

ВЛАДИМИР ДРОНОВ



# PHP, MySQL и Dreamweaver

## разработка интерактивных Web-сайтов

### + ВИДЕОКУРС



НАЧАЛА WEB-ДИЗАЙНА

ВВЕДЕНИЕ В БАЗЫ ДАННЫХ

НАПИСАНИЕ  
WEB-ПРИЛОЖЕНИЙ  
НА ЯЗЫКЕ PHP

СОЗДАНИЕ СЕРВЕРНЫХ  
СТРАНИЦ СРЕДСТВАМИ  
DREAMWEAVER

ОБУЧЕНИЕ  
WEB-ПРОГРАММИРОВАНИЮ  
НА ПРИМЕРАХ

**PRO**

ПРОФЕССИОНАЛЬНОЕ  
ПРОГРАММИРОВАНИЕ

+ CD

УДК 681.3.06  
ББК 32.973.26-018.2  
Д75

**Дронов В. А.**

Д75 PHP, MySQL и Dreamweaver. Разработка интерактивных Web-сайтов. — СПб.: БХВ-Петербург, 2007. — 480 с.: ил.  
+ Videокурс (на CD-ROM) — (Профессиональное программирование)  
ISBN 978-5-9775-0125-5

Рассмотрены приемы разработки на языке PHP интерактивных Web-сайтов, извлекающих данные из баз MySQL. Для написания простейших страниц используется популярный программный пакет визуального Web-редактора Dreamweaver, попутно дается краткое описание языка HTML. Приводится сжатое описание языка PHP, а также подробно разбираются все сценарии PHP, созданные Dreamweaver, и анализируется их работа. Изложены принципы написания специализированных Web-страниц без использования Dreamweaver. В результате читатель создает полностью работоспособный и весьма развитый блог — интернет-дневник. Прилагаемый CD содержит видеокурс по основам работы в Adobe Dreamweaver CS3.

*Для 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>Ирина Иноземцева</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 28.08.07.

Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 38,7.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию  
№ 77.99.02.953.Д.006421.11.04 от 11.11.2004 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 978-5-9775-0125-5

© Дронов В. А., 2007  
© Оформление, издательство "БХВ-Петербург", 2007

# Оглавление

|   |           |
|---|-----------|
| <b>Введение</b> .....                                     | <b>11</b> |
| О чем вообще идет речь?.....                              | 11        |
| Dreamweaver: "Делай, как я!" .....                        | 12        |
| Типографские соглашения .....                             | 13        |
| Благодарности .....                                       | 14        |
| <br>  |           |
| <b>ЧАСТЬ I. ОСНОВЫ WEB-ДИЗАЙНА</b> .....                  | <b>17</b> |
| <br>  |           |
| <b>Глава 1. Современные интернет-технологии</b> .....     | <b>19</b> |
| Принципы работы Интернета .....                           | 19        |
| Что такое Интернет. Сервисы Интернета .....               | 19        |
| Клиенты и серверы.....                                    | 22        |
| Протоколы .....   | 25        |
| Интернет-адреса .....                                     | 28        |
| Базовые понятия WWW .....                                 | 30        |
| Web-страницы и Web-сайты.....                             | 30        |
| Web-обозреватели .....                                    | 33        |
| Web-серверы.....  | 35        |
| Публикация Web-сайта в Интернете. Хостинг-провайдеры..... | 36        |
| Что дальше? .....   | 37        |
| <br>  |           |
| <b>Глава 2. HTML — язык написания Web-страниц</b> .....   | <b>39</b> |
| Введение в язык HTML .....                                | 39        |
| Теги HTML. Форматирование текста.....                     | 40        |
| Графика на Web-страницах. Внедренные элементы .....       | 45        |
| Гиперссылки .....   | 48        |
| Интернет-адреса .....                                     | 50        |
| Правильно оформленные Web-страницы.....                   | 52        |
| Иерархия тегов HTML .....                                 | 53        |
| Кодирование текста. Проблема русских кодировок .....      | 54        |

|  |            |
|--|------------|
| Начала сайтостроения.....  | 56         |
| Планирование Web-сайта .....   | 56         |
| Логическая структура Web-сайта .....   | 58         |
| Проектируем наш первый Web-сайт .....  | 60         |
| Что дальше? .....  | 62         |
| <b>Глава 3. Adobe Dreamweaver — пакет для создания Web-страниц<br/>и Web-сайтов.....</b> | <b>63</b>  |
| Предварительная настройка Dreamweaver .....  | 64         |
| Основы работы в Dreamweaver .....  | 68         |
| Создание новой Web-страницы .....  | 68         |
| Набор текста .....   | 69         |
| Форматирование фрагментов текста .....   | 71         |
| Форматирование абзацев.....  | 76         |
| Специальные символы и нетекстовые элементы.....  | 78         |
| Таблицы .....  | 82         |
| Создание таблиц.....   | 83         |
| Работа с таблицей.....   | 86         |
| Формирование таблиц .....  | 87         |
| Объединение ячеек.....   | 89         |
| Вставка графических изображений .....  | 91         |
| Создание гиперссылок.....  | 94         |
| Предварительный просмотр Web-страниц .....   | 97         |
| Вызов справки .....  | 97         |
| Что дальше? .....  | 99         |
| <b>Глава 4. CSS — язык оформления Web-страниц.....</b>                                   | <b>101</b> |
| Введение в CSS .....   | 102        |
| Создание стилей CSS .....  | 102        |
| Три способа задания стилей.....  | 105        |
| Контейнеры.....  | 106        |
| Почему "каскадные"?.....   | 107        |
| Псевдостили.....   | 109        |
| Работа со стилями в Dreamweaver .....  | 110        |
| Вызов справочника по CSS .....   | 119        |
| Что дальше? .....  | 120        |
| <b>Глава 5. Работа с Web-сайтом в Dreamweaver .....</b>                                  | <b>121</b> |
| Подготовка к публикации сайта.....   | 122        |
| Регистрация сайта в Dreamweaver .....  | 122        |
| Работа с файлами сайта. Панель <i>Files</i> .....  | 125        |
| Проверка Web-страниц .....   | 129        |
| Проверка правильности HTML-кода .....  | 129        |
| Проверка гиперссылок .....   | 131        |
| Взаимодействие панели <i>Files</i> и окна документа.....                                 | 133        |

