

Денис Колисниченко

PHP 5/6 и MySQL 6

Разработка Web-приложений

3-е издание

Санкт-Петербург

«БХВ-Петербург»

2011

УДК 681.3.06
ББК 32.973.26-018.2
К60

Колисниченко Д. Н.

К60 PHP 5/6 и MySQL 6. Разработка Web-приложений. —
3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2011. — 528 с.: ил.
+ CD-ROM — (Профессиональное программирование)

ISBN 978-5-9775-0704-2

Даны начала программирования на PHP: установка и настройка PHP и MySQL, выбор редактора PHP-кода, основы синтаксиса и самые полезные функции PHP. Рассмотрено создание дополнительных модулей — фотогалереи, RSS-граббера, гостевой книги, собственного шаблонизатора, применение шаблонизатора Smarty, работа с SMTP, POP, MP3 и сервисом geCAPTCHA. В качестве хранилища данных использованы два сервера — самая современная версия MySQL 6 и "суперлегкий" сервер баз данных SQLite. Показано, как с помощью технологии Ajax добиться обновления данных на странице без ее перезагрузки. Представлены два реальных проекта: универсальная CMS и система продажи недвижимости.

В третьем издании подробно рассмотрена программа phpMyAdmin, интеграция PHP-сценариев и Flash-приложений, оптимизация PHP-кода, добавлен новый материал в 12 глав. Прилагаемый компакт-диск содержит дополнительные главы, все листинги из книги, а также необходимое программное обеспечение.

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

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 31.03.11.

Формат 70×100¹/₁₆. Печать офсетная. Усл. печ. л. 42,57.

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

ISBN 978-5-9775-0704-2

© Колисниченко Д. Н., 2011
© Оформление, издательство "БХВ-Петербург", 2011

Оглавление

Введение.....	1
Что нового в 3-м издании.....	1
Немного истории.....	2
PHP 6.....	3
MySQL	4
Поддержка читателей	4
 <u>ЧАСТЬ I. ТЕОРИЯ</u>	5
 РАЗДЕЛ 1. БЫСТРЫЙ СТАРТ	7
 Глава 1. Установка необходимого программного обеспечения	9
1.1. Нужно ли устанавливать программное обеспечение.....	9
1.2. Выбор PHP-редактора и FTP-клиента.....	10
1.3. Установка Apache + PHP + MySQL в Windows.....	13
1.4. Установка Apache + PHP + MySQL в Linux.....	20
1.5. Несколько советов	21
 Глава 2. Первая PHP-программа.....	22
2.1. Ваша первая программа	22
2.2. Запуск PHP-программы	23
2.3. Вывод текста без <i>echo</i>	24
2.4. Совместимость с PHP 4	25
 Глава 3. Основы синтаксиса PHP	27
3.1. Переменные.....	27
3.1.1. Правила объявления переменных. Имена переменных	27
3.1.2. Типы данных переменных	28
3.1.3. Булевы переменные	29
3.1.4. Операции над переменными	30
3.1.5. Ссылки	31
3.2. Константы.....	32

3.3. Выражения и операции.....	33
3.3.1. Что такое выражение	33
3.3.2. Арифметические операции	33
3.3.3. Логические выражения.....	34
3.3.4. Приоритеты операций	34
3.3.5. Операторы эквивалентности == и ===.....	35
3.3.6. Операции со строками.....	36
3.4. Условный оператор.....	37
3.5. Циклы.....	38
3.5.1. Цикл со счетчиком.....	38
3.5.2. Цикл <i>while</i>	38
3.5.3. Цикл <i>do-while</i>	39
3.5.4. Принудительное завершение цикла и пропуск итерации	39
3.6. Оператор выбора <i>switch-case</i>	40

