

Владимир Дронов



Adobe Dreamweaver CS4

+ВИДЕОКУРС



- Основные принципы Web-дизайна
- Фреймовый, табличный и контейнерный Web-дизайн
- Каскадные таблицы стилей
- Web-сценарии
- Использование эффектов и компонентов Spry
- Написание серверных Web-приложений
- Работа с наборами данных HTML и XML

+  cd

Наиболее
полное
руководство

В ПОДЛИННИКЕ®

УДК 681.3.06
ББК 32.973.26-018.2
Д75

Дронов В. А.

Д75 Adobe Dreamweaver CS4. — СПб.: БХВ-Петербург,
2009. — 832 с.: ил. + Видеокурс (на CD-ROM) — (В подлиннике)
ISBN 978-5-9775-0412-6

Книга посвящена созданию Web-страниц и Web-сайтов в русской версии визуального Web-редактора Adobe Dreamweaver CS4. Подробно рассказывается об основных принципах Web-дизайна, о верстке Web-страниц с использованием фреймов, таблиц и контейнеров, использовании каскадных таблиц стилей CSS, Web-сценариев и эффектов и компонентов Spry. Особое внимание уделяется написанию в среде Dreamweaver серверных Web-приложений с использованием технологии ASP и работе с наборами данных HTML и XML. Подробно рассказывается о публикации готовых Web-сайтов в Интернете. Прилагаемый CD содержит видеокурс по основам работы в Adobe Dreamweaver CS4.

Для Web-дизайнеров и Web-программистов

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>Игорь Цырульников</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 22.02.09.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 67,08.

Тираж 2500 экз. Заказ №

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

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.60.953.Д.003650.04.08 от 14.04.2008 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 978-5-9775-0412-6

© Дронов В. А., 2009
© Оформление, издательство "БХВ-Петербург", 2009

Оглавление

Введение	1
Dreamweaver CS4 — что нового?	2
Типографские соглашения	3
Как мы будем изучать Dreamweaver.....	3
Благодарности	4
ЧАСТЬ I. СОЗДАЕМ ПРОСТЕЙШИЕ WEB-СТРАНИЦЫ	5
Глава 1. Введение в современные интернет-технологии	7
Основные принципы работы Интернета	7
Что такое Интернет	7
Сервисы Интернета	9
Клиенты и серверы.....	9
Протоколы	13
Интернет-адреса	15
Основные понятия WWW	17
Web-страницы и Web-сайты.....	17
Web-обозреватели	21
Web-серверы.....	23
Публикация Web-сайта в Интернете. Хостинг-провайдеры.....	23
Как создаются Web-страницы?	25
Язык HTML и его теги.....	25
Вложенность тегов.....	27
Две секции Web-страницы	29
Гиперссылки	30
Интернет-адреса в WWW	32
Каскадные таблицы стилей CSS	33
Физическое и логическое форматирование	35
Будущее HTML.....	36
Что дальше?	37

Глава 2. Основные принципы работы с Web-редактором Dreamweaver CS4.....	38
Web-редакторы.....	38
Среда Adobe Dreamweaver CS4.....	40
Главное окно программы.....	40
Окна документов.....	42
Панели.....	46
Раскладки.....	52
Предварительная настройка Dreamweaver.....	56
Учим русский.....	56
Настраиваем параметры создаваемых Web-страниц.....	59
Задаем параметры создаваемого HTML-кода.....	60
Файловые операции.....	61
Инструменты для работы с Web-страницами.....	63
Инструменты, присутствующие в окнах документов.....	64
Просмотр Web-страницы.....	67
Поиск и замена текста.....	70
Вызов справки.....	80
Что дальше?.....	81
Глава 3. Создание простейших Web-страниц.....	82
Создание новой Web-страницы.....	82
Работа с текстом.....	84
Набор текста.....	84
Форматирование абзацев.....	87
Форматирование фрагментов текста.....	94
Вставка специальных символов.....	101
Работа с гиперссылками.....	108
Обычные гиперссылки.....	108
Почтовые гиперссылки.....	113
Якоря.....	114
Общие параметры Web-страницы.....	117
Дополнительные возможности Dreamweaver.....	123
Вставка и чтение комментариев.....	123
Вставка текущей даты.....	124
"Чистка" HTML-кода.....	126
"Чистка" HTML-кода, порожденного Microsoft Word.....	127
Что дальше?.....	131
Глава 4. Работа с кодом HTML.....	132
Основные средства Dreamweaver для работы с HTML-кодом.....	132
Три режима отображения Web-страницы.....	133
Вызов справки по HTML.....	135

Простейшие средства для работы с кодом HTML	136
Свертка кода HTML	140
Расширенные средства работы с HTML-кодом.....	142
Вставка тегов в HTML-код.....	142
Правка и удаление тегов	145
Настройки, влияющие на создаваемый HTML-код	147
Что дальше?	149

ЧАСТЬ II. ПРОДОЛЖАЕМ СОЗДАВАТЬ WEB-СТРАНИЦЫ 151

Глава 5. Графика и мультимедиа 153

Введение в интернет-графику	154
Внедренные элементы	154
Растровые и векторные изображения. Сжатие графики	155
Форматы интернет-графики	156
Простые графические изображения	157
Вставка графического изображения	157
Параметры графического изображения	162
Правка изображения в Dreamweaver	164
Специальные изображения.....	168
Изображения-гиперссылки.....	168
Активные изображения.....	170
Карты-изображения	172
Фоновые изображения	178
Мультимедиа	180
Поддержка мультимедийных данных.....	180
Графика Shockwave/Flash	183
Фильмы Flash Video	190
Прочее мультимедийное содержимое	195
Что дальше?	201