---

|  |            |
|--|------------|
| Публикация сайта.....  | 133        |
| Публикация сайта на локальном Web-сервере .....                        | 134        |
| Публикация сайта на удаленном Web-сервере .....                        | 138        |
| Использование протокола FTP .....                                      | 138        |
| Настройка Dreamweaver для публикации сайта по FTP .....                | 139        |
| Публикация сайта по протоколу FTP.....                                 | 143        |
| Что дальше?.....   | 144        |
| <b>ЧАСТЬ II. ОСНОВЫ WEB-ПРОГРАММИРОВАНИЯ .....</b>                     | <b>145</b> |
| <b>Глава 6. Принципы Web-программирования .....</b>                    | <b>147</b> |
| Недостатки статических Web-страниц и их преодоление .....              | 147        |
| Данные и их представление.....   | 148        |
| Недостатки статических Web-страниц.....                                | 149        |
| Серверные программы — способ отделить информацию от представления..... | 150        |
| Технологии создания серверных программ.....                            | 152        |
| Серверные Web-страницы .....   | 152        |
| Другие технологии серверного программирования .....                    | 155        |
| Второй Web-сайт. Использование серверных страниц.....                  | 156        |
| Что дальше?.....   | 156        |
| <b>Глава 7. Базы данных.....</b>                                       | <b>157</b> |
| Введение в реляционные базы данных.....                                | 157        |
| Что такое реляционные базы данных .....                                | 157        |
| Составные части реляционной базы данных .....                          | 158        |
| Таблицы, поля и записи.....  | 159        |
| Правила.....   | 161        |
| Индексы и ключи.....   | 162        |
| Связи.....   | 166        |
| Настольные и серверные реляционные СУБД.....                           | 168        |
| Язык обработки данных SQL .....  | 171        |
| Зачем нужен SQL .....  | 171        |
| Выборка данных.....  | 172        |
| Простейшие запросы выборки данных.....                                 | 172        |
| Сортировка данных .....  | 174        |
| Фильтрация данных.....   | 175        |
| Задание связей между таблицами.....                                    | 177        |
| Псевдонимы полей .....   | 179        |
| Агрегатные функции SQL.....  | 179        |
| Изменение данных .....   | 181        |
| Добавление записи.....   | 181        |
| Изменение записи .....   | 182        |
| Удаление записи .....  | 183        |
| Другие запросы SQL .....   | 183        |

|  |            |
|--|------------|
| Разграничение доступа. Права.....                          | 184        |
| Сервер данных MySQL и его возможности.....                 | 186        |
| Создаем базу данных для нашего сайта.....                  | 189        |
| Что дальше?.....   | 191        |
| <b>Глава 8. Краткий курс языка PHP.....</b>                | <b>193</b> |
| Основные понятия PHP.....                                  | 193        |
| Написание сценариев PHP.....                               | 194        |
| Операторы, аргументы и выражения.....                      | 196        |
| Переменные.....  | 197        |
| Типы данных.....   | 199        |
| Логический.....  | 199        |
| Целочисленный.....   | 199        |
| С плавающей точкой.....                                    | 200        |
| Строковый.....   | 200        |
| NULL.....  | 201        |
| Операторы.....   | 202        |
| Арифметические.....  | 202        |
| Оператор объединения строк.....                            | 203        |
| Операторы присваивания.....                                | 203        |
| Операторы сравнения.....                                   | 204        |
| Логические операторы.....                                  | 205        |
| Вычисление выражений, содержащих логические операторы..... | 206        |
| Совместимость и преобразование типов данных.....           | 207        |
| Приоритет операторов.....                                  | 209        |
| Сложные выражения PHP.....                                 | 210        |
| Блоки.....   | 211        |
| Условные выражения.....                                    | 211        |
| Выражения выбора.....                                      | 213        |
| Циклы.....   | 214        |
| Цикл со счетчиком.....                                     | 215        |
| Цикл с постусловием.....                                   | 216        |
| Цикл с предусловием.....                                   | 217        |
| Прерывание цикла.....                                      | 217        |
| Функции.....   | 218        |
| Создание функций.....                                      | 218        |
| Вызов функций.....   | 220        |
| Использование переменных внутри тела функции.....          | 221        |
| Встроенные функции PHP.....                                | 222        |
| Массивы.....   | 223        |
| Создание массивов и работа с ними.....                     | 223        |
| Цикл просмотра.....  | 225        |
| Константы.....   | 226        |
| Комментарии.....   | 227        |
| Что дальше?.....   | 228        |

