

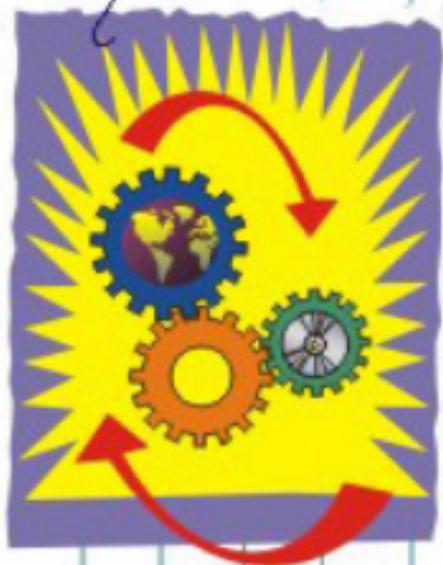


www.bhw.ru
www.bhw.kiev.ua

Д. Котеров

PHPR4

САМОУЧИТЕЛЬ



Технология Web-программирования

Web-сервер Apache

Стандартные функции

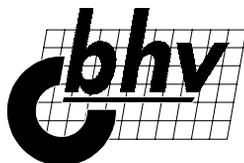
Объектно-ориентированное программирование

Язык программирования для создания эффективных Web-приложений

Дмитрий Котеров

САМОУЧИТЕЛЬ

PHR 4



Санкт-Петербург

Дюссельдорф ♦ Киев ♦ Москва ♦ Санкт-Петербург

УДК 681.3.06

Учебное пособие по использованию языка PHP версии 4 содержит обширную информацию о приемах, призванных в кратчайшие сроки сделать новичка, владеющего хотя бы одним алгоритмическим языком, Web-программистом. Рассматриваются основы протоколов HTTP и CGI, схемы разработки крупных сценариев на PHP, синтаксис языка и работа с простейшими функциями, объектно-ориентированное программирование на PHP с применением идеологии интерфейсов, манипуляции со строками и массивами, создание баз данных и многое другое.

Для программистов и Web-разработчиков

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зав. редакцией	<i>Наталья Таркова</i>
Редактор	<i>Евгений Васильев</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Дизайн обложки	<i>Игоря Цырульников</i>
Зав. производством	<i>Николай Тверских</i>

Котеров Д. В.

Самоучитель PHP 4. — СПб.: БХВ-Петербург, 2001. — 576 с.: ил.

ISBN 5-94157-071-6

© Д. В. Котеров, 2001

© Оформление, издательство "БХВ-Петербург", 2001

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 13.09.01.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 46,44.

Доп. тираж 4000 экз. Заказ №

"БХВ-Петербург", 198005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар, № 77.99.1.953.П.950.3.99 от 01.03.1999 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов

в ордена Трудового Красного Знамени ФГУП "Техническая книга"

Министерства Российской Федерации по делам печати,
телерадиовещания и средств массовых коммуникаций.

198005, Санкт-Петербург, Измайловский пр., 29.

Содержание

Предисловие	1
Чего хочет программист от своей профессии	2
Временные затраты	3
О чем эта книга	4
Общая структура книги	5
ЧАСТЬ I. ОСНОВЫ WEB-ПРОГРАММИРОВАНИЯ	9
Глава 1. Принципы работы Интернета	11
Протоколы передачи данных	11
Семейство TCP/IP	13
Адресация с Сети	14
IP-адрес	14
Доменное имя	16
Порт	19
Терминология	20
Сервер	20
Узел	21
Порт	21
Сетевой демон	22
Провайдер	22
Хост	22
Виртуальный хост	23
Хостинг-провайдер (хостер)	23
Хостинг	24
Сайт	24
HTML-документ	24
Страница (или HTML-страница)	24
Web-программирование	25
World Wide Web и URL	25
Протокол	26
Имя хоста	26
Порт	26
Путь к странице	27
Глава 2. Интерфейс CGI	28
Что такое CGI?	28
Секреты URL	29
Заголовки и метод <i>GET</i>	30
<i>GET</i>	32
<i>POST</i>	32
<i>Content-type</i>	32
<i>User-Agent</i>	33

<i>Referer</i>	33
<i>Content-length</i>	33
<i>Cookie</i>	34
<i>Accept</i>	34
Эмуляция браузера через telnet.....	34
Метод <i>POST</i>	35
Кодировки и форматы данных.....	36
Что такое формы и для чего они нужны.....	37
Передача параметров "вручную".....	38
Использование формы.....	38
Абсолютный и относительный путь к сценарию.....	39
Метод <i>POST</i> и формы.....	40
Глава 3. CGI изнутри	42
Передача документа пользователю.....	43
Заголовки ответа.....	44
Пример CGI-сценария.....	46
Передача информации CGI-сценарию.....	48
Переменные окружения.....	48
Передача параметров методом <i>GET</i>	50
Передача параметров методом <i>POST</i>	51
Расшифровка URL-кодированных данных.....	53
Формы.....	56
Тэг <i><input></i> — различные поля ввода.....	57
Тэг <i><textarea></i> — многострочное поле ввода текста.....	61
Тэг <i><select></i> — список.....	62
Загрузка файлов.....	64
Формат данных.....	64
Тэг загрузки файла (<i>file</i>).....	66
Что такое Cookies и с чем их едят.....	67
Установка Cookie.....	69
Получение Cookies из браузера.....	71
Пример программы для работы с Cookies.....	71
Авторизация.....	73
ЧАСТЬ II. ВЫБОР И НАСТРОЙКА ИНСТРУМЕНТАРИЯ.	
WEB-СЕРВЕР APACHE	77
Глава 4. Установка Apache	79
Введение: зачем нужен домашний сервер?.....	79
Дистрибутивы и ссылки.....	80
От слов к делу: установка Apache.....	80
Этап первый: установка.....	81
Этап второй: настройка файла конфигурации Apache.....	82
Этап третий: тестирование Apache.....	86
Виртуальные хосты Apache.....	88