Глава 6. Таблицы 202

Текст фиксированного формата	202
Работа с таблицами	205
Создание таблицы	205
Как формируются таблицы.....	208
Задание размеров таблицы, ее строк и столбцов.....	211
Выделение элементов таблиц.....	214
Работа с таблицей и ее элементами	216
Форматирование таблиц	219
Объединение ячеек таблиц	223
Практикум по работе с таблицами.....	225
Специальные возможности по работе с таблицами	227
Сортировка таблиц.....	227
Вставка табличных данных	230

Составные изображения	232
Фиксация размеров ячеек таблицы	234
Недостатки таблиц и их преодоление	235
Что дальше?	237
Глава 7. Работа с Web-сайтом	238
Начала сайтостроения.....	238
Планирование сайта	239
Публикация Web-сайта	245
Регистрация сайта в Dreamweaver	247
Процесс регистрации сайта	248
Настройка прокси-сервера.....	255
Работа с локальной копией сайта	256
Управление файлами и папками локальной копии	256
Управление Web-сайтами, зарегистрированными в Dreamweaver	260
Взаимодействие панели <i>Файлы</i> и окна документа	263
Проверка гиперссылок и HTML-кода	264
Замена гиперссылок	268
Публикация сайта.....	269
Соединение с FTP-сервером.....	269
Простая публикация сайта и отдельных его файлов	270
Синхронизация копий сайта.....	271
Разъединение с FTP-сервером.....	275
Работа с удаленной копией сайта	275
Поиск обновленных файлов	278
Работа с Web-страницами на FTP-сервере напрямую	279
Дополнительные возможности работы с Web-страницами.....	280
Заметки.....	280
Активы	286
Избранные элементы	288
Библиотека	290
Что дальше?	293
ЧАСТЬ III. ЗАНИМАЕМСЯ WEB-ДИЗАЙНОМ	295
Глава 8. Фреймовый Web-дизайн.....	297
Введение во фреймы	297
Фреймы и наборы фреймов	298
Схемы наборов фреймов	299
Работа с фреймами в Dreamweaver	303
Создание фреймов.....	303
Как формируются фреймы	309
Параметры фреймов и наборов фреймов.....	311
Создание содержимого фреймов	315
Заполнение фреймов.....	316

Использование цели гиперссылки для открытия страниц в заданном фрейме	318
Создание полосы навигации.....	319
Оптимизация фреймов.....	324
Уменьшение объема и сложности HTML-кода фреймов	324
Ускорение обработки фреймов	326
Недостатки фреймов и их преодоление	327
Что дальше?	328
Глава 9. Табличный Web-дизайн.....	329
Схемы табличного дизайна	329
Построение таблиц разметки	333
Использование полосы навигации на страницах, построенных на основе таблиц разметки	337
Недостатки табличного Web-дизайна	338
Недостатки таблиц	339
Недостатки табличного Web-дизайна	342
Что дальше?	343
Глава 10. Шаблоны.....	344
Введение в шаблоны Dreamweaver	344
Работа с шаблонами.....	345
Создание шаблона	345
Правка шаблона.....	348
Создание изменяемых областей.....	350
Создание Web-страниц на основе шаблонов	352
Правка шаблонов, на основе которых уже созданы Web-страницы.....	355
Применение шаблонов к уже созданным Web-страницам	356
Использование полосы навигации в шаблонах и основанных на них страницах	359
Открепление Web-страниц от шаблонов.....	360
Управление шаблонами в списке панели <i>Активы</i>	361
Дополнительные возможности шаблонов.....	362
Изменяемые атрибуты	362
Необязательные области.....	365
Повторяющиеся области	368
Необязательные изменяемые области	371
Табличная повторяющаяся область	372
Вложенные шаблоны	373
Недостатки шаблонов и их преодоление	374
Что дальше?	375
Глава 11. Работа со стилями CSS.....	376
Введение в каскадные таблицы стилей	376
Стили	376
Таблицы стилей	379

Правила каскадности и приоритет стилей	381
Псевдостили.....	383
Работа с таблицами стилей в Dreamweaver.....	384
Создание стилей	385
Определение стиля	388
Привязка стилей	406
Панель <i>Стили CSS</i>	407
Управление стилями	410
Работа с CSS-кодом таблиц стилей	414
Управление таблицами стилей.....	417
Вызов справки по CSS	419
Контейнеры	420
Что дальше?.....	422
ЧАСТЬ IV. ИСПОЛЬЗУЕМ НОВЕЙШИЕ ТЕХНОЛОГИИ.....	423
Глава 12. Плавающие контейнеры.....	425
Простейший случай контейнерного Web-дизайна	426
Атрибуты стиля CSS, предназначенные для управления контейнерами.....	428
Пример контейнерного Web-дизайна.....	431
Пример реализации "резиновых" контейнеров	436
Стандартные разметки Dreamweaver	444
Недостатки контейнерного Web-дизайна	445
Что дальше?.....	446
Глава 13. Свободно позиционируемые контейнеры	447
Введение в свободно позиционируемые контейнеры.....	447
Что такое свободно позиционируемый контейнер.....	448
Как создаются свободно позиционируемые контейнеры	449
Работа со свободно позиционируемыми контейнерами	453
Создание свободно позиционируемых контейнеров.....	454
Параметры свободно позиционируемых контейнеров	457
Работа с группой свободно позиционируемых контейнеров	459
Использование панели <i>Элементы AP</i>	461
Пример использования свободно позиционируемых контейнеров	463
Прямое задание параметров свободно позиционируемых контейнеров	468
Инструменты позиционирования Dreamweaver	470
Преобразование свободно позиционируемых контейнеров в таблицы разметки и обратно.....	476
Недостатки свободно позиционируемых контейнеров и их преодоление	478
Что дальше?	479
Глава 14. Использование Web-сценариев	480
Начала Web-программирования	480
Web-сценарии	481