РАЗДЕЛ 2. ПЕРЕДАЧА ПАРАМЕТРОВ PHP-ПРОГРАММАМ..... 43

Глава 4. Методы GET и POST..... 45

4.1. Интерфейс CGI.....	45
4.2. Метод GET	47
4.3. Метод POST	47

Глава 5. Протокол HTTP и интерфейс CGI..... 49

5.1. Специальные переменные окружения CGI.....	49
5.2. Заголовки протокола HTTP	50
5.3. Коды ответов протокола HTTP	51

Глава 6. Передача параметров посредством HTML-формы..... 53

6.1. Создание простейшей формы и ее обработка в сценарии	53
6.2. Создание пользовательского интерфейса с помощью формы	56
6.2.1. Ввод текста. Теги <i>INPUT</i> и <i>TEXTAREA</i>	57
6.2.2. Зависимые и независимые переключатели	58
6.2.3. Списки выбора	59
6.2.4. Форма для передачи файлов	61
6.2.5. Кнопки	61
6.3. Проверка параметров формы.....	62
6.3.1. Проверка корректности e-mail	63
6.3.2. Проверка правильности номера кредитной карты.....	64
6.3.3. Удаление лишних пробелов	65
6.4. Защита от спама с помощью CAPTCHA.....	65
6.5. Форма поиска в Яндексе	68

Глава 7. Запоминание параметров с помощью Cookies и сессий..... 69

7.1. Что такое Cookies и как с ними работать.....	69
7.2. Механизм сессий.....	71
7.2.1. Сессии и Cookies: преимущества и недостатки.....	71
7.2.2. Для чего нужны сессии.....	71
7.2.3. Как работает механизм сессий.....	73
7.2.4. Обход Cookies	74
7.3. Массивы и Cookies.....	75

Глава 8. Отдельное слово о директиве <i>register_globals</i>	76
8.1. Почему опасно использовать <i>register_globals</i>	76
8.2. Если <i>register_globals</i> отключена	80
8.3. PHP 6 и <i>register_globals</i>	82
РАЗДЕЛ 3. МАССИВЫ И СПИСКИ	83
Глава 9. Основные операции над массивами и списками	85
9.1. Массив и список. Цикл <i>foreach</i>	85
9.2. Функции <i>list()</i> и <i>array()</i>	87
9.3. Удаление массива	88
9.4. Слияние массивов	88
9.5. Функция <i>print_r()</i>	90
Глава 10. Функции сортировки массивов	92
10.1. Функции для сортировки массивов	92
10.2. Функция <i>sort()</i> — сортировка списка	92
10.3. Функция <i>asort()</i> — сортировка массива по значениям	93
10.4. Функция <i>ksort()</i> — сортировка по ключам	94
10.5. Функции <i>array_reverse()</i> и <i>shuffle()</i>	95
10.6. Собственная функция сортировки	95
10.7. Натуральная сортировка	96
Глава 11. Особые операции над массивами	98
11.1. Добавление и удаление элементов массива	98
11.2. Упаковка переменных в массив и их извлечение	99
11.3. Получение части массива	101
11.4. Функции автоматического заполнения массива	101
11.5. Сравнение массивов	102
11.6. Полезные операции над массивом	103
11.6.1. Вычисление суммы и произведения всех элементов массива	103
11.6.2. Проверка существования элемента в массиве	103
11.6.3. Получение случайного элемента из массива	104
11.6.4. Удаление дубликатов из массива	104
11.6.5. Получение значений и ключей массива	105
11.6.6. Замена местами значений и ключей	105
11.6.7. Подсчет значений в массиве	105
11.6.8. Замена в массиве	106
11.6.9. Поиск в массиве	106
11.6.10. Прогулка по массиву	107
РАЗДЕЛ 4. ФУНКЦИИ В PHP	109
Глава 12. Полезные стандартные функции	111
12.1. Генератор случайных чисел	111
12.2. Дата и время	112
12.2.1. Кратко о <i>timestamp</i>	112
12.2.2. Функции <i>strtotime()</i> и <i>checkdate()</i>	112
12.2.3. Вывод даты	113
12.3. Математические функции	115

