

# PHP, MySQL И DREAMWEAVER MX 2004

## РАЗРАБОТКА ИНТЕРАКТИВНЫХ WEB-САЙТОВ

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

**Владимир Дронов**

# **PHP, MySQL и DREAMWEAVER MX 2004**

**РАЗРАБОТКА ИНТЕРАКТИВНЫХ  
WEB-САЙТОВ**

Санкт-Петербург

«БХВ-Петербург»

2005

УДК 681.3.06  
ББК 32.973.26-018.2  
Д75

**Дронов В. А.**

Д75 PHP, MySQL и Dreamweaver MX 2004. Разработка интерактивных Web-сайтов. — СПб.: БХВ-Петербург, 2005. — 448 с.: ил.

ISBN 5-94157-598-X

В качестве базового средства разработки интерактивных Web-сайтов на платформе PHP-MySQL выбран популярный Web-редактор Macromedia Dreamweaver MX 2004. В качестве примера рассмотрено создание сайта — архива программ и электронных статей. Изложение построено по принципу: от простого — к сложному. Простейшие статичные Web-страницы создаются в редакторе Dreamweaver, попутно приводится краткое описание языка HTML. Простейшие серверные страницы, извлекающие данные из базы MySQL, также создаются в редакторе Dreamweaver, при этом подробно разбираются все сценарии PHP, созданные Dreamweaver, и описывается их работа. Параллельно дается введение в базы данных и приводится краткое описание языка PHP. Наиболее сложные Web-страницы создаются средствами PHP-MySQL без использования Dreamweaver. Приводятся примеры разработки элементов развития Web-портала: выбираемой цветовой схемы, управления файлами через Web-интерфейс, собственного списка рассылки и др. Книга ориентирована на читателя, имеющего базовые понятия об интернет-технологиях.

*Для Web-программистов*

УДК 681.3.06  
ББК 32.973.26.-018.2

### **Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. гл. редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Екатерина Капальгина</i>
Компьютерная верстка	<i>Натальи Караваевой</i>
Корректор	<i>Виктория Пиотровская</i>
Дизайн серии	<i>Инна Тачина</i>
Оформление обложки	<i>Игоря Цырульникова</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 21.01.05.

Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 36,12.

Тираж 4000 экз. Заказ №

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

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 5-94157-598-X

© Дронов В. А., 2005  
© Оформление, издательство "БХВ-Петербург", 2005

# Оглавление

<b>Введение</b> .....	<b>1</b>
О чем вообще идет речь?.....	1
Dreamweaver: "Делай, как я!".....	2
Благодарности.....	3
<b>ЧАСТЬ I. НАШ ПЕРВЫЙ WEB-САЙТ</b> .....	<b>5</b>
<b>Глава 1. Современные интернет-технологии</b> .....	<b>7</b>
Основные принципы работы Интернета.....	7
Что такое Интернет. Сервисы Интернета .....	7
Клиенты и серверы.....	10
Протоколы .....	13
Интернет-адреса.....	15
Основные понятия WWW.....	18
Web-страницы и Web-сайты.....	18
Web-обозреватели .....	20
Web-серверы.....	23
Публикация Web-сайта в Интернете. Хостинг-провайдеры.....	24
Что дальше?.....	25
<b>Глава 2. HTML — язык написания Web-страниц</b> .....	<b>26</b>
Введение в язык HTML.....	26
Теги HTML. Форматирование текста.....	27
Графика на Web-страницах. Внедренные элементы .....	31
Гиперссылки.....	34
Правильно оформленные Web-страницы .....	36
Иерархия тегов HTML .....	37
Средства оформления Web-страниц .....	38

Каскадные таблицы стилей (CSS) .....	38
Создание стилей .....	39
Три способа задания стилей .....	42
Почему "каскадные"? .....	43
Теги физического форматирования HTML .....	45
Кодирование текста. Проблема русских кодировок .....	47
Начала сайтостроения .....	49
Планирование сайта .....	49
Логическая структура Web-сайта .....	50
Проектируем наш первый Web-сайт .....	52
Что дальше? .....	53
<b>Глава 3. Работа с Macromedia Dreamweaver MX 2004 .....</b>	<b>54</b>
Предварительная настройка Dreamweaver .....	55
Основы работы в Dreamweaver .....	57
Создание новой Web-страницы .....	58
Набор текста .....	59
Форматирование фрагментов текста .....	61
Форматирование абзацев .....	66
Специальные символы и нетекстовые элементы .....	68
Работа с таблицами .....	71
Создание таблиц .....	71
Работа с таблицей .....	74
Как формируются таблицы .....	74
Более сложные таблицы .....	76
Вставка графических изображений .....	79
Создание гиперссылок .....	82
Работа со стилями CSS .....	84
Предварительный просмотр Web-страниц .....	89
Вызов справки .....	89
Что дальше? .....	91
<b>Глава 4. Работа с Web-сайтом в Dreamweaver .....</b>	<b>92</b>
Подготовка к публикации сайта .....	93
Регистрация сайта в Dreamweaver .....	93
Работа с файлами сайта. Панель <i>Files</i> .....	97
Проверка Web-страниц .....	100
Проверка правильности HTML-кода .....	100
Проверка гиперссылок .....	102
Взаимодействие панели <i>Files</i> и окна документа .....	103
Публикация сайта .....	104
Публикация сайта на локальном Web-сервере .....	105