|   |            |
|---|------------|
| <b>Глава 9. Простейший вывод данных .....</b>                   | <b>229</b> |
| Подготовка к созданию серверных страниц .....                   | 229        |
| Регистрация базы данных в Dreamweaver .....                     | 232        |
| Создание простейших серверных страниц .....                     | 238        |
| Создание набора записей .....                                   | 238        |
| Создание серверной страницы .....                               | 242        |
| Разбор сценариев PHP, выводящих данные из базы .....            | 246        |
| Передача данных между серверными страницами .....               | 249        |
| Метод передачи данных GET .....                                 | 250        |
| Создание Web-страниц, передающих данные друг другу .....        | 251        |
| Разбор сценариев PHP, принимающих и обрабатывающих данные ..... | 254        |
| Более сложные серверные страницы .....                          | 258        |
| Реализация постраничного вывода записей .....                   | 258        |
| Создание навигатора .....                                       | 259        |
| Вывод сведений о наборе записей .....                           | 260        |
| Разбор кода, реализующего постраничный просмотр .....           | 261        |
| Вывод элементов Web-страницы в зависимости от условия .....     | 266        |
| Реализация возврата на нужную страницу .....                    | 268        |
| Что дальше? .....   | 269        |
| <br>  |            |
| <b>Глава 10. Ввод и правка данных .....</b>                     | <b>271</b> |
| Реализация ввода и передачи данных .....                        | 271        |
| Ввод данных. Формы .....  | 272        |
| Кодирование данных .....  | 274        |
| Передача данных .....   | 275        |
| Простые серверные Web-страницы для ввода и правки данных .....  | 277        |
| Страница для добавления записи .....                            | 277        |
| Разбор сценариев PHP, добавляющих запись .....                  | 287        |
| Страница для правки заметки .....                               | 291        |
| Разбор сценариев PHP, изменяющих запись .....                   | 296        |
| Страница для удаления записи .....                              | 297        |
| Страница для работы с комментариями .....                       | 300        |
| Более сложные Web-страницы для ввода и правки данных .....      | 302        |
| Что дальше? .....   | 311        |
| <br>  |            |
| <b>Глава 11. Более сложный вывод данных .....</b>               | <b>313</b> |
| Правильный вывод значений даты .....                            | 313        |
| Особые случаи вывода элементов Web-страницы .....               | 314        |
| Создание сложных наборов записей .....                          | 317        |
| Создание страницы статистики .....                              | 319        |
| Реализация поиска .....   | 325        |
| Что дальше? .....   | 330        |

|  |            |
|--|------------|
| <b>ЧАСТЬ III. БЕЗОПАСНОСТЬ И ЦЕЛОСТНОСТЬ ДАННЫХ.....</b>             | <b>331</b> |
| <b>Глава 12. Введение в безопасность и целостность данных.....</b>   | <b>333</b> |
| Безопасность и разграничение доступа.....                            | 333        |
| Целостность данных.....  | 335        |
| Что дальше?.....   | 337        |
| <b>Глава 13. Разграничение доступа.....</b>                          | <b>339</b> |
| Создание таблицы списка пользователей.....                           | 340        |
| Создание страницы входа на сайт.....                                 | 341        |
| Процесс создания страницы входа на сайт в Dreamweaver.....           | 341        |
| Сессии. Переменные уровня сессии.....                                | 344        |
| Разбор кода PHP, выполняющего вход.....                              | 347        |
| Разграничение доступа к Web-страницам.....                           | 350        |
| Процесс разграничения доступа к страницам в Dreamweaver.....         | 350        |
| Разбор кода PHP, выполняющего разграничение доступа.....             | 352        |
| Создание страницы выхода с сайта.....                                | 356        |
| Процесс создания страницы выхода с сайта в Dreamweaver.....          | 356        |
| Разбор кода PHP, выполняющего выход.....                             | 357        |
| Создание административных страниц для управления пользователями..... | 359        |
| Разграничение доступа к фрагментам Web-страниц.....                  | 361        |
| Что дальше?.....   | 365        |
| <b>Глава 14. Обеспечение ссылочной целостности данных.....</b>       | <b>367</b> |
| Простой способ обеспечения ссылочной целостности.....                | 368        |
| Сложный способ обеспечения ссылочной целостности.....                | 369        |
| Недостаток простого способа и попытка его устранить.....             | 369        |
| Блокировка таблиц MySQL и ее использование.....                      | 372        |
| Реализация сложного способа.....                                     | 373        |
| Каскадное удаление записей.....                                      | 374        |
| Что дальше?.....   | 375        |
| <b>ЧАСТЬ IV. ПОСЛЕДНИЕ ШТРИХИ.....</b>                               | <b>377</b> |
| <b>Глава 15. Обработка текста средствами PHP.....</b>                | <b>379</b> |
| Разбиение текста заметки на абзацы.....                              | 379        |
| Форматирование текста.....   | 382        |
| Недопустимость HTML-форматирования в блогах. Внутренние теги.....    | 382        |
| Реализация форматирования текста.....                                | 385        |
| Вставка графических изображений и гиперссылок.....                   | 386        |
| Что дальше?.....   | 391        |
| <b>Глава 16. Управление файлами через Web-интерфейс.....</b>         | <b>393</b> |
| Просмотр содержимого папки.....                                      | 394        |

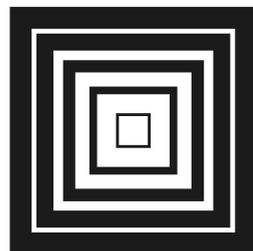
---

|  |            |
|--|------------|
| Отправка файлов на Web-сайт .....  | 398        |
| Как отправить файл из Web-обозревателя .....                                       | 399        |
| Как принять отправленный файл .....  | 400        |
| Реализация отправки файла .....  | 403        |
| Удаление файлов .....  | 405        |
| Что дальше? .....  | 407        |
| <b>Глава 17. Хранение данных на стороне клиента .....</b>                          | <b>409</b> |
| Задание цветовой гаммы сайта .....   | 410        |
| Хранение настроек посетителя .....   | 412        |
| Способы хранения настроек .....  | 412        |
| Cookie и их использование .....  | 414        |
| Реализация хранения настроек в cookie .....  | 416        |
| Какие данные стоит хранить в cookie .....  | 418        |
| <b>Заключение .....</b>  | <b>421</b> |
| <b>ПРИЛОЖЕНИЯ .....</b>  | <b>425</b> |
| <b>Приложение 1. Установка Web-сервера Apache .....</b>                            | <b>427</b> |
| Установка .....  | 427        |
| Запуск и остановка .....   | 432        |
| Настройка .....  | 433        |
| Доступ к документации по Apache .....  | 434        |
| <b>Приложение 2. Установка сервера данных MySQL .....</b>                          | <b>435</b> |
| Установка .....  | 435        |
| Настройка .....  | 440        |
| Запуск и остановка .....   | 441        |
| Запуск и остановка под Windows 95, 98 и Me .....                                   | 441        |
| Запуск и остановка под Windows NT .....  | 442        |
| Запуск и остановка под Windows 2000, XP, 2003, Vista .....                         | 443        |
| Доступ к документации по MySQL .....   | 443        |
| <b>Приложение 3. Установка платформы PHP .....</b>                                 | <b>445</b> |
| Установка .....  | 445        |
| Настройка .....  | 446        |
| Запуск и остановка .....   | 448        |
| Доступ к документации по PHP .....   | 448        |
| <b>Приложение 4. Установка и использование клиента данных<br/>phpMyAdmin .....</b> | <b>451</b> |
| Установка и настройка .....  | 451        |
| Использование .....  | 452        |
| Вход .....   | 452        |

