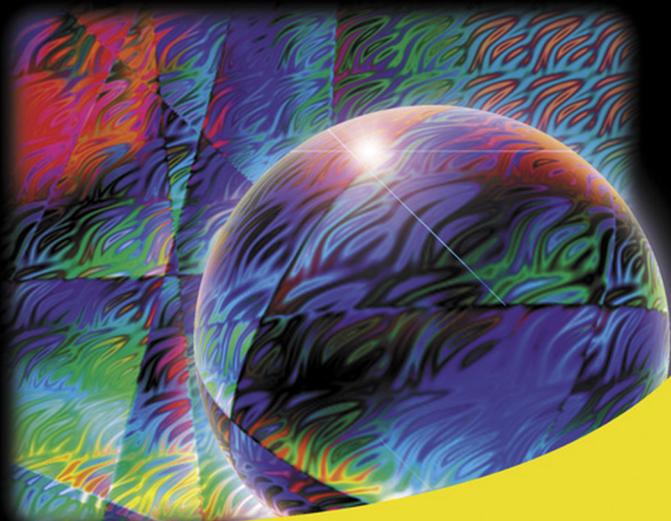




Самоучитель

Владимир Дронов

Adobe Dreamweaver CS5.5



Самые современные интернет-технологии — HTML 5 и CSS 3

Контейнерный Web-дизайн

Эффекты и компоненты Spry

Мини-приложения

Наборы данных XML

Аудио и видео в формате HTML 5

Кроссплатформенные Web-страницы. Медиазапросы

Семантическая разметка

Многоколоночная верстка и гибкая разметка CSS 3

Владимир Дронов

Самоучитель
Adobe
Dreamweaver CS5.5

Санкт-Петербург

«БХВ-Петербург»

2012

УДК 681.3.06
ББК 32.973.26-018.2
Д75

Дронов В. А.

Д75 Самоучитель Adobe Dreamweaver CS5.5. — СПб.: БХВ-Петербург, 2012. — 384 с.: ил.

ISBN 978-5-9775-0771-4

Книга посвящена созданию Web-страниц и Web-сайтов в новой версии визуального Web-редактора Adobe Dreamweaver CS5.5 с применением HTML 5 и CSS 3. Рассказывается об основных принципах Web-дизайна, об оформлении Web-страниц с применением стилей CSS, о верстке на основе принципов контейнерного Web-дизайна, об использовании эффектов и компонентов Spry и мини-приложений Dreamweaver, о наборах данных XML. Особое внимание уделяется применению самых современных на данный момент технологий Web-дизайна, в число которых входят помещение на Web-страницы аудио- и видеороликов, создание кроссплатформенных Web-страниц и медиазапросов, семантическая разметка и новые возможности CSS 3 по форматированию — многоколоночная верстка и гибкая разметка. Также рассказывается о публикации готовых Web-сайтов в Интернете.

Для Web-дизайнеров

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

| | |
|-------------------------|-----------------------------|
| Главный редактор | <i>Екатерина Кондукова</i> |
| Зам. главного редактора | <i>Евгений Рыбаков</i> |
| Зав. редакцией | <i>Григорий Добин</i> |
| Редактор | <i>Екатерина Капалыгина</i> |
| Компьютерная верстка | <i>Ольги Сергиенко</i> |
| Корректор | <i>Зинаида Дмитриева</i> |
| Дизайн серии | <i>Инны Тачиной</i> |
| Оформление обложки | <i>Марины Дамбиевой</i> |
| Зав. производством | <i>Николай Тверских</i> |

Подписано в печать 31.10.11.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 30,96.

Тираж 1500 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

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 978-5-9775-0771-4

© Дронов В. А., 2011
© Оформление, издательство "БХВ-Петербург", 2011

Оглавление

| | |
|---|-----------|
| Введение..... | 11 |
| Dreamweaver CS5.5 — что нового? | 11 |
| Чем мы будем заниматься | 12 |
| Типографские соглашения | 13 |
| Благодарности | 14 |
| | |
| ЧАСТЬ I. ВВЕДЕНИЕ В СОВРЕМЕННЫЕ ИНТЕРНЕТ-ТЕХНОЛОГИИ..... | 15 |
| | |
| Глава 1. Как все это работает | 16 |
| Как работает Интернет | 16 |
| Программы, обеспечивающие работу Интернета. Клиенты и серверы | 16 |
| Протоколы | 18 |
| Порты TCP..... | 20 |
| Интернет-адреса | 21 |
| IP-адреса и доменные имена | 21 |
| Формирование интернет-адресов | 24 |
| Web-страницы и Web-сайты | 26 |
| Что такое Web-страница и Web-сайт..... | 26 |
| Содержимое, представление и поведение Web-страницы..... | 27 |
| Web-сайты и Web-серверы: технические детали | 28 |
| Корневая папка Web-сервера | 28 |
| Web-страница по умолчанию..... | 29 |
| Публикация Web-сайта в Интернете. Хостинг-провайдеры..... | 30 |
| Как создаются Web-страницы и Web-сайты. Web-редакторы | 31 |
| Альтернативный вариант — CMS | 32 |
| Что дальше?..... | 34 |
| | |
| Глава 2. Принципы построения Web-сайтов | 36 |
| Основные этапы планирования Web-сайта..... | 36 |
| Структурирование Web-сайта..... | 37 |
| Логическая структура Web-сайта | 38 |
| Физическая структура Web-сайта..... | 41 |
| Что дальше?..... | 44 |

| | |
|---|-----------|
| ЧАСТЬ II. СОЗДАНИЕ ПРОСТЕЙШИХ WEB-СТРАНИЦ, ПУБЛИКАЦИЯ WEB-САЙТА..... | 45 |
| Глава 3. Введение в Adobe Dreamweaver CS5.5 | 46 |
| Рабочее окружение Adobe Dreamweaver CS5.5 | 46 |
| Главное окно программы | 46 |
| Окна документов | 48 |
| Панели | 50 |
| Тестирование Web-страниц | 55 |
| Интерактивный просмотр..... | 56 |
| Тестирование Web-страниц в Web-обозревателе..... | 57 |
| Вызов справки | 57 |
| Предварительная настройка Dreamweaver..... | 59 |
| Регистрация Web-сайта в Dreamweaver | 61 |
| Что дальше?..... | 63 |
| Глава 4. Форматирование текста | 64 |
| Создание новой Web-страницы | 64 |
| Название Web-страницы | 66 |
| Набор текста..... | 66 |
| Файловые операции..... | 68 |
| Форматирование текста..... | 70 |
| Панель <i>Свойства</i> | 70 |
| Форматирование фрагментов текста..... | 70 |
| Создание заголовков..... | 72 |
| Создание примечаний..... | 73 |
| Создание списков..... | 74 |
| Текст фиксированного формата | 75 |
| Вставка специальных символов..... | 76 |
| Разрыв строк..... | 76 |
| Неразрывный пробел | 78 |
| Обозначения | 78 |
| Горизонтальная линия | 80 |
| Комментарии..... | 80 |
| Что дальше?..... | 82 |
| Глава 5. Графические изображения и таблицы..... | 83 |
| Графические изображения | 83 |
| Введение в интернет-графику. Внедренные элементы | 83 |
| Форматы интернет-графики..... | 84 |
| Вставка графического изображения..... | 85 |
| Как изображения ведут себя на Web-странице. Блочные и встроенные элементы..... | 88 |
| Параметры графического изображения | 89 |
| Таблицы | 91 |
| Создание таблицы | 91 |
| Ввод данных в таблицу..... | 92 |
| Ячейки шапки..... | 94 |
| Объединение ячеек | 95 |
| Использование таблиц для разметки Web-страниц..... | 96 |
| Что дальше?..... | 98 |