Глава 5. Установка PHP и MySQL	93
Установка PHP	93
Настройка Apache для работы с PHP	95
Тестирование PHP	96
Установка дополнительных модулей	97
Установка MySQL	98
Тестирование MySQL	100
ЧАСТЬ III. ОСНОВЫ ЯЗЫКА PHP.....	103
Глава 6. Характеристика языка PHP	105
Интерпретатор или компилятор?	106
Достоинства и недостатки интерпретатора	108
Пример PHP-программы.....	110
Использование PHP в Web	114
Глава 7. Переменные, константы, выражения	117
Переменные	117
Типы переменных	118
Действия с переменными	121
Определение типа переменной	122
Установка типа переменной.....	123
Оператор присваивания.....	123
Ссылочные переменные	124
Жесткие ссылки.....	124
Символические ссылки	126
Некоторые условные обозначения.....	126
<i>string</i>	127
<i>int, long</i>	127
<i>double, float</i>	127
<i>bool</i>	127
<i>array</i>	128
<i>list</i>	128
<i>object</i>	128
<i>void</i>	128
<i>mixed</i>	128
Константы	129
Предопределенные константы	129
Определение констант	130
Проверка существования константы.....	130
Выражения	130
Логические выражения.....	132
Строковые выражения.....	133
Операции.....	135
Арифметические операции.....	136
Строковые операции.....	136
Операции присваивания.....	136

Операции инкремента и декремента.....	137
Битовые операции.....	137
Операции сравнения.....	138
Операция эквивалентности.....	138
Логические операции.....	140
Оператор отключения предупреждений.....	140
Глава 8. Работа с данными формы.....	143
Передача данных командной строки.....	143
Формы.....	145
Трансляция полей формы в переменные.....	146
Трансляция переменных окружения и Cookies.....	148
Трансляция списков.....	149
Трансляция массивов.....	151
Глава 9. Конструкции языка.....	153
Инструкция <i>if-else</i>	153
Использование альтернативного синтаксиса.....	154
Цикл с предусловием <i>while</i>	155
Цикл с постусловием <i>do-while</i>	156
Универсальный цикл <i>for</i>	156
Инструкции <i>break</i> и <i>continue</i>	157
Нетрадиционное использование <i>do-while</i> и <i>break</i>	159
Цикл <i>foreach</i>	160
Конструкция <i>switch-case</i>	161
Инструкция <i>require</i>	162
Инструкция <i>include</i>	163
Трансляция и проблемы с <i>include</i>	163
Инструкции однократного включения.....	164
Глава 10. Ассоциативные массивы.....	167
Создание массива "на лету". Автомассивы.....	168
Инструкция <i>list()</i>	170
Списки и ассоциативные массивы: путаница?.....	170
Инструкция <i>array()</i> и многомерные массивы.....	171
Операции над массивами.....	172
Доступ по ключу.....	172
Функция <i>count()</i>	173
Слияние массивов.....	173
Косвенный перебор элементов массива.....	175
Прямой перебор массива.....	177
Списки и строки.....	178
Сериализация.....	179
Глава 11. Функции и области видимости.....	181
Пример функции.....	182
Общий синтаксис определения функции.....	184
Инструкция <i>return</i>	185

Параметры по умолчанию.....	186
Передача параметров по ссылке.....	187
Переменное число параметров.....	188
Локальные переменные.....	190
Глобальные переменные.....	191
Массив <i>\$GLOBALS</i>	192
Статические переменные.....	194
Рекурсия.....	195
Вложенные функции.....	195
Условно определяемые функции.....	197
Передача функций "по ссылке".....	198
Возврат функцией ссылки.....	199
Пример функции: <i>Dump()</i>	201
Несколько советов по использованию функций.....	202
ЧАСТЬ IV. СТАНДАРТНЫЕ ФУНКЦИИ PHP.....	203
Глава 12. Строковые функции.....	206
Конкатенация строк.....	206
О сравнении строк и инструкции <i>if-else</i>	207
Функции для работы с одиночными символами.....	209
Функции отрезания пробелов.....	210
Базовые функции.....	212
Работа с блоками текста.....	213
Функции для преобразований символов.....	214
Функции изменения регистра.....	216
Установка локали (локальных настроек).....	217
Преобразование кодировок.....	218
Функции форматных преобразований.....	219
Работа с бинарными данными.....	221
Хэш-функции.....	223
Сброс буфера вывода.....	225
Глава 13. Работа с массивами.....	226
Сортировка массивов.....	226
Сортировка массива по значениям (<i>asort()/arsort()</i>).....	226
Сортировка по ключам (<i>ksort()/krsort()</i>).....	227
Сортировка по ключам при помощи функции <i>uksort()</i>	227
Сортировка по значениям при помощи функции <i>uasort()</i>	228
Переворачивание массива <i>array_reverce()</i>	228
Сортировка списка <i>sort()/rsort()</i>	228
Сортировка списка при помощи функции <i>usort()</i>	229
Перемешивание списка <i>shuffle()</i>	229
Ключи и значения.....	230
Комплексная замена в строке.....	231
Слияние массивов.....	232