События.....	483
Краткий курс языка JavaScript	484
Как пишутся Web-сценарии	496
Простейший Web-сценарий.....	497
Более сложный Web-сценарий.....	498
Web-сценарии — подход Dreamweaver. Поведения.....	500
Работа с поведением.....	502
Создание поведений.....	502
Поведения, поддерживаемые Dreamweaver	506
Написание своих Web-сценариев	530
Привязка Web-сценариев к фрагментам текста и произвольным тегам	532
Дополнительные инструменты Dreamweaver для работы со сценариями.....	533
Недостатки Web-сценариев и их преодоление	535
Альтернативные технологии. Апплеты Java.....	536
Что дальше?	539
Глава 15. Использование эффектов и компонентов Spry	540
Эффекты Spry	541
Эффект <i>Всплеск</i>	542
Эффект <i>Высвечивание</i>	542
Эффект <i>Жалюзи</i>	543
Эффект <i>Появление/Растворение</i>	545
Эффект <i>Расширить/Сжать</i>	546
Эффект <i>Скольжение</i>	548
Эффект <i>Тряска</i>	549
Компоненты Spry	550
Меню (<i>Панель меню Spry</i>).....	550
"Блокнот" (<i>Панель со вкладками Spry</i>).....	556
Связанные панели (<i>Набор вкладок Spry</i>)	560
Сворачиваемая панель (<i>Сворачивающаяся панель Spry</i>).....	563
Что дальше?	567
Глава 16. Метатеги и серверные директивы.....	568
Реклама в Интернете.....	568
Поисковые машины	568
Как работают поисковые агенты	572
Метатеги	573
Пассивная интернет-реклама	574
Работа с метатегами в среде Dreamweaver.....	576
Ключевые слова.....	577
Описание Web-страницы.....	578
Перезагрузка и перенаправление.....	579
Базовый интернет-адрес	580
Связи между Web-страницами.....	581
Специальные метатеги.....	584

Серверные директивы.....	585
Введение в серверные директивы.....	585
Вставка серверных директив в HTML-код	587
Стандартный набор серверных директив.....	587
Как использовать серверные директивы	589
Поддержка серверных включений в Dreamweaver	590
Что дальше?	591
ЧАСТЬ V. ПИШЕМ СЕРВЕРНЫЕ WEB-СТРАНИЦЫ.....	593
Глава 17. Введение в серверное программирование.....	595
Начала серверного программирования	595
Как работают серверные программы	595
Разновидности серверных программ.....	597
Как реализуется ввод данных. Web-формы	600
Как данные передаются по Сети.....	602
Серверное программирование — подход Dreamweaver. Серверные поведения	604
Какую технологию создания серверных страниц нам выбрать?.....	606
Введение в базы данных.....	607
Что дальше?.....	609
Глава 18. Web-формы.....	610
Работа с Web-формами в Dreamweaver	610
Создание Web-формы	611
Создание элементов управления.....	613
Основные принципы разработки Web-форм	631
Практикум по созданию Web-форм	633
Простейшая Web-форма	633
Использование таблиц и стилей для создания Web-форм	636
Список гиперссылок	638
Поведения, предназначенные для работы с Web-формами.....	640
Создание списка гиперссылок (<i>Меню переходов</i>)	641
Создание кнопки перехода для списка гиперссылок (<i>Выполнение меню переходов</i>).....	641
Задание нового значения поля ввода (<i>Задать текст текстового поля</i>).....	642
Проверка данных, введенных в форму (<i>Проверить форму</i>)	643
Компоненты Spry, предназначенные для работы с элементами управления.....	645
Проверка данных, введенных в поле ввода (<i>Текстовое поле проверки Spry</i>)	645
Проверка данных, введенных в область редактирования (<i>Текстовая область проверки Spry</i>)	651
Проверка данных, заданных с помощью флажков (<i>Флажок проверки Spry</i>)	654
Проверка данных, заданных в списке (<i>Выбор проверки Spry</i>)	658
Проверка данных, введенных в поле ввода пароля (<i>Пароль проверки Spry</i>)	661
Проверка подтверждения введенного пароля (<i>Подтверждение проверки Spry</i>)	664
Что дальше?.....	667

Глава 19. Простейшие серверные Web-страницы.....	668
Подготовка к созданию серверных Web-страниц	669
Установка соединения с базой данных	673
Создание источника данных ODBC.....	674
Регистрация базы данных в Dreamweaver	677
Создание серверных страниц в Dreamweaver	681
Страница добавления записи.....	681
Простейшая страница для просмотра данных	686
Страница для просмотра нескольких записей	699
Создание динамических списков	702
Создание сложных наборов записей.....	705
Динамические атрибуты	708
Передача данных другой Web-странице	710
Создание фильтров.....	711
Необязательные области серверной страницы	714
Что дальше?	717
Глава 20. Создание Web-сайтов, основанных на серверных Web-страницах.....	718
Принципы создания динамических сайтов	718
База данных нашего сайта	720
Административные страницы сайта	721
Как администрируются Web-сайты	721
Инструменты для работы со списком категорий.....	722
Инструменты для работы со списком статей.....	730
Средства разграничения доступа	734
Страницы общего доступа.....	742
Страница списка категорий	742
Страница списка статей	745
Страница регистрации нового посетителя	747
Реализация поиска статей.....	749
Что дальше?	750
ЧАСТЬ VI. ИСПОЛЬЗОВАНИЕ НАБОРОВ ДАННЫХ	751
Глава 21. Наборы данных HTML	753
Понятие набора данных	754
Работа с наборами данных HTML в Dreamweaver	756
Подключение набора данных HTML.....	758
Простые случаи вывода данных из наборов HTML.....	763
Более сложные случаи вывода данных из наборов HTML	769
Достоинства и недостатки наборов данных.....	780
Что дальше?	781