| | |
|--|------------|
| Глава 6. Средства навигации | 99 |
| Введение в средства навигации | 99 |
| Создание гиперссылок в Dreamweaver..... | 100 |
| Создание текстовых гиперссылок | 100 |
| Создание графических гиперссылок | 102 |
| Цель гиперссылки и ее задание..... | 103 |
| Принципы формирования интернет-адресов, используемых в WWW..... | 104 |
| Создание почтовых гиперссылок | 107 |
| Якоря..... | 107 |
| Иерархическая система навигации..... | 109 |
| Что дальше?..... | 110 |
| | |
| Глава 7. Работа с HTML-кодом..... | 111 |
| Три режима отображения, поддерживаемые окнами документов..... | 111 |
| Введение в язык HTML | 113 |
| Теги | 113 |
| Атрибуты тегов | 116 |
| Литералы..... | 117 |
| Вложенность тегов..... | 118 |
| Секции заголовка и тела Web-страницы. Невидимые теги | 119 |
| Метаданные Web-страницы | 120 |
| Инструменты Dreamweaver для работы с HTML-кодом | 120 |
| Секция тегов | 120 |
| Средства, доступные в режиме отображения HTML-кода | 121 |
| Справочник по HTML | 124 |
| Что дальше?..... | 125 |
| | |
| Глава 8. Публикация Web-сайта | 126 |
| Как выполняется публикация Web-сайта..... | 126 |
| Задание параметров публикации Web-сайта | 128 |
| Задание места публикации Web-сайта | 128 |
| Задание в качестве места публикации FTP-сервера..... | 130 |
| Задание основных параметров FTP-сервера..... | 130 |
| Задание дополнительных параметров FTP-сервера | 131 |
| Задание параметров прокси-сервера (брандмауэра) | 131 |
| Задание в качестве места публикации локальной папки | 133 |
| Задание дополнительных параметров публикации | 134 |
| Управление местами публикации | 134 |
| Публикация Web-сайта..... | 135 |
| Выбор публикуемого Web-сайта | 135 |
| Соединение с FTP-сервером | 135 |
| Простая публикация Web-сайта и отдельных его файлов | 136 |
| Синхронизация копий Web-сайта..... | 138 |
| Разъединение с FTP-сервером | 141 |
| Получение файлов из места публикации | 141 |
| Управление файлами и папками с помощью панели <i>Файлы</i> | 142 |
| Расширенный режим панели <i>Файлы</i> | 144 |
| Что дальше?..... | 145 |

| | |
|---|------------|
| ЧАСТЬ III. ОФОРМЛЕНИЕ WEB-СТРАНИЦ | 147 |
| Глава 9. Введение в таблицы стилей CSS..... | 148 |
| Как создается представление Web-страниц..... | 148 |
| Что такое представление Web-страницы. Язык CSS | 148 |
| Стили, атрибуты стиля, селекторы и таблицы стилей | 149 |
| Внутренние и внешние таблицы стилей. Организация стилей в таблицы стилей..... | 151 |
| Работа со стилями и таблицами стилей в среде Dreamweaver | 152 |
| Создание стилей | 152 |
| Привязка стилей | 155 |
| Просмотр стилей, действующих на элемент Web-страницы, с помощью панели <i>Стили CSS</i> | 156 |
| Просмотр всех созданных стилей с помощью панели <i>Стили CSS</i> | 157 |
| Управление стилями | 157 |
| Управление таблицами стилей..... | 160 |
| Работа с CSS-кодом таблиц стилей | 161 |
| Введение в язык CSS | 161 |
| Средства HTML для привязки стилей и таблиц стилей..... | 162 |
| Средства Dreamweaver для доступа к CSS-коду таблиц стилей..... | 162 |
| Справочник по CSS..... | 164 |
| Что дальше?..... | 164 |
| Глава 10. Задание параметров текста..... | 165 |
| Параметры шрифта текста | 165 |
| Параметры блока текста..... | 169 |
| Параметры пунктов списков..... | 172 |
| Параметры, задаваемые только вручную в CSS-коде..... | 173 |
| Степень полупрозрачности элемента Web-страницы | 173 |
| Создание тени у текста | 174 |
| Практикум по созданию стилей..... | 174 |
| Что дальше?..... | 177 |
| Глава 11. Задание параметров фона, размеров, отступов, рамки и курсора..... | 178 |
| Параметры фона..... | 178 |
| Параметры размеров..... | 180 |
| Параметры отступов | 181 |
| Параметры рамки..... | 182 |
| Параметры курсора..... | 184 |
| Параметры, задаваемые только вручную в CSS-коде..... | 185 |
| Создание рамок со скругленными углами | 186 |
| Объединение рамок соседних ячеек таблицы..... | 187 |
| Практикум по созданию стилей..... | 188 |
| Что дальше?..... | 189 |
| Глава 12. Селекторы, каскадность и контейнеры | 190 |
| Селекторы CSS..... | 190 |
| Разновидности стилей CSS..... | 190 |

| | |
|--|-----|
| Специальные компоненты комбинированных стилей | 193 |
| Комбинаторы | 193 |
| Псевдоэлементы | 194 |
| Псевдоклассы | 194 |
| Псевдоселекторы по атрибутам тега | 196 |
| Применение к элементу Web-страницы сразу нескольких стилей | |
| Правила каскадности | 197 |
| Практикум по созданию стилей | 199 |
| Контейнеры | 201 |
| Встроенные контейнеры | 201 |
| Блочные контейнеры | 202 |
| Что дальше? | 204 |

ЧАСТЬ IV. РАЗМЕТКА WEB-СТРАНИЦ 205

Глава 13. Контейнерный Web-дизайн 206

| | |
|---|-----|
| Введение в контейнерный Web-дизайн | 206 |
| Недостатки текстового Web-дизайна | 206 |
| Разметка Web-страниц | 207 |
| Разметка посредством блочных контейнеров. Контейнерный Web-дизайн | 209 |
| Реализация контейнерного Web-дизайна | 210 |
| Управление местоположением контейнеров. Плавающие контейнеры | 210 |
| Указание ширины для плавающих контейнеров | 212 |
| Искажение дизайна Web-страницы и способ его предотвратить | 212 |
| Плоская система навигации | 213 |
| Реализация разметки, основанной на блочных контейнерах | 214 |
| Создание контейнеров, формирующих разметку | 214 |
| Задание местоположения и размеров контейнеров, формирующих разметку | 215 |
| Создание полосы навигации | 219 |
| Создание остальных Web-страниц Web-сайта | 223 |
| Что дальше? | 223 |

Глава 14. Шаблоны 224

| | |
|---|-----|
| Введение в шаблоны Dreamweaver | 224 |
| Реализация шаблонного Web-дизайна | 225 |
| Создание шаблона | 225 |
| Создание изменяемых областей | 227 |
| Создание Web-страниц на основе шаблонов | 230 |
| Правка шаблонов, на основе которых уже созданы Web-страницы | 231 |
| Дополнительные возможности шаблонов | 232 |
| Повторяющиеся области | 232 |
| Необязательные области | 234 |
| Необязательные изменяемые области | 237 |
| Табличная повторяющаяся область | 238 |
| Изменяемые атрибуты | 239 |
| Открепление Web-страниц от шаблонов | 242 |
| Что дальше? | 242 |

ЧАСТЬ V. СОЗДАНИЕ ИНТЕРАКТИВНЫХ WEB-СТРАНИЦ..... 243**Глава 15. Простейшие интерактивные элементы.****Поведения Dreamweaver 244**

Введение в создание интерактивных элементов 244

Web-сценарии 245

События 245

Имена элементов Web-страниц 247

Web-программирование — подход Dreamweaver. Поведения 247

Работа с поведением 249

Создание поведений 249

Некоторые поведения, поддерживаемые Dreamweaver, и их параметры 251

Переход к URL-адресу 251*Изменение значения у атрибута результирующего стиля* 253

Управление поведением 255

Интерактивный просмотр Web-страниц, дополнение 255

Теги HTML, предназначенные для поддержки Web-сценариев 256

Что дальше? 257

Глава 16. Эффекты и компоненты Spry 258

Понятие об эффектах и компонентах Spry 258

Подготовка к созданию эффектов и компонентов Spry 260

Работа с эффектами Spry 261

Создание эффектов Spry 261

Некоторые эффекты Spry, поддерживаемые Dreamweaver, и их параметры 262

Плавное появление или исчезновение элемента Web-страницы
(*Появление/Растворение*) 263Плавное увеличение или уменьшение размеров элемента Web-страницы
(*Расширить/Сжать*) 264

