pre>[[{percentage&gt;   <length>   left   center   right][ <percentage>   <length>   top   center   bottom]?]   [[left   center   bottom]]   [top   center   bottom]]  </length></percentage></length></pre>	*0		Нет	Относительно размера самого блока	Визуальный
'background-repeat'	repeat   repeat-x   repeat-y   no-repeat   inherit	repeat		Нет		Визуальный
'background'	['background-color'    'background-tmage'    'background-repeat'    'background-attachment'    'background-position']   ihherit	См. отдельные свойства		Нет	Paзрешены для 'background-position'	Визуальный
'border-collapse'	collapse   separate   inherit	separate	Элементы 'table'и 'inline-table'	Да		Визуальный
'border-color'	<pre>[<color>   transparent ]{1,4}   inherit</color></pre>	См. отдельные свойства		Нет		Визуальный
'border-spacing'	<length> <length>?   inherit</length></length>	0	Элементы 'table'и 'inline-table'	Да		Визуальный
'border-style'	<pre><bounder-style> {1,4}   inherit</bounder-style></pre>	См. отдельные свойства		Нет		Визуальный
'border-top' 'border-right' 'border-bottom' 'border-left'	<pre>[  cborder-width&gt;    <border- style="">    'border-top-color']   inherit</border-></pre>	См. отдельные свойства		Нет		Визуальный

**Таблица П2.1.** Свойства CSS 2.1

Приложение 2

648

#### Свойства CSS 2.1 и специализированные дополнения

					<b>Таблица П2.1</b> (про	должение)
Название	Значения	Начальная величина	Применяется к (по умолчанию: а11)	Насле- дуется?	Проценты (по умолчанию: N/A)	Типы вывода
<pre>'border-top-color' 'border-right-color' 'border-bottom-color' 'border-left-color' 'border-left-color'</pre>	<color>   transparent   inherit</color>	Значение свойства ' color'		Нет		Визуальный
<pre>'border-top-style' 'border-right-style' 'border-bottom-style' 'border-left-style'</pre>	<pre><border-style>   inherit</border-style></pre>	none		Нет		Визуальный
<pre>'border-top-width' 'border-right-width' 'border-bottom-width' 'border-left-width' 'border-left-width'</pre>	<pre><border-width>   inherit</border-width></pre>	medium		Нет		Визуальный
'border-width'	<pre><bounder-width> {1,4}   inherit</bounder-width></pre>	См. отдельные свойства		Нет		Визуальный
'border'	<pre>[<bodder-width>    <bodder- style="">    &gt;'border-top- color']   inherit</bodder-></bodder-width></pre>	См. отдельные свойства		Нет		Визуальный
'bottom'	<pre><length>   <percentage>   auto   inherit</percentage></length></pre>	auto	Позиционированные элементы	Нет	Относительно высоты блока-контейнера	Визуальный
'caption-side'	top   bottom   inherit	top	<b>Элементы</b> 'table- caption'	Да		Визуальный
'clear'	none   left   right   both   inherit	none	Элементы блочного уровня	Нет		Визуальный
'clip'	<shape>   auto   inherit</shape>	auto	Элементы с абсолют- ным позиционирова- нием	Нет		Визуальный
'color'	<color>   inherit</color>	Зависит от агента пользователя		Да		Визуальный

649

Название	Значения	Начальная величина	Применяется к (по умолчанию: аll)	Насле- дуется?	Проценты (по умолчанию: N/A)	Типы вывода
'content'	<pre>normal   none   [<string>   <uri>   <counter>   attr(<identifier>)   open-quote   close-quote   no-open-quote   no-close- quote]+  inherit</identifier></counter></uri></string></pre>	normal	Псевдоэлементы :before и :after	Нет		Bce
'counter-increment'	[ <identifier> <integer>?]+  none   inherit</integer></identifier>	none		Нет		Bce
'counter-reset'	[ <identifier> <integer>?]+  none   inherit</integer></identifier>	none		Нет		Bce
'cursor'	[[ <uri>]*[auto   crosshair   defailt   pointer   move   e-resize   ne-resize   nw-resize   n-resize   se-resize   sw-resize   s-resize   w-resize   text   wait   help   progress] ] </uri>	auto		Да		Визуаль- ный, инте- рактивный
'direction'	ltr   rtl   inherit	ltr	Все элементы, см. текст в книге	Да		Визуальный
display'	<pre>inline   block   list-item   run-in   inline-block   table   inline-table   table- group   table-feader- group   table-footer-group   table-column   table- group   table-column - table-column   table- cell   table-caption  none   inherit</pre>	inline		Нет		Bce
'empty-cells'	show   hide   inherit	show	Элементы 'table-cell'	Да		Визуальный
'float'	left   right   none   inherit	none	Bce	Нет		Визуальный
'font-family'	<pre>[[family-name&gt;   <generic- family&gt; ] [, <family-name>  <generic-family>]*]  inherit</generic-family></family-name></generic- </pre>	Зависит от агента пользователя		Да		Визуальный

30	йства	a CSS 2.1	и спе	циа.	пизирован	ные дополнени	я					
	Типы вывода	Визуальный	Визуальный	Визуальный	Визуальный	Визуальный	Визуальный	Визуальный	Визуальный	Визуальный	Визуальный	Визуальный
	Проценты (по умолчанию: N/A)	Задается относительно размера шрифта родительского элемента				См. отдельные свойства	См. текст в книге	Относительно ширины блока-контейнера		Относительно размера шрифта элемента		
	ісле- ется?					_	5	h			_	