Глава 22. Наборы данных XML	782
Язык XML.....	782
Краткий курс языка XML	782
Наборы данных XML.....	784
Два способа обработки данных XML. Таблицы стилей XSL и XSLT	785
Три способа обработки данных XML, поддерживаемые Dreamweaver.....	786
Вывод данных XML с помощью компонентов Spry	788
Вывод данных XML средствами Web-обозревателя.....	792
Вывод данных XML средствами Web-сервера.....	798
Вызов справки по XML и XSLT	801
Заключение.....	803
Приложение. Описание компакт-диска.....	806
Предметный указатель	807

ГЛАВА 1



Введение в современные интернет-технологии

У читателей наверняка скопилось много вопросов. Что представляют собой те красивые Web-странички, которые выводит нам Web-обозреватель? Как они создаются? Какие программы нужны для этого? Наверно, все это очень сложно...

Стоп-стоп-стоп! Разумеется, автор ответит на все эти вопросы. И начнет он с самой что ни на есть голой теории...

Основные принципы работы Интернета

Говорят, в первой польской энциклопедии, изданной, кажется, в XVII столетии, термин "лошадь" описывался так: "что такое лошадь, знают все". То же самое можно сейчас сказать об Интернете. (Вот только можно ли сейчас сказать то же самое о лошади?..)

Но дать определение Интернету мы все-таки должны. Просто для того, чтобы понимать, о чем идет речь. А поняв это, можем приступить к изучению принципов, по которым этот самый Интернет работает.

Что такое Интернет

В самом деле, что такое *Интернет*? Всемирная компьютерная сеть. Ее, кстати, так часто и называют: Всемирная сеть, или даже просто Сеть с большой буквы. Протянутая по всему земному шару паутина медных проводов, волоконно-оптических линий и радиоканалов, связывающих друг с другом многочисленные компьютеры, — вот что такое Интернет. Разумеется, все здесь подчиняется общим стандартам (о которых мы поговорим далее) — иначе эта суперсеть просто не будет работать.

Если же быть совсем точным, то Интернет — это не единая сеть, а совокупность более мелких сетей, связанных друг с другом общими каналами и стандартами. Таких сетей превеликое множество: огромные территориальные сети, раскинувшиеся на целые области, штаты и государства, и ведомственные сети, объединяющие родственные организации, и локальные компьютерные сети отдельных организаций, и так называемые кампусные сети — сети, объединяющие компьютеры одного или нескольких близлежащих районов города. Благодаря проложенным между ними каналам высокоскоростной связи они составляют единое целое, имя которому Интернет.

Даже частные пользователи, подключающиеся к Интернету по модему, выделенной линии, радиоканалу или поддерживающему такую возможность сотовому телефону, тоже по сути дела являются частью Сети. Так что когда мы включаем наш модем и дозваниваемся до нашего *интернет-провайдера* (организации, предоставляющей доступ в Интернет), то приобщаемся к единому целому. А что, разве это не повод для законной гордости?

Сеть Интернет имеет одну замечательную особенность — она очень устойчива к сбоям. Так, если где-то порвется провод, мы этого не заметим. А все потому, что данные, которые мы запрашиваем, пойдут в этом случае по другому проводу. Специалисты говорят, что Интернет децентрализован — он не имеет единого центра, из которого ведется управление пересылкой данных, поэтому в случае аварии автоматически переконфигурируется и продолжает нормально работать.

Еще одна замечательная особенность Интернета — его глобальность, всемирность. Не вставая из-за компьютера, мы можем совершить путешествие по всему миру, побывать в США, Австралии, Германии, Танзании, на Огненной Земле и даже в Антарктиде! Для этого нужно всего лишь набрать нужный нам интернет-адрес.

Интернет имеет достаточно долгую и бурную историю. Он появился еще в конце 60-х годов XX века, когда Министерство обороны США финансировало проект создания компьютерной сети, устойчивой к сбоям. Разумеется, создавалась эта сеть для нужд обороны, да и название имела другое — *ARPANET*. Позднее, в начале 80-х, эта сеть отошла к ученым, а военные приступили к созданию другой сети, которой пользуются до сих пор. И в то же самое время ARPANET был переименован в *Internet*, или, если по-русски, Интернет.

Первоначально, еще во времена ARPANET, эта сеть использовалась для пересылки электронной почты и обмена файлами. Web-странички, ради которых мы, в основном, и путешествуем по Сети, появились только в конце 80-х. Именно тогда Интернет и "пошел в народ", перестав быть сетью ученых и превратившись в сеть для всех.

В Россию, точнее, в СССР, Интернет официально пришел в 1991 году, но популярность среди широких масс компьютерщиков приобрел только в середине 90-х. В настоящее же время в России, наверно, и не найти человека, не слышавшего об Интернете. Вы такого встречали? Автор — еще нет.

Сервисы Интернета

Раз уж мы заговорили об услугах, предоставляемых Интернетом, или, как говорят профессионалы, *сервисах* Интернета, то давайте узнаем о них побольше. В конце концов, нам ими пользоваться...

Самый старый и самый популярный до сих пор сервис Интернета — это электронная почта (e-mail). Ежедневно в мире отправляются и принимаются сотни миллионов электронных писем, и это количество в будущем будет только увеличиваться. В самом деле, электронная почта доступна, удобна, быстра и бесплатна, в отличие от почты "бумажной", которую пользователи Интернета уже успели презрительно прозвать "улиточной" (по-английски — snail mail). Конечно, эти доступность, удобство, быстрота и бесплатность имеют и некоторые недостатки, вроде спама — несанкционированных рекламных рассылок, но эти недостатки вполне можно стерпеть.