Получение части массива	232
Вставка/удаление элементов	232
Переменные и массивы	234
Создание списка — диапазона чисел	236
Глава 14. Математические функции	238
Встроенные константы	238
Функции округления	239
Случайные числа	239
Перевод в различные системы счисления	241
Минимум и максимум	242
Степенные функции	242
Тригонометрия	243
Глава 15. Работа с файлами	244
О текстовых и бинарных файлах	244
Открытие файла	245
Конструкция <i>or die()</i>	249
Безмянные временные файлы	249
Закрытие файла	250
Чтение и запись	250
Блочные чтение/запись	251
Построчные чтение/запись	251
Чтение CSV-файла	252
Положение указателя текущей позиции	253
Функции для определения типов файлов	254
Определение типа файла	254
Определение возможности доступа	255
Определение параметров файла	255
Специализированные функции	256
Функции для работы с именами файлов	257
Функции манипулирования целыми файлами	258
Другие функции	260
Блокирование файла	261
Типы блокировок	262
Блокировки с запретом "подвисания"	265
Пример счетчика	266
Глава 16. Работа с каталогами	268
Манипулирование каталогами	268
Работа с записями	269
Пример: печать дерева каталогов	271
Глава 17. Каналы и символические ссылки	273
Каналы	273
Символические ссылки	275
Жесткие ссылки	276

Глава 18. Запуск внешних программ	277
Глава 19. Работа с датами и временем	280
Представление времени в формате timestamp	280
Работа с датами	281
Григорианский календарь	283
Глава 20. Посылка писем через PHP	285
Функция отправки письма	285
Проблема с кодировками	286
Посылка в указанной кодировке	286
Динамическая смена кодировки	287
Проблема с заголовками	287
Перспективы: создание "умной" функции для отправки писем	288
Глава 21. Работа с WWW	289
Установка заголовков ответа	289
Вывод заголовка	289
Запрет кэширования	290
Получение заголовков запроса	290
Работа с Cookies	291
Немного теории	291
Установка Cookie	292
Получение Cookie	293
SSI и функция <i>virtual()</i>	294
Эмуляция функции <i>virtual()</i>	294
Глава 22. Основы регулярных выражений в формате RegEx	296
Начнем с примеров	296
Пример первый	296
Пример второй	297
Выводы	297
Терминология	298
Использование регулярных выражений в PHP	298
Сопоставление	298
Сопоставление с заменой	299
Язык RegEx	299
Простые символы	300
Квантификаторы повторений	302
Мнимые символы	304
Оператор альтернативы	304
Группирующие скобки	305
"Карманы"	305
Дополнительные функции	308
Примеры использования регулярных выражений	309
Имя и расширение файла	309
Имя каталога и файла	309
Проверка на идентификатор	310

Модификация тэгов	310
Преобразование гиперссылок	310
Преобразование адресов E-mail.....	311
Выделение всех уникальных слов из текста.....	311
Заключение	312
Глава 23. Работа с изображениями	314
Универсальная функция <i>getImageSize()</i>	315
Работа с изображениями и библиотека GD.....	316
Пример	316
Создание изображения	317
Определение параметров изображения.....	318
Сохранение изображения	319
Работа с цветом в формате RGB	320
Создание нового цвета.....	320
Получение ближайшего цвета.....	320
Эффект прозрачности	321
Получение RGB-составляющих.....	322
Графические примитивы	322
Копирование изображений	322
Прямоугольники.....	323
Линии	324
Дуга сектора.....	324
Закраска произвольной области	324
Многоугольники.....	325
Работа с пикселями	325
Работа с фиксированными шрифтами.....	326
Загрузка шрифта.....	326
Параметры шрифта	327
Вывод строки.....	327
Работа со шрифтами TrueType.....	327
Вывод строки	328
Определение границ строки.....	329
Пример	329
Глава 24. Управление интерпретатором.....	332
Информационные функции.....	332
Настройка параметров PHP	333
<i>error_reporting</i>	334
<i>magic_quotes_gpc on/off</i>	334
<i>max_execution_time</i>	335
<i>track_vars on/off</i>	335
Контроль ошибок.....	335
Оператор отключения ошибок	336
Пример использования оператора @	337
Принудительное завершение программы	337
Финализаторы.....	338