Управление эффектами Spry 266

Работа с компонентами Spry 266

Создание компонентов Spry 266

Свертываемая панель Spry (спойлер) 267*Набор вкладок Spry (гармошка)* 269

Управление компонентами Spry 272

Настройка внешнего вида компонентов Spry 272

Что дальше? 274

Глава 17. Наборы данных 275

Понятие о наборах данных 275

Создание наборов данных XML 277

Краткий курс языка XML 277

Формирование наборов данных 279

Создание набора данных XML в Dreamweaver 282

Использование наборов данных XML 282

Подключение набора данных 283

Вывод данных, хранящихся в наборах 287

Область Spry 287

| | |
|--|------------|
| <i>Динамический текст Spry</i> | 289 |
| <i>Повторяющаяся область Spry</i> | 291 |
| Оформление компонентов Spry с помощью стилей CSS..... | 293 |
| Что дальше?..... | 294 |
| Глава 18. Мини-приложения (виджеты) Dreamweaver | 295 |
| Что такое мини-приложение | 295 |
| Программа Adobe Widget Browser | 296 |
| Главное окно программы | 297 |
| Вход в магазин приложений | 298 |
| Получение подробных сведений о мини-приложениях..... | 299 |
| Установка мини-приложения | 303 |
| Управление установленными мини-приложениями | 304 |
| Выход из магазина приложений | 305 |
| Работа с мини-приложениями в Dreamweaver..... | 305 |
| Помещение мини-приложения на Web-страницу..... | 306 |
| Задание содержимого для мини-приложения..... | 307 |
| Настройка мини-приложений | 309 |
| Создание пользовательского пресета | 309 |
| Управление пользовательскими пресетами..... | 313 |
| Что дальше?..... | 314 |
| ЧАСТЬ VI. ИСПОЛЬЗУЕМ НОВЕЙШИЕ ТЕХНОЛОГИИ..... | 315 |
| Глава 19. Вывод аудио и видео на Web-страницах | 316 |
| Три способа поместить на Web-страницу аудио- или видеоролик..... | 316 |
| Перекодирование мультимедийных файлов с помощью утилиты SUPER | 318 |
| Запуск | 319 |
| Настройка | 320 |
| Использование..... | 322 |
| Завершение работы | 324 |
| Размещение аудио- и видеороликов на Web-странице средствами HTML 5 | 325 |
| Подготовка мультимедийных файлов | 325 |
| Размещение аудио и видео на Web-странице | 326 |
| Размещение аудиоролика на Web-странице | 327 |
| Размещение видеоролика на Web-странице | 328 |
| Альтернативные источники | 330 |
| Размещение на Web-странице видеороликов Flash Video..... | 331 |
| Подготовка видеофайлов..... | 331 |
| Размещение видеоролика на Web-странице | 331 |
| Размещение на Web-странице видеороликов, хранящихся на сервисе видеохостинга | 334 |
| Что дальше?..... | 335 |
| Глава 20. Разработка кроссплатформенных Web-страниц | 336 |
| Введение в кроссплатформенный Web-дизайн | 336 |
| Что такое кроссплатформенность..... | 336 |
| Реализация кроссплатформенного Web-дизайна. Медиазапросы | 337 |
| Работа с медиазапросами в Dreamweaver | 339 |
| Создание файла медиазапросов | 339 |

| | |
|--|------------|
| Просмотр Web-страницы при различных разрешениях экрана | 343 |
| Правка таблиц стилей, определенных в медиазапросах | 345 |
| Режим многоэкранного просмотра | 347 |
| Что дальше? | 349 |
| Глава 21. Семантическая разметка Web-страниц | 350 |
| Введение в семантическую разметку | 350 |
| Что такое семантическая разметка | 350 |
| Теги семантической разметки | 352 |
| Реализация семантической разметки | 356 |
| Что дальше? | 357 |
| Глава 22. Многоколоночная верстка и гибкая разметка | 358 |
| Многоколоночная верстка | 358 |
| В каких случаях многоколоночная верстка может пригодиться | 358 |
| Атрибуты стиля, предназначенные для реализации многоколоночной верстки | 359 |
| Основные параметры колонок | 359 |
| Дополнительные параметры колонок | 360 |
| Растягивание элемента Web-страницы на все колонки | 362 |
| Реализация многоколоночной верстки | 362 |
| Гибкая разметка | 363 |
| Понятие гибкой разметки | 363 |
| Атрибуты стиля, предназначенные для реализации гибкой разметки | 364 |
| Собственно реализация гибкой разметки | 364 |
| Указание направления гибкой разметки | 366 |
| Указание выравнивания для элементов | 366 |
| Реализация гибкой разметки | 368 |
| Заключение | 371 |
| Предметный указатель | 373 |

Введение

Популярность традиционного Web-дизайна в настоящее время несколько снизилась. Виною тому стали многочисленные CMS — программные инструменты, позволяющие создать Web-сайт, не то что не написав ни единой строчки кода, но и вообще не занимаясь Web-дизайном как таковым. В результате профессии специалистов по настройке и доработке CMS стали популярнее профессии Web-дизайнера, а Интернет оказался наводненным множеством шаблонных Web-сайтов, практически неотличимых друг от друга.

Значит ли это, что старые добрые времена, когда каждая Web-страница создается вручную, когда Web-дизайнеры стремятся к уникальности своих Web-произведений, а иные Web-сайты действительно являются произведениями искусства, ушли в прошлое?

Отнюдь! И это не только точка зрения автора данной книги. Корпорация Adobe тоже так не считает. Иначе чем объяснить выпуск весной этого года новой версии Dreamweaver, носящей обозначение CS5.5!

Dreamweaver CS5.5 — что нового?

Dreamweaver — настоящий "ветеран" среди программ Web-редакторов. Первая его версия была разработана фирмой Macromedia еще в 1998 году и сразу стала популярной благодаря удобному интерфейсу, богатым возможностям и "благосклонностью" даже к неопытным Web-дизайнерам. Более того, Dreamweaver стал своего рода эталоном среди Web-редакторов, и впоследствии все программы такого рода, так или иначе, сравнивались с ним.

С тех пор утекло много воды. Dreamweaver рос и развивался. В версии 4.0 появилось увесистое электронное руководство по языкам HTML, CSS и JavaScript — истинное благодеяние для начинающих Web-дизайнеров, желающих повысить свою квалификацию. Dreamweaver MX получил новый интерфейс и средства для разработки Web-серверных программ, ранее поставлявшиеся в виде отдельного продукта Macromedia UltraDev. А Dreamweaver MX 2004 щеголял многочисленными дополнительными инструментами, в частности, для мелкой правки изображений.

В 2006 году фирма Macromedia вместе со всеми разработками, в том числе и Dreamweaver, была куплена корпорацией Adobe. Dreamweaver 8 — первая версия Dreamweaver, вышедшая под эгидой Adobe, — мог похвастаться улучшением всего того, что было в нем раньше, и еще кое-какими мелкими нововведениями. Dreamweaver CS3 преподнес Web-дизайнерам набор весьма впечатляющих компонентов Spry — меню, панель с вкладками, сворачивающиеся панели и пр., — и средства для взаимодействия с наборами данных; плюс очередные улучшения всего уже имеющегося.

Dreamweaver CS4 отличался от предыдущих версий обновленным в очередной раз пользовательским интерфейсом (достаточно спорное нововведение, на взгляд автора...) и режимом интерактивного просмотра Web-страниц. А Dreamweaver CS5 впервые получил поддержку языков HTML 5 и CSS 3 (правда, сильно ограниченную) и мини-приложений.

И вот — Dreamweaver CS5.5. Что он может нам предложить?

- Улучшенная (ненамного) поддержка HTML 5 и CSS 3.
- Улучшенная поддержка мини-приложений. Теперь программа Adobe Widget Browser, служащая для просмотра и загрузки мини-приложений с Web-сайта Adobe, поставляется в его составе. (Ранее ее приходилось загружать и устанавливать отдельно.)
- Средства для создания кроссплатформенных Web-страниц, в том числе режим многоэкранного просмотра.
- Несколько нововведений, которые наверняка порадуют разработчиков Web-приложений для мобильных устройств. (Впрочем, эта тема выходит за рамки данной книги.)