Публикация сайта на удаленном Web-сервере.....	108
Использование для публикации Web-сайта протокола FTP.....	108
Настройка Dreamweaver для публикации сайта по FTP.....	110
Публикация сайта по протоколу FTP.....	113
Что дальше?.....	113
<b>ЧАСТЬ II. НАШИ ПЕРВЫЕ СЕРВЕРНЫЕ ПРОГРАММЫ.....</b>	<b>115</b>
<b>Глава 5. Введение в Web-программирование.....</b>	<b>117</b>
Недостатки статичных Web-страниц и их преодоление.....	117
Данные и их представление.....	118
Недостатки статических Web-страниц.....	119
Серверные программы — радикальный способ отделить информацию от представления.....	120
Технологии создания серверных программ.....	122
Активные серверные Web-страницы.....	122
Другие технологии серверного программирования.....	124
Наш второй Web-сайт будет использовать активные серверные страницы.....	125
Что дальше?.....	126
<b>Глава 6. Базы данных.....</b>	<b>127</b>
Введение в реляционные базы данных.....	127
Что такое реляционные базы данных.....	127
Составные части реляционной базы данных.....	128
Таблицы, поля и записи.....	128
Правила.....	131
Индексы и ключи.....	131
Связи.....	135
Настольные и серверные реляционные СУБД.....	137
Язык обработки данных SQL.....	140
Зачем нужен SQL.....	140
Выборка записей из таблицы.....	141
Простейшие запросы выборки данных.....	141
Сортировка данных.....	143
Фильтрация данных.....	144
Задание связей между таблицами.....	146
Псевдонимы полей.....	147
Агрегатные функции SQL.....	148
Изменение записей таблицы.....	150
Добавление записи.....	150
Изменение записи.....	150
Удаление записи.....	152

Другие запросы SQL.....	152
Разграничение доступа. Права.....	152
Сервер данных MySQL и его возможности .....	154
Создаем базу данных для нашего сайта.....	157
Что дальше?.....	159
<b>Глава 7. PHP — технология написания серверных приложений .....</b>	<b>160</b>
Основные понятия PHP .....	160
Написание сценариев PHP.....	161
Операторы, аргументы и выражения.....	163
Переменные.....	164
Типы данных.....	166
Логический.....	166
Целочисленный.....	166
С плавающей точкой .....	166
Строковый.....	167
NULL.....	168
Операторы.....	168
Арифметические.....	169
Оператор объединения строк.....	170
Операторы присваивания.....	170
Операторы сравнения .....	171
Логические операторы.....	172
Как PHP вычисляет выражения, содержащие логические операторы .....	173
Совместимость и преобразование типов данных.....	173
Приоритет операторов.....	175
Сложные выражения PHP.....	177
Блоки.....	177
Условные выражения .....	177
Выражения выбора .....	179
Циклы.....	181
Цикл со счетчиком.....	181
Цикл с постусловием.....	182
Цикл с предусловием.....	183
Прерывание цикла .....	184
Функции .....	185
Создание функций.....	185
Вызов функций .....	186
Использование переменных внутри тела функции .....	188
Встроенные функции PHP .....	189
Массивы .....	190
Создание массивов и работа с ними .....	190