Название	Значения	Начальная величина	Применяется к (по умолчанию: а11)	Насле- дуется?	Проценты (по умолчанию: N/A)	
'font-size'	<pre><absolute-size>  <relative- size&gt;   <length>   <per- centage&gt;   inherit</per- </length></relative- </absolute-size></pre>	medium		Да	Задается относительно размера шрифта родительского элемента	ш
'font-style'	normal   italic   oblique   inherit	normal		Да		ш
'font-variant'	normal   small-caps   inherit	normal		Да		ш
'font-weight'	normal   bold   bolder   lighter   100   200   300   400   500   600   700   800   900   inherit	normal		Да		ш
'font'	<pre>[['font-style'  'font- variant'   'font-weight']? 'font-size'[/'line- height']?'font-family']   caption   icon   menu   message-box   small-caption   status-bar   inherit</pre>	См. отдельные свойства		Да	См. отдельные свойства	ш
'height'	<le><le><li><li><li><li><li><li><li><li><li><li< td=""><td>auto</td><td>Все элементы, кроме неперемещаемых строчных элементов, табличных столбцов и групп столбцов</td><td>Нет</td><td>См. текст в книге</td><td>ш</td></li<></li></li></li></li></li></li></li></li></li></le></le>	auto	Все элементы, кроме неперемещаемых строчных элементов, табличных столбцов и групп столбцов	Нет	См. текст в книге	ш
'left'	<le><le>clength&gt;   <percentage>  <li>auto   inherit</li></percentage></le></le>	auto	Позиционированные элементы	Нет	Относительно ширины блока-контейнера	ш (
'letter-spacing'	normal   <length>  inherit</length>	normal		Да		ш
'line-height'	normal   <number>   <length>   <percentage>   inherit</percentage></length></number>	normal		Да	Относительно размера шрифта элемента	ш
'list-style-image'	<ur><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li><li></li></ur>	none	<b>Элементы с</b> 'display: list-item'	Да		ш
'list-style-position'	inside   outside   inherit	outside	<b>Элементы с</b> 'display: list-item'	Да		ш

Название	Значения	Начальная величина	Применяется к (по умолчанию: аll)	Насле- дуется?	Проценты (по умолчанию: N/A)	Типы вывода
'list-style-type'	disc   circle   square  decimal   decimal-leading- zero   lower-roman   upper- roman   lower-greek   lower- latin   upper-latin   armenian   georgian   lower- alpha   upper-alpha   none  inherit	disc	<b>Эпементы</b> c'display: list-item'	Да		Визуальный
'list-style'	['list-style-type'   'list- style-position'    'list- style-image' ]   inherit	См. отдельные свойства	<b>Элементы с</b> 'display: list-item'	Да		Визуальный
'margin-right' 'margin-left'	<pre><margin-width>   inherit</margin-width></pre>	0	Все элементы, кроме элементов с типом табличного вывода, но не элементы table и inline-table	Нет	Относительно ширины блока-контейнера	Визуальный
'margin-top' 'margin-bottom'	<pre><margin-width>   inherit</margin-width></pre>	0	Все элементы, кроме элементов с типом табличного вывода, но не элементы table и inline-table	Нет	Относительно ширины блока-контейнера	Визуальный
'margin'	<pre><margin-width> {1,4}   inherit</margin-width></pre>	см. отдельные свойства	Все элементы, кроме элементов с типом табличного вывода, но не элементы table и inline-table	Нет	Относительно ширины блока-контейнера	Визуальный
'max-height'	<pre><li><li><li><li><li><li><li><li><li><li< td=""><td>euou</td><td>Все элементы, кроме неперемещаемых строчных элементов, табличных столбцов и групп столбцов</td><td>Нет</td><td>См. текст в книге</td><td>Визуальный</td></li<></li></li></li></li></li></li></li></li></li></pre>	euou	Все элементы, кроме неперемещаемых строчных элементов, табличных столбцов и групп столбцов	Нет	См. текст в книге	Визуальный
'max-width'	<pre><length>   <percentage>   none   inherit</percentage></length></pre>	епоп	Все элементы, кроме неперемещаемых строчных элементов, табличных строк и групп строк	Нет	Относительно ширины блока-контейнера	Визуальный

					<b>Таблица П2.1</b> (про	должение)
Название	Значения	Начальная величина	Применяется к (по умолчанию: аll)	Насле- дуется?	Проценты (по умолчанию: N/A)	Типы вывода
'min-height'	<ledition <li=""><le><le><li><li><li><li><li><li><li><li><li><li< td=""><td>0</td><td>Все элементы, кроме неперемещаемых строчных элементов, табличных столбцов и групп столбцов</td><td>Нет</td><td>См. текст в книге</td><td>Визуальный</td></li<></li></li></li></li></li></li></li></li></li></le></le></ledition>	0	Все элементы, кроме неперемещаемых строчных элементов, табличных столбцов и групп столбцов	Нет	См. текст в книге	Визуальный
'min-width'	<length>   <percentage>   inherit</percentage></length>	0	Все элементы, кроме неперемещаемых строчных элементов, табличных строк и групп строк	Нет	Относительно ширины блока-контейнера	Визуальный
'orphans'	<integer>   inherit</integer>	2	Элементы блочного уровня	Да		Визуаль- ный, стра- ничный
'outline-color'	<color>  invert   inherit</color>	invert		Нет		Визуаль- ный, инте- рактивный
'outline-style'	<pre><body></body></pre> <pre>cborder-style&gt;   inherit</pre>	none		Нет		Визуаль- ный, инте- рактивный
'outline-width'	<pre><border-width>   inherit</border-width></pre>	medium		Нет		Визуаль- ный, инте- рактивный
'outline'	<pre>['outline-color'    'outline- style'    'outline-width']   inherit</pre>	См. отдельные свойства		Нет		Визуаль- ный, инте- рактивный
'overflow'	visible   hidden   scroll   auto   inherit	visible	Неперемещаемые элементы блочного уровня, табличные ячейки и встроенные блочные элементы	Нет		Визуальный
'padding-top' 'padding-right' 'padding-bottom' 'padding-left'	< padding-width >   inherit	0	Bce элементы, кроме элементов с типом табличного вывода, но не элементы table, inline-table и table- cell	Нет	Относительно ширины блока-контейнера	Визуальный