Налицо, в основном, эволюционное развитие старых наработок. Крупные нововведения обещаются в версии CS6, которая должна выйти весной следующего года. В частности, Adobe "грозится" наконец-то реализовать в новой версии Dreamweaver полную поддержку HTML 5 и CSS 3. (Насколько полной будет эта поддержка, пока неизвестно.)

Чем мы будем заниматься

В этой книге мы будем изучать инструменты, предлагаемые Dreamweaver CS5.5 для создания Web-страниц и Web-сайтов и публикации их в Интернете. Попутно мы разберемся с современными интернет-технологиями и принципами Web-дизайна — куда же без этого...

В процессе изучения Dreamweaver мы создадим два Web-сайта для гипотетической фирмы "Веники и швабры", занимающейся торговлей вениками, метлами, щетками и швабрами. Первый Web-сайт будет совсем простеньким и незатейливым, а второй — уже значительно сложнее: будет использовать таблицы стилей, интерактивные элементы, аудио и видео.

В работе над Web-сайтами мы будем максимально использовать инструменты, предоставляемые Dreamweaver. Если для реализации какой-либо возможности в соста-

ве Dreamweaver не окажется подходящего средства, мы будем набирать необходимый HTML- и CSS-код вручную — тут уж ничего не поделаешь.

Для создания Web-страниц наших Web-сайтов мы используем самые последние на данный момент версии языков HTML и CSS — HTML 5 и CSS 3 соответственно. Хотя они все еще находятся в разработке, современные программы Web-обозревателей уже поддерживают ограниченный набор возможностей, предусмотряемых этими версиями.

Чтобы потом не возникало лишних вопросов, давайте перечислим версии популярных Web-обозревателей, в которых появилась поддержка HTML 5 и CSS 3 на минимально необходимом уровне.

❑ Microsoft Internet Explorer — 9.0.

❑ Mozilla Firefox — 4.0.

❑ Google Chrome — 9.0.

❑ Opera — 12.0.

❑ Apple Safari — 5.1.

Следовательно, если посетитель захочет наведаться на наш Web-сайт, ему потребуется любая из этих версий Web-обозревателя или более новая.

Впрочем, Web-обозреватели с поддержкой HTML 5 и CSS 3 ныне занимают большую часть рынка. Так что никаких проблем у посетителей наших Web-сайтов возникнуть не должно.

Устаревшие технологии Web-дизайна, такие как фреймы и табличный Web-дизайн, мы применять не будем. Станем двигаться в будущее, не оглядываясь на прошлое.

Типографские соглашения

Прежде чем начать изучение Dreamweaver, давайте кое о чем условимся.

В этой книге будут приведены фрагменты кода на языках HTML, CSS и XML. При их написании используются типографские соглашения, уже ставшие своего рода стандартами в компьютерном книгоиздании. Нам необходимо их знать.

❑ В угловые скобки (<>) заключаются наименования значений различных параметров (атрибутов тегов и атрибутов стилей), которые, вместе со скобками, дополнительно выделяются курсивом. В реальный код, разумеется, должно быть подставлено реальное значение. Например:

```
column-rule-color: <значение цвета>
```

Здесь вместо строки <значение цвета> должно быть подставлено реальное значение цвета.

❑ В квадратные скобки ([]) заключаются необязательные фрагменты кода. Например:

```
htm[1]
```

Последняя буква 1 может присутствовать, а может и не присутствовать.

- ❑ Вертикальной чертой (|) разделяются несколько фрагментов кода, из которых в данном месте должен присутствовать только один. Например:

```
column-span: none | all
```

Здесь должно присутствовать либо слово `none`, либо слово `all`, но не оба сразу.

Весь остальной код набирается "как есть".

Благодарности

Автор приносит благодарности своим родителям, знакомым и коллегам по работе.

- ❑ Белову Алексею Васильевичу, начальнику отдела АСУ Волжского гуманитарного института (г. Волжский Волгоградской обл.), где работает автор, — за понимание и поддержку.
- ❑ Всем работникам отдела АСУ — за понимание и поддержку.
- ❑ Родителям — за терпение, понимание и поддержку.
- ❑ Архангельскому Дмитрию Борисовичу — за дружеское участие.
- ❑ Шапошникову Игорю Владимировичу — за содействие.
- ❑ Рыбакову Евгению Евгеньевичу, заместителю главного редактора издательства "БХВ-Петербург", — за неоднократные побуждения к работе, без которых автор давно бы обленился.
- ❑ Издательству "БХВ-Петербург" — за издание моих книг.
- ❑ Всем своим читателям и почитателям — за прекрасные отзывы о моих книгах.
- ❑ Всем, кого я забыл здесь перечислить, — за все хорошее.



ЧАСТЬ I

Введение в современные интернет-технологии

Глава 1. Как все это работает

Глава 2. Принципы построения Web-сайтов



ГЛАВА 1

Как все это работает

Первая глава будет в основном теоретической. Мы рассмотрим основные принципы работы Интернета, познакомимся с программами, обеспечивающими его работу, узнаем, что собой представляют Web-страницы и Web-сайты, и еще много вещей, без которых в современном Web-дизайне просто никуда. А еще мы изучим несколько специальных терминов, которыми будем оперировать в дальнейшем.

Что такое Интернет, или Всемирная сеть, или просто Сеть с большой буквы, думать, объяснять не надо — сейчас это знают все. Так что сразу перейдем к разговору о принципах его работы. Разговор этот будет довольно долгим...

Как работает Интернет

Когда мы набираем в поле ввода интернет-адреса в окне Web-обозревателя адрес нужной нам Web-страницы, она тотчас загружается и выводится на экран. Но как Web-обозреватель находит именно ту Web-страницу, что мы запрашиваем? Каким образом он ее загружает? И где вообще она хранится?

Программы, обеспечивающие работу Интернета. Клиенты и серверы

Любая компьютерная информация — тексты, Web-страницы, графика, аудио- и видеоролики, программы, базы данных, всяческие служебные сведения и пр. — хранится в файлах. Это мы уже знаем.

Информация, которая опубликована в Интернете, также хранится в виде обычных файлов. Эти файлы располагаются на дисках компьютеров, подключенных к Интернету, — вполне обычных компьютеров, наподобие нашего.

Так, Web-страница, которую мы открыли ранее в Web-обозревателе, хранится на каком-то компьютере, подключенном к Интернету, в виде файла (или нескольких файлов; подробнее о хранении Web-страниц мы поговорим потом). И Web-обозреватель просто загрузил этот файл по Сети, прочитал его и вывел нам его содержимое — собственно Web-страницу.

Но как собственно он ее загрузил? Очень просто — в этом ему "помогли" особые программы, о которых сейчас пойдет речь.

Вообще, все программы, работающие с Интернетом и через Интернет, делятся на две большие группы. Рассмотрим их.

Первая группа — *клиентские программы*, или *клиенты*. Это те программы, с которыми мы работаем непосредственно, которые принимают от нас запрос на вывод каких-либо хранящихся в Интернете данных, загружают и выводят эти данные. Типичный пример клиентской программы — знакомый нам Web-обозреватель, который еще иногда называют Web-клиентом.

Мы пользуемся клиентами электронной почты для отправки и получения писем. Для общения в сетях мгновенных сообщений мы пользуемся клиентом мгновенных сообщений. Существуют клиенты для работы с чатами, хранилищами файлов FTP (если кто-то еще помнит, что это такое) и прочими сервисами Интернета.

Отметим важную деталь. Клиентские программы — и Web-обозреватели, и "почтовики", и клиенты мгновенных сообщений — не имеют прямого доступа к файлам, в которых хранится запрашиваемая нами информация. Это сделано из соображений безопасности. Вместо этого они отправляют запросы на получение этих файлов или их фрагментов другим программам, которые относятся ко второй группе.