Глава 13. Функции для работы со строками.....	116
13.1. Основные строковые функции.....	116
13.2. Специальные функции замены	119
13.3. Преобразование строки	120
13.4. Функции преобразования кодировок	122
13.5. Функции для работы с отдельными символами строки	122
13.6. Функция <i>md5()</i> и другие функции шифрования/хэширования.....	123
13.7. Функция <i>explode()</i> : выделение подстрок.....	124
13.8. Статистические функции.....	124
13.9. Функции вывода текста	125
13.10. Установка локали.....	127
13.11. Форматирование чисел и денежных величин.....	128
Глава 14. Работаем с файлами и каталогами	129
14.1. Права доступа в UNIX.....	129
14.2. Чтение файла	131
14.2.1. Использование функций <i>fopen()</i> и <i>fread()</i>	132
14.2.2. Использование функции <i>file()</i> : построчное чтение файла	134
14.2.3. Чтение всего файла: функция <i>file_get_contents()</i>	135
14.3. Запись файла	135
14.4. Создание временных файлов	136
14.5. Работа с CSV-файлами	136
14.6. Специальные функции для работы с файлами	139
14.6.1. Функции для работы с именами файлов	139
14.6.2. Работа с правами доступа.....	139
14.6.3. Копирование, переименование и удаление файлов	140
14.6.4. Время доступа к файлу	141
14.6.5. Другие полезные функции	142
14.7. Совместный доступ к файлу	142
14.8. Функции для работы с каталогами	143
Глава 15. Вывод графических изображений средствами PHP.....	145
15.1. Библиотека GD	145
15.1.1. Получение информации об изображении	145
15.1.2. Конвертирование графических форматов	148
15.1.3. Вывод текста поверх картинки	150
15.1.4. Прозрачность.....	153
15.2. Изменение размера изображения	153
15.3. Создание водяных знаков.....	155
Глава 16. Работа с сетевыми сокетами в PHP. Сетевые функции.....	157
16.1. Еще раз о том, что такое сокет	157
16.2. Использование сокетов.....	157
16.3. Пример использования сокетов	158
16.4. Блокирующий и неблокирующий режимы сокета	161
16.5. DNS-функции	162
Глава 17. Собственные функции	163
17.1. Зачем нужны собственные функции	163
17.2. Особенности функций в PHP	163

17.3. Объявление функции	164
17.4. Области видимости функции	165
17.5. Вложенность функций	165
17.6. Переменное число аргументов	167
17.7. Передача массивов в качестве параметров	168

РАЗДЕЛ 5. СЕРВЕРЫ БАЗ ДАННЫХ MYSQL 6 И SQLITE 169

Глава 18. Основы SQL 171

18.1. Немного истории	171
18.2. Преимущества SQL	171
18.3. Как выглядят запросы	172
18.4. Что такое база данных	173
18.5. Создание таблиц	174
18.6. Добавление записей в таблицу	177
18.7. Обновление записей	178
18.8. Выборка записей	179
18.9. Удаление записей	180
18.10. Встроенные функции	181
18.11. Группировка записей. Сложные запросы	181

Глава 19. Функции для работы с MySQL 185

19.1. Подключение к серверу MySQL	185
19.2. Несколько MySQL-соединений	187
19.3. Передача запросов серверу	188
19.4. Работа с базой данных. Создание базы данных	191
19.5. Функция <i>mysql_real_escape_string(\$content)</i>	192

Глава 20. Альтернативная база данных SQLite..... 193

20.1. MySQL vs SQLite: что лучше	193
20.2. Открытие базы данных	195
20.3. Передача запросов	195
20.4. Работа с результатом запроса	196
20.5. Список PHP-функций для работы с SQLite	197

Глава 21. Полезные приемы при работе с MySQL 6..... 199

21.1. Выбор кодировки	199
21.2. Сортировка: вывод новинок. Вывод случайных записей	200
21.3. Постраничный вывод таблицы	201