---

|  |            |
|--|------------|
| Создание базы данных .....                                   | 454        |
| Создание таблиц .....  | 455        |
| Создание полей .....   | 455        |
| Создание индексов .....                                      | 458        |
| Правка и удаление полей, индексов, таблиц и баз данных ..... | 459        |
| Правка и удаление полей .....                                | 459        |
| Правка и удаление индексов .....                             | 460        |
| Правка и удаление таблиц .....                               | 460        |
| Правка и удаление баз данных .....                           | 461        |
| Управление пользователями .....                              | 461        |
| Средства управления пользователями phpMyAdmin .....          | 461        |
| Создание пользователя .....                                  | 462        |
| Правка и удаление пользователей .....                        | 466        |
| Работы с данными .....                                       | 467        |
| Выход .....  | 468        |
| Доступ к документации по phpMyAdmin .....                    | 468        |
| <b>Предметный указатель .....</b>                            | <b>469</b> |

# ГЛАВА 1



## Современные интернет-технологии

Да, без теории не обойтись, ведь прежде чем приниматься что-то делать, нужно выяснить, зачем, как и почему именно так это "что-то" делается. А иначе у нас ничего толкового не выйдет. Так что давайте выключим компьютер — пусть отдохнет! — и почитаем.

### Принципы работы Интернета

Сначала мы поговорим о том, что такое Интернет и как он работает — рассмотрим некоторые общие вопросы.

### Что такое Интернет. Сервисы Интернета

И первый же вопрос, на который нам нужно получить ответ, — это что, собственно, такое Интернет и что он может нам дать. А разбором принципов его работы мы займемся потом.

Итак, *Интернет* — это Всемирная компьютерная сеть. (Конечно, это всем известно, но ведь автор должен дать определение.) 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 это подчиняется общим стандартам (о которых мы поговорим далее), а 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 через аналоговый или цифровой модем, по выделенной линии 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 общаемся 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, соединяющий два участка (или, как говорят профессионалы-сетевики, сегмента) Сети, мы этого не заметим. А все потому, что данные, которые мы запрашиваем, пойдут в этом случае по другому каналу. Специалисты говорят, что Интернет децентрализован — он не имеет единого центра, из которого ведется управление пересылкой данных, поэтому в случае аварии автоматически переконфигурируется и продолжит нормально работать.

Еще одна замечательная особенность Интернета — его, так сказать, глобальность. Не вставая из-за компьютера, мы можем совершить путешествие по всему миру: побывать в США, Австралии, Германии, Зимбабве, Огненной Земле и даже в Антарктиде (да, и туда протянулись вездесущие провода!). Для этого нужно всего лишь набрать нужный нам интернет-адрес.

Итак, что такое Интернет, мы выяснили. Теперь совершим небольшое путешествие в прошлое и посмотрим, как все начиналось.

Интернет имеет достаточно долгую историю. Он появился еще в первой половине 70-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 в ведение ученых, а военные приступили к созданию другой сети, которой пользуются до сих пор. И в то же самое время ARPANET был переименован в Internet, или, по-русски, Интернет.

Первоначально, еще во времена ARPANET, эта сеть использовалась для пересылки электронной почты и файлов. 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 Интернет официально пришел в 1991 году, но популярность среди широких масс компьютерщиков приобрел только в середине 90-х годов XX века. В настоящее время в России количество пользователей Интернета исчисляется десятками миллионов.

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...

Итак, 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" (от англ. *"snail mail"*). Конечно, доступность, удобство, быстрота и бесплатность имеют оборотную сторону, вроде "спама" — несанкционированных рекламных рассылок, но с ними вполне можно бороться.

Еще один сервис Интернета, почти такой же старый, как почта, — это пересылка файлов. Он называется *FTP* (File Transfer Protocol, протокол передачи файлов). Сейчас FTP отошел на второй план, уступив место более новым сервисам, но все еще весьма активно используется.

Третий сервис Интернета, который мы подробно рассмотрим, — это Всемирная паутина, или *WWW* (World Wide Web), или просто *Web*. Это и есть те самые Web-страницы и Web-сайты, которые мы просматриваем в *Web-обозревателе* (программе для просмотра Web-страниц). Пожалуй, это самый впечатляющий и самый востребованный сервис, собственно, и приведший к тому, что Интернет и "пошел в народ". Вот о нем-то и пойдет речь в этой книге.

Осталось упомянуть о нескольких сервисах Интернета, уже (или еще) не имеющих такой широкой популярности.

