

**ВЛАДИМИР ДРОНОВ**



# **PHP 5/6, MySQL 5/6 и Dreamweaver CS4**

## **Разработка интерактивных Web-сайтов**



НАЧАЛА WEB-ДИЗАЙНА

ВВЕДЕНИЕ В БАЗЫ ДАННЫХ

НАПИСАНИЕ  
WEB-ПРИЛОЖЕНИЙ  
НА ЯЗЫКЕ PHP

ОСОБЕННОСТИ НОВЕЙШИХ  
ВЕРСИЙ PHP 6 И MYSQL 6

СОЗДАНИЕ ИНТЕРАКТИВНЫХ  
WEB-САЙТОВ В СРЕДЕ ADOBE  
DREAMWEAVER CS4

ОБУЧЕНИЕ  
WEB-ПРОГРАММИРОВАНИЮ  
НА ПРИМЕРАХ

**PRO**  
ПРОФЕССИОНАЛЬНОЕ  
ПРОГРАММИРОВАНИЕ

УДК 681.3.06  
ББК 32.973.26-018.2  
Д75

**Дронов В. А.**

Д75 PHP 5/6, MySQL 5/6 и Dreamweaver CS4. Разработка интерактивных Web-сайтов. — СПб.: БХВ-Петербург, 2009. — 544 с.: ил. — (Профессиональное программирование)

ISBN 978-5-9775-0439-3

Рассмотрены приемы разработки на языке PHP интерактивных Web-сайтов, извлекающих данные из баз MySQL. Отмечены особенности новейших версий PHP 6 и MySQL 6. Для написания простейших страниц используется популярный Web-редактор Adobe Dreamweaver CS4, попутно дается краткое описание языка HTML. Приводится 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.

*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>Анна Кузьмина</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</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 30.06.09.

Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 43,86.

Тираж 2000 экз. Заказ №

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

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

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

# Оглавление

<b>ВВЕДЕНИЕ .....</b>	<b>1</b>
О чем вообще идет речь?.....	1
Dreamweaver: "Делай, как я!" .....	3
Типографские соглашения .....	4
Благодарности .....	5
<b>ЧАСТЬ I. ОСНОВЫ WEB-ДИЗАЙНА .....</b>	<b>7</b>
<b>ГЛАВА 1. СОВРЕМЕННЫЕ ИНТЕРНЕТ-ТЕХНОЛОГИИ .....</b>	<b>9</b>
Принципы работы Интернета .....	9
Что такое Интернет. Сервисы Интернета .....	9
Клиенты и серверы.....	12
Протоколы.....	15
Интернет-адреса .....	18
Базовые понятия WWW.....	21
Web-страницы и Web-сайты.....	21
Web-обозреватели .....	25
Web-серверы .....	26
Публикация Web-сайта в Интернете. Хостинг-провайдеры .....	27
Что дальше?.....	29
<b>ГЛАВА 2. HTML — ЯЗЫК НАПИСАНИЯ WEB-СТРАНИЦ .....</b>	<b>30</b>
Введение в язык HTML .....	31
Теги HTML. Форматирование текста.....	31
Графика на Web-страницах. Внедренные элементы.....	36
Гиперссылки .....	40
Интернет-адреса в WWW .....	42
Правильно оформленные Web-страницы .....	44
Иерархия тегов HTML.....	46
Кодирование текста. Проблема русских кодировок.....	47
Начала сайтостроения.....	49
Планирование Web-сайта .....	50
Логическая структура Web-сайта .....	51
Проектируем наш первый Web-сайт .....	54
Что дальше?.....	56