Генерация кода во время выполнения.....	339
Выполнение кода.....	339
Генерация функций.....	341
Проверка синтаксической корректности кода.....	343
Другие функции.....	344
Глава 25. Управление сессиями.....	345
Зачем нужны сессии?.....	345
Механизм работы сессий.....	346
Инициализация сессии.....	347
Регистрация переменных.....	348
Идентификатор сессии и имя группы.....	349
Имя группы сессий.....	349
Идентификатор сессии.....	350
Другие функции.....	351
Установка обработчиков сессии.....	352
Обзор обработчиков.....	352
Регистрация обработчиков.....	354
Пример: переопределение обработчиков.....	354
Сессии и Cookies.....	357
Явное использование константы <i>SID</i>	357
Неявное изменение гиперссылок.....	358
Неявное изменение формы.....	359
Так использовать Cookies в сессиях или нет?.....	360
Глава 26. Работа с базой данных MySQL.....	361
Неудобство работы с файлами.....	362
Устройство MySQL.....	363
Соединение с базой данных.....	364
Обработка ошибок.....	365
Выполнение запросов к базе данных.....	365
Язык запросов MySQL.....	366
Создание таблицы.....	366
Удаление таблицы.....	371
Вставка записи.....	371
Удаление записей.....	371
Поиск записей.....	371
Обновление записей.....	372
Получение числа записей, удовлетворяющих выражению.....	372
Получение уникальных значений столбцов.....	372
Получение результата.....	373
Параметры результата.....	373
Получение поля результата.....	374
Получение целой строки результата.....	374
Получение информации о результате.....	375
Пример использования функций поддержки MySQL.....	377
Уникальные идентификаторы в MySQL.....	378
Работа с таблицами.....	379

Глава 27. Сетевые функции	381
Работа с сокетами.....	381
Функции для работы с DNS.....	382
Разрешение IP-адреса в доменное имя и наоборот.....	383
Корректный перевод IP-адреса в доменное имя.....	383
ЧАСТЬ V. ПРИЕМЫ ПРОГРАММИРОВАНИЯ НА РНР	387
Глава 28. Загрузка файлов на сервер	389
Multipart-формы.....	390
Тэг выбора файла.....	390
Закачка файлов и безопасность.....	391
Поддержка закачки в РНР.....	392
Простые имена полей закачки.....	392
Пример: фотоальбом.....	393
Сложные имена полей.....	395
Проблемы со сложными именами.....	396
Глава 29. Модульность программы. Написание "библиотекаря"	397
Наши требования.....	397
Библиотекарь.....	398
Работа с библиотекарем.....	404
Автоматическое подключение библиотекаря.....	405
Способ первый: использование <i>auto_prepend_file</i>	406
Способ второй: установка обработчика Apache.....	407
Обработчики Apache.....	407
Перехват обращений к несуществующим страницам.....	411
Связывание РНР с другим расширением.....	412
Решение проблемы заикливания обработчика.....	413
Глава 30. Код и шаблон страницы	415
Идеология.....	416
Двухуровневая схема.....	417
Шаблон страницы.....	417
Генератор данных.....	419
Взаимодействие генератора данных и шаблона.....	421
Недостатки.....	422
Трехуровневая схема.....	423
Шаблон страницы.....	423
Диаграммы двухуровневой и трехуровневой моделей.....	425
Интерфейс.....	426
Ядро.....	427
Проверка корректности входных данных.....	428
Шаблонизатор.....	429
Традиционное построение страниц.....	431
Что такое шаблонизатор?.....	433

Описание шаблонизатора	434
Обработчик Apache для шаблонизатора	441
Главный модуль шаблонизатора	443
"Перехват" выходного потока	449
Стек буферов	450
Проблемы с отладкой	451
Глава 31. Объектно-ориентированное программирование на PHP	453
Классы и объекты	454
Свойства объекта	454
Методы	456
Класс таблицы MySQL	456
Доступ объекта к своим свойствам	459
Инициализация объекта. Конструкторы	460
Деструктор	461
Наследование	462
Полиморфизм	464
Полноценный класс таблицы MySQL	465
Копирование объектов	478
Ссылки и интерфейсы	479
Возврат ссылки на объект	481
Возврат интерфейса	482
Глава 32. Почтовые шаблоны	486
Мини-шаблонизатор	486
Отправка и перекодирование писем	488
Пример	493
Глава 33. Разные советы	496
Разделенные вычисления	496
Использование самопереадресации	501
Запрет кэширования страниц	503
Несколько слов о флажках checkbox	504
ЧАСТЬ VI. ПРИЛОЖЕНИЯ	507
Приложение 1. Файл конфигурации Apache <i>httpd.conf</i>	509
Приложение 2. Файл конфигурации PHP <i>php.ini</i>	536
Предметный указатель	555

Предисловие

В последнее время в связи с бурным развитием сети Интернет в программировании начинает все более резко выделяться отдельная отрасль. Поначалу она не могла даже и сравниться по своей сложности с другими областями программистского ремесла, не "дотягиваясь" не только до системного, но даже и до прикладного программирования. Речь идет, конечно, о программировании сценариев для Web, или, как часто говорят, *Web-программировании*. В наши дни, однако, роль этой отрасли в структуре Интернета все более возрастает, соответственно растет и средняя оценка сложности сценариев. Многие системы (например, поисковые) по объему кода приближаются к размеру исходных кодов серьезных пакетов прикладных программ.

Представляю, как эти слова тут же вызовут бурю протеста со стороны прикладных и системных программистов, лишь мельком глянувших на программирование в Web. "Как, — заявят они, — неужели написание простейших программ на "бейсикоподобных" интерпретаторах вообще можно назвать серьезным программированием? Да с этим же справится любой начинающий изучать программирование студент, потому что эта область не вносит и не может внести каких-либо новшеств, не "изобретает" алгоритмов, и, кстати, в ней нет ничего творческого. Да и вообще, скука-то, наверное, какая..." Обычно с такими людьми можно спорить часами. Действительно, какую бы задачу им ни привели, они начинают утверждать, что решить ее очень просто, хотя на самом деле это в контексте Web, мягко говоря, оказывается не совсем так.