Они называются *серверными программами*, или *серверами*. Их назначение — получить от клиентской программы запрос на какой-либо файл или фрагмент информации, хранящийся в файле, возможно, как-то его обработать и отправить этому самому клиенту. С пользователями же они никак не взаимодействуют.

Можно сказать, что серверы выступают посредниками между клиентами и информацией, которую они запрашивают.

Web-обозреватель при получении Web-страниц "общается" с серверной программой, называемой *Web-сервером*. Клиент электронной почтой отправляет письма серверу электронной почты и от него же принимает новые письма. Функционирование других сервисов Интернета — систем мгновенных сообщений, чатов, FTP — также обеспечивают соответствующие серверы.

В общем, процесс запроса информации через Интернет выглядит так:

1. Клиент принимает от пользователя интернет-адрес, однозначно идентифицирующий фрагмент информации, которую тот желает получить. Таким фрагментом информации может быть файл Web-страницы. Как вариант, клиент получает от пользователя команду на получение определенного фрагмента информации по заранее заданному интернет-адресу (это справедливо для клиентов электронной почты).
2. Клиент отправляет серверу, находящемуся по этому интернет-адресу, запрос на получение данного фрагмента информации. Этот запрос включает собственно интернет-адрес, введенный пользователем, команду, указывающую, что именно клиенту нужно от сервера, и, возможно, дополнительные данные, необходимые серверу для выполнения запроса.

3. Сервер принимает запрос от клиента, проверяет, существует ли на его дисках запрошенный данный фрагмент информации и имеет ли данный клиент к нему доступ. Если это так, Web-сервер отправляет фрагмент информации клиенту. В противном случае он посылает клиенту сообщение об ошибке, включающее код, однозначно идентифицирующий возникшую проблему, и, возможно, текстовое описание ошибки.
4. Клиент получает от сервера либо запрошенный фрагмент информации, либо сообщение об ошибке и выводит его на экран.

Примерно в такой же последовательности клиент отправляет данные серверу. Отправляемые данные передаются в особом блоке, следующем сразу за запросом.

Как видим, работу Интернета обеспечивают клиентские и серверные программы, функционирующие в тесном контакте. Такая *архитектура* (принцип построения компьютерных систем) носит название "*клиент-сервер*", или *двухзвенной*.

НА ЗАМЕТКУ

Существуют и другие разновидности компьютерной архитектуры. Так, *трехзвенная* архитектура предусматривает еще одну серверную программу, находящуюся между клиентом и сервером и выполняющую обработку запрашиваемых данных. А *однозвенная* архитектура наделяет интернет-программы функциями и клиента, и сервера; по такому принципу работают программы клиентов пиринговых, или файлообменных, сетей.

Осталось ввести еще пару терминов. Компьютер, на котором пользователь работает с клиентской программой, мы назовем *клиентским*. А компьютер, где работает программа серверная, пусть называется *серверным*.

ВНИМАНИЕ!

Часто серверный компьютер называют сервером, как и серверную программу. Это может породить путаницу, т. к. на одном серверном компьютере могут работать сразу несколько серверных программ. Поэтому в данной книге словом "сервер" всегда будет называться именно серверная программа.

Протоколы

Люди, чтобы понять друг друга, должны разговаривать на одном языке. Аналогично, программы клиента и сервера, чтобы успешно взаимодействовать, также должны использовать один "язык", или, в терминологии Интернета, *протокол*.

Сегодняшний Интернет использует огромное множество всяких протоколов. Одни протоколы функционируют на самом низком уровне и определяют базовые возможности передачи данных. Другие протоколы работают поверх первых и определяют возможности более высокого уровня.

Самый, можно сказать, фундаментальный протокол Интернета носит незамысловатое название Internet Protocol (протокол Интернета), или *IP*. Он определяет базовые функции Интернета, касающиеся простой передачи данных: способ написания интернет-адресов компьютеров, формат, в котором данные передаются через Интернет, и простейшие средства защиты от сбоев, которые могут при этом возникнуть (и, надо сказать, возникают).

Протокол IP просто передает данные от одного компьютера к другому. Что это за данные и для какой программы они предназначены, его не касается.

Другой фундаментальный протокол — Transfer Control Protocol (протокол управления передачей), или *TCP*. Он работает поверх протокола IP, пользуясь его услугами по передаче данных.

Прежде всего, протокол TCP выполняет передачу данных от программы программе. Этим он отличается от IP, для которого нет разницы, предназначен ли данный фрагмент информации Web-серверу или клиенту электронной почты. Для этого TCP вводит поддержку портов (о которых мы скоро поговорим) и реализующие ее дополнительные структуры данных, передаваемых в составе основной информации.

Кроме этого, TCP обеспечивает возможность гарантированной доставки данных. Он строго следит, чтобы ни один фрагмент данных не был потерян при передаче, а если это все-таки случится, он будет отправлен повторно.

НА ЗАМЕТКУ

Протоколы TCP и IP настолько "срослись" друг с другом, что их часто обозначают одной аббревиатурой — *TCP/IP*.

Здесь мы рассмотрели два *протокола низкого уровня*, которые используются всеми интернет-программами и реализуются операционной системой. Далее речь пойдет о *протоколах высокого уровня*; они используются только определенными клиентами и серверами и реализуются ими же. Эти протоколы работают "поверх" TCP (и, соответственно, IP).

Протоколы высокого уровня определяют:

- вид передаваемых с их помощью данных (Web-страницы, письма, мгновенные сообщения и др.);
- формат интернет-адресов, указывающих на фрагменты информации;
- набор команд, с помощью которых клиент указывает серверу, что ему нужно получить;
- формат дополнительных данных, отправляемых в некоторых случаях клиентом серверу или, наоборот, сервером клиенту.

Протоколов высокого уровня очень много. Каждый сервис Интернета использует свой протокол высокого уровня, а то и сразу несколько — либо конкурирующих, либо предназначенных для разных задач.

Самый популярный сервис Интернета — WWW — использует для работы протокол HyperText Transfer Protocol (протокол передачи гипертекста), или *HTTP*. Он определяет формат, используемый для указания интернет-адресов получаемых файлов, и набор команд, которыми Web-обозреватель "командует" Web-сервером. (Интересно, что этих команд всего три.)

HTTP — самый известный протокол Интернета. Пожалуй, всем более-менее грамотным интернетчикам известны эти четыре буквы.

Электронная почта использует сразу три протокола. Протокол Simple Mail Transfer Protocol (простой протокол передачи почты), или *SMTP*, используется для отправки писем клиентом серверу. Протокол Post-Office Protocol ver. 3 (почтовый протокол версии 3), или *POP3*, применяется для передачи писем от сервера клиенту. А протокол Internet Message Access Protocol (протокол доступа к почте Интернета), или *IMAP*, создан в качестве замены устаревающему *POP3*, но все никак не вытеснит его.

Также свои собственные протоколы используют различные системы мгновенных сообщений, FTP, службы интернет-телефонии, чаты и пр. Кроме того, существует ряд протоколов, с помощью которых различные серверные программы обмениваются друг с другом служебными данными. Впрочем, это уже выходит за рамки данной книги...

Порты TCP

Говоря о протоколах, мы упомянули несколько не знакомых нам вещей. Настала пора рассмотреть их.

Мы уже знаем, что на одном серверном компьютере могут работать сразу несколько серверных программ. Также на одном клиентском компьютере могут одновременно работать несколько программ-клиентов. При этом все они могут в одно и то же время передавать и принимать данные, не мешая друг другу. Как это достигается?

Дело в том, что протокол TCP делит канал доступа к Интернету на множество отдельных "канальчиков", называемых *портами TCP*, или просто *портами*. Каждый порт TCP имеет уникальный номер, а всего их 65 535.

Так вот, данные, предназначенные для разных программ, передаются через разные порты; при этом данные, передаваемые через один порт, не смешиваются с данными, передаваемые через другой. Собственно, поддержка портов и введена для того, чтобы обеспечить передачу данных от программы программе.

Но какие именно порты используются для передачи данных между программами? Это зависит от протокола, по которому эти программы работают.