#### Свойства CSS 2.1 и специализированные дополнения

653

Название	Значения	Начальная величина	Применяется к (по умолчанию: аll)	Насле- дуется?	Проценты (по умолчанию: N/A)	Типы вывода
'padding'	<pre><pre><pre>cpadding-width&gt; {1,4}   inherit</pre></pre></pre>	См. отдельные свойства	Bce элементы, кроме элементов с типом табличного вывода, но не элемен- ты table, inline- table и table-cell	Нет	Относительно ширины блока-контейнера	Визуальный
'page-break-after'	auto   always   avoid   left   right   inherit	auto	Элементы блочного уровня	Нет		Визуальный, страничный
page-break-before'	auto   always   avoid   left   right   inherit	auto	Элементы блочного уровня	Нет		Визуальный, страничный
'page-break-inside'	avoid   auto   inherit	auto	Элементы блочного уровня	Да		Визуальный, страничный
'position'	static   relative   absolute   fixed   inherit	static		Нет		Визуальный
quotes	[ <string> <string>]+ none   inherit</string></string>	Зависит от агента пользователя		Да		Визуальный
'right'	<li><li><li><li><li><li><li><li><li><li></li></li></li></li></li></li></li></li></li></li>	auto	Позиционированные элементы	Нет	Относительно ширины блока-контейнера	Визуальный
table-layout'	auto   fixed   inherit	auto	Элементы 'table'и 'inline-table'	Нет		Визуальный
'text-align'	left   right   center   justify   inherit	'left' ecnu 'direction' paBHO 'ltr'; 'right' ecnu 'direction' paBHO 'rtl'	Элементы блочного уровня, табличные ячейки и встроенные блоки	Да		Визуальный
'text-decoration'	<pre>none   [underline    overline    line-through    blink]   inherit</pre>	none		Нет (см. текст)		Визуальный
'text-indent'	<pre><length>   <percentage>   inherit</percentage></length></pre>	0	Элементы блочного уровня, табличные ячейки и встроенные блоки	Да	Относительно ширины блока-контейнера	Визуальный

## **Таблица П2.1** (окончание)

Название	Значения	Начальная величина	Применяется к (по умолчанию: аll)	Насле- дуется?	Проценты (по умолчанию: N/A)	Типы вывода
text-transform'	capitalize   uppercase   lowercase   none   inherit	none		Да		Визуальный
'top'	<le><li><li><li><li><li><li><li><li><li><li< td=""><td>auto</td><td>Позиционированные элементы</td><td>Нет</td><td>Относительно высоты блока-контейнера</td><td>Визуальный</td></li<></li></li></li></li></li></li></li></li></li></le>	auto	Позиционированные элементы	Нет	Относительно высоты блока-контейнера	Визуальный
'unicode-bidi'	normal   embed   bidi- override   inherit	normal	Все элементы, но см. текст	Нет		Визуальный
'vertical-align'	<pre>baseline   sub   super   top   text-top   middle   bottom   text-bottom   <percentage>   <length>   inherit</length></percentage></pre>	baseline	Строчные элементы и элементы 'table- cell'	Нет	Относительно величи- Ны 'line-height' само- го элемента	Визуальный
'visibility'	visible   hidden   collapse   inherit	visible		Да		Визуальный
'white-space'	normal   pre   nowrap   pre-wrap   pre-line   inherit	normal		Да		Визуальный
'widows'	<integer>   inherit</integer>	5	Элементы блочного уровня	Да		Визуальный, страничный
'width'	<pre><length>   <percentage>   auto   inherit</percentage></length></pre>	auto	Все элементы, кроме неперемещаемых строчных элементов, табличных строк и групп строк	Нет	Относительно ширины блока-контейнера	Визуальный
'word-spacing'	normal   <length>   inherit</length>	normal		Да		Визуальный
'z-index'	auto   <integer>   inherit</integer>	auto	Позиционированные элементы	Нет		Визуальный

#### Свойства CSS 2.1 и специализированные дополнения

Название	Значения	Начальная величина	Применяется к (по умолчанию: а11)	Насле- дуется?	Проценты (по умол- чанию: N/A)	Тип вывода
'filter'	CM. http://tr.im/filterproperty	N/A (не определено)		Нет		Свойства фильтра
'overflow-x'	visible   scroll   hidden   auto	visible (за исключени- ем элемента textarea, в этом случае началь- ное значение становит- ся hidden)		Нет		Визуальный
'overflow-y'	visible   scroll   hidden   auto	visible (за исключени- ем элемента textarea, в этом случае началь- ное значение становит- ся auto)				Визуальный
'scrollbar-3dlight-color'	<color></color>	Цвет по умолчанию	Элемент с полосой прокрутки	Да		Визуальный
'scrollbar-arrow-color'	<color></color>	Цвет по умолчанию	Элемент с полосой прокрутки	Да		Визуальный
'scrollbar-base-color'	<color></color>	Цвет по умолчанию	Элемент с полосой прокрутки	Да		Визуальный
'scrollbar-darkshadow- color'	<color></color>	Цвет по умолчанию	Элемент с полосой прокрутки	Да		Визуальный
'scrollbar-face-color'	<color></color>	Цвет по умолчанию	Элемент с полосой прокрутки	Да		Визуальный
'scrollbar-highlight- color'	<color></color>	Цвет по умолчанию	Элемент с полосой прокрутки	Да		Визуальный
'scrollbar-shadow-color'	<color></color>	Цвет по умолчанию	Элемент с полосой прокрутки	Да		Визуальный
'zoom'	normal   <number>   <percentage></percentage></number>	normal		Нет	Да	Визуальный
'background-position-x'	<li><li><li><li><li><li><li><li><li><li></li></li></li></li></li></li></li></li></li></li>	0%		Нет	Да	Визуальный