Цикл просмотра .....	192
Константы .....	193
Комментарии .....	194
Что дальше?.....	195
<b>Глава 8. Простейшие серверные Web-страницы. Вывод данных.....</b>	<b>196</b>
Подготовка к созданию серверных страниц .....	196
Регистрация базы данных в Dreamweaver.....	199
Создание простейших серверных страниц.....	203
Создание набора записей.....	204
Создание самой серверной страницы .....	207
Разбор сценариев PHP, используемых для вывода записей.....	212
Передача данных между серверными страницами.....	215
Метод передачи данных GET.....	215
Создание Web-страниц, передающих данные друг другу .....	217
Разбор PHP-кода, обеспечивающего обработку принятых данных.....	220
Более сложные серверные страницы .....	222
Написание сценариев PHP вручную .....	222
Одновременный вывод значений из нескольких полей .....	224
Страницы с несколькими наборами данных.....	227
Правильный вывод значений даты.....	229
Условный вывод элементов Web-страницы.....	230
Вывод сведений о наборе записей.....	232
Что дальше?.....	234
<b>Глава 9. Реализация ввода и правки данных .....</b>	<b>235</b>
Как реализуется ввод и передача данных.....	235
Ввод данных. Формы.....	236
Кодирование данных.....	238
Передача данных.....	239
Административные и пользовательские Web-страницы.....	241
Создание простых серверных Web-страниц для ввода и правки данных...	243
Простая страница для добавления записи.....	243
Разбор сценариев PHP, используемых для добавления записи.....	249
Задание значений по умолчанию для элементов управления.....	253
Простая страница для правки записи .....	256
Разбор сценариев PHP, используемых для правки записи .....	261
Простая страница для удаления записи.....	262
Более сложные серверные страницы для ввода и правки данных.....	264
Частный случай ввода данных — поисковая машина.....	271
Что дальше?.....	273

<b>ЧАСТЬ III. РЕШАЕМ ВОПРОСЫ БЕЗОПАСНОСТИ И ЦЕЛОСТНОСТИ ДАННЫХ.....</b>	<b>275</b>
<b>Глава 10. Введение в безопасность и целостность данных .....</b>	<b>277</b>
Безопасность и разграничение доступа .....	277
Целостность данных.....	279
Что дальше?.....	280
<b>Глава 11. Разграничение доступа .....</b>	<b>281</b>
Создание таблицы списка пользователей.....	282
Создание страницы входа на сайт .....	283
Процесс создания страницы входа на сайт в Dreamweaver .....	283
Сессии. Переменные уровня сессии .....	286
Разбор кода PHP, выполняющего вход.....	288
Разграничение доступа к закрытым Web-страницам .....	292
Процесс разграничения доступа к страницам в Dreamweaver.....	292
Разбор кода PHP, выполняющего разграничение доступа.....	294
Создание страницы выхода с сайта.....	297
Процесс создания страницы выхода с сайта в Dreamweaver .....	297
Разбор кода PHP, выполняющего выход .....	299
Создание административных страниц для управления пользователями....	300
Разграничение доступа к фрагментам Web-страниц.....	301
Что дальше?.....	305
<b>Глава 12. Поддержание ссылочной целостности данных .....</b>	<b>306</b>
Простейший способ поддержания ссылочной целостности .....	307
Второй — более сложный — способ поддержания ссылочной целостности .....	308
Недостаток первого способа поддержания ссылочной целостности и попытка его устранить.....	309
Блокировка таблиц MySQL и ее использование.....	311
Реализация второго способа поддержания ссылочной целостности.....	312
Что дальше?.....	313
<b>ЧАСТЬ IV. НАНОСИМ ПОСЛЕДНИЕ ШТРИХИ .....</b>	<b>315</b>
<b>Глава 13. Вывод сообщений об ошибках .....</b>	<b>317</b>
Страница, сообщающая об ошибке 404.....	318
Сообщения об ошибках в сценариях PHP .....	320
Что дальше?.....	326

<b>Глава 14. Хранение данных на стороне клиента .....</b>	<b>327</b>
Задание порядка сортировки записей.....	327
Хранение настроек посетителя.....	330
Способы хранения настроек.....	330
Cookie и их использование.....	332
Реализация хранения настроек в cookie.....	334
Какие данные стоит хранить в cookie.....	335
Что дальше?.....	336
<b>Глава 15. Управление файлами через Web-интерфейс.....</b>	<b>337</b>
Два способа управления файлами на сайте .....	338
Отправка файлов на Web-сервер .....	340
Как отправить файл из Web-обозревателя.....	340
Как принять отправленный файл.....	341
Реализация отправки файла .....	345
Управление файлами, находящимися на Web-сервере .....	347
Средства PHP для управления файлами .....	347
Выбор папки .....	348
Просмотр содержимого папки.....	348
Получение сведений о файлах и папках .....	350
Копирование, перемещение, переименование и удаление файлов .....	351
Создание и удаление папок .....	352
Создание Web-интерфейса для управления файлами .....	352
Просмотр содержимого папки.....	352
Переходы между папками .....	355
Управление файлами .....	358
Создание и удаление папок .....	360
Что дальше?.....	363
<b>Глава 16. Организация почтовой рассылки.....</b>	<b>364</b>
Введение в почтовые рассылки .....	365
Что такое почтовая рассылка .....	365
Как осуществляются почтовые рассылки.....	366
Реализация службы рассылки на PHP.....	367
Два способа реализовать службу рассылок на PHP.....	367
Средства PHP для отправки почты .....	369
Отправка почты через собственный почтовый сервер .....	369
Прямая отправка письма на почтовый сервер .....	371
Как отправить письмо сразу по нескольким адресам .....	373
Создание соответствующих Web-страниц .....	374