<b>ГЛАВА 3. ADOBE DREAMWEAVER — ПАКЕТ ДЛЯ СОЗДАНИЯ WEB-СТРАНИЦ И WEB-САЙТОВ .....</b>	<b>57</b>
Запуск Dreamweaver. Главное окно Dreamweaver.....	58
Предварительная настройка Dreamweaver.....	59
Основы работы в Dreamweaver .....	63
Создание новой Web-страницы .....	63
Набор текста .....	65
Форматирование фрагментов текста .....	68
Форматирование абзацев.....	70
Специальные символы и нетекстовые элементы .....	72
Работа с таблицами .....	76
Создание таблиц.....	77
Работа с таблицей.....	80
Формирование таблиц .....	81
Объединение ячеек.....	83
Вставка графических изображений.....	86
Создание гиперссылок.....	89
Предварительный просмотр Web-страниц .....	92
Вызов справки .....	93
Что дальше?.....	96
<b>ГЛАВА 4. CSS — ЯЗЫК ОФОРМЛЕНИЯ WEB-СТРАНИЦ .....</b>	<b>97</b>
Простейшие приемы работы со стилями CSS.....	98
Встроенные стили CSS. Основные понятия языка CSS .....	98
Стили переопределения тегов CSS. Таблицы стилей CSS.....	100
Форматирование текста средствами CSS .....	104
Стилевые классы CSS. Привязка стилиевых классов .....	110
Комбинированные стили CSS.....	112
Развитые средства Dreamweaver для работы со стилями CSS.....	114
Почему таблицы стилей CSS называются каскадными.	
Правила каскадности .....	117
Псевдостили.....	120
Контейнеры.....	123
Встроенные контейнеры.....	123
Блочные контейнеры.....	124
Создание рамок .....	125
Привязка внешней таблицы стилей к Web-странице .....	128
Средства Dreamweaver для работы со стилями.....	129
Вызов справочника по CSS.....	131
Что дальше?.....	132
<b>ГЛАВА 5. РАБОТА С WEB-САЙТОМ В DREAMWEAVER.....</b>	<b>133</b>
Подготовка к публикации сайта .....	134
Регистрация сайта в Dreamweaver .....	134

Работа с файлами сайта. Панель <i>Файлы</i> .....	137
Проверка Web-страниц .....	141
Проверка правильности HTML-кода .....	141
Проверка гиперссылок .....	143
Взаимодействие панели <i>Файлы</i> и окна документа .....	145
Публикация сайта .....	145
Публикация сайта на локальном Web-сервере .....	146
Публикация сайта на удаленном Web-сервере .....	150
Использование для публикации Web-сайта протокола FTP .....	151
Настройка Dreamweaver для публикации сайта по FTP .....	152
Публикация сайта по протоколу FTP .....	156
Управление Web-сайтами, зарегистрированными в Dreamweaver .....	158
Что дальше? .....	159
<b>ЧАСТЬ II. ОСНОВЫ WEB-ПРОГРАММИРОВАНИЯ .....</b>	<b>161</b>
<b>ГЛАВА 6. ПРИНЦИПЫ WEB-ПРОГРАММИРОВАНИЯ .....</b>	<b>163</b>
Недостатки статических Web-страниц и их преодоление .....	163
Данные и их представление .....	164
Недостатки статических Web-страниц .....	165
Серверные программы — радикальный способ отделить информацию от представления .....	166
Технологии создания серверных программ .....	169
Серверные Web-страницы .....	169
Другие технологии серверного программирования .....	172
Второй Web-сайт. Использование серверных страниц .....	173
Что дальше? .....	173
<b>ГЛАВА 7. БАЗЫ ДАННЫХ .....</b>	<b>174</b>
Введение в реляционные базы данных .....	174
Что такое реляционные базы данных .....	175
Составные части реляционной базы данных .....	176
Таблицы, поля и записи .....	176
Правила .....	178
Индексы и ключи .....	179
Связи .....	183
Настольные и серверные реляционные СУБД .....	186
Язык обработки данных SQL .....	189
Зачем нужен SQL .....	189
Выборка данных .....	191
Простейшие запросы выборки данных .....	191
Сортировка данных .....	193

Фильтрация данных.....	194
Задание связей между таблицами .....	196
Псевдонимы полей .....	198
Агрегатные функции SQL.....	198
Изменение данных .....	201
Добавление записи .....	201
Изменение записи.....	201
Удаление записи .....	202
Другие запросы SQL .....	203
Разграничение доступа. Права.....	203
Сервер данных MySQL и его возможности.....	206
База данных для нашего сайта .....	209
Что дальше?.....	212
<b>ГЛАВА 8. КРАТКИЙ КУРС ЯЗЫКА PHP .....</b>	<b>213</b>
Основные понятия PHP .....	213
Написание сценариев PHP .....	214
Операторы, аргументы и выражения.....	216
Переменные .....	218
Типы данных.....	220
Логический.....	220
Целочисленный.....	220
С плавающей точкой .....	220
Строковый .....	221
NULL.....	222
Операторы.....	223
Арифметические.....	223
Оператор объединения строк .....	224
Операторы присваивания .....	224
Операторы сравнения.....	225
Логические операторы .....	226
Вычисление выражений, содержащих логические операторы .....	228
Совместимость и преобразование типов данных.....	228
Приоритет операторов .....	230
Сложные выражения PHP .....	232
Блоки .....	232
Условные выражения.....	233
Выражения выбора.....	235
Циклы .....	237
Цикл со счетчиком .....	237
Цикл с постусловием.....	238
Цикл с предусловием .....	239
Прерывание цикла .....	240