Таблица П2.2. Дополнения к CSS, разработанные Microsoft

				F	аблица П2.2	(окончание)
Название	Значения	Начальная величина	Применяется к (по умолчанию: а11)	Насле- дуется?	Проценты (по умол- чанию: N/A)	Тип вывода
'background-position-y'	<pre><length>   <percentage>   top   center   bottom</percentage></length></pre>	%0		Нет	Да	Визуальный
'ime-mode'	auto   active   inactive   disabled	auto		Да		Визуальный
'layout-grid'	mode  type   line   char	both loose none none		Да		Визуальный
'layout-grid-char'	<pre><length>  <percentage>   none   auto</percentage></length></pre>	none		Нет	Да	Визуальный
'layout-grid-line'	<li><li><li><li><li><li><li><li><li><li></li></li></li></li></li></li></li></li></li></li>	none		Нет	Да	Визуальный
'layout-grid-mode'	both   none   line   char	both		Да		Визуальный
'layout-grid-type'	loose   strict   fixed	loose		Да		Визуальный
'line-break'	normal   strict	normal		Да		Визуальный
'text-autospace'	none   ideograph-alpha   ideograph-numeric   ideograph- parenthesis   ideograph-space	none		Нет		Визуальный
'text-justify'	auto   distribute   distribute- all-lines   distribute-center- last   inter-cluster   inter- ideograph   interword   kashida   newspaper	auto		Да		Визуальный
'text-kashida-space'	<pre><pre>centage&gt;   inherit</pre></pre>	%0		Да		Визуальный
'text-underline-position'	above   below   auto   auto-pos	auto		Да		Визуальный
'word-break'	normal   break-all   keep-all	normal		Да		Визуальный
'word-wrap'	normal   break-word	normal		Да		Визуальный
'writing-mode'	lr-tb   tb-rl	lr-tb		Нет		Визуальный

#### Свойства CSS 2.1 и специализированные дополнения

657

Табл	<b>пица П2.3</b> . Дополі	нения к свой	ствам CSS для программных продуктс	e Mozilla
ви		Начальная величина	Применяется к (по умолчанию: <sub>а1</sub> 1)	Наследу- ется?

Название	Значения	Начальная величина	Применяется к (по умолчанию: аll)	Наследу- ется?
-moz-appearance'	<pre>none   button   button-small   checkbox   checkbox-container   checkbox-small   dialog   listbox   menu   menuitem   menulist   menulist-button   menulist- textfield   progressbar   radio   radio-container   radio-small   resizer   scrollbar   scrollbar-button-down   scrollbar-button-left   scrollbarthuton- right   scrollbarbutton-up   scrollbartrack- norizontal   scrollbartutton-up   scrollbartrack- dge   tabpanels   textfield   toolbar   treeheadercell   treeheadersortarrow   treeitem   treetwisty   treetwistyopen   treeview   window</pre>	onon		Нет
'-moz-binding'	uri   none	none		Нет
'-moz-background-clip'	border   padding	border		Нет
'-moz-background-inline-policy'	bounding-box   continuous   each-box	continuous	Строчные элементы	Нет
-moz-background-origin'	border   padding   content	padding		Нет
'-moz-background-image'	<pre>[<uri>   none]   [number   <percentage>] [stretch   repeat   round] none   <image/> [<number>  <percentage>] [<humber>  <percentage>] [<bodder-width>] [stretch   repeat   round]</bodder-width></percentage></humber></percentage></number></percentage></uri></pre>	none		Нет
'-moz-border-bottam-colors' '-moz-border-left-colors' '-moz-border-right-colors' '-moz-border-top-colors'	<color>   transparent</color>	Не опреде- лено		Нет
'-moz-border-radius'	<leafletter<li><le><le></le></le></leafletter<li>	0		Нет

Наследу- ется?	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Нет	Да
Применяется к (по умолчанию: аіі)		Элементы со значением CSS display в дополнениях -moz-box или -moz-inline-box	Элементы со значением CSS display в дополнениях -moz-box или -moz-inline-box	Элементы со значением CSS display в дополнениях -moz-box или -moz-inline-box	Элементы со значением CSS display в дополнениях -moz-box или -moz-inline-box	Элементы со значением CSS display в дополнениях -moz-box или -moz-inline-box			Элементы image языка XUL <sup>1</sup> и псевдоэле- менты :-moz-tree-image, :-moz-tree- twisty и :-moz-tree-checkbox
Начальная величина	0	stretch	normal	0	horizontal	start	none	content-box	auto
Значения	<li><li><percentage></percentage></li></li>	start   center   end   baseline   stretch	normal   reverse	0~ - 0	horizontal   vertical	start   center   end   justify	<pre>none   [<color> <length> <length> <length> <length>   <length> <length> <length> <length> ? <length> <length> <length>? <color>]   inherit</color></length></length></length></length></length></length></length></length></length></length></length></color></pre>	content-box   border-box   padding box	для значений rect(), rect состоит из четырех вычисленных длин
Название	<pre>'-moz-horder-radius-bottomleft' '-moz-border-radius-bottomright' '-moz-border-radius-topleft' '-moz-border-radius-topright'</pre>	'-moz-box-align'	'-moz-box-direction'	'-moz-box-flex'	'-moz-box-orient'	'-moz-box-pack'	'-moz-box-shadow'	'-moz-box-sizing'	'-moz-image-region'