Страница регистрации нового подписчика .....	374
Страница выполнения рассылки .....	376
Страница отписки от рассылки .....	380
Что дальше? .....	381
<b>Заключение</b> .....	<b>383</b>
<b>ПРИЛОЖЕНИЯ</b> .....	<b>387</b>
<b>Приложение 1. Установка Web-сервера Apache</b> .....	<b>389</b>
Установка .....	389
Запуск и остановка .....	394
Предварительная настройка .....	395
Доступ к справочной системе Apache .....	396
<b>Приложение 2. Установка сервера данных MySQL</b> .....	<b>397</b>
Установка .....	397
Предварительная настройка .....	401
Запуск и остановка .....	402
Запуск и остановка под Windows 95, 98 и Me .....	402
Запуск и остановка под Windows NT .....	403
Запуск и остановка под Windows 2000, XP, 2003 .....	404
Доступ к справочной системе MySQL .....	404
Вспомогательные программы .....	405
<b>Приложение 3. Установка обработчика PHP</b> .....	<b>406</b>
Установка .....	406
Предварительная настройка .....	407
Запуск и остановка .....	408
Доступ к справочной системе PHP .....	408
<b>Приложение 4. Установка и использование     клиента данных phpMyAdmin</b> .....	<b>409</b>
Установка и настройка .....	409
Использование .....	411
Вход .....	411
Создание базы данных .....	412
Создание таблиц .....	413
Создание полей .....	413

---

Создание индексов .....	415
Правка и удаление полей, индексов, таблиц и баз данных.....	416
Правка и удаление полей .....	416
Правка и удаление индексов .....	417
Правка и удаление таблиц.....	418
Правка и удаление баз данных.....	418
Управление пользователями.....	419
Средства управления пользователями phpMyAdmin.....	419
Создание пользователя .....	420
Правка и удаление пользователей.....	424
Работы с данными .....	424
Выход.....	425
Другие программы клиентов данных MySQL.....	426
<b>Предметный указатель .....</b>	<b>427</b>

# Введение

"Вот еще одна книга о PHP и MySQL. Боже, сколько их уже издано!.. Так нет, нужно опять тратить бумагу, чтобы рассказать о том же самом".

Так может сказать какой-нибудь знаток компьютерных и интернет-технологий, разглядывая разноцветные книжные обложки на прилавке магазина. Действительно, книг о создании Web-сайтов с использованием серверных Web-страниц PHP и баз данных MySQL сейчас написано очень (возможно, даже слишком) много. "PHP!", "MySQL!", "PHP и MySQL!", "MySQL и PHP!!!" — кричат обложки. Стоит ли писать еще одну книгу на ту же самую тему?

Вопрос не в том, *стоит ли* писать. И даже не в том, *о чем* писать. А *как!* Да-да, именно как писать подобные книги!

Но стоп, давайте обо всем по порядку. Потенциальный читатель наверняка подготовил уйму вопросов, и автору придется на них отвечать. А отвечать лучше всего не торопясь, спокойно, без лишнего шума. Итак...

## О чем вообще идет речь?

Действительно, что это за штуки такие — PHP, MySQL и Dreamweaver? И с чем их едят?

Начнем с самого начала.

MySQL — это сервер данных. Особая программа, позволяющая хранить упорядоченные данные в особых структурах, называемых базами данных. Например, в такую базу можно записать инвентарную книгу, список работников, экзаменационные ведомости за несколько лет, список книг, хранящихся в библиотеке, и многое другое. И не просто записать и сохранить, а еще и получать эти данные по первому запросу. И не просто получать, а быстро, без особого труда и только те данные, которые были запрошены.

PHP — это платформа для создания серверных Web-страниц. Грубо говоря, с ее помощью можно писать программы, которые по запросу посетителя сайта создают Web-страницы и отправляют их ему. Такие Web-страницы могут содержать данные, хранящиеся в базе данных, в том числе, и MySQL.