- ◆ *Новости*, или *USENET*. Чем-то этот сервис похож на электронную почту: пользователь пишет письмо в так называемую *группу новостей* — своего рода электронную доску объявлений. Другие пользователи читают это письмо и пишут ответы, которые помещаются на эту же самую "доску". Совокупность первоначального письма и ответов на него образуют *обсуждение* (по-англ. — *thread*), участие в котором может принять любой пользователь, подписавшийся на эту группу новостей. Когда-то сервис новостей был весьма популярен, но сейчас, после появления различных Web-форумов, прозябает на задворках Интернета.
- ◆ *Потоковое вещание*. Это своего рода теле- и радиовещание через Интернет появилось совсем недавно, несколько лет назад, но быстро набирает

популярность. К сожалению, чтобы слушать интернет-радио, и в особенности смотреть интернет-телевидение, нужен достаточно быстрый канал связи, но такие каналы сейчас есть у многих, даже домашних пользователей Интернета.

- ◆ *Интернет-пейджеры*. Этот сервис также похож на электронную почту: пользователи пересылают друг другу короткие "записки" по аналогии с обычным пейджером. Интернет-пейджеры работают, как правило, быстрее, чем обычная электронная почта, и временами создают иллюзию непосредственного общения. В качестве примера можно вспомнить популярнейший ICQ и его менее известных "коллег": Miranda, Odigo и пр.
- ◆ *Чаты* (от англ. *chat* — "болтовня"). Это своего рода "разговор" через Интернет, еще более напоминающий непосредственное общение. Пользователь набирает на клавиатуре текст, который в мгновение ока пересылается его собеседнику или собеседникам. По популярности чаты превосходят интернет-пейджеры и приближаются к WWW.

Ну, вот и все. Совсем уж малоизвестные и узкоспециализированные сервисы Интернета мы рассматривать не будем. В конце концов, сведения о них (как и о многом другом) можно найти в том же самом Интернете. А тема этой книги совсем иная.

## Клиенты и серверы

Продолжим наше путешествие в электронные дебри Интернета. На этот раз речь пойдет о двух разновидностях программ, с помощью которых предоставляются интернет-услуги.

В самом деле, каким образом мы пользуемся всем тем богатством, что дает нам Всемирная сеть? С помощью особых программ! Такие программы делятся на две принципиально разные категории, и мы сейчас о них поговорим.

Программы, относящиеся к первой категории, взаимодействуют непосредственно с пользователями Интернета и помогают им получать различную информацию: электронные письма, Web-страницы, сообщения интернет-пейджеров, чатов и пр. Это Web-обозреватели, клиенты электронной почты, чатов, интернет-пейджеры — все те, с которыми мы имеем дело на своих компьютерах. Такие программы называются программами-клиентами (а компьютеры, на которых они работают, — наши компьютеры! — клиентскими).

Информация, с которой мы работаем посредством программ-клиентов, все эти Web-сайты, письма, звуковые и видеофайлы, хранится на других компьютерах — серверных. За выдачу ее клиентским программам, а значит, и нам, отвечают программы, относящиеся ко второй категории, — серверы. Для каждого сервиса (и для каждой соответствующей им разновидности программ-

клиентов) Интернета существует свой класс серверов: Web-серверы, серверы электронной почты, чата, интернет-пейджеров, потокового вещания и пр.

### Замечание

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

Теперь поговорим подробнее о том, как же клиенты взаимодействуют с серверами. Причем процессы приема и отправки данных мы рассмотрим отдельно.

Процесс получения информации от сервера клиентами состоит из пяти шагов.

1. Пользователь запрашивает с помощью программы-клиента некую информацию.
2. Клиент устанавливает *соединение* (своего рода воображаемый канал связи) с сервером и посылает тому особый информационный блок, называемый *клиентским запросом*. Структура этого запроса жестко стандартизирована, чтобы сервер его понял.
3. Сервер принимает запрос и расшифровывает его.
4. Сервер извлекает нужный клиенту файл или фрагмент данных, записанных в файле, и посылает его клиенту в виде другого информационного блока — *серверного ответа*. Если же нужных данных нет, или сервер почему-то не смог обработать клиентский запрос, он возвращает в составе ответа *сообщение об ошибке* — особый информационный блок, содержащий описание возникшей ошибки. Разумеется, и серверный ответ, и сообщение об ошибке также жестко стандартизированы.
5. Клиент получает ответ от сервера, расшифровывает его и выдает полученную информацию пользователю. Если получено сообщение об ошибке, клиент уведомляет об этом пользователя либо предпринимает какие-то действия самостоятельно. После принятия ответа от сервера клиент разрывает соединение с ним.

Процесс отправки клиентом данных серверу также состоит из пяти шагов.

1. Пользователь каким-то образом передает программе-клиенту отправляемую информацию.
2. Клиент устанавливает соединение с сервером и посылает тому отправляемую информацию в составе клиентского запроса. При этом отправляемая информация, как правило, особым образом шифруется.

3. Сервер принимает запрос, расшифровывает его и извлекает отправленную информацию.
4. Сервер записывает отправленную клиентом информацию в файл или каким-то образом ее обрабатывает. После этого в случае успешной записи он отправляет клиенту в составе ответа так называемое *подтверждение* — особый информационный блок, сообщающий о том, что все прошло нормально. В случае неудачи отправляется 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. После принятия ответа от сервера клиент разрывает соединение с ним.

Весь процесс "общения" клиента и сервера, начиная с отправки клиентом запроса и заканчивая принятием им ответа от сервера, называется *сеансом*.

Ранее было сказано, что перед отправкой клиентом запроса серверу, т. е. перед началом сеанса, клиент должен установить соединение с сервером. Так вот, это соединение существует ровно столько времени, сколько требуется для сеанса, и поэтому называется *сеансовым*, или *временным*.

Каждое соединение требует компьютерных ресурсов. Серверу нужно хранить в памяти сведения о клиенте, установившем это соединение, а при большом количестве таких соединений (а на сильно загруженных корпоративных серверах так часто и бывает) памяти расходуется очень много. Поэтому сеансовые соединения имеют большое преимущество — они длятся очень недолго, ровно столько, сколько нужно для отправки серверного ответа, после чего разрываются, и отведенная им компьютерная память автоматически освобождается.