#### Свойства CSS 2.1 и специализированные дополнения

**Таблица П2.3** (окончание)

Название	Значения	пачальная величина	применяется к (по умолчанию: аll)	паследу- ется?
'-moz-opacity'	0 (хлли меньше)   0 < number < 1   1 (клли больше)	1		Нет
'-moz-outline'	-moz-outline-color  -moz-outline-style   -moz-outline-width	См. отдель- ные свойства		Нет
'-moz-outline-color'	<color>   invert</color>	invert		Нет
-moz-outline-offsef	<number></number>	0		Нет
'-moz-outline-radius'	<pre><length>   <percentage></percentage></length></pre>	0		Нет
-moz-outline-radius-bottomleft	<leafth>   <percentage></percentage></leafth>	0		Нет
-moz-outline-radius-bottomright				
-moz-outline-radius-topleft				
'-moz-outline-radius-topright'				
'-moz-outline-style'	none   dotted   dashed   solid   double   groove ridge   inset   outset	none		Нет
'-moz-outline-width'	<width></width>	medium		Нет
'-moz-user-focus'	Ignore   normal			
'-moz-outline-input'	none   enabled   disabled   inherit	none		Да
'-moz-outline-select'	none   text   -moz-none			Нет
'-moz-window-shadow'	default   none	default	Элементы, формирующие собственные окна, например, такие как window, panel	Нет

## Селекторы, псевдоклассы и псевдоэлементы CSS 2.1

В этом приложении содержатся три справочные таблицы, в которых показаны способы применения стилей к соответствующим элементам.

В табл. ПЗ.1 включены селекторы спецификации CSS 2.1. Селекторы помогают сообщать Web-обозревателю о том, к каким фрагментам страницы применять объявления CSS.

#### Примечание

В столбце "Общий шаблон" табл. ПЗ.1 обозначения с, к и s стоят на месте селекторов типа.

Таблица П3.2 содержит список псевдоклассов. Псевдокласс — это скрытый класс, с помощью которого Web-обозреватель ссылается на свой собственный класс. Например, с помощью псевдоклассов можно определить свойства разных состояний: visited, active и hover, повсюду встречающихся ссылок.

В табл. ПЗ.3 приведен перечень псевдоэлементов. Схожий по природе с псевдоклассом псевдоэлемент заключает содержимое Web-страницы в скрытые теги и затем к этим элементам применяет стили. Поскольку структура таких элементов больше похожа на типовой элемент, а не класс, их называют псевдоэлементами.

Селектор	Общий шаблон	Описание	Пример
Universal (универ- сальный)	*	Сочетается с любым эле- ментом	* { text-decoration: none; }
Туре (типа)	С	Сочетается с любым эле- ментом; в приведенном примере со всеми эле- ментами h2	h2 { font-weight: normal; }
Descendant (потомка)	CRS	Сочетается с любым эле- ментом s, являющимся потомком элемента к, потомка элемента с	<pre>div#content p em { background- color: yellow; }</pre>
Child (дочернего элемента)	c > s	Выбирается любой эле- мент s, дочерний по от- ношению к элементу с	<pre>li &gt; ul { list-style-type: circle;}</pre>
Adjacent Sibling (смежного эле- мента)	c + s	Выбирается любой эле- мент s, следующий сразу же за элементом с	<pre>div#content+p { text-indent: 0;}</pre>

Таблица ПЗ.1. Селекторы CSS 2.1

#### Таблица ПЗ.1 (окончание)

Селектор	Общий шаблон	Описание	Пример
Grouping (группи- ровки)	C, R, S	Несколько селекторов используют одно и то же объявление (объявления)	hi, h2, h3, h4 { color: #Ocf;}
Class (класса)	C.classR	Выбирается любой эле- мент с, содержащий атри- бут class со значением classR	<pre>img.content { padding: 4px; border: lpx solid black; }</pre>
ID (идентифика- тора)	C#idR	Выбирается любой эле- мент с, содержащий атри- бут id со значением idR	<pre>div#content { color: #333;}</pre>
Attribute Selector (атрибута)	C[attribute]	Выбирается любой эле- мент с, содержащий attribute	<pre>a[link] {text-decoration: none;}</pre>
Attribute Selector	C[attribute=" valueR"]	Выбирается любой эле- мент с, содержащий attribute СО Значением valueR	<pre>input[type="text"] { width: 33%; }</pre>
Attribute Selector	C[attribute~= "valueR"]	Выбирается любой эле- мент с, содержащий attribute, значение кото- рого представляет собой список отделенных друг от друга пробелом слов, одно из них соответствует значению valueR	<pre>div.advertisement form[class~="login"] { float: left; margin-left: 7px; }</pre>
Attribute Selector	C[attribute  ="valueR"]	Выбирается любой эле- мент с, содержащий attribute, значение кото- рого представляет собой список отделенных друг от друга дефисом слов, пер- вое из которых соответст- вует значению valueR	<pre>warning[lang="uk"]:after {   content: "Blimey!"}</pre>

#### Таблица ПЗ.2. Псевдоклассы CSS 2.1

Псевдокласс	Общий шаблон	Описание	Пример
:first-child	C:first-child	Соответствует элементу с, первому дочернему эле- менту другого элемента	<pre>divs p:first-chlld {color: white; background-color: red; }</pre>
:link	C:link	Соответствует любой не- посещенной ссылке эле- мента с	<pre>a:link {text-decoration: none; }</pre>
:visited	C:visited	Соответствует любой посещенной ссылке элемента с	a:visited {font-weight: normal; }