РАЗДЕЛ 6. ИНСТРУМЕНТЫ ДЛЯ СОЗДАНИЯ СЛОЖНЫХ ПРОЕКТОВ.... 207

Глава 22. Разработка собственного шаблонизатора 209

22.1. Организация файлов и каталогов проекта	209
22.2. Выносим параметры в отдельный файл	212
22.3. Подключение дополнительных файлов	213
22.3.1. Инструкции <i>include</i> и <i>require</i>	213
22.3.2. Альтернативный способ подключения сценариев	213
22.3.3. Инструкции <i>include_once</i> и <i>require_once</i>	215
22.4. Шаблоны	215

Глава 23. Шаблонизатор Smarty	219
23.1. Что такое Smarty	219
23.2. Установка Smarty	219
23.3. Создание setup.php	222
23.4. Разработка шаблонов Smarty	223
23.4.1. Комментарии в шаблонах	223
23.4.2. Переменные в Smarty.....	224
23.4.3. Файлы конфигурации шаблонов.....	225
23.4.4. Служебная переменная <code>{<i>\$smarty</i>}</code>	226
23.4.5. Модификаторы переменных	227
23.4.6. Стандартные (встроенные) функции Smarty	230
Функции <code>{include}</code> и <code>{insert}</code>	230
Функция <code>{foreach}</code>	230
Функции <code>{if}</code> , <code>{elseif}</code> , <code>{else}</code>	232
Функция <code>{capture}</code>	233
Функция <code>{php}</code>	233
Функция <code>{strip}</code>	233
23.4.7. Пользовательские функции Smarty	234
Функция <code>{assign}</code>	234
Функция <code>{cycle}</code>	234
Функция <code>{fetch}</code>	235
Функции <code>{html_checkboxes}</code> и <code>{html_radios}</code>	235
Функция <code>{html_image}</code>	236
Функция <code>{html_select_date}</code>	236
Функция <code>{html_select_time}</code>	237
Функция <code>{html_table}</code>	238
23.5. Smarty для программиста	239
23.5.1. Специальные переменные	239
23.5.2. Полезные методы класса <i>Smarty</i>	241
Глава 24. Объектно-ориентированное программирование.....	242
24.1. Основы ООП	242
24.2. Классы и объекты	243
24.3. Конструкторы и деструкторы класса	245
24.4. Наследование классов. Полиморфизм	246
24.5. Новые возможности PHP 5/6	247
24.5.1. Область видимости членов класса.....	247
24.5.2. Абстрактные классы и методы	248
24.5.3. Служебное слово <i>final</i>	248
24.5.4. Клонирование объектов	249
24.5.5. Обработка исключительных ситуаций	250
24.5.6. Константы-члены класса	251
24.5.7. Статические члены класса.....	251
24.5.8. Оператор <i>instanceof</i>	252
24.5.9. Итераторы.....	252
24.5.10. Пространства имен	252
Глава 25. Механизм сессий	254
25.1. Для чего нужны сессии.....	254
25.2. Как работает механизм сессий.....	255