Функции .....	241
Объявление функций .....	241
Вызов функций .....	243
Использование переменных внутри тела функции .....	244
Встроенные функции PHP .....	245
Массивы .....	246
Создание массивов и работа с ними .....	246
Цикл просмотра .....	249
Константы .....	250
Комментарии .....	251
Что дальше? .....	252
<b>Глава 9 . Простейший вывод данных .....</b>	<b>253</b>
Подготовка к созданию серверных страниц .....	253
Регистрация базы данных в Dreamweaver .....	256
Создание простейших серверных страниц .....	263
Создание набора записей .....	263
Создание серверной страницы .....	266
Разбор сценариев PHP, выводящих данные из базы .....	271
Передача данных между серверными страницами .....	275
Метод передачи данных GET .....	276
Создание Web-страниц, передающих данные друг другу .....	277
Разбор PHP-кода, принимающего и обрабатывающего данные .....	280
Более сложные серверные страницы .....	286
Реализация постраничного вывода записей .....	286
Создание навигатора .....	286
Вывод сведений о наборе записей .....	288
Разбор кода, реализующего постраничный просмотр .....	290
Вывод элементов Web-страницы в зависимости от условия .....	295
Реализация возврата на нужную "страницу" .....	297
Что дальше? .....	299
<b>Глава 10. Ввод и правка данных .....</b>	<b>300</b>
Реализация ввода и передачи данных .....	300
Ввод данных. Формы .....	301
Кодирование данных .....	303
Передача данных .....	304
Простые серверные Web-страницы для ввода и правки данных .....	306
Страница для добавления записи .....	306
Разбор сценариев PHP, добавляющих запись .....	316
Страница для правки записи .....	321
Разбор сценариев PHP, изменяющих запись .....	326

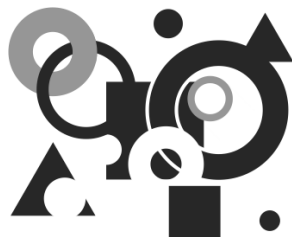
Страница для удаления записи .....	328
Страница для работы с комментариями.....	331
Более сложные Web-страницы для ввода и правки данных .....	333
Что дальше?.....	344
<b>ГЛАВА 11. БОЛЕЕ СЛОЖНЫЙ ВЫВОД ДАННЫХ .....</b>	<b>345</b>
Правильный вывод значений даты.....	345
Особые случаи вывода элементов Web-страницы.....	346
Создание сложных наборов записей .....	349
Создание страницы статистики .....	351
Реализация поиска.....	358
Что дальше?.....	363
<b>ЧАСТЬ III. БЕЗОПАСНОСТЬ И ЦЕЛОСТНОСТЬ ДАННЫХ.....</b>	<b>365</b>
<b>ГЛАВА 12. ВВЕДЕНИЕ В БЕЗОПАСНОСТЬ И ЦЕЛОСТНОСТЬ ДАННЫХ .....</b>	<b>367</b>
Безопасность и разграничение доступа.....	367
Целостность данных .....	370
Что дальше?.....	371
<b>ГЛАВА 13. РАЗГРАНИЧЕНИЕ ДОСТУПА .....</b>	<b>372</b>
Создание таблицы списка пользователей.....	373
Создание страницы входа на сайт .....	374
Процесс создания страницы входа на сайт в Dreamweaver.....	374
Сессии. Переменные уровня сессии.....	378
Разбор кода PHP, выполняющего вход.....	381
Разграничение доступа к Web-страницам .....	384
Процесс разграничения доступа к страницам в Dreamweaver .....	384
Разбор кода PHP, выполняющего разграничение доступа.....	387
Создание страницы выхода с сайта.....	391
Процесс создания страницы выхода с сайта в Dreamweaver .....	391
Разбор кода PHP, выполняющего выход .....	392
Создание административных страниц для управления пользователями.....	394
Разграничение доступа к фрагментам Web-страниц .....	397
Что дальше?.....	402
<b>ГЛАВА 14. ОБЕСПЕЧЕНИЕ ССЫЛОЧНОЙ ЦЕЛОСТНОСТИ ДАННЫХ .....</b>	<b>403</b>
Простой способ обеспечения ссылочной целостности .....	404
Сложный способ обеспечения ссылочной целостности.....	405
Недостаток простого способа и попытка его устранить .....	406
Блокировка таблиц MySQL и ее использование .....	408
Реализация сложного способа.....	409