Прежде всего, все программы, работающие по одному протоколу, — и клиенты, и серверы — используют для передачи служебной информации — запросов от клиента серверу и ответов сервера клиенту — специально выделенный порт, называемый *управляющим*. Именно через этот порт Web-обозреватель передает все запросы Web-серверу, а Web-сервер возвращает Web-обозревателю сообщения об успешной передаче запрошенных файлов или сообщения о возникших ошибках. Именно через этот порт клиент электронной почты передает серверу команды на получение списка новых сообщений, а сервер отправляет клиенту ответы, содержащие количество новых сообщений или сигнализирующие, что новых сообщений нет.

Каждый протокол определяет номер порта, который используется в качестве управляющего по умолчанию; он так и называется — *порт по умолчанию*. Так, для протокола HTTP портом по умолчанию является 80-й, для протокола SMTP — 25-й, а для протокола POP3 — 110-й.

А что же сами данные? Они могут быть переданы через любой порт, который в данный момент не занят передачей других данных и не является портом по умолчанию для какого-либо протокола. Порт для передачи данных выбирается сервером, который сообщает его номер клиенту в ответном сообщении.

Вернемся к Web-странице, которую мы загрузили ранее. Когда мы запустили ее открытие, Web-обозреватель передал запрос Web-серверу через управляющий порт 80 — порт по умолчанию для протокола HTTP. Получив этот запрос, Web-сервер загрузил файл и выяснил, что для его передачи Web-обозревателю можно использовать порт с номером, скажем, 1237, который не занят передачей другого файла и не является портом по умолчанию для других протоколов. После этого Web-сервер, опять же, через управляющий порт 80, отправил Web-обозревателю ответ, содержащий номер выбранного порта — 1237. Web-обозреватель после получения ответа сможет без труда загрузить запрошенный файл через этот порт.

Передача данных через порт, отличный от управляющего, — великолепная идея! Благодаря этому мы можем загружать с сервера сразу несколько массивов данных. Так, возможна одновременная загрузка с одного и того же Web-сервера нескольких больших файлов, и все они будут загружаться, не мешая друг другу. А управляющий порт всегда будет свободен.

Почти все серверные программы позволяют в своих настройках указать в качестве управляющего другой порт, отличный от порта по умолчанию. Это может пригодиться в случае, если на серверном компьютере планируется установить две однотипные серверные программы, например, два Web-сервера, которые должны работать одновременно. В таком случае один Web-сервер будет использовать в качестве управляющего порт по умолчанию (80-й), а второй задействует другой порт, скажем, 8000-й. (Автору, кстати, встречались Web-серверы, использующие в качестве управляющего именно этот порт.)

Интернет-адреса

Осталось только поговорить об интернет-адресах. Упомянули мы их очень часто, но обстоятельного разговора так и не было...

В общем случае *интернет-адрес* — это комбинация чисел и букв, однозначно идентифицирующая фрагмент информации, запрашиваемый клиентом. Так, когда мы собираемся открыть в Web-обозревателе какую-либо Web-страницу, мы набираем ее интернет-адрес.

IP-адреса и доменные имена

Интернет-адреса, указывающие на компьютеры, делятся на две разновидности, о которых нам обязательно следует узнать.

IP-адрес — самая старая разновидность интернет-адресов. Такие интернет-адреса были в ходу на заре эпохи Интернета.

IP-адрес представляет собой шестнадцатеричное число, однозначно идентифицирующее компьютер. IP-адреса непосредственно используются протоколом IP —

ключевым протоколом Интернета, — из-за чего они и получили свое название. Примеры IP-адресов: **45.37.122.80** и **54.89.6.201.72.10**.

Каждый компьютер, подключенный к Интернету, должен иметь уникальный IP-адрес, в противном случае он просто не сможет работать в сети. IP-адрес либо указывается в настройках протокола IP на этом компьютере, либо автоматически присваивается особой программой, работающей на серверном компьютере, через который данный компьютер выходит в Интернет (*шлюзе*). Технология автоматического присвоения IP-адреса и используемый ей служебный протокол носят название Dynamic Host Configuration Protocol (протокол динамического конфигурирования узла), или *DHCP*, а серверная программа, занимающаяся "раздачей" IP-адресов, — *DHCP-сервером*.

Существуют два стандарта формирования IP-адресов, различающиеся по количеству формирующих их шестнадцатеричных цифр, и, соответственно, две версии протокола IP, которые их поддерживают. Рассмотрим их.

□ Стандарт *IPv4* предусматривает для формирования IP-адреса четыре цифры. Так, упомянутый ранее IP-адрес **45.37.122.80** относится именно к этому стандарту (посчитаем количество цифр в этом IP-адресе — их как раз четыре).

Стандарт IPv4 позволяет сформировать 4 294 967 296 уникальных интернет-адресов (в реальности, из-за некоторых ограничений протокола IP, — чуть меньше). Он возник, когда компьютеров, подключенных к нему, было относительно мало, и даже сами создатели Всемирной сети не предполагали, что их детище станет таким популярным.

□ Стандарт *IPv6* предлагает формировать IP-адреса из шести цифр. Пример такого интернет-адреса — **54.89.6.201.72.10**.

В настоящее время к Интернету, помимо обычных компьютеров, подключаются мобильные телефоны, планшеты и чуть ли не бытовая техника. Естественно, свободных IP-адресов, сформированных на основе стандарта IPv4, стало катастрофически не хватать; более того, специалисты предрекают, что такие IP-адреса закончатся совсем скоро. Стандарт IPv6 призван устранить эту проблему — он позволяет сформировать $5 \cdot 10^{28}$ интернет-адресов (по 300 млн интернет-адресов на каждого жителя планеты Земля), что более чем достаточно.

В настоящее время, однако, наибольшей популярностью пользуется стандарт IPv4. Его конкурент и преемник IPv6, если и используется, то не очень широко, хотя в ближайшем будущем положение наверняка изменится.

Достоинство у IP-адресов одно — их без всякого преобразования "понимает" протокол IP. Недостаток тоже один — их плохо понимают люди, пользователи Интернета. В самом деле, попробуйте-ка запомнить пару десятков IP-адресов ваших любимых Web-сайтов!..

Именно поэтому с расширением Интернета была введена новая система его структурирования и, кроме этого, новая разновидность интернет-адресов, называемая доменными именами.

Прежде всего, вся сеть Интернет была разделена на несколько десятков крупных фрагментов — *доменов*, или *доменных зон*. В каждый такой домен попали все ком-

пьютеры, либо относящиеся к организации определенной разновидности (коммерческой, образовательной, военной, правительственной и пр.), либо находящиеся в одной стране. В первом случае получится *интернациональный* домен, а во втором — *национальный*.

Каждый домен получил уникальное имя, состоящее из букв латинского алфавита. Так, интернациональный домен, включающий компьютеры коммерческих организаций, получил имя *com*, интернациональный домен, объединяющий компьютеры правительственных структур, — имя *gov*, а национальный домен, объединяющий российские компьютеры, — имя *ru*.

Внутри получившихся доменов были сформированы другие домены, более мелкие. Они объединили компьютеры, принадлежащие отдельным организациям, находящиеся в одной территориальной единице или вообще в одном городе, и также получили имена, уникальные в пределах домена, в котором они были сформированы. Так, домен, объединяющий компьютеры, что находятся в Волгограде, получил "говорящее" имя *volgograd*, домен с компьютерами, находящимися в Новосибирске, — имя *nsk*, а домен с компьютерами издательства "БХВ-Петербург" — имя *bhv*.

А внутри этих доменов были сформированы еще более мелкие и совсем микроскопические. Они объединили компьютеры, находящиеся в отделах организации, городах территориальной единицы, районах или организациях города, и также получили имена, уникальные в пределах "внешнего" домена. Самые мелкие из этих доменов вообще содержат только один компьютер. Так, домен с единственным компьютером, на котором установлен Web-сервер издательства "БХВ-Петербург", получил имя *www*.