#### Таблица ПЗ.2 (окончание)

Псевдокласс	Общий шаблон	Описание	Пример
:hover	C:hover	Соответствует элементу с, выбранному пользовате- лем (обычно перемещени- ем пиктограммы курсора над ссылкой), но еще не активизированному	a:hover { background-color: orange; }
:active	C:active	Соответствует элементу с, активизированному поль- зователем	a:active { color: green; }
:focus	C:focus	Соответствует элементу с, обладающему фокусом (обычно поле ввода формы)	<pre>input:focus {background-color: #F7F7D5;}</pre>
:lang	C:lang(R)	Соответствует элементу с, использующему язык R	<pre>p:lang(en) {font-weight: bold;}</pre>

#### Таблица ПЗ.З. Псевдоэлементы CSS 2.1

Псевдоэлемент	Общий шаблон	Описание	Пример
:first-line	C:first-line	В элементе с выбирается первая строка текста	h2+c:fiist-line {color: #727977;}
:first-letter	C:first-letter	В элементе с выбирается первая буква	<pre>hl:first-letter { font-size: 66%; text-transform: lowercase; }</pre>
:before	C:before	Помещает генерируемое содержимое перед эле- ментом, применяется со свойством content	<pre>ul.tracklisting li: before { content: "Song title: ";}</pre>
:after	C:after	Помещает генерируемое содержимое после элемен- та, применяется со свойст- вом content	<pre>div#footer p. copyright:after {content: "Double true!";}</pre>

#### приложение 4 Селекторы и псевдоклассы CSS3

Несмотря на то, что к моменту написания книги работа над спецификацией CSS3 все еще продолжается, некоторые разработчики Web-обозревателей начинают поддерживать свойства из незавершенной спецификации. В этом приложении для справки приведен перечень новых селекторов CSS3.

Для того чтобы точно знать, какие средства задания стилей в документе есть в вашем распоряжении, смотрите *приложение 3* (содержащее селекторы CSS 2.1) вместе с данным перечнем.

В табл. П4.1 описан новый селектор CSS3, обобщенный селектор смежных элементов.

#### Примечание

В столбце "Общий шаблон" табл. П4.1 обозначения с, к и s стоят на месте селекторов типа.

Селектор	Общий шаблон	Описание	Пример
Обобщенный селектор смежных элементов	C~R	Соответствует любому элементу в, которому предшествует элемент с	<pre>#content ~ img {padding: 2px; border 2px solid black;}</pre>

Таблица П4.1. Новый селектор CSS3

Таблица П4.2 содержит список новых селекторов атрибута. Эти селекторы расширяют возможности выбора элементов в документе на основе значения атрибута и даже небольшого фрагмента этого значения.

Таблица	П4.2.	Селекторы	і атриб	vma CSS3

Селектор	Общий шаблон	Описание	Пример
Attribute Selector	C[attribute^= "valueR"]	Выбирается любой элемент C, содержащий attribute, значение которого начина- ется с valueR	<pre>Img[alt^="mugshot"] {width: 100px;}</pre>
Attribute Selector	C[attribute\$= "valueR"]	Выбирается любой элемент C, содержащий attribute, значение которого точно совпадает с valueR	<pre>Img[alt\$="photo"] {border: 10px solid red;}</pre>

#### Таблица П4.2 (окончание)

Селектор	Общий шаблон	Описание	Пример
Attribute Selector	C[attribute*= "valueR"]	Выбирается любой элемент с, содержащий attribute, в значении которого есть подстрока valueR	<pre>Img[alt*="executive"] { }</pre>
Attribute Selector	C[attribute = "valueR"]	Выбирается любой элемент C, содержащий attribute, значение которого содер- жит valueR в списке слов, разделенных дефисом	<pre>Img[alt ="non"] {opacity: .5;}</pre>

В табл. П4.3 представлен список структурных псевдоклассов. Они позволяют выбирать элементы, исходя из порядка их следования. Например, можно указать элементы списка 11 с нечетными номерами, используя селектор nth-child вместо селектора атрибута class.

Псевдокласс	Общий шаблон	Описание	Пример
:last-child	C:last-child	Соответствует элементу с, последнему дочернему эле- менту другого элемента	<pre>divs p:last-chlld {color: white; background-color: black; }</pre>
:target	C:target	Соответствует элементу с, при активации ссылки на фрагмент (например, #section)	<pre>#section:target {background-color:yellow;}</pre>
:enabled	C:enabled	Соответствует элементу с, если он в состоянии enabled	<pre>input[type="age"]:enabled {background-color: green;}</pre>
:disabled	C:disabled	Соответствует элементу с, если он в состоянии disabled	<pre>input[type="password"]:disabled {background-color: #999;}</pre>
:root	:root	Соответствует корневому элементу документа; в документах HTML4 это элемент нтмL	<pre>:root {display: block;}</pre>
:nth-child()	C:nth- child(an+b)	Соответствует элементам в дереве документа, кото- рым предшествует опреде- ленное число родственных элементов одного с ними уровня; если n целое, :nth-child (an+b) будет	<pre>tr:nth-child(2n) {background-color: #99ff99;}</pre>

соответствовать элементу, перед которым есть an+b-1 родственных элементов одного с ним уровня