Еще один сервис Интернета, почти такой же старый, как почта, — это пересылка файлов. Пользователи Интернета называют его *FTP* (File Transfer Protocol, протокол передачи файлов; почему так — мы узнаем чуть позже). Сейчас FTP уже не имеет той популярности, как на заре существования Интернета, но все еще довольно часто используется. Так, все крупные корпорации — Microsoft, Adobe, Intel и др. — помимо Web-сайта, имеют и сервер FTP.

Третий сервис Интернета — это Всемирная паутина, или *WWW* (World Wide Web, повсеместно протянутая паутина), или просто *Web*, те самые Web-страницы и Web-сайты, которые мы просматриваем в Web-обозревателе. Появившийся значительно позже электронной почты и FTP, WWW стала самым популярным сервисом и, собственно, превратила Интернет из сети ученых в сеть для всех.

Об остальных сервисах Интернета (а их немало) мы только упомянем. Это потоковое вещание, интернет-пейджеры, чаты, нашумевшие в последние несколько лет файлообменные сети и некоторые другие, менее известные или устаревшие сервисы.

Клиенты и серверы

Но каким образом мы пользуемся всем тем богатством, что дает нам Всемирная сеть? С помощью особых программ! Это Web-обозреватель, клиент элек-

тронной почты, программа просмотра интернет-телевидения и прослушивания интернет-радио, интернет-пейджер и "чатилка". Все они очень хорошо нам знакомы.

Но программ, используемых для предоставления нам сервисов Интернета, гораздо больше. И очень многие из них нам, если так можно сказать, "не видны", т. е. мы не общаемся с ними напрямую. Вообще, существуют два совершенно разных вида интернет-программ. И сейчас мы о них поговорим.

Программы, относящиеся к первому виду, — это Web-обозреватели, клиенты электронной почты, чатов, интернет-пейджеры, в общем, все те, с которыми мы имеем дело непосредственно. Мы получаем с их помощью различную информацию из Сети и работаем с ней. Такие программы называются программами-клиентами, а компьютеры, на которых они работают, — наши с вами компьютеры! — клиентскими.

Да, но как программы-клиенты получают из Сети нужную нам информацию (Web-страницы, файлы, письма и пр.)? Очень просто — для этого они обращаются к другим программам, относящимся ко второму виду. Это программы-серверы, работающие на серверных компьютерах, где также хранится и запрашиваемая клиентами информация. Существуют Web-, FTP-серверы, серверы электронной почты, чата, интернет-пейджеров, потокового вещания и пр.

На заметку

Очень часто понятие "сервер" распространяется и на серверный компьютер, и на саму программу-сервер. Это, вообще-то, неправильно, т. к. на одном серверном компьютере может быть установлено несколько различных программ-серверов, но вошло в практику.

Процесс получения информации клиентами от сервера включает шесть шагов.

1. Пользователь запрашивает с помощью программы-клиента некую информацию, введя в нее интернет-адрес сервера. (Об интернет-адресах мы поговорим потом, а пока что будем знать, что это особый адрес, однозначно идентифицирующий нужную нам программу-сервер, работающую на определенном компьютере, который подключен к Интернету.)
2. Клиент устанавливает *соединение* (воображаемую линию связи) с сервером и посылает тому особый информационный блок, называемый *клиентским запросом*. Этот запрос содержит описание требуемых клиенту данных и должен быть определенным образом оформлен, чтобы сервер его понял.
3. Сервер принимает запрос и расшифровывает его.

4. Сервер извлекает запрошенный файл или фрагмент данных, записанных в файле, и посылает его клиенту в составе другого информационного блока — *серверного ответа*. Разумеется, этот ответ также должен быть составлен определенным образом. Если же запрашиваемые клиентом данные не были найдены, или сервер почему-то не смог понять клиентский запрос, он возвращает *сообщение об ошибке* — информационный блок, содержащий *код* (числовой номер) и, возможно, текстовое описание возникшей ошибки. Так, если Web-сервер не найдет запрошенную Web-обозревателем Web-страницу, он вернет сообщение об ошибке с кодом 404 — "запрошенная Web-страница не найдена".
5. Клиент получает ответ от сервера, расшифровывает его и выдает полученную информацию пользователю. Если получено сообщение об ошибке, клиент сообщает об этом пользователю либо предпринимает какие-то действия самостоятельно. Так, получив сообщение об ошибке с кодом 404, Web-обозреватель выведет соответствующее сообщение.
6. Клиент разрывает соединение с сервером.

Процесс отправки клиентом данных серверу также включает шесть шагов.

1. Пользователь вводит в программу-клиент информацию и интернет-адрес сервера, которому она должна быть отправлена.
2. Клиент устанавливает соединение с сервером и посылает тому отправляемую информацию в составе клиентского запроса. При этом отправляемая информация, как правило, особым образом кодируется.
3. Сервер принимает запрос, расшифровывает его и извлекает отправленную информацию.
4. Сервер записывает отправленную клиентом информацию в файл, помещает в базу данных или обрабатывает каким-то образом. В случае успешной записи или обработки он отправляет клиенту так называемое *подтверждение* — информационный блок, сообщающий о том, что все прошло нормально. Если у сервера возникли проблемы с приемом информации, он отправляет сообщение об ошибке.
5. Клиент получает ответ от сервера, расшифровывает его и уведомляет пользователя об успешной или неуспешной отправке данных либо предпринимает какие-то действия самостоятельно.
6. Клиент разрывает соединение с сервером.

Весь процесс "общения" клиента и сервера, начиная с отправки клиентом запроса и заканчивая принятием им ответа от сервера, называется *сеансом*. А соединение между клиентом и сервером, устанавливаемое на время этого сеанса и разрываемое после его окончания, называется *сеансовым*, или *временным*.