Как видим, весь Интернет разделен на домены, вложенные друг друга, как матрешки. Крупные домены содержат миллионы компьютеров, более мелкие — тысячи, еще более мелкие — единицы, десятки или сотни. А самые мелкие домены вообще получили в свое "ведение" один компьютер. И каждый такой домен имеет уникальное имя.

Что же мы получим? Набор имен доменов, последовательно вложенных друг в друга, который однозначно идентифицирует компьютер, подключенный к Интернету. Совсем как IP-адрес!

Если мы запишем имена всех доменов, к которым принадлежит нужный нам компьютер, в порядке от самого мелкого к самому крупному в направлении слева направо, разделив их точками, мы получим *доменное имя* компьютера. Это вторая разновидность интернет-адресов.

Предположим, что мы хотим записать доменное имя компьютера, на котором установлен Web-сервер издательства "БХВ-Петербург". Самый мелкий домен, в котором он находится, называется *www*; он вложен в более крупный домен *bhv*, а он, в свою очередь, в огромный доменище *ru*. В результате мы получим доменное имя **www.bhv.ru**. Согласитесь — запомнить это несравнимо проще, чем невразумительный IP-адрес!

Но протокол IP не "понимает" доменных имен — ему подавай только IP-адреса! Что же делать?

Было найдено следующее решение. В Интернете был запущен особый сервис Domain Name System (система доменных имен), или *DNS*. Он включает, прежде всего, набор *DNS-серверов*, работающих на специально выделенных для этого серверных компьютерах. Каждый из этих DNS-серверов обслуживает определенный домен и хранит список доменных имен компьютеров, находящихся в этом домене, и соответствующих им IP-адресов. Все DNS-серверы связаны друг с другом.

А операционная система включает в свой состав программу DNS-клиента. Он, получив доменное имя, отправляет его DNS-серверу и получает в ответ IP-адрес, соответствующий этому имени, который и передает программам, входящим в состав операционной системы и обеспечивающим работу протокола IP. Все это работает абсолютно прозрачно для пользователя.

Формирование интернет-адресов

Ранее мы много говорили о двух разновидностях интернет-адресов: IP-адресах и доменных именах. Но все эти интернет-адреса идентифицировали компьютеры. Нам же нужно идентифицировать фрагменты информации, которые на них хранятся. Ведь мы, как Web-дизайнеры, будем иметь дело именно с фрагментами информации, а именно с файлами.

Так что настала пора рассмотреть процесс формирования интернет-адресов, указывающих на фрагменты информации, или *полных* интернет-адресов.

Возьмем для примера несколько интернет-адресов компьютеров. Используем для этого только доменные имена — они привычнее.

www.bhv.ru

mail.ru

www.somesite.ru

Эти интернет-адреса обрабатываются протоколом IP.

Прежде всего, нам нужно указать на программу, с помощью которой мы будем получать требуемые фрагменты информации. Для этого требуется добавить к интернет-адресу компьютера название протокола. Оно записывается в начале интернет-адреса и отделяется от него двоеточием и двумя символами слэша (/).

http://www.bhv.ru

Здесь мы указали в интернет-адресе протокол HTTP, получив тем самым интернет-адрес программы Web-сервера.

ftp://www.somesite.ru

А здесь мы указали протокол FTP, получив интернет-адрес программы FTP-сервера. Похоже, что эта программа работает на том же компьютере, что и программа Web-сервера; об этом говорит имя домена — *www*.

Единственное исключение из этого правила — интернет-адреса серверов электронной почты. В них не указывается наименование протокола. Например, ниже представлен такой интернет-адрес:

mail.ru

Вообще, электронная почта — весьма специфический сервис Интернета, стоящий особняком; даже в списках DNS-серверов, где перечисляются доменные имена и соответствующие им IP-адреса, предусмотрены специальные разделы для интернет-адресов серверов электронной почты.

Бывает так, что серверная программа, к которой мы хотим обратиться, использует управляющий порт, отличный от порта по умолчанию для данного протокола. (О портах TCP было рассказано в соответствующем параграфе.) В этом случае мы можем указать номер нужного порта прямо в интернет-адресе, поставив его в конце и отделив от остального адреса двоеточием.

<http://www.somesite.ru:8000>

Здесь мы обратились к другому Web-серверу, работающему на компьютере с интернет-адресом **www.somesite.ru** и использующему в качестве управляющего порт с номером 8000.

Осталось сказать, что полученный интернет-адрес будет обрабатываться уже протоколом TCP, что, в принципе, и так понятно.

Теперь укажем в интернет-адресе ссылку на фрагмент информации, который нам требуется получить: путь к файлу или папке, имя ящика электронной почты или что-то еще. Интернет-адрес, который мы получим, будет обрабатываться протоколом высокого уровня: HTTP, SMTP, POP3, FTP и др.

Формат указания в интернет-адресе фрагмента информации зависит от конкретного протокола. Например, протокол HTTP требует помещать путь к запрашиваемому файлу в конце интернет-адреса.

<http://www.somesite.ru/page1.htm>

Получив такой интернет-адрес, Web-обозреватель обратится к Web-серверу, расположенному по интернет-адресу **<http://www.somesite.ru>**, и запросит файл `page1.htm`, находящийся в корневой папке Web-сайта, работу которого обеспечивает вышеупомянутый Web-сервер. (О корневой папке и прочих технических деталях, связанных с Web-сайтами, мы узнаем позже.)

<http://www.somesite.ru/folder1/folder2/page45.htm>

А здесь мы запрашиваем файл `page45.htm`, находящийся в папке `folder2`, вложенной в папку `folder1`, которая, в свою очередь, вложена в корневую папку Web-сайта, работу которого обеспечивает Web-сервер с интернет-адресом **<http://www.somesite.ru>**. Или, если короче, файл `folder1/folder2/page45.htm` Web-сайта **<http://www.somesite.ru>**.

<http://www.somesite.ru/folder1/>

Протокол HTTP допускает указание интернет-адресов папок. Например, сейчас мы обратились к папке `folder1` Web-сайта **<http://www.somesite.ru>**. Что случится в этом случае, мы выясним потом.

<http://www.bhv.ru/>

А здесь мы обратились к самой корневой папке Web-сайта **<http://www.bhv.ru>**. Символ слэша (`/`), находящийся в конце интернет-адреса, обозначает как раз корневую папку.

Протоколы SMTP и POP3 требуют указывать имя почтового ящика перед интернет-адресом программы почтового сервера, отделяя его символом @.

mailbox22@mail.ru

Обращаемся к почтовому ящику mailbox22, зарегистрированному на почтовом сервере **mail.ru**.

Мы еще вернемся к интернет-адресам далее в этой главе, когда будем рассматривать технические детали, связанные с Web-сайтами, и в *главе 6*, посвященной средствам навигации. А пока что закончим с ними, и вообще с принципами работы Интернета.

Web-страницы и Web-сайты

Настало время поговорить о Web-страницах и Web-сайтах, выяснить, что они собой представляют, и рассмотреть принципы, на основе которых они создаются. Точнее, начать рассмотрение основных принципов — это разговор весьма долгий, и растянется он на несколько последующих глав.

Что такое Web-страница и Web-сайт

Дадим определение Web-страницы и Web-сайта, чтобы знать, о чем вообще идет речь. Итак, *Web-страница* — это документ, предназначенный для распространения в Интернете посредством сервиса World Wide Web (повсеместно протянутая паутина), или WWW, или просто Всемирная паутина.

Web-страница представляет собой обычный текстовый файл, имеющий расширение htm[1]. В этом файле содержится код Web-страницы — текст, являющийся ее содержимым, который отформатирован с помощью специальных команд.

Некоторые части содержимого Web-страницы могут храниться в отдельных файлах. Так, в отдельных файлах хранятся графические изображения, опубликованные на данной Web-странице, внешние таблицы стилей, аудио- и видеоролики и некоторые другие данные. Подробнее обо всем этом мы поговорим в следующих главах книги.

Для просмотра Web-страниц используются специализированные клиентские программы — Web-обозреватели. Для собственно распространения их в Интернете, т. е. выдачи в ответ на запросы клиентов, применяются специализированные серверные программы — Web-серверы. Собственно, мы уже это знаем.