Каскадное удаление записей .....	411
Что дальше? .....	412
<b>ЧАСТЬ IV. ПОСЛЕДНИЕ ШТРИХИ .....</b>	<b>413</b>
<b>ГЛАВА 15. ОБРАБОТКА ТЕКСТА СРЕДСТВАМИ PHP .....</b>	<b>415</b>
Разбиение текста заметки на абзацы .....	415
Форматирование текста .....	419
Недопустимость HTML-форматирования в блогах. Внутренние теги .....	419
Реализация форматирования текста .....	421
Вставка графических изображений и гиперссылок .....	423
Вставка видеоклипов .....	429
Что дальше? .....	440
<b>ГЛАВА 16. УПРАВЛЕНИЕ ФАЙЛАМИ ЧЕРЕЗ WEB-ИНТЕРФЕЙС .....</b>	<b>441</b>
Просмотр содержимого папки .....	443
Отправка файлов на Web-сайт .....	447
Как отправить файл из Web-обозревателя .....	447
Как принять отправленный файл .....	449
Реализация отправки файла .....	452
Удаление файлов .....	454
Страница для управления видеофайлами .....	456
Страница для управления прочими файлами .....	459
Что дальше? .....	459
<b>ГЛАВА 17 . ХРАНЕНИЕ ДАННЫХ НА СТОРОНЕ КЛИЕНТА .....</b>	<b>460</b>
Задание цветовой гаммы сайта .....	461
Хранение настроек посетителя .....	464
Способы хранения настроек .....	464
Cookies и их использование .....	466
Реализация хранения настроек в cookies .....	468
Какие данные стоит хранить в cookies .....	470
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>472</b>
<b>ПРИЛОЖЕНИЯ .....</b>	<b>475</b>
<b>ПРИЛОЖЕНИЕ 1. УСТАНОВКА WEB-СЕРВЕРА APACHE .....</b>	<b>477</b>
Установка .....	477
Запуск и остановка .....	483
Настройка .....	483
Доступ к документации по Apache .....	484

<b>ПРИЛОЖЕНИЕ 2. УСТАНОВКА СЕРВЕРА ДАННЫХ MYSQL .....</b>	<b>486</b>
Установка .....	486
Запуск и остановка .....	488
Запуск и остановка MySQL из Командной строки .....	488
Запуск и остановка MySQL как службы .....	489
Изменение пароля пользователя <i>root@localhost</i> .....	491
Доступ к документации по MySQL .....	492
<b>ПРИЛОЖЕНИЕ 3. УСТАНОВКА ПЛАТФОРМЫ PHP .....</b>	<b>493</b>
Установка .....	493
Настройка .....	495
Запуск и остановка .....	497
Доступ к документации по PHP .....	497
<b>ПРИЛОЖЕНИЕ 4. УСТАНОВКА И ИСПОЛЬЗОВАНИЕ КЛИЕНТА ДАННЫХ PHPMYADMIN .....</b>	<b>498</b>
Установка и настройка .....	498
Использование .....	499
Вход .....	499
Создание базы данных .....	501
Создание таблиц .....	502
Создание полей .....	502
Создание индексов .....	505
Правка и удаление полей, индексов, таблиц и баз данных .....	507
Правка и удаление полей .....	507
Правка и удаление индексов .....	509
Правка и удаление таблиц .....	509
Правка и удаление баз данных .....	510
Управление пользователями .....	510
Средства управления пользователями phpMyAdmin .....	510
Создание пользователя .....	512
Правка и удаление пользователей .....	516
Работы с данными .....	516
Выход .....	518
Доступ к документации по phpMyAdmin .....	518
<b>ПРИЛОЖЕНИЕ 5. ИСПОЛЬЗОВАНИЕ УТИЛИТЫ SUPER ДЛЯ ПЕРЕКОДИРОВАНИЯ ВИДЕОФАЙЛОВ .....</b>	<b>519</b>
Запуск .....	520
Использование .....	521
Завершение работы .....	523
<b>ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ .....</b>	<b>525</b>

## ГЛАВА 1



# Современные интернет-технологии

Да, без теории не обойтись. Ведь прежде чем приниматься за что-то, 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 приобретаем 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 продолжает 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.

Не менее популярен и другой сервис Интернета — 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 "пошел в народ". Вот о нем-то и пойдет речь в этой книге.

Остальные популярные ныне сервисы Интернета мы рассмотрим вкратце. Все равно в этой книге мы заниматься ими не будем.