Любое соединение между клиентом и сервером устанавливается только клиентом. Сервер установить соединение с клиентом не может. Можно сказать, что серверу здесь отведена подчиненная роль.

Мы только что познакомились с особой *архитектурой* (принципом построения компьютерных систем), называемой *двухзвенной*, или архитектурой "*клиент-сервер*". Эта архитектура использует два вида программ — клиенты и серверы, — выполняющие разные роли. Она используется для реализации почти всех современных интернет-сервисов и пока что себя оправдывает.

На заметку

Некоторые интернет-сервисы, в частности файлообменные сети (BitTorrent и др.), используют другую архитектуру — *однозвенную*. Здесь все компьютеры, подключенные к Интернету и реализующие этот сервис, фактически равны между собой; любой из них может выступать в роли как клиентского (запрашивать информацию у других компьютеров), так и серверного (предоставлять хранящуюся на нем информацию другим компьютерам). Само собой, здесь используется особое программное обеспечение, которое может работать и как клиент, и как сервер.

В отличие от клиента, "имеющего дело" с одним-единственным пользователем, сервер работает сразу с множеством пользователей, причем одновременно. Сведения о соединениях, данные, пересылаемые клиентам и принимаемые от клиентов, — все это активно отнимает системные ресурсы компьютера, и чем больше соединений и данных проходят через сервер, тем больше требуется ресурсов. Поэтому на серверных компьютерах, как правило, не экономят.

Серверные компьютеры — настоящие монстры, содержащие несколько процессоров, дисковые массивы впечатляющей емкости, быстрые каналы связи с Интернетом и специальное программное обеспечение, способное "ворочать" огромными массивами данных. Все в них нацелено на то, чтобы обслужить как можно больше клиентов за минимальное время. Но часто, если клиентов и запросов оказывается слишком много, ресурсов серверного компьютера не хватает, и начинаются проблемы. Они могут проявляться в том, что сервер просто отказывается обслужить "лишних" клиентов, предлагая им подождать немного, когда нагрузка немного снизится, а то и в том, что могучий серверный компьютер просто-напросто "зависает". Такое тоже случается, и не так уж редко...

Ну да не будем о грустном! Не стоит начинать знакомство с таким притягательным миром интернет-технологий со столь печальных вещей, как системные сбои. Чем их меньше, и чем реже они случаются, тем лучше для всех нас.

Протоколы

Люди, чтобы понимать друг друга, должны разговаривать на одном языке. Точно так и с компьютерами, подключенными к сети, неважно какой — всемирной или локальной. Обмен данными по этим сетям должен проходить по единым стандартам, иначе начнется новое вавилонское столпотворение.

Стандарт, согласно которому организуются передаваемые по сети данные и команды, управляющие передачей этих данных, называется *протоколом*. В Интернете для обмена данными используются довольно много протоколов, и некоторые мы здесь вкратце рассмотрим.

Самый фундаментальный протокол Интернета — *IP* (Internet Protocol, межсетевой протокол). Он занимается тем, что разбивает подготовленные к передаче данные на порции (*пакеты*) определенной длины и определенного формата, помещает в каждую такую порцию интернет-адреса компьютера-отправителя и компьютера-получателя и предусматривает простейшие средства защиты от сбоев, которые могут возникнуть при пересылке данных.

Можно сказать, что протокол IP выполняет "грязную" работу по пересылке данных, работая на самом низком уровне. Поэтому его называют *протоколом низкого уровня*.

На IP базируется протокол *TCP* (Transfer Control Protocol, протокол управления передачей). Он обеспечивает гарантированную доставку данных, т. е. отвечает за то, чтобы все отправленные данные дошли до компьютера-получателя. Но это только первая из его обязанностей.

А вторая обязанность протокола TCP заключается в том, что он делит один реальный, физический, канал связи Интернета (кабель, волоконно-оптическую линию или радиоканал) на несколько воображаемых, виртуальных, "каналчиков", называемых *портами TCP*. Делается это для того, чтобы по одному физическому каналу можно было передавать сразу несколько потоков данных, принадлежащих разным программам, — для этого используются разные порты TCP. Всего таких портов предусмотрено 65 535, и все они пронумерованы.

TCP также относится к протоколам низкого уровня и так тесно связан с IP, часто эту парочку называют одним словом *TCP/IP*. А иногда даже считают за один протокол.

TCP/IP используется другими протоколами, уже *высокого уровня*. Эти протоколы описывают способы оформления клиентских запросов, серверных ответов, подтверждений и сообщений об ошибках, команды, пересылаемые клиентом серверу при запросе или передаче данных, и способ кодирования передаваемой информации.

На заметку

Строго говоря, существуют еще *протоколы физического уровня*, располагающиеся "ниже" даже IP. Они определяют электрические параметры сигнала, кабелей, разъемов и пр.

Каждый сервис Интернета использует свой собственный протокол высокого уровня, а то и несколько, предназначенных для разных задач или разработанных конкурирующими организациями. Давайте рассмотрим протоколы, с которыми мы столкнемся в будущем.

Начнем мы, конечно, с WWW. Для передачи данных Всемирная паутина использует протокол *HTTP* (HyperText Transfer Protocol, протокол передачи гипертекста). Он задает набор команд, отправляемых клиентом (Web-обозревателем) Web-серверу, и способы представления пересылаемых данных. Пожалуй, это самый широкоизвестный протокол Интернета — всем более-менее грамотным интернетчикам знакомы эти четыре буквы.

Сервис пересылки файлов FTP использует протокол, который так и называется — FTP. Он также определяет набор команд для управления файлами на сервере (загрузка с сервера, отправка на сервер, копирование, перемещение, удаление, создание папки на сервере и т. д.) и способы кодирования файлов для пересылки по каналам связи. В этом смысле протоколы HTTP и FTP весьма похожи.