MySQL и PHP впечатляют даже по отдельности. А уж вместе — это просто мечта! Например, серверная программа, написанная на PHP и являющаяся частью сайта интернет-магазина, может извлекать из базы данных MySQL список товаров и формировать из него красивую Web-страницу. А другая серверная программа, другая часть того же интернет-магазина, спрашивает посетителя, какой товар он хочет заказать и по какому адресу этот товар нужно выслать, и сама записывает в ту же базу данных сведения о новом заказе.

Неудивительно, что PHP и MySQL так популярны. Мало того, что это весьма мощные программы, на которых можно реализовать практически все что угодно. Они еще прекрасно работают в связке и — внимание, сторонники использования легальных программ! — абсолютно бесплатны. находка для создателя некоммерческого Web-сайта с большим информационным наполнением. Да и вполне коммерческий интернет-магазин они тоже "потянут".

Dreamweaver же — мощнейший пакет для создания Web-страниц, обычных (статичных) и серверных, в том числе и написанных на языке PHP. С его помощью мы можем создавать простейшие Web-страницы PHP, вообще не зная этого языка и принципов программирования на нем. А возможностей у Dreamweaver столько, что о нем можно написать несколько толстенных книг, и все равно какие-то секреты наверняка останутся неописанными.

Dreamweaver создан фирмой Macromedia и — увы!.. — является платным. Однако доступна демонстрационная версия, работающая 30 дней. Так что все желающие могут попробовать его в действии, так сказать, "повертеть в руках".

Но знаток, собаку съевший на PHP и MySQL, недовольно кривит губы. "Прекрасно, — бормочет он. — PHP и MySQL — это понятно. Но причем тут Dreamweaver?"

## **Dreamweaver: "Делай, как я!"**

А вот здесь мы вплотную подошли к ответу на вопрос — как писать. Вообще, на взгляд автора, это главный вопрос, на который автору любой книги, не только компьютерной тематики, нужно всегда знать ответ.

Писать книги о PHP и MySQL можно по-разному. Можно начать с описания языка, продолжить примерами простейших страничек, а закончить описанием работы с базами данных MySQL. Можно сразу начать с описания работы с базами данных, продолжить тонкостями составления запросов к ним, а закончить таким дебрями, что читателей дрожь пробьет. А еще можно почти полностью сосредоточиться на MySQL, а PHP описать поверхностно, только "чтобы было".

А можно и совсем по-другому — вот так. Значит, причем здесь Dreamweaver? А вот причем...

Ранее упоминалось, что Dreamweaver может сам создавать простейшие серверные Web-страницы PHP, извлекающие данные из баз MySQL. Так вот,

мы начнем с того, что будем пользоваться для создания своих первых страничек услугами Dreamweaver и внимательно разбирать код PHP, который он создаст. В этом нам поможет руководство по PHP, которое можно найти на официальном сайте этой платформы — <http://www.php.net>.

Когда мы приобретем кое-какой опыт, то сами начнем писать серверные страницы. В конце концов, возможности Dreamweaver в PHP-программировании не очень велики, многие нужные вещи он просто не сможет для нас сделать. А мы сможем!

Фактически мы будем учиться писать серверные страницы PHP на готовых примерах. Это, пожалуй, лучший способ обучения программированию. Автор сам точно так же изучал в свое время PHP — читая код, созданный Dreamweaver. "Делай, как я! — командует Dreamweaver. — Учись у меня!" И этому призыву стоит последовать. Хотя бы первое время.

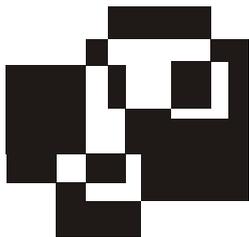
Знарок компьютеров пожимает плечами: "Программирование — это, прежде всего, искусство, и искусству компьютеры не обучены". Что ж, спору нет. Но не будем об этом забывать, что программы создают люди. А ведь именно люди создали искусство...

В качестве примера мы создадим Web-сайт — архив статей и файлов (программ, архивов и пр.). Списки статей и файлов мы будем хранить в базе данных MySQL. А извлекать их оттуда будут серверные страницы PHP, которые нам поможет создать Dreamweaver.

## Благодарности

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

- Губине Наталье Анатольевне, начальнику отдела АСУ Волжского гуманитарного института (г. Волжский Волгоградской обл.), где работает автор, — за понимание и поддержку.
- Всем работникам отдела АСУ — за понимание и поддержку.
- Родителям — за терпение, понимание и поддержку.
- Архангельскому Дмитрию Борисовичу — за дружеское участие.
- Шапошникову Игорю Владимировичу — за содействие.
- Рыбакову Евгению Евгеньевичу, заместителю главного редактора издательства "ВНУ-Петербург", — за неоднократные побуждения к работе, без которых автор давно бы обленился.
- Издательству "ВНУ-Петербург" — за издание моих книг.
- Всем российским программистам, занятым в разработке MySQL и PHP, — за прекрасные программные продукты.
- Всем, кого я забыл здесь перечислить, — за все хорошее.