25.3. Обход Cookies	257
25.4. Сценарий аутентификации	257
Глава 26. Введение в PEAR.....	262
26.1. Серьезные проекты и PEAR.....	262
26.2. Пример использования класса <i>DB</i>	263
Глава 27. Контроль версий	267
27.1. Выбор системы контроля версий	267
27.2. Практическое использование TortoiseHG (Mercurial)	268
27.3. Просмотр внесенных изменений	269
Глава 28. Тестирование PHP-сценариев	273
28.1. Программа работает, но не так, как нам нужно	273
28.2. "Самодельные" точки останова	274
28.3. Система автоматического тестирования.....	275
28.4. Директива <i>error_reporting</i>	279
<u>ЧАСТЬ II. ПРАКТИКА</u>.....	281
РАЗДЕЛ 7. РАЗРАБОТКА ОСНОВНЫХ ЭЛЕМЕНТОВ САЙТА	283
Глава 29. Загрузка файлов на сервер.....	285
29.1. Что нужно знать о загрузке файлов на сервер.....	285
29.2. Реализация загрузки файла	288
29.3. Загрузка нескольких файлов	290
29.4. Проблемы при загрузке файлов.....	292
Глава 30. Использование FTP-функций.....	293
30.1. Функции для работы с FTP	293
30.2. Примеры использования FTP-функций	296
Глава 31. Отправка и прием почты	299
31.1. Отправка почты средствами PHP — функция <i>mail()</i>	299
31.2. Отправка писем с вложениями — класс <i>HtmlMimeMail</i>	300
31.2.1. Отправка сообщения.....	300
31.2.2. Проблемы при отправке сообщения	304
31.2.3. MIME-типы	305
31.3. Класс <i>PHPMailer</i> . Разработка сценария автоматической рассылки прайс-листа.....	306
31.4. Получение писем по протоколу POP3	310
Глава 32. Работа с RSS: получаем новости на сайт	314
32.1. Краткие сведения о RSS	314
32.2. Формат RSS-файла.....	314
32.3. Написание сценария импорта новостей.....	317
32.4. Подключение файла <i>import.php</i> к сайту.....	319
32.5. Создание граббера новостей	320

Глава 33. Поиск с использованием регулярных выражений.....	324
33.1. Нужно что-то найти.....	324
33.2. Язык регулярных выражений RegEx.....	325
33.3. Управляющие конструкции	327
33.3.1. Квантификаторы	327
33.3.2. Альтернативный оператор 	327
33.3.3. Скобки.....	327
33.4. Псевдосимволы	328
33.5. Практическое использование RegEx-функций	328
33.6. Фильтры, или Конец эры регулярных выражений.....	332
Глава 34. Работаем с MP3.....	334
34.1. Формат MP3	334
34.2. Библиотека PEAR	335
34.3. Вывод ID3-тегов.....	336
34.4. Редактирование ID3-тегов.....	337
34.5. Удаление тега	338
Глава 35. Технология Ajax	339
35.1. Что такое Ajax	339
35.2. Ваше первое Ajax-приложение	340
РАЗДЕЛ 8. РАЗРАБОТКА ТИПИЧНОГО САЙТА.....	345
Глава 36. Создание простейшего движка сайта.....	347
36.1. Планирование движка	347
36.1.1. Зачем нужно разрабатывать собственный движок	347
36.1.2. Необходимые нам функции движка	348
36.1.3. "Принципиальная схема" движка	350
36.2. Основные функции движка	351
36.2.1. Разработка TPL-шаблона	351
36.2.2. Файл настроек	352
36.2.3. Основной файл CMS — index.php	353
36.2.4. Проектирование базы данных.....	355
Таблица static	355
Таблица cats.....	356
Таблица pages.....	357
36.2.5. Иерархическая структура сайта	358
Алгоритм работы меню.....	358
Сценарий menu.php.....	360
Вывод содержимого раздела и страницы	365
36.3. Дополнительные функции движка	369
36.3.1. Вывод информации из таблицы static	369
36.3.2. Функция вывода содержимого HTML-файла.....	370
36.3.3. Версия для печати.....	371
36.4. Где взять листинги этой главы.....	372
Глава 37. Создание фотогалереи	373
37.1. Постановка задачи	373