Что ж, отчасти такие люди правы. Поначалу все мы так считали, пока не столкнулись вплотную с тем, что называется Web-программированием. Да, в большинстве своем все программы удивляют своей кажущейся простотой. Но везде есть "подводные камни", и Web-программирование особенно ярко это доказывает. Обычно на написание сценариев уходят не месяцы и годы, а дни и недели. Но особо сложные сценарии могут потребовать значительно большего времени на разработку. Наконец, на первый взгляд работа Web-программиста кажется на редкость скучной. Но...

Все это обстоит именно таким образом, если вы программируете, что называется, "для себя", и при этом не пытаетесь каким-либо образом "автоматизировать" и упростить этот процесс. Действительно, можно получать удовольствие от написания прикладных программ (особенно нетривиальных), даже если их никто, кроме автора и его ближайших знакомых, потом не увидит. Здесь привлекает сам процесс. Вот этим-то и отличается программирование в Web: нельзя писать сценарии "для себя", это занятие действи-

тельно покажется (а возможно, так оно и есть) скучным. Зато если вы создали программу, прекрасно работающую в Интернете, через которую "проходят" сотни человек в день, и к тому же с удобным и оригинальным интерфейсом — вот тут-то и начинает вам нравиться ваша профессия.

Примечание

Лу Гринзо, один из программистов IBM, говорил: "Все программисты немного чокнутые. Это как бесконечная компьютерная игра: мы должны получать удовольствие от своей работы. Какие бы деньги нам ни платили, если в нашем ремесле нет ничего увлекательного, никто из нас не станет работать". Думаю, нам всем иногда стоит задумываться над этими словами.

Чего хочет программист от своей профессии

Давайте попробуем разобраться, чего хочет Web-программист, когда он выбирает свою профессию. Возможно, он считает, что эта стезя довольно прибыльна? Но деньги приходят, что называется, "сами собой" с накоплением опыта и получением определенных навыков, по мере того, как человек становится профессионалом. Так происходит с любой профессией, а не только с программированием. Кстати, как я немного выше упоминал, никто из профессиональных программистов не работает исключительно за деньги, основной стимул — это все-таки интерес к работе.

Идем дальше. Может быть, ему нужна известность? Конечно, этот фактор не является третьестепенным, учитывая то, что известность — гарантия, что программист всегда легко сможет найти работу. Однако, как и деньги, слава и известность также не бывают самими по себе — их необходимо заслужить. И, к тому же, много ли вы знаете известных имен программистов, действительно заслуживших свое признание практикой (Билл Гейтс не в счет, потому что он уже давно этим не занимается)? Правильно — ни одного. Разве что, может быть, кто-нибудь вспомнит доблестных создателей игры Doom, ставшей уже историей.

Но есть нечто такое, на что я уже намекал, и именно этим Web-программирование (да и вообще любая работа, происходящая в Web) резко отличается в лучшую сторону от всех доселе известных видов программирования. Вы можете быть очень хорошим прикладным или системным программистом. Однако вряд ли ваши программы будут использовать такое количество людей, которое ежедневно посещает даже и не самую популярную страничку в Интернете, "подкрепленную" Web-сценарием. Вряд ли вы получите такое количество отзывов, приобретете такое число бесплатных тестеров, усердно шлющих вам гору писем с сообщениями о неточностях и ошибках в вашем продукте, а также с отзывами и предложениями. А ведь, как известно,

заметить ошибку в программе означает "отрубить ей голову". Наконец, иногда приятно отметить для себя, что сценарий, написанный вами несколько лет назад, о котором вы почти уже и забыли думать, продолжает исправно работать "сам по себе", без всякого человеческого вмешательства.

Временные затраты

Да, я уже слышу очередные протесты "системщиков". Конечно, операционная система — безусловный "долгожитель" на множестве компьютеров. Вместе с тем, согласитесь, написать работоспособную ОС, действительно пригодную для использования (без всяких там оговорок) — довольно тяжелая работа, если не сказать большего. Под силу ли это одиночке? Сомневаюсь, что с ней в приемлемые сроки справится не то чтобы один, а десяток или даже пятьдесят человек.

Проведем несложные расчеты. На одной из конференций представитель фирмы Sun Microsystems заявил (видимо, в качестве порицания), что исходный текст последних версий Windows насчитывает порядка 50 миллионов строк. Думаю, он не очень сильно ошибся в своей оценке (как мы увидим, даже если он завысил цифру хоть в 10 раз, все равно результат будет неутешительный). В сумме это составляет около $50 \text{ млн} \times 20 \text{ байт} = 1000 \text{ Мбайт}$ (из расчета в среднем 20 символов в строке). Предположим, программист может печатать со скоростью 30 символов в минуту (разумеется, скорость собственно печати значительно выше, но ведь прежде чем что-то набирать, нужно сначала все спланировать и разработать). Таким образом, работая непрерывно, он в этом темпе создаст ОС за $1000 \text{ Мбайт} / (30/60 \text{ мин}) / 3600 \text{ с} = 555 \text{ 555 часов}$, что составит $555 \text{ 555} / 24 = 23 \text{ 148}$ дня или ровным счетом $23 \text{ 148} / 365 = 63$ года непрерывной круглосуточной работы! А ведь мы значительно завысили реальную скорость печати, да и, к тому же, нельзя 24 часа заниматься только тем, что набирать программу на клавиатуре.