А вот электронная почта использует целых два протокола. Первый протокол — *SMTP* (Simple Mail Transfer Protocol, простой протокол пересылки почты) — используется для пересылки почты клиентом серверу. Для получения же почты от сервера клиент общается с ним по протоколу *POP3* (Post-Office Protocol ver. 3, протокол почты версии 3).

Существует еще один почтовый протокол — *IMAP* (Internet Message Access Protocol, протокол доступа к почте Интернета). "Коллега" и "наследник" более старого POP3, он предоставляет больше возможностей, но распространен не так широко.

Чуть раньше мы узнали о портах TCP. Так вот, каждый существующий протокол высокого уровня использует для передачи данных свой собственный порт (так называемый *порт по умолчанию*). В табл. 1.1 перечислены некоторые протоколы и используемые ими порты по умолчанию.

Порт по умолчанию может быть изменен — такую возможность предоставляют все более-менее серьезные серверы. Так, Web-сервер может быть настроен так, чтобы использовать для "общения" с клиентами не 80-й порт, а, скажем, 8000-й. Это применяется, например, если на одном серверном компьютере работают два Web-сервера; тогда один из них настраивают на порт по умолчанию — 80-й, — а другой — да хотя бы и на 8000-й.

Таблица 1.1. Порты TCP, используемые по умолчанию для передачи данных некоторых протоколов высокого уровня

Протокол	Используемый порт TCP
FTP	21
HTTP	80
POP3	110
SMTP	25

Интернет-адреса

Теперь давайте поговорим о том, каким образом идентифицируются компьютеры, подключенные к Интернету. А именно — об интернет-адресах.

Интернет-адрес — это числовое или строковое значение, позволяющее точно идентифицировать компьютер в Сети. Именно такой интернет-адрес (точнее, два — отправителя и получателя) подставляется в каждый отправляемый по Сети пакет IP, чтобы он успешно дошел до места назначения.

На заметку

Существует, правда, возможность дать одному компьютеру сразу несколько интернет-адресов. Но используется это нечасто и в особых случаях. И в дальнейшем для простоты мы будем считать, что один интернет-адрес — это один компьютер.

На заре эпохи Интернета в качестве интернет-адреса использовался *IP-адрес* — числовое значение, идентифицирующее компьютер для протокола IP. IP-адрес замечательно подходит для компьютеров, но очень плохо — для людей. Вот пример интернет-адреса:

192.168.1.10

Не очень-то наглядно, правда? Именно поэтому с расширением Интернета была введена в строй новая система интернет-адресов, которой мы пользуемся до сих пор. Это так называемые доменные имена, о которых стоит поговорить подробно.

Но прежде чем мы начнем разговор о доменных именах, давайте выясним, что такое домен. *Домен*, или *доменная зона*, — это участок Интернета, созданный для удобства управления им. Такой участок может быть крупным, мелким или вообще состоять из одного компьютера. Каждому домену присваивается имя, состоящее из латинских букв и цифр; также могут быть использованы символы дефиса, подчеркивания и некоторые другие.

Структура доменов похожа на матрешку: мелкие домены "вложены" внутрь крупных, а крупные, в свою очередь, — внутрь гигантских. Гигантские домены называются *доменами верхнего уровня*, а вложенные в них более мелкие — *доменами нижнего уровня*.

Домены верхнего уровня бывают интернациональными и национальными. *Интернациональные домены* объединяют компьютеры по какому-то признаку; к ним относятся домены с именами com и biz (коммерческие организации), edu (образовательные), mil (военные), org (организации, не занимающиеся компьютерами и Интернетом), net (организации, занимающиеся компьютерами и Интернетом), travel (туристические организации) и некоторые другие. *Национальные домены* объединяют компьютеры по территориальному признаку и выдаются отдельным странам; это домены с именами us (США), uk (Великобритания), fr (Франция), de (Германия), ru (Россия) и др.

Домены нижнего уровня выдаются, как правило, отдельным организациям или, опять же, по территориальному признаку. Их текстовое обозначение часто совпадает с названием этой организации или района.

Если теперь записать обозначения всех доменов, в которых находится нужный нам компьютер, в порядке от более мелких к более крупным, разделив их точками, мы получим *доменное имя* этого компьютера. Так, если у нас сам компьютер имеет имя comp45, отдел, в котором он стоит, — buh (бухгалтерия), организация, включающая этот отдел, — office, а страна — ru (Россия), то мы получим такое доменное имя:

comp45.buh.office.ru

Согласитесь — запомнить это гораздо проще, чем невразумительный IP-адрес.

Да, но проблема в том, что протокол IP не понимает доменные имена! Что делать? Как преобразовать доменное имя в понятный ему IP-адрес?

Для этого используется особый сервис Интернета, называемый *DNS* (Domain Name System, система доменных имен). Клиент отправляет *серверу DNS* запрос, содержащий доменное имя, и получает в виде ответа IP-адрес, соответствующий этому доменному имени. А уж с IP-адресом он знает, что делать.

Такие серверы DNS имеются в каждом домене; кроме того, несколько самых мощных в мире серверов DNS (*корневые серверы DNS*) находятся как бы "выше" всех доменов, даже доменов верхнего уровня.

Всем хороши доменные имена, кроме одного, — они не позволяют задать номер порта TCP или хотя бы протокол. Они только задают сам серверный компьютер, а ведь на одном серверном компьютере могут работать несколько программ-серверов. Что делать? Просто указать перед доменным именем

обозначение протокола, реализуемого нужным сервером, вот так (обозначение протокола подчеркнуто):

http://comp45.buh.office.ru

ftp://comp45.buh.office.ru

В первом случае мы обращаемся к Web-серверу, а во втором — к серверу FTP, находящемуся на одном и том же компьютере **comp45.buh.office.ru**.