## **Часть I**

# **Наш первый Web-сайт**

**Глава 1. Современные интернет-технологии**

**Глава 2. HTML — язык написания Web-страниц**

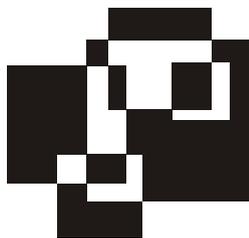
**Глава 3. Работа с Macromedia Dreamweaver MX 2004**

**Глава 4. Работа с Web-сайтом в Dreamweaver**

В первой части мы займемся тем, что создадим свой самый первый Web-сайт. Да-да! Сначала, конечно, будет вводный курс интернет-технологий, потом мы выясним, как же создаются Web-страницы, далее изучим сам пакет Macromedia Dreamweaver MX 2004, — он очень поможет нам в работе. А уж вооружившись столь фундаментальными знаниями, мы приступим к работе над сайтом.

Он будет совсем простеньким, наш первый сайт. Ведь для нас сейчас главное — научиться, не так ли?

# Глава 1



## Современные интернет-технологии

Каждая практика должна предваряться теорией. Без теории никуда, ведь, прежде чем приниматься что-то делать, нужно выяснить, зачем, как и почему именно так это "что-то" делается. А иначе у нас ничего не получится.

Так и с интернет-технологиями. Без необходимого багажа знаний мы просто ничего не сможем сотворить (по крайней мере, ничего стоящего у нас точно не получится). Так давайте же отключим компьютер — пусть отдохнет! — и читаем.

### Основные 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 далее.

Итак, *Интернет* — это всемирная компьютерная сеть. (Вы уже наверняка это знаете.) 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 позже), 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 по модему, 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 пользователям 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 продолжает нормально работать.

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

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

Интернет имеет достаточно долгую и бурную историю. Он появился еще в первой половине 70-х годов XX века, когда американское Министерство обороны финансировало проект создания компьютерной сети, устойчивой к сбоям. Разумеется, создавалась эта сеть для нужд обороны, да и название имела другое — ARPANET. Позднее же, в начале 80-х, эта сеть отошла к ученым, а военные приступили к созданию другой сети, которой пользуются до сих пор. И в то же самое время ARPANET был переименован в Internet, или, если по-русски, Интернет.

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

В Россию, точнее, в СССР, Интернет пришел в 1990 году, но популярность среди широких масс компьютерщиков приобрел только в середине 90-х.

В настоящее время в России миллионы людей пользуются услугами Сети для общения и поиска информации.

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" (по-английски — snail mail). Конечно, эти доступность, удобство, быстрота и бесплатность несут некоторые неудобства, вроде "спама" — несанкционированных рекламных рассылок, но с ними можно бороться.

Еще один сервис Интернета, почти такой же старый, как почта, — это пересылка файлов. Пользователи Интернета называют его FTP (File Transfer Protocol, протокол передачи файлов; почему так — мы узнаем чуть позже). Сейчас FTP уже не имеет той популярности, как на заре существования Интернета, но все еще часто используется для пересылки файлов.

Третий сервис Интернета, который мы подробно рассмотрим, — это Всемирная Паутина, или WWW (World Wide Web, повсеместно протянутая паутина), или просто Web. Это и есть те самые Web-страницы и Web-сайты, которые мы просматриваем в Web-обозревателе. Пожалуй, самый впечатляющий сервис, собственно, и приведший к тому, что сеть ученых стала сетью для всех и получила такую популярность. Вот о нем-то и пойдет речь в этой книге.

Осталось упомянуть еще о нескольких сервисах Интернета, не имеющих такой широкой популярности.

- **Новости.** Чем-то этот сервис похож на электронную почту: пользователь пишет письмо в так называемую *группу новостей* — своего рода электронную доску объявлений. Другие пользователи прочитывают это письмо и пишут ответы, которые помещаются на эту же самую "доску". Совокупность первоначального письма и ответов на него образуют *обсуждение* (по-английски — thread), участие в котором может принять любой пользователь, подписавшийся на эту группу новостей. Когда-то сервис новостей был весьма популярен, но сейчас постепенно сдает свои позиции.
- **Потоковое вещание.** Это своего рода теле- и радиовещание через Интернет появилось совсем недавно, несколько лет назад, но быстро набирает популярность. Единственный серьезный недостаток — необходимость иметь достаточно быстрый канал — сейчас перестает быть актуальным.
- **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 английского 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 другая.