Ко всему прочему, нужно еще компилировать программу, исправлять ошибки, еще раз компилировать и так до бесконечности (как это может показаться непривычному человеку). Наконец, "Нет ошибок в данной трансляции", но вдруг — логическая ошибка, и начинай все заново?.. Допустим даже ОС будет занимать не 50 миллионов строк, а только 5 миллионов. Предположим, что в команде не один, а 1000 человек. И пусть рабочий день программиста составляет 6 часов непрерывной работы. Итак, мы получим, что на написание нашей ОС этой командой уйдет $555 \text{ 555} / 10 / 1000 \times (24/6) = 222$ дня, или около семи месяцев. Что ж... Вполне неплохо, но какой ценой?.. К тому же совершенно неизвестно, получится ли в конце концов система, которая кому-то будет нужна. Представляете, полгода работы — и все напрасно?!

Разумеется, в системном и прикладном программировании существуют и другие направления. Например, можно написать какую-нибудь полезную

программу, вроде текстового процессора или браузера. Кстати, вы знаете достоверно, сколько человек писало Internet Explorer? Лучше бы и я этого не знал...

И вот мы вернулись к тому, с чего начинали: чем же так привлекательна профессия Web-программиста. Все-таки понять это в полной мере можно, лишь достаточно поработав в этой области. Самое привлекательное в ней то, что результат своей работы можно видеть через довольно короткий срок.

О чем эта книга

Книга, которую вы держите в руках, является в некотором роде *учебником* по Web-программированию. Я сделал попытку написать ее так, чтобы даже самый неподготовленный читатель, владеющий лишь основами программирования на одном из алгоритмических языков, смог овладеть большинством необходимых знаний и в минимальные сроки начать профессиональную работу в Web.

Конечно, нельзя вести разговор о программировании, не подкрепляя его конкретными примерами на том или ином алгоритмическом языке. Поэтому главная задача книги — подробное описание языка PHP версии 4, а также некоторых удобных приемов, позволяющих создавать качественные Web-программы за очень короткие сроки, получая продукты, легко модифицируемые и поддерживаемые в будущем. И хотя язык PHP постоянно изменяется, я уверен, что ему обеспечено долгое доминирование в области языков для программирования в Web, по крайней мере, в ближайшее время.

Попутно описываются наиболее часто используемые и полезные на практике приемы Web-программирования, не только на PHP. Я постарался рассказать практически обо всем, что потребуется в первую очередь для освоения профессии Web-программиста. Но это вовсе не значит, что книга переполнена всякого рода точной технической информацией. Технического материала не так много, основной "упор" сделан не на "низкий уровень", а на те методы, которые позволят в значительной степени облегчить труд программиста, начинающего работать в области Web.

В тексте много "общефилософских" рассуждений на тему "как могло бы быть, если..." или "как бы сделал я сам в этой ситуации...", они обычно оформлены в виде примечаний. Иногда я позволяю себе писать не о том, что есть *на самом деле*, а о том, как это *могло бы быть* в более благоприятных обстоятельствах. Здесь применяется метод: "расскажи сначала просто, пусть и не совсем строго и точно, а затем постепенно детализируй, освещая подробности, опущенные в прошлый раз". По своему опыту знаю, что такой стиль повествования чаще всего оказывается гораздо более плодотворным, чем строгое и сухое описание фактов. Еще раз: я не ставил себе целью написать исчерпывающее руководство в определенной области, и не стремил-

ся описывать все максимально точно, как в учебнике по математике, — наоборот, во многих местах я пытаюсь отталкиваться от умозрительных рассуждений, возможно, немного и не соответствующих истине. Основной подход — от частного к общему, а не наоборот. Как-никак, "изобретение велосипеда" испокон веков считалось лучшим приемом педагогики.

Возможно, многие детали (даже важные) я опустил, если они *не* относятся к категориям приемов:

- которые наиболее часто применяются;
- без которых нельзя обойтись в Web-программировании.

Может быть, я уделил чему-то незаслуженно мало внимания. Наконец, в этой книге, как и в любой другой (за исключением разве что старого энциклопедического словаря), есть ошибки и неточности — сразу приношу за них свои извинения. Признаюсь честно: многие примеры простых программ могут содержать синтаксические "огрехи", т. к. из-за своей простоты они никогда не были протестированы. Это не относится к крупным программам, приведенным в пятой части книги — как раз они были тщательно отлажены. Везде, где можно, присутствуют подробные комментарии практически к каждой строке программы, поэтому в основном логика описываемых действий должна быть предельно ясна.

Общая структура книги

Книга состоит из пяти частей, содержащих в общей сложности 33 главы, и двух приложений. Непосредственное описание языка PHP начинается с третьей части. Это объясняется необходимостью прежде узнать кое-что о CGI (Common Gateway Interface — Общий шлюзовой интерфейс) — первая часть, а также выбрать подходящий инструментарий и Web-сервер для программирования — вторая часть. В четвертой части разобраны наиболее полезные стандартные функции языка. Пятая часть посвящена различным приемам программирования на PHP с множеством примеров. Приложения содержат техническую информацию, которая может иногда пригодиться Web-программисту.