Также имеется возможность указать номер порта TCP, через который производится обмен данными. Номер порта записывается после доменного имени серверного компьютера через двоеточие, вот так (подчеркнуто):

http://comp45.buh.office.ru:8000

Многие серверы (почтовые, FTP и др.) требуют от пользователя ввода его имени и, возможно, пароля. Имя пользователя помещается между названием протокола и самим доменным именем и отделяется от последнего знаком амперсанда (@). Вот два примера задания имени пользователя в доменном имени сервера (подчеркнуто):

ftp://user@comp45.buh.office.ru

account@server.ru

Последний пример демонстрирует нам обычный адрес электронной почты. Заметим, что название протокола здесь не указывается — почтовый клиент и почтовый сервер сами знают, какой протокол использовать.

Ну а пароль пользователя помещается между именем и знаком @ и отделяется от имени двоеточием — вот так (подчеркнуто):

ftp://user:password@comp45.buh.office.ru

Ну вот, с основными принципами работы Интернета мы ознакомились. Теперь давайте сосредоточимся на WWW — в основном, именно этим сервисом мы будем пользоваться на протяжении всей книги.

Основные понятия WWW

Здесь мы узнаем все о Web-страницах и Web-сайтах, выясним, чем сайт отличается от страницы, поговорим о Web-обозревателях и Web-серверах и изучим множество новых терминов.

Web-страницы и Web-сайты

Что такое Web-страница? Ответить на этот вопрос могут многие. Это интернет-документ, предназначенный для распространения через Интернет по-

средством сервиса WWW. А если уж говорить по-простонародному, это то, что показывает в своем окне программа-клиент для просмотра Web-страниц — Web-обозреватель.

С технической точки зрения Web-страница — это обычный текстовый файл, который можно создать в любом текстовом редакторе, например Блокноте, стандартно поставляемом в составе Windows. Этот файл содержит собственно текст Web-страницы и команды форматирования этого самого текста. Команды форматирования называются *тегами*, а описывает их особый язык *HTML* (HyperText Markup Language, язык гипертекстовой разметки). Файл Web-страницы обязательно должен иметь расширение `htm[1]`.

А что такое Web-сайт? Это набор Web-страниц, подчиненных общей тематике и объединенных в единое целое (как — будет рассказано далее в этой книге). Как видим, сугубо технических отличий у Web-страницы и Web-сайта не слишком много.

Web-сайт сохраняется на жестких дисках серверного компьютера, на котором работает Web-сервер (серверная программа, обеспечивающая работу сервиса WWW), в виде набора различных файлов. Прежде всего, это, разумеется, файлы Web-страниц, составляющих сайт. Многие сайты включают файлы графических изображений, помещенных на страницы (почему изображения хранятся отдельно от самих страниц, мы узнаем потом). Также сайт может содержать файлы архивов и дистрибутивов программ и некоторые другие файлы, о которых мы тоже поговорим потом.

Зачастую различные файлы, составляющие сайт, хранятся в папках. Конечно, папки использовать необязательно, но так удобнее, особенно если файлов много и все они разных типов.

Для хранения всех файлов, составляющих сайт, на диске серверного компьютера создается особая папка, называемая *корневой*. Все файлы и папки сайта должны находиться только в этой папке, без малейших исключений.

Корневую папку сайта на серверном компьютере создает человек, занимающийся настройкой и обслуживанием программы Web-сервера (или же всего серверного компьютера), — *администратор*. При этом он заносит полный путь этой папки в настройки Web-сервера, чтобы последний "знал", где ее найти.

На заметку

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

Но как нам получить нужный файл (страницу, архив или дистрибутив) с Web-сайта? Правильно — для этого нужно указать Web-обозревателю интернет-адрес этого файла, введя его в специальное поле ввода. Web-обозреватель извлечет из введенного нами интернет-адреса путь к нужному файлу и отправит его Web-серверу, управляющему сайтом. Web-сервер получит этот путь, найдет корневую папку, отыщет в ней запрошенный файл и пришлет Web-обозревателю, т. е. нам.

Предположим, мы ввели в Web-обозреватель вот такой интернет-адрес:

`http://www.somesite.ru/somepage.html`

В этом случае Web-обозреватель сразу выделит из него интернет-адрес Web-сервера

`http://www.somesite.ru`

и путь к запрошенному нами файлу

`/somepage.html`

Как мы видим, в начале пути стоит символ слэша (/). Он обозначает корневую папку сайта. Именно в корневой папке Web-сервер будет искать файл `somepage.html`.

Далее Web-обозреватель отправит Web-серверу **`http://www.somesite.ru`** такой запрос:

`/somepage.html`

то есть путь к нужному нам файлу. Web-сервер, получив этот запрос, найдет в корневой папке файл `somepage.html`, загрузит его и отправит Web-обозревателю. Если же такого файла нет или Web-сервер почему-то не сможет его загрузить, он отправит Web-обозревателю сообщение об ошибке.

Мы уже знаем, что любой пакет IP содержит в себе, кроме всего прочего, интернет-адрес отправителя. Кроме того, интернет-адрес клиентского компьютера посылается в составе клиентского запроса HTTP. Так что Web-сервер всегда сможет узнать, куда ему отправить запрошенный файл.

Если же мы наберем в Web-обозревателе вот такой интернет-адрес:

`http://www.somesite.ru/download/archive.zip`

Web-обозреватель выделит из него такой путь:

`/download/archive.zip`

Сейчас мы запросили архивный файл `archive.zip`, находящийся в папке `download`, вложенной в корневую папку сайта. Web-обозреватель pošлет Web-серверу **`http://www.somesite.ru`** вот такой запрос:

`/download/archive.zip`