Но сеансовые соединения имеют огромный недостаток — с их помощью начать сеанс обмена данными может только клиент. А в случае, например, чата или интернет-пейджера серверу самому может понадобиться начать сеанс, чтобы отправить новое сообщение "отдыхающему" клиенту. В этом случае используется *постоянное*, или *многосеансовое*, подключение с другим сценарием работы.

1. Клиент устанавливает постоянное соединение с сервером.
2. Клиент и сервер ожидают прихода отправляемых данных.
3. Если данные пришли, клиент или сервер выполняют сеанс обмена.
4. Если клиент еще не разорвал соединение, выполняется переход к пункту 2.

Здесь нужно обратить внимание на то, что соединение устанавливается и завершается только клиентом. Сервер установить соединение с клиентом не может.

В отличие от клиента, "имеющего дело" с одним-единственным пользователем, сервер работает сразу с множеством пользователей, причем одновременно. Данные о соединениях, данные, пересылаемые клиентам и принимаемые от клиентов, — все это активно потребляет 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, у которого достаточно "сил", чтобы управлять всей этой мощью. Все в них нацелено на то, чтобы обслужить как можно больше клиентов, обработать как можно больше запросов, чтобы пользователи получили запрошенную информацию за приемлемое 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.

Но 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.

Итак, мы только что познакомились с особой *архитектурой* (принципом построения компьютерных систем), называемой *двухзвенной*, или архитектурой *"клиент-сервер"*, разделяющей все интернет-программы на клиенты и серверы. Эта архитектура используется для реализации практически всех современных интернет-сервисов и пока что себя оправдывает.

#### Замечание

Некоторые интернет-сервисы, в частности, так называемые *файлообменные сети* (Napster, Gnutella, Kazaa и пр.), используют принципиально 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, реализующее функции и клиента, и сервера.

## Протоколы

Люди, чтобы понимать друг друга, должны разговаривать на одном языке. Точно так и с компьютерами, подключенными к сети, неважно, какой — все-

мирной или локальной. Обмен данными по этим сетям должен проходить по единым стандартам, иначе случится новое вавилонское столпотворение.

Стандарт, по которому кодируются данные для отправки по сети, называется *протоколом*. В Интернете для обмена данными используются несколько протоколов, которые мы здесь вкратце рассмотрим.

Самый фундаментальный, если так можно сказать, протокол Интернета — это *TCP/IP* (Transfer Control Protocol/Internet Protocol, протокол управления передачей/протокол Интернета). Это так называемый *протокол низкого уровня*, определяющий только самые основные параметры передаваемых данных: длину отдельных порций (*пакетов*) данных, формат указания адресов получателя и отправителя, а также простейшие средства защиты от ошибок. Можно сказать, что TCP/IP занимается исключительно передачей данных по каналам Интернета, не вникая, что же именно он передает.

На протоколе TCP/IP базируются другие протоколы, уже *высокого уровня*. Эти протоколы описывают способы построения клиентских запросов и серверных ответов, особые команды, пересылаемые клиентом серверу при запросе или передаче данных, и способы кодирования передаваемой информации. Сам процесс передачи данных они не затрагивают — для этого существует "чернорабочий" TCP/IP.

#### Замечание

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

Каждый сервис Интернета использует свой собственный протокол высокого уровня (а то и сразу несколько, предназначенных для разных задач или работанных 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 для запроса и отправки данных, пересылаемых клиентом (Web-обозревателем) Web-серверу, и способы представления пересылаемых в обе стороны данных. Пожалуй, это самый широкоизвестный протокол Интернета (конечно, после TCP/IP).

#### Замечание

Протокол HTTP для управления обменом данными предусматривает всего три команды: загрузки файла, отправки файла и получения сведений о файле.

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 и т. д.) и способы кодирования файлов для пересылки по каналам связи. В этом смысле протоколы HTTP и FTP — "родственники".

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. При получении же почты от сервера клиент общается с ним по протоколу *POP3* (Post-Office Protocol, протокол почты).

### Замечание

Нужно также упомянуть протокол *IMAP* (Internet Message Access Protocol, протокол доступа к почте Интернета), применяемый также для получения клиентом почты от сервера. По сравнению с более старым *POP3* он предоставляет больше возможностей, но распространен не так широко.

Сервис новостей использует для работы протокол *NNTP* (Network News Transfer Protocol, протокол передачи сетевых новостей). Остальные сервисы используют свои протоколы. Но мы не будем на них останавливаться.

Мы уже знаем, что перед началом сеанса связи клиент должен установить соединение с сервером и что соединение это можно представить как воображаемый канал связи, существующий "внутри" физического канала (кабеля, волоконно-оптической линии или радиоканала). Так вот, протокол TCP/IP позволяет создать всего 65 535 таких воображаемых каналов, и называются эти каналы *портами* TCP/IP. Соединение с сервером может быть установлено через любой из этих портов; соединения, установленные через разные порты, действуют независимо, не "мешая" друг другу. Это позволяет сразу нескольким программам (например, Web-обозревателю, почтовому клиенту и интернет-пейджеру) обмениваться данными с соответствующими серверами через один физический канал (модем, сетевой кабель, радиоканал и пр.).

Каждый существующий протокол высокого уровня использует для передачи данных свой собственный порт TCP/IP (так называемый *порт по умолчанию*). В табл. 1.1 перечислены некоторые протоколы и используемые ими порты по умолчанию.

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

| Протокол | Порт по умолчанию |
|----------|-------------------|
| HTTP     | 80                |
| FTP      | 21                |
| SMTP     | 25                |
| POP3     | 110               |

Но почему такое странное название — "порт по умолчанию"? Давайте разберемся.