□ *Потоковое вещание.* Это своего рода теле- и радиовещание через Интернет появилось лет десять назад и сейчас весьма популярно. В качестве примеров можно упомянуть сверхпопулярный сайт YouTube и его русского "коллегу" RuTube. К сожалению, чтобы слушать интернет-радио и, в особенности, смотреть интернет-телевидение, нужен достаточно быст-

рый 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.
- Сервис пересылки файлов *FTP* (File Transfer Protocol, протокол передачи файлов). Сейчас он отошел на второй план, уступив место более новым сервисам, но все еще весьма активно используется.

Ну, вот и все. Устаревшие, 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.

### **НА ЗАМЕТКУ**

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 передачи информации от сервера клиентами состоит из шести шагов.

1. Пользователь запрашивает с помощью программы-клиента некую информацию, указав при этом 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* (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 запрошенные клиентом данные отсутствуют, 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.
6. Клиент разрывает соединение с сервером.

Процесс отправки клиентом данных серверу также состоит из шести шагов.

1. Пользователь вводит в программу-клиент отправляемую 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 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 неудачной 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 в начале 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.

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 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 мощностью. 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 большинства современных интернет-сервисов и пока что себя оправдывает.

### **НА ЗАМЕТКУ**

Некоторые интернет-сервисы, в частности, так называемые *файлообменные сети* (BitTorrent 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 —

всемирной 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 довольно много протоколов, и некоторые из них мы 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 в самом низу иерархии протоколов. 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, "канальчиков", 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 используется другими протоколами, уже *высокого уровня*. Эти протоколы описывают способы оформления клиентских запросов, серверных ответов, подтверждений и сообщений об ошибках, определяют команды, пересылаемые клиентом серверу при запросе или передаче данных, и способы кодирования передаваемой информации.

### **НА ЗАМЕТКУ**

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

Каждый сервис Интернета использует собственный протокол высокого уровня (а то и сразу 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 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) — всем более-менее грамотным интернетчикам знакомы эти четыре буквы.

### **НА ЗАМЕТКУ**

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, создание папки, копирование, перемещение, удаление файлов и папок и пр.) 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).

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 собственный порт (так называемый *порт по умолчанию*). В табл. 1.1 перечислены некоторые протоколы и используемые ими порты по умолчанию.

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

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

Порт по умолчанию может быть изменен — такую возможность предоставляют все более-менее серьезные серверы. Скажем, 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.

## 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 IP. Как мы помним, 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. Вот пример IP-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 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. 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 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), **travel** (туристические 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` (бухгалте-

рия), 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 IP не понимает доменные имена! Что делать? Как преобразовать доменное имя в понятный ему IP-адрес?

Для этого используются 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.

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, которое отделяется от доменного имени двоеточием и двумя слэшами. При этом для передачи данных будет задействован порт по умолчанию этого протокола. Вот пара примеров интернет-адресов, указывающих на программы-серверы (обозначение протокола подчеркнута):

**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 подчеркнут):

**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 особый 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 же 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. Файлы, хранящие Web-страницы, должны иметь расширение htm или html.

#### **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.

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

Работу сервиса WWW реализует особая серверная программа — *Web-сервер*. Она занимается тем, что принимает от Web-обозревателей запросы и "раздает" им запрошенные файлы: Web-страницы, изображения, архивы, дистрибутивы и пр. Собственно, мы уже это знаем.

Все файлы, составляющие сайт, должны храниться в специальной папке, называемой *корневой*. Именно в ней Web-сервер будет искать файлы, запрашиваемые Web-обозревателями. Как правило, корневая папка создается автоматически при установке программы Web-сервера.

### **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 Web-страниц, графических изображений, архивов и дистрибутивов программ и различные служебные файлы. Обычно это делается для удобства управления большим Web-сайтом.

### **НА ЗАМЕТКУ**

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, в отличие от корневой, создаются вручную *администратором* — человеком, занимающимся настройкой и обслуживанием серверной программы (в нашем случае — Web-сервера) или всего серверного компьютера.

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**

т. е. 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 может 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 нам понадобится архивный файл archive.zip, находящийся в папке download, вложенной в корневую папку сайта, 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**

и pošлет Web-серверу такой клиентский запрос:

**/download/archive.zip**

Web-сервер в ответ найдет в корневой папке папку downloads, в ней — файл archive.zip и его отправит Web-обозревателю.

### **НА ЗАМЕТКУ**

Для запроса файла, находящегося в виртуальной папке, используется аналогичный интернет-адрес:

**http://www.somesite.ru/pictures/picture.jpg**