## Клиенты и серверы

Продолжим наше погружение 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! Это Web-обозреватель, клиент электронной почты, программа просмотра интернет-телевидения и интернет-радио, ICQ и "чатилка". Все они очень хорошо нам знакомы.

Но программ, используемых для предоставления нам сервисов Интернета, гораздо больше. И очень многие из них нам, если так можно сказать, "не видны". То есть мы не общаемся с ними напрямую. Вообще, существуют две принципиально разные категории интернет-программ. И сейчас мы их рассмотрим.

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 с вами 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 Интернета существует свой класс серверов (и, как мы уже знаем, клиентов): Web-серверы, серверы электронной почты, чата, интернет-пейджеров, потокового вещания и пр.

### Примечание

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

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

В общем, процесс получения информации от сервера клиентов включает пять шагов.

1. Пользователь запрашивает с помощью программы-клиента некую информацию.
2. Клиент устанавливает соединение 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*. Разумеется, этот ответ также жестко стандартизирован. Если же нужных данных нет, или сервер почему-то не смог понять клиентский запрос, он возвращает в составе ответа *сообщение об ошибке* — особый информационный блок, содержащий описание возникшей ошибки.
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 включает примерно те же пять 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. 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 неуспешной записи 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, то есть перед началом сеанса, клиент должен соединиться с сервером, иначе говоря — установить *соединение*. Это соединение длится ровно столько времени, сколько нужно для завершения сеанса, и поэтому называется *сеансовым*, или *временным*.

Каждое соединение требует компьютерных ресурсов. Серверу нужно хранить в памяти сведения о клиенте, установившем это соединение, а если таких соединений было установлено много (так часто и бывает), то 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 приходится начинать сеансы, чтобы отправить новые сообщения "отдыхающим" клиентам. В этом случае используются *постоянные*, или *многосеансовые*, подключения с другим сценарием работы.

1. Клиент устанавливает постоянное соединение с сервером.
2. Клиент и сервер обмениваются данными. При этом сеанс обмена может быть начат как клиентом, так и сервером.
3. Клиент разрывает соединение с сервером. Обмен данными окончен.

Здесь нужно обратить внимание на то, что соединение устанавливается только клиентом. Сервер же работает через уже установленное соединение.

Как правило, серверные компьютеры — настоящие монстры, содержащие несколько процессоров, огромные дисковые массивы впечатляющей емкости, мощные средства для подключения к Интернету 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, начинаются проблемы. Они могут проявляться в том, что сервер просто отказывается обслужить "лишних" клиентов, предлагая им подождать немного, когда нагрузка немного снизится, а то и в том, что могучий серверный компьютер просто-напросто "зависает". Такое тоже случается, и не так уж редко.

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

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

### Примечание

Некоторые интернет-сервисы, в частности, так называемые *файлобменные 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, имеющее 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 начнется 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 главный, 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: длина отдельных порций (*пакетов*) данных, способ кодирования, указания адресов получателя и отправителя, а также защита 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 формат клиентских запросов и серверных ответов: особые команды, пересылаемые клиентом серверу при запросе или передаче данных, и способ представления передаваемой информации. Кодированием передаваемой информацией особым помехоустойчивым кодом, разбиением ее на пакеты и собственно передачей занимается "чернорабочий" TCP/IP.

### Примечание

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

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

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

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

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

С понятием протокола очень тесно связано понятие порта TCP/IP. *Порт* — это своего рода воображаемый канал, по которому передаются данные с использованием одного из протоколов высокого уровня. Можно сказать, что любой канал передачи данных Интернета разделен на 65 535 небольших пронумерованных "канальчика" — именно столько портов предусматривает протокол TCP/IP.

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

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

Протокол	Используемый порт
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, который они обслуживают, на другой. Например, Web-сервер может быть настроен так, чтобы использовать 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 вполне возможно, но не рекомендуется, чтобы не обескураживать пользователей, и поэтому применяется очень редко, только в крайних случаях.

Если же сервер специально не настраивать на использование другого порта, то он будет использовать именно порт по умолчанию. Так изначально задано в его настройках.

## 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, созданный для удобства управления им. Такой участок может быть крупным или мелким или вообще состоять из одного компьютера. Каждый домен обозначается строкой текста, состоящей из английских букв.

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

Домены верхнего уровня бывают интернациональными и национальными. *Интернациональные домены* объединяют компьютеры по какому-то признаку; к ним относятся домены com (коммерческие серверы), edu (образовательные), mil (военные), org (организации, не занимающиеся компьютерами и Интернетом), net (организации, занимающиеся компьютерами и Интернетом) 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 этой организации или района.