#### Таблица П4.3. Структурные псевдоклассы CSS3

#### Таблица П4.3 (окончание)

Псевдокласс	Общий шаблон	Описание	Пример
:nth-last- child()	C:nth-last- child(an+b)	Соответствует элементам в дереве документа, после которых есть определенное число родственных элемен- тов одного с ними уровня; если п целое, :nth-last- child(an+b) будет соответ- ствовать элементу, после которого есть an+b-1 родст- венных элементов одного с ним уровня	<pre>tr:nth-last-child(-2n) {background-color: #99ff99;}</pre>
:nth-of-type	C:nth- of-type(an+b)	Соответствует элементам в дереве документа, которым предшествует определенное число родственных элемен- тов одного с ними типа; если n целое, inth-of-type (an+b) будет соответствовать эле- менту, перед которым есть an+b-1 родственных элемен- тов одного с ним типа	<pre>tr:nth-of-type(2n) {float:right;}</pre>
:nth-last- of-type	C:nth-last- of-type(an+b)	Соответствует элементам в дереве документа, после которых есть определенное число родственных элемен- тов одного с ними типа; если п целое, :nth-last-of- type (an+b) будет соответст- вовать элементу, после которого есть an+b-1 родст- венных элементов одного с ним типа	<pre>tr:nth-last-of-type(2n) {float:right;}</pre>
:first- of-type	C:first- of-type	Соответствует первому до- чернему элементу элемента с заданным типом	<pre>p:first-of-type {font-weight: bold;}</pre>
:last-of-type	C:last- of-type	Соответствует последнему дочернему элементу эле- мента с заданным типом	<pre>p:last-of-type {background-color: black;}</pre>
:only-child	C:only-child	Соответствует дочернему элементу, если он единст- венный потомок своего родительского элемента	<pre>li:only-child {font-size: 2em;}</pre>
:only-of-type	C:only- of-type	Соответствует дочернему элементу, если он единст- венный потомок своего родительского элемента	<pre>li:only-of-type {font-size: 2em;}</pre>
:empty	C:empty	Соответствует любому элементу без потомков	*:empty {background: red; height: 100px;}
:not()	C:not(R)	Соответствует всем элемен- там в элементе с за исклю- чением элементов R	<pre>*:not(:hover) {opacity: .7;}</pre>

### Стилевое оформление элементов форм

Формы оказывают большое влияние на наш повседневный образ жизни, неразрывно связанный с Интернетом и электронной коммерцией, поэтому они занимают видное место в Web-дизайне.

Web-разработчики хотят управлять внешним видом элементов форм в их проекте Webстраницы для того, чтобы они были более привлекательны для посетителей и при этом соответствовали остальным компонентам дизайна.

Проблема заключается в том, что разные Web-обозреватели по-разному отображают элементы формы. Даже в одной версии Web-обозревателя элементы формы могут выглядеть по-разному в различных операционных системах.

В этом приложении описывается, как несогласованно Web-обозреватели отображают элементы управления форм. Поскольку дается обзор 10 Web-обозревателей и 20 свойств CSS вместе с 8 HTML-элементами форм, полное приложение слишком объемно для печатной версии книги. Мы открыли доступ к нему в Интернете.

Полную версию *приложения 5* (Appendix E) на английском языке вы можете найти по адресу http://oreilly.com/catalog/9780596155933/ (выберите ссылку Examples).

#### Предметный указатель

#### Α

Adobe BrowserLab 562 Adobe Dreamweaver 543 Ajax 297

#### В

Boot Camp 561 BrowserCam 562 BrowserShots 562

#### С

CSS 25, 54

#### D

Debian Linux 561 DOCTYPE 31 Document Object Model, DOM 468

#### Ε

Escape-последовательность 491

#### F

FIR-метод 229, 230, 385

#### G

GIF-файл 263

#### Η

HyperText Markup Language 25

#### 

Internet Explorer для Windows 556, 557

#### Κ

Knoppix 561

#### L

LIR-метод 385 Luminosity Contrast Ratio Analyser 572 Lynx 563 Lynx Viewer 562

#### Ν

Nifty Corners Cube 248

#### Ρ

Parallels Desktop 561 Phark-метод 231 Pixy-метод 386 PNG-файл 226 Prototype Framework 297

#### Q

Quirks mode 31

#### S

Sans serif 54 Scriptaculous 297

#### U

URI 37, 489, 500

#### V

VMWare Workstation 561

#### W

Web-безопасные шрифты 125 Wine 561

#### Y

YUI Reset CSS 277

#### Α

Абсолютное позиционирование 519, 539, 568 Автоматически генерируемое содержимое 313 Альтернативная таблица стилей 106 Альфа-прозрачность 38, 224 Асимметричный макет 543

- Атрибут
- ◊ alt 38
- ◊ autosave 413
- ♦ cite 37
- ♦ class 49
- ♦ controls 40
- ◊ id 47
- ♦ placeholder 412
- ♦ poster 42
- ♦ results 413
- ♦ title 38, 392

#### Б

Базовая линия 463 Базовая линия шрифта 150 Библиотека jQuery 622 Блок-контейнер 521 Буквица 146 Букмарклет 53

#### В

Валидатор 31, 641 Внешняя таблица стилей 89 Внутренняя таблица стилей 89 Встроенные стили 89 Выноска 260 Выравнивание по ширине 143 Выступающий инициал 146

#### Г

Генерируемое содержимое 489

#### 3

Заголовки разных уровней 35 Значение 57

#### И

Идентификатор фрагмента 395 Изображение-карта 395

#### К

Каскадные таблицы стилей 25, 54 Кернинг 185 Клавиши доступа 377, 429 Класс vevent 51 Колонки фиксированной ширины 509, 515, 521 Контекстный селектор 459 Контент 25, 54 Курсор 354

#### Μ

Маркер 307, 313, 315, 318, 327 Маркированный список 366 Межстрочный интервал 499 Метод ◊ Гилдера — Левина 229 ◊ горной вершины 245 ◊ замещения изображением 567 ◊ раздвижных дверей 241 Микроформат ◊ hCalendar 51 ◊ hCard 49

Модуль mod\_rewrite 560

#### Н

Назначение устройств вывода 481

#### 0

Обобщенное название семейства шрифтов 122 Обратный отступ 169, 326 Обтекание текста 506 Объявление !important 93, 549 Относительные единицы 136 Отступ 82