Дело в том, что все более-менее серьезные серверы предоставляют возможность изменить порт, используемый протоколом, которые они обслуживают, на другой. Например, Web-сервер может быть настроен так, чтобы использоваться для "общения" с клиентами не 80-й порт, а, скажем, 8000-й. (Автору этой книги время от времени встречаются Web-серверы, настроенные таким образом.) Это весьма редко, только в особых случаях, но все же применяется. (Например, если на одном серверном компьютере работают два Web-сервера, один из них настраивают на порт по умолчанию — 80-й, — а другой — да хотя бы и на 8000-й.)

#### Замечание

Вообще, специально настраивать сервер для использования порта по умолчанию не нужно — он уже настроен соответствующим образом.

## Интернет-адреса

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

*Интернет-адрес* — это уникальное числовое или строковое значение, позволяющее точно идентифицировать компьютер в Сети. Именно по интернет-адресу клиент находит нужный ему сервер. Именно по интернет-адресу происходит отправка данных. Интернет-адрес — это своего рода "имя" сервера.

Изначально, на заре эпохи Интернета, в качестве интернет-адреса использовался *IP-адрес* — числовое значение, идентифицирующее компьютер для протокола TCP/IP. Как мы помним, TCP/IP разбивает передаваемую информацию на пакеты. Так вот: в каждом таком пакете содержатся IP-адреса компьютера-отправителя и компьютера-получателя.

IP-адрес замечательно подходит для компьютеров, но очень плохо — для людей. Он имеет такой вид:

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 адреса, о которых стоит поговорить подробно.

Но прежде чем мы начнем разговор о доменных адресах, давайте выясним, что такое домен. *Домен*, или *доменная зона*, — это участок Интернета, выделенный по какому-либо принципу, например, территориальному. Домен мо-

жет быть крупным, или мелким, или вообще состоять из одного компьютера. Он обозначается особой строкой текста.

### Внимание!

В обозначении домена могут присутствовать только латинские буквы, цифры, знаки "минус" и подчеркивания.

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 нетерриториальному признаку; к ним относятся домены com (коммерческие серверы), edu (образовательные), mil (военные), org (организации, не занимающиеся компьютерами и Интернетом), net (организации, занимающиеся компьютерами и Интернетом), biz (коммерческие организации) и некоторые другие. *Национальные домены* объединяют компьютеры по территориальному признаку и выдаются целым странам; это домены us (США), uk (Великобритания), fr (Франция), de (Германия), ru (Россия) и др.

Что касается доменов нижнего уровня, то они выдаются, как правило, отдельным организациям или, опять же, по территориальному признаку. Их текстовое обозначение часто совпадает с названием организации или района — владельца домена.

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, — department, а страна — ru (Россия), то мы получим такое доменное имя:

```
comp45.buh.department.ru
```

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

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

Для этого используются особые программы, называемые серверами *DNS* (Domain Name System, система доменных имен). Занимаются они тем, что принимают от компьютеров, которым нужно куда-то отправить данные по протоколу TCP/IP, доменные имена и возвращают соответствующие этим именам IP-адреса. Такие серверы DNS имеются в каждом домене; кроме того,

несколько самых мощных в мире серверов DNS находятся как бы "выше" всех доменов, даже доменов верхнего уровня, — это глобальные серверы DNS.

Но вернемся к доменным именам. Доменное имя идентифицирует сам серверный компьютер, а не выполняющуюся на нем программу-сервер. А таких серверов на одном компьютере может быть несколько: Web, FTP, почта, чат и пр. Чтобы обратиться к нужному серверу, не "беспокоя" остальных, перед доменным именем указывается обозначение протокола, по которому этот сервер "общается" с клиентами; при этом для передачи данных будет задействован порт по умолчанию этого протокола. Вот пара примеров интернет-адресов, указывающих на программы-серверы (обозначение протокола выделено полужирным шрифтом):

**http**://comp45.buh.department.ru

**ftp**://comp45.buh.department.ru

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

Если же какой-либо сервер использует порт, отличный от порта по умолчанию, то номер нужного порта записывается после доменного имени серверного компьютера и отделяется от него двоеточием. Вот так мы можем обратиться к Web-серверу, использующему порт 8000 (номер порта выделен полужирным шрифтом):

http://comp45.buh.department.ru:**8000**

Ну вот, с основными принципами работы Интернета и соответствующими им понятиями и терминами мы ознакомились. Конечно, кое-что новое мы узнаем потом, в процессе чтения книги, но пока что полученных знаний нам хватит. Давайте сосредоточимся на WWW — в основном, именно этим сервисом мы будем пользоваться на протяжении всей книги.

## Базовые понятия WWW

Здесь мы выясним все о Web-страницах и Web-сайтах, узнаем, чем сайт отличается от страницы, поговорим о Web-клиентах и Web-серверах и освоим несколько новых понятий.

## Web-страницы и Web-сайты

Что такое *Web-страница*? Ответить на этот вопрос могут многие. Это интернет-документ, предназначенный для распространения через Интернет посредством сервиса WWW. А если уж говорить совсем упрощенно, это то, что

показывает в своем окне программа для просмотра Web-страниц — Web-обозреватель.

С технической точки зрения Web-страница — это текстовый файл, содержащий собственно текст наряду со специальными командами, выполняющими форматирование текста и создающими элементы, не относящиеся к тексту (изображения, таблицы и пр.), и сохраненный на жестких дисках серверного компьютера. Получив от Web-обозревателя запрос по протоколу HTTP, *Web-server* (серверная программа, обеспечивающая работу сервиса WWW) извлекает этот файл и отправляет его Web-обозревателю.

Файлы, хранящие Web-страницы, должны иметь расширение htm или html. Оба этих расширения вполне равноправны.