Web-сайт — это совокупность Web-страниц, объединенных общей тематикой и связанных друг с другом гиперссылками. (О гиперссылках и вообще средствах навигации мы поговорим в *главе 6*.)

Web-сайт может быть большим, включающим десятки, а то и сотни и даже тысячи Web-страниц. Web-сайт может быть и маленьким, насчитывая десяток Web-страниц. Встречаются также Web-сайты, представляющие собой одну-единственную Web-страницу; обычно это Web-сайты — "заглушки", временно заменяющие "полноразмерный" Web-сайт, над которым в настоящее время ведется работа.

Содержимое, представление и поведение Web-страницы

Все, что находится на любой Web-странице, можно разделить на три категории: содержимое, представление и поведение.

Содержимое Web-страницы — это, собственно, информация, которая на ней выводится. Это текст, разбитый на абзацы и блоки, возможно, предваренные заголовками. Это таблицы. Это графические изображения, аудио- и видеоролики. Это гиперссылки. Это всевозможные конструкции, не относящиеся к тексту, например горизонтальные линии. В общем, все то, что называется *элементами Web-страницы*.

Согласно современным принципам Web-дизайна, содержимое — это и есть Web-страница. То есть в текстовом файле с расширением `htm[1]`, который и представляет собой Web-страницу, хранится как раз ее содержимое. Как уже говорилось, некоторые части содержимого хранятся в отдельных файлах; это, прежде всего, графические изображения, аудио- и видеоролики.

Содержимое создается с помощью особого языка, называемого HyperText Markup Language (язык гипертекстовой разметки), или *HTML*. Этот язык определяет набор команд, называемых *тегами* и предназначенных для разбиения текста на абзацы, формирования таблиц, вставки графических изображений и пр.

Представление Web-страницы — это набор правил, определяющих внешний вид различных элементов Web-страницы: абзацев, заголовков, таблиц и др. Говоря проще, представление позволяет указать, каким шрифтом должен быть набран вот этот заголовок, каким цветом должно быть выведено вот это слово в абзаце, и должна ли вот эта таблица иметь рамку, и если должна, то какой толщины и какого цвета.

Представление Web-страницы создается с помощью так называемых таблиц стилей. Таблицы стилей могут располагаться как в самом файле Web-страницы, так и в отдельных файлах. Подробнее о таблицах стилей и их возможностях по оформлению Web-страниц мы поговорим в *части III*.

Представление, т. е. таблицы стилей, пишутся на другом языке — Cascading Style Sheets (каскадные таблицы стилей), или *CSS*. Он предлагает богатые средства задания нужного нам оформления Web-страницы, например имени, размера и начертания шрифта, цвета текста и фона, выравнивания абзаца, параметров рамки и т. д.

Поведение Web-страницы — это набор правил, указывающих, как должна вести себя Web-страница в ответ на определенные действия пользователя. Так, мы можем указать, чтобы при наведении курсора мыши на какой-либо фрагмент текста менялся цвет его рамки или чтобы при щелчке на заголовке расположенный ниже его абзац появлялся на экране. Другими словами, мы можем создать *интерактивные элементы Web-страницы*.

Поведение Web-страницы в настоящее время реализуется с помощью Web-сценариев — небольших программ, управляющих Web-страницей и ее элементами.

Web-сценарии могут располагаться как в самом файле Web-страницы, так и в отдельных файлах.

Поведение, т. е. Web-сценарии, пишутся на языке программирования *JavaScript*. Его возможностей вполне хватает для реализации даже весьма сложных интерактивных элементов: раскрывающихся абзацев, "гармошек", меню, анимации и даже полнофункциональных приложений, размещаемых прямо на Web-странице.

Мы не будем рассматривать написание Web-сценариев на языке JavaScript — это довольно большая тема, которой впору посвящать отдельную книгу. Вместо этого мы воспользуемся средствами современных Web-редакторов по созданию поведения Web-страницы — разнообразных интерактивных элементов. Эти средства мы рассмотрим в *части V*.

Одно из важнейших правил современного Web-дизайна гласит: содержимое, представление и поведение должны быть отделены друг от друга. А в более конкретном приложении — содержимое (сами Web-страницы), представление (таблицы стилей) и поведение (Web-сценарии) должны быть разнесены по разным файлам. Смешивать их настоятельно не рекомендуется.

Есть, по крайней мере, две причины так сделать.

- Мы можем применить одни и те же представление и поведение, хранящиеся в отдельных файлах, к любой части содержимого — к любой Web-странице нашего Web-сайта, — без необходимости каждый раз создавать их копии.
- Если возникнет необходимость исправить представление или поведение, нам будет достаточно внести требуемые правки в один-единственный файл — тот, где хранится это представление или поведение. Править сами файлы Web-страниц нам не потребуется.

К сожалению, если представление практически всегда удастся безболезненно отделить от содержимого, то с поведением это получается не всегда. Такова специфика интерактивных элементов, создаваемых средствами Web-редактора. В *части V* мы сами в этом убедимся.

Web-сайты и Web-серверы: технические детали

Теперь рассмотрим некоторые технические детали, касающиеся Web-сайтов и Web-серверов. Впоследствии знание их нам обязательно пригодится.

Корневая папка Web-сервера

Предположим, что у нас есть готовый Web-сайт и установленный, настроенный и полностью работоспособный Web-сервер. Что нам нужно сделать, сделать Web-сайт доступным в Интернете?

При установке Web-сервера на жестком диске компьютера создается особая папка, называемая *корневой*. В эту папку помещаются все файлы и папки, составляющие Web-сайт, который должен быть опубликован в Интернете с помощью данного Web-сервера. Все файлы и папки, находящиеся вне этой папки, в Интернете доступны не будут.

Получив запрос от Web-обозревателя, Web-сервер будет искать требуемый им файл в этой самой корневой папке. Если он там не обнаружится, Web-сервер сформирует и отправит Web-обозревателю сообщение об ошибке.

Так, если мы наберем вот такой интернет-адрес

<http://www.somesite.ru/page1.htm>

Web-сервер будет искать файл `page1.htm` в корневой папке Web-сайта.

А в случае набора интернет-адреса

<http://www.somesite.ru/folder1/folder2/page45.htm>

Web-сервер будет искать файл `page45.htm` в папке `folder2`, вложенной в папку `folder1`, которая, в свою очередь, находится, опять же, в корневой папке.

НА ЗАМЕТКУ

Нужно отметить, что все серьезные программы Web-серверов позволяют создавать так называемые *виртуальные папки*. Виртуальная папка может находиться в любом месте файловой системы компьютера, но Web-сервер считает ее частью Web-сайта, словно она находится в его корневой папке.

Так, если мы наберем интернет-адрес

<http://www.somesite.ru/virtualfolder1/picture2.jpg>

Web-сервер обратится к реальной папке, соответствующей виртуальной папке `virtualfolder1`, и отыщет там файл `picture2.jpg`

Web-страница по умолчанию

Но мы ведь далеко не всегда открываем в Web-обозревателе какие-либо определенные файлы. Гораздо чаще мы набираем в нем интернет-адреса вида

<http://www.somesite.ru/>

Здесь символ слэша (`/`), присутствующий в конце интернет-адреса, обозначает корневую папку Web-сайта. То есть мы фактически обратились к папке, а не к файлу. (Как мы выяснили ранее, протокол HTTP такое допускает.)

Как поступит Web-сервер, получив такой запрос? Какой файл он отправит Web-обозревателю?

Файл, хранящий так называемую *Web-страницу по умолчанию*. Этот файл, имеющий строго определенное имя, Web-сервер будет искать в папке, к которой был выполнен запрос.

<http://www.somesite.ru/folder1/>

А в этом случае Web-сервер отправит Web-страницу по умолчанию, хранящуюся в папке `folder1` корневой папки.

Файлы с Web-страницами по умолчанию должны иметь строго определенное имя — обычно вида `default.htm[1]` или `index.htm[1]`; эти имена поддерживаются большинством современных Web-серверов. Впрочем, современные Web-серверы позволяют задать в своих настройках для файла с Web-страницей по умолчанию произвольное имя.