37.2. Загрузка изображений на сервер	373
37.3. Вывод галереи.....	377
Глава 38. Гостевая книга	380
38.1. Пережиток прошлого?	380
38.2. Разработка базы данных и структура гостевой книги	381
38.3. Вывод гостевой книги	381
38.4. Добавление записей в гостевую книгу	384
38.5. Сервисный сценарий gb_service.php	387
Глава 39. Интеграция галереи LiveJournal и вашего сайта	389
39.1. Что такое Живой журнал.....	389
39.2. Интеграция фотогалереи LiveJournal и сайта	390
39.3. Настройка внешнего вида галереи	394
Глава 40. Создание счетчика сайта	396
40.1. Постановка задачи	396
40.2. Файл конфигурации	396
40.3. Разработка таблиц counter и ipaddr.....	397
40.4. Сценарий counter.php.....	398
40.5. Сценарий reset_counter.php	402
Глава 41. Статистика сайта	404
41.1. Методы сбора статистики	404
41.2. Программы-анализаторы журналов Web-сервера.....	405
41.3. Системы статистики	407
Глава 42. Голосования (опросы).....	410
42.1. Разработка собственной системы голосования	410
42.2. Разработка сценария poll_form.php	412
42.3. Сценарий poll_process.php.....	413
42.4. Сценарий poll_results.php	414
РАЗДЕЛ 9. СЛОЖНЫЙ ПРОЕКТ: САЙТ ПО ПРОДАЖЕ НЕДВИЖИМОСТИ	419
Глава 43. Постановка задачи.....	421
43.1. Функции будущего проекта	421
43.2. Разработка базы данных	422
Глава 44. Разработка основной части сайта.....	431
44.1. С чего начать	431
44.2. Реализация основных функций системы.....	432
44.2.1. Аутентификация пользователей	432
44.2.2. Вывод VIP-объявлений	435
44.2.3. Вывод объявлений об услугах	437
44.2.4. Вывод рекламных баннеров.....	438
44.2.5. Постраничный вывод объявлений	438
44.3. На что обратить внимание	441

Глава 45. Разработка панели администрирования	447
45.1. Функции панели управления	447
45.2. Управление VIP-объявлениями	448
45.3. Массовая отправка электронного сообщения	449
45.4. Общие операции с базой данных	450
45.5. Редактирование статей, новостей и контактов	450
РАЗДЕЛ 10. ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ САЙТА	451
Глава 46. SSL-соединения	453
46.1. Защищаем передаваемые данные	453
46.2. Настройка SSL в DirectAdmin	454
46.3. SSL-переменные	456
Глава 47. Защита PHP с помощью конфигурационного файла	458
47.1. Конфигурационный файл <code>php.ini</code>	458
47.2. Отключение потенциально опасных функций	460
47.3. Рекомендованные значения некоторых конфигурационных директив	460
Глава 48. Защита сайта от атак	461
48.1. Сайт в опасности	461
48.2. Два самых распространенных метода взлома	462
48.3. Межсайтовый скриптинг	462
48.4. SQL-инъекции	464
48.5. Флуд	468
48.6. Защита форума PHPBB2 от спамеров	469
ПРИЛОЖЕНИЯ	471
Приложение 1. Программа phpMyAdmin	473
Действия над таблицами	474
Создание новой таблицы. Изменение структуры таблицы	476
Вставка записей	477
Обзор таблицы	477
Выполнение произвольного SQL-кода	477
Резервное копирование БД	479
Приложение 2. Flash и PHP	480
Приложение 3. Оптимизация PHP-кода	484
ПЗ.1. Правильное использование <code>echo</code>	484
ПЗ.2. Длина переменных	485
ПЗ.3. Правильное обращение к элементам массива	485
ПЗ.4. Циклы	486
ПЗ.5. Чтение файлов	487
ПЗ.6. Вывод текста: <code>echo</code> против <code>printf</code>	489
ПЗ.7. "Лишние" вызовы функций	491

ПЗ.8. Контрольные замеры	492
ПЗ.8.1. Чтение файла	492
ПЗ.8.2. Вывод содержимого файла	493
ПЗ.9. Выводы	495
 Приложение 4. Описание компакт-диска	496
 Предметный указатель	499

Введение

PHP (Hypertext Preprocessor) — один из самых популярных языков программирования, используемый для разработки Web-приложений. В настоящее время PHP поддерживается подавляющим большинством хостинг-провайдеров, что делает его чуть ли не основным языком, с помощью которого можно разработать любой интернет-