### Внимание!

Web-страницы, созданные с использованием какой-либо технологии серверных страниц, должны иметь расширение, соответствующее используемой технологии. Так, страницы, созданные с использованием PHP, должны иметь расширение php. (О серверных Web-страницах будет рассказано в *главе 6*.)

А что такое *Web-сайт*? Это набор Web-страниц, подчиненных общей тематике и объединенных в единое целое (как — будет рассказано в *главе 2*). Как видим, чисто технических отличий у Web-страницы и Web-сайта не слишком много.

### Замечание

Отдельные Web-страницы и целые Web-сайты также могут быть сохранены на жестком диске клиентского компьютера и открываться в Web-обозревателе прямо с него. В этом случае роль своеобразного Web-сервера выполняет файловая система.

Теперь поговорим о некоторых технических деталях размещения сайта на серверном компьютере. Эти детали очень важны.

Прежде всего, для хранения всех файлов, составляющих Web-сайт, на диске серверного компьютера создается особая папка, называемая *корневой*. Эта папка создается человеком, занимающимся настройкой и обслуживанием программы Web-сервера (или же всего серверного компьютера), — *администратором*. Полный путь к этой папке тот же самый администратор заносит в настройки Web-сервера, чтобы последний смог ее найти.

### Внимание!

Все без исключения файлы, являющиеся содержимым Web-сайта, должны находиться в корневой папке. Все файлы, не находящиеся в корневой папке, автоматически исключаются Web-сервером из состава сайта.

Внутри корневой папки могут быть созданы другие папки, хранящие файлы второстепенных Web-страниц, графических изображений, архивов и дистрибутивов программ. Обычно это делается для удобства управления большим Web-сайтом.

### Замечание

Вообще-то, все серьезные программы Web-серверов предоставляют возможность создания так называемых *виртуальных папок*. Виртуальная папка — это папка, находящаяся в любом месте файловой системы компьютера, но считаемая Web-сервером частью сайта. Виртуальные папки также создаются администратором Web-сервера.

Одна из страниц Web-сайта должна быть так называемой *страницей по умолчанию*. Она отправляется Web-обозревателю, если мы обратимся напрямую к Web-серверу, отправив запрос на его интернет-адрес, скажем:

<http://www.somesite.ru>

Имя файла этой страницы задается администратором Web-сервера в его настройках. Обычно файл страницы по умолчанию называется default.htm[1] или index.htm[1].

Если же нам нужна какая-то конкретная страница сайта, мы отправим Web-серверу интернет-адрес, указывающий на файл этой страницы. Например:

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

Как мы видим, имя файла нужной нам страницы (в нашем случае — somepage.html) указывается сразу после интернет-адреса Web-сервера. Получив его, Web-сервер ищет этот файл в корневой папке сайта и, если он там есть, отправляет его клиенту.

Если же нам понадобится файл, находящийся не в самой корневой папке сайта, а в одной из вложенных в нее папок, мы отправим Web-серверу запрос вида:

<http://www.somesite.ru/download/archive.zip>

В этом случае Web-сервер отправит Web-обозревателю архивный файл archive.zip, находящийся в папке download, вложенной в корневую папку сайта.

### Замечание

Для обращения к файлу, находящемуся в виртуальной папке, используется аналогичный запрос:

<http://www.somesite.ru/pictures/somepicture.jpg>

Здесь pictures — виртуальная папка.

Пожалуй, на этом знакомство с интернет-адресами можно закончить. Далее, изучая создание Web-страниц и, в частности, гиперссылок в *главе 2*, мы узна-

ем о них кое-что еще. А сейчас временно расстанемся с ними и поговорим о различных программах Web-обозревателей и Web-серверов, имеющих хождение в настоящее время.

## Web-обозреватели

Мы уже знаем, что Web-обозреватели — это программы для просмотра Web-страниц и Web-сайтов. Их основная задача — это отправить Web-серверу корректно, в соответствии со всеми стандартами сформированный клиентский запрос, принять серверный ответ и обработать полученные в составе этого ответа данные (скажем, вывести на экран полученную Web-страницу или сохранить на жестком диске архивный файл). Для этого окно Web-обозревателя содержит поле ввода интернет-адреса и область, в которую собственно выводится Web-страница. (Разумеется, оно также содержит заголовок, главное меню и панели инструментов, как и многие окна приложений Windows.)

А теперь — нечто новенькое. После получения от сервера файла Web-страницы (и всех связанных с ней файлов, т. к. страница может состоять из множества файлов; подробнее об этом мы поговорим в главе 2) Web-обозреватель сохраняет их на жестком диске клиентского компьютера в особой области, называемой *кэшем*. Этот кэш может иметь вид как обычной папки (кэш Microsoft Internet Explorer и Opera), так и большого файла (кэш Netscape Navigator и Firefox).

Зачем это нужно? Да хотя бы затем, чтобы мы смогли впоследствии просмотреть данную страницу, не подключаясь к Интернету. Все современные Web-обозреватели поддерживают так называемый *автономный режим* (offline mode), когда они отображают только те страницы, что находятся в кэше. (Кстати — исключительно удобная вещь, особенно для тех, кто выходит в Интернет через модем.) Если же мы попытаемся просмотреть страницу, которой нет в кэше, Web-обозреватель предложит нам подключиться к Интернету и загрузить ее.

Теперь познакомимся с программами Web-обозревателей, имеющими в настоящее время наибольшую популярность. Все они, в общем, следуют одним и тем же стандартам и отличаются друг от друга только деталями, не оговоренными в этих стандартах, и удобством для пользователей.

Король виртуальных пространств — это, конечно, Microsoft Internet Explorer. Он имеется на любом компьютере, работающем под управлением Windows (что, как говорят злые языки, и обусловило его популярность). Однако это очень мощная, быстрая, весьма нетребовательная к ресурсам и исключительно удобная программа. Автор данной книги для просмотра Web-страниц пользуется именно Internet Explorer. В настоящее время доступна версия 7.0.