#### Π

Панель инструментов Image 264 Параллакс 578 Плавающая модель 514, 517, 526 Поле 82 ◊ поиска 411, 413 Порядок отображения колонок 507, 534 Потерянная ссылка 552 Правило (*a*) import 481 (a) media 96, 482
 V/HA 350 Приподнятая буквица 146 Псевдокласс 72, 350 ♦ visited 347 ◊ :focus 281, 351, 405, 408 hover 193, 357, 376, 420, 435, 469 Iast-of-type 340 Ilink 347 ◊ :not 398 :target 396 Псевдоколонки 538 Псевдоэлемент 73, 489, 490 ♦ ::moz-selection 176 ♦ ::selection 175 333, 437, 489 ◊ :before 314, 381, 575 ♦ :first-letter 146

◊ :first-line 171

#### Ρ

Разделитель 311 Рамка 82, 194 Расширения языка HTML 411 Режим обратной совместимости 31

#### С

Свойство 57 ◊ animation-duration 581 ♦ animation-iteration-count 582 ♦ animation-name 581 ♦ background 204, 319 ♦ background-attachment 209, 583 ♦ background-color 175, 303 ♦ background-image 155, 173, 199, 263, 383 ♦ background-position 156, 201, 267, 342, 386 ♦ background-repeat 156, 199, 200, 213 ♦ background-size 212, 214 ◊ border 193, 289, 311, 454 ♦ border-bottom 154 ♦ border-collapse 456 ♦ border-image 207 ♦ border-radius 195 ♦ border-spacing 456 ♦ border-style 456 ◊ border-top 154 ♦ bottom 113 ◊ box-shadow 252 ◊ clear 109, 180, 476  $\diamond$  color 403 ◊ content 231, 576 ◊ counter-increment 334  $\diamond$  counter-reset 333  $\diamond$  cursor 354 ◊ display 136, 295, 364, 365, 381 ♦ empty-cells 457 ♦ filter 183, 192, 303 ♦ font 152 ◊ font-family 122, 131 ◊ font-size 135, 139, 566 ♦ font-style 175 ◊ font-weight 175 ◊ image-rendering 217 ♦ left 114 ♦ letter-spacing 183 ♦ line-height 177, 499 ♦ list-style 295, 318 ♦ list-style-image 316

♦ list-style-position 328 ♦ list-style-type 307, 317 ♦ margin 504 ◊ max-width 216 moz-image-rect 269 ◊ –ms-interpolation-mode 217 ◊ opacity 300, 581 ♦ outline 351 ◊ overflow 112 opage-break-before 492 ◊ position 112, 114, 116, 287, 519 ♦ text-align 142, 285 text-decoration 347 ♦ text-indent 147, 165, 314, 326 ♦ text-overflow 144 text-shadow 181 ♦ top 114  $\diamond$  transition 359, 360 ♦ transition-delay 360 transition-duration 359 transition-property 360 vertical-align 464 visibility 136 width 320
 ♦ word-spacing 183 ♦ word-wrap 134 ♦ z-index 118 ◊ zoom 112, 304 ◊ с сокращенной формой записи 104, 152 Селектор 57 Inth-child() 168, 468 Inth-child(odd) 411 ID 64, 77, 352, 364, 473 ◊ атрибута 70, 404, 487 ◊ класса 61, 77, 402, 407, 439, 459, 462, 467 Подстроки 490 ◊ потомка 65, 368, 430, 459 прямого потомка 67 ◊ соседнего элемента 68, 167 ◊ типа 60, 408, 410 Синтаксическая запись CSS 57 Скругление углов 239 Спецификация CSS3 231, 398 Специфичность правила 94 Списки 44 Список определений 44 Средник 532

Ссылка 350, 352, 353 Ссылки-цепочки 379

#### Т

Тег 25 Тень 253 Типы носителей информации 481 Трекинг 185

#### У

Универсальный селектор 66 Условные комментарии 556

#### Φ

Фильтры CSS 559 Фоновое изображение 199, 203, 320 Функция ◊ attr(x) 489 ◊ counters() 333 ◊ -moz-linear-gradient() 222 ◊ -moz-radial-gradient() 222 ◊ transition-timing-function 359

- ♦ -webkit-gradient() 220
- ♦ url() 208

#### Χ

Хостинг Google 622

#### Ш

Шестнадцатеричный еscape-эквивалент Unicode 315 Шрифт ◊ без засечек 495 ◊ с засечками 495

#### Э

Экранный считыватель 385 Элемент ◊ <style> 88 ◊ anchor 45 ◊ audio 39 ◊ blockquote 36, 157 ◊ body 28 Элемент (прод.)

- ♦ caption 48
- ♦ dialog 336
- ◊ em 42, 175
- ♦ head 28, 87
- ♦ heading 29
- ◊ html 117
- ♦ HTML 28
- ♦ hr 293
- ♦ iframe 215
- ◊ img 38
- ◊ input 402
- ♦ form 402
- $\diamond$  legend 432
- ◊ li 43
- ◊ noscript 618
- ♦ ol 44
- $\diamond$  option 410
- ◊ paragraph 29
- ◊ q 36
- $\diamond$  select 410
- $\diamond$  source 40

- ◊ strong 42, 174
- $\diamond$  style 58
- ◊ sub 189
- ♦ sup 189
- $\diamond$  table 48
- $\diamond$  tbody 48
- ♦ td 49
- $\diamond~$  th 49, 462
- $\diamond$  thead 48
- ♦ title 29
- ♦ tr 48
- ♦ ul 43
- ♦ video 40
- ♦ fieldset 431
- ◊ блочного уровня 123, 511
- 👌 страницы 274
- Эффект обрезанной в край страницы 275

#### Я

Язык разметки гипертекста 25 Якорь 47