Теперь чуть подробнее о каждой части книги. В первой рассматриваются теоретические аспекты программирования в Web, а также основы того механизма, который позволяет писать программы в Сети. Если вы уже знакомы с этим материалом (например, занимались программированием на Perl или других языках), можете ее смело пропустить. Вкратце я опишу, на чем базируется Web, что такое интерфейс CGI, как он работает на низком уровне, как используются возможности языка HTML при программировании Web, как происходит взаимодействие CGI и HTML и многое другое. В принципе, вся теория по Web-программированию коротко изложена именно в этой части книги. Так как CGI является независимым от платформы интерфейс-

сом, материал не "привязан" к конкретному языку (хотя в примерах используется Си как наиболее универсальный язык). Если вы не знаете языка Си, не стоит отчаиваться: немногочисленные примеры на этом языке не настолько сложны, чтобы в них можно было "запутаться". К тому же, каждое действие подробно комментируется. Большинство описанных идей будет повторно затронуто в последующих главах, посвященных уже языку PHP.

Вторая часть книги довольно небольшая и состоит из разного рода дополнительной информации, связанной по большей части с серверным программным обеспечением Apache. Сервер Apache — один из самых популярных в мире, на нем построено около двух третей хостов Интернета (по крайней мере, на настоящий момент). Главное его достоинство — простое и в то же время универсальное конфигурирование, что позволяет создавать довольно сложные и большие серверы на его основе. Думаю, вряд ли в ближайшее время кто-либо будет серьезно использовать PHP под управлением какого-то другого сервера, нежели Apache. Основное внимание во второй части уделено установке и использованию Apache для Windows, поскольку, как мы увидим ниже, это очень сильно облегчает программирование и отладку сценариев. Не секрет, что подчас выбор неверного и неудобного инструментария только из-за того, что "им пользуются все", является серьезной помехой при программировании. Именно из-за этого многие "закаленные" Web-программисты "старого образца" не принимают PHP всерьез. Вторая часть книги призвана раз и навсегда решить эту проблему.

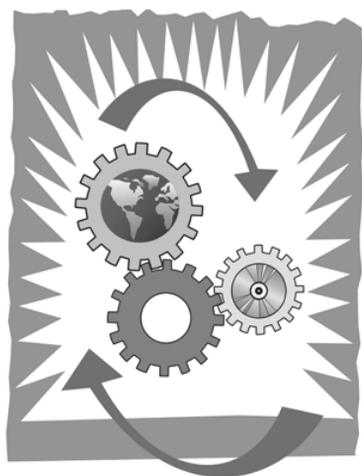
Третья часть целиком посвящена основам PHP. Язык PHP — сравнительно молодой, но в то же время удивительно удобный и гибкий язык для программирования Web. С помощью него можно написать 99% программ, которые обычно требуются в Интернете. Для оставшегося 1% придется использовать Си или Perl (или другой универсальный язык). Впрочем, даже это необязательно: вы сильно облегчите себе жизнь, если интерфейсную оболочку будете разрабатывать на PHP, а ядро — на Си, особенно, если ваша программа должна работать быстро, например, если вы пишете поисковую систему. Последняя тема в этой книге не рассматривается, поскольку требует довольно большого опыта низкоуровневого программирования на языке Си, а потому не вписывается в концепцию данной книги.

Четвертая часть может быть использована не только как своеобразный учебник, но также и в справочных целях — ведь в ней рассказано о большинстве стандартных функций, встроенных в PHP. Я группировал функции в соответствии с их назначением, а не в алфавитном порядке, как это иногда бывает принято в технической литературе. Что ж, думаю, книга от этого только выиграла. Содержание части во многих местах дублирует документацию, сопровождающую PHP, но это ни в коей мере не означает, что она является лишь ее грубым переводом. Наоборот, я пытался взглянуть на "кухню" Web-программирования, так сказать, свежим взглядом, еще помня свои собственные ошибки и изыскания. Конечно, все функции PHP описать невоз-

можно (потому что они добавляются и совершенствуются от версии к версии), да этого и не требуется, но львиная доля предоставляемых РНР возможностей все же будет нами рассмотрена.

Пятая часть книги целиком посвящена различным приемам программирования на РНР. Она насыщена всевозможными примерами программ и библиотек, облегчающими работу программиста. Если первые три части, да и четвертая в известной мере, касались Web-программирования в основном теоретически, то здесь как раз основной упор сделан на практику. Как известно, грамотное программирование и написание повторно используемого кода может сильно облегчить жизнь, поэтому один из первых приемов, рассматриваемых в пятой части — это написание системы управления модулями и библиотеками. Кроме того, вряд ли вы станете разрабатывать сайты в одиночку — скорее всего, в вашей команде будет дизайнер, HTML-верстальщик и представители других профессий. Поэтому на передний план выходит техника отделения кода от шаблона страницы сценария, чему также уделяется довольно много внимания. Дополнительно рассматриваются: загрузка (upload) файлов, реализация почтовых шаблонов, техника разделенных вычислений и т. д.