Если теперь записать обозначения всех доменов, в которых находится нужный нам компьютер, в порядке от более мелких к более крупным, разделив их точками, мы получим *доменное имя* этого компьютера. Так, если у нас сам компьютер имеет имя comp45, отдел, в котором он стоит, — buh (бухгалтерия), организация, включающая этот отдел, — 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 сетевых 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. И всем им хватает работы.

Но вернемся к доменным именам и рассмотрим некоторые детали, которые помогут нам в дальнейшем.

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. Вот так (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.

### Примечание

Существует, правда, возможность дать одному компьютеру сразу несколько доменных имен и даже IP-адресов. Но используется это нечасто и в особых случаях. В дальнейшем для простоты мы будем считать, что одно доменное имя (и один IP-адрес) — это один компьютер.

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 (номер порта выделен полужирным шрифтом):

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

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

## Основные понятия WWW

Здесь мы выясним все о Web-страницах и Web-сайтах, узнаем, чем сайт отличается от страницы, поговорим о Web-клиентах и Web-серверах и уясним 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 по-простонародному, 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, некоторые команды его форматирования и сохраненный на жестких дисках серверного компьютера. Получив от Web-обозревателя запрос по протоколу HTTP, *Web-сервер* (серверная программа, обеспечивающая работу сервиса WWW) извлекает этот файл и отправляет его Web-обозревателю.

Но как Web-обозреватель дает понять Web-серверу, какая Web-страница ему нужна? Очень просто — он пересылает в составе запроса имя и полный путь файла, в котором она сохранена. Скажем, вот так:

```
http://comp45.buh.department.ru/somepage.html
```

Этот запрос заставит Web-сервер извлечь и отправить Web-обозревателю файл `somepage.html`.

А что такое *Web-сайт*? Это набор Web-страниц, подчиненных общей тематике и объединенных в единое целое (как — будет рассказано в *главе 2*). Web-сайт также сохраняется на жестких дисках серверного компьютера в виде набора файлов, находящихся в папках. (Конечно, папки использовать необязательно, но так удобнее, особенно если файлов много и все они разных типов.) Как видим, чисто технических отличий у Web-страницы и Web-сайта не слишком много.

А теперь поговорим о некоторых технических деталях сохранения сайта на дисках серверного компьютера.

Прежде всего, для хранения всех файлов, составляющих сайт (или Web-страницу, если рассматривать ее как "вырожденный" случай сайта), на диске серверного компьютера создается особая папка, называемая *корневой*. Все, что не находится в этой папке, автоматически исключается Web-сервером из состава сайта.

Все файлы, составляющие Web-сайт, да и Web-страницу тоже, если рассматривать ее как "вырожденный" случай сайта, сохраняются на дисках сер-

верного компьютера в особой папке. Эту папку создает для сайта человек, занимающийся настройкой и обслуживанием программы Web-сервера (или же всего серверного компьютера), — *администратор*. Полный путь данной папки заносится в настройки Web-сервера, чтобы последний смог ее найти. Все содержимое сайта должно находиться в корневой папке.

### Примечание

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. В дальнейшем и мы будем считать виртуальные папки частью сайта, если специально не сказано иначе.

Когда Web-обозреватель присылает Web-серверу запрос вида:

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

Web-сервер находит файл `somepage.html` в корневой папке сайта и отправляет его Web-обозревателю. Если же Web-обозревателю понадобится файл, находящийся не в самой корневой папке сайта, а в одной из вложенных в нее папок, он должен прислать такой запрос:

```
http://www.somesite.ru/somefolder1/somefolder2/somepage.html
```

В этом случае Web-сервер отправит Web-обозревателю файл `somepage.html`, находящийся в папке `somefolder1/somefolder2`, вложенной, опять же, в корневую папку сайта.

### Примечание

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

```
http://www.somesite.ru/somevirtualfolder1/somepage.html
```

Так, все прекрасно, все замечательно и все исключительно ясно! Но ведь мы крайне нечасто набираем в поле ввода интернет-адреса Web-обозревателя такие запросы, указывающие непосредственно на нужную нам Web-страницу. Много чаще наши запросы выглядят чуть "скромнее", например, так:

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

То есть они не указывают на файл. Как поступает Web-сервер в таком случае?

Дело в том, что одна из страниц сайта задается в качестве так называемой *страницы по умолчанию*. Именно она отправляется Web-обозревателю, если он не прислал запрос на конкретную страницу (и вообще на конкретный файл). Имя файла этой страницы задается администратором Web-сервера в его настройках — как правило, `default.htm[1]` или `index.htm[1]`.