В приложениях приведена дополнительная информация, касающаяся Web-программирования. В Приложении 1 содержится полный перевод на русский язык комментариев в файле конфигурации Apache `httpd.conf`. Она может очень пригодиться вам, если вы собираетесь тесно взаимодействовать с этим сервером в своих сценариях. Приложение 2 включает аналогичный перевод комментариев, сопровождающих файл конфигурации интерпретатора РНР. Оно призвано помочь лучше систематизировать сведения о конфигурировании РНР, полученные из других глав книги (и увидеть реальный пример использования многих описанных директив).



ЧАСТЬ I.

ОСНОВЫ WEB-ПРОГРАММИРОВАНИЯ

Глава 1



Принципы работы Интернета

Протоколы передачи данных

Как и любая компьютерная сеть, Интернет основан на множестве компьютеров, соединенных друг с другом проводами, через спутниковый канал связи и т. д. Однако, как известно, одних проводов для передачи информации недостаточно — передающей и принимающей сторонам необходимо также придерживаться ряда соглашений, позволяющих строго регламентировать передачу данных, а также гарантировать, что эта передача пройдет без искажений. Такой набор правил называется *протоколом передачи*. Грубо говоря, протокол — это набор правил, который позволяет системам, взаимодействующим в рамках Интернета, обмениваться данными в наиболее удобной для них форме. Следуя сложившейся в книгах подобного рода традиции, я вкратце расскажу, что же из себя представляют основные протоколы Интернета.

Замечание

Иногда я буду называть Интернет Сетью с большой буквы, в отличие от "сети" с маленькой буквы, которой обозначается вообще любая сеть, локальная или глобальная. Тут ситуация сходна со словом "галактика": наша галактика называется Галактикой с прописной буквы, а "галактика" со строчной буквы соответствует любой другой звездной системе подобных размеров. На самом деле, сходство Сети и Галактики идет несколько дальше орфографии, и, я думаю, вы скоро также проникнетесь этой мыслью.

Необходимость некоторой стандартизации возникла чуть ли не с самого момента возникновения компьютерных сетей. Действительно, подчас одной сетью объединены компьютеры, работающие под управлением не только различных операционных систем, но нередко имеющие и совершенно различную архитектуру процессора, организацию памяти и т. д. Именно для того, чтобы обеспечивать возможность передачи между такими компьютерами, и предназначены всевозможные протоколы. Давайте рассмотрим этот вопрос чуть подробнее.

Разумеется, для разных целей существуют различные протоколы. К счастью, нам не нужно иметь представление о каждом из них — достаточно знать только тот, который мы будем использовать в Web-программировании. Та-

ковым для нас является *протокол TCP* (Transmission Control Protocol — Протокол управления передачей данных), а точнее, *протокол HTTP* (Hypertext Transfer Protocol — Протокол передачи гипертекста), базирующийся на TCP. Протокол HTTP как раз и задействуется браузерами и Web-серверами.

Заметьте, что уже в самом начале первой главы я упомянул о том, что один протокол может использовать в своей работе другой. В мире Интернета эта ситуация является совершенно обычной. Чаще всего каждый из протоколов, участвующих в передаче данных по сети, реализуется в виде отдельного и по возможности независимого программного обеспечения или драйвера. Среди них существует некоторая иерархия, когда один протокол является всего лишь "надстройкой" над другим, тот, в свою очередь — над третьим, и т. д. до самого "низкоуровневого" драйвера, работающего уже непосредственно на физическом уровне с сетевыми картами или модемами. На рис. 1.1 приведена примерная схема того, что происходит при отправке запроса браузером пользователя на некоторый Web-сервер в Интернете. Прямоугольниками обозначены программные компоненты: драйверы протоколов и программы-абоненты (последние выделены жирным шрифтом), направление передачи данных указано стрелками. Конечно, в действительности процесс гораздо более сложен, но нам сейчас нет необходимости на этом останавливаться.

Обратите внимание, что в пределах каждой системы протоколы на схеме расположены в виде "стопки", один над другим. Такая структура обуславливает то, что часто семейство протоколов обмена данными в сети Интернет называют *стеком TCP/IP* (стек в переводе с английского как раз и обозначает "стопку").

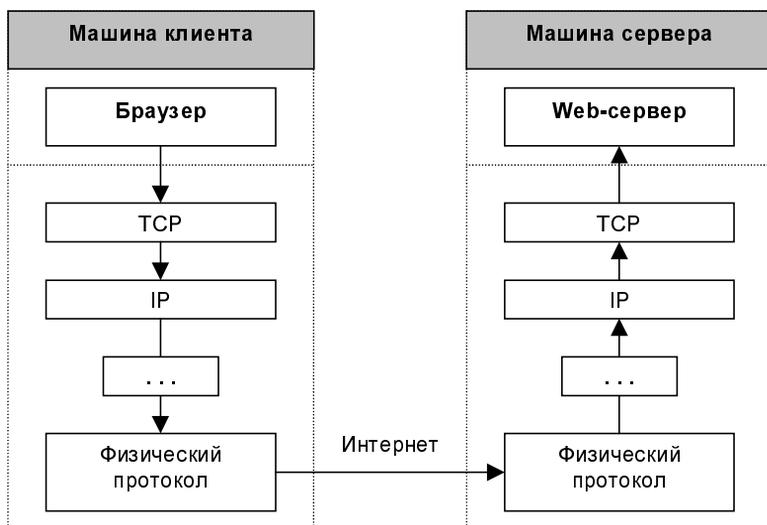


Рис. 1.1. Организация обмена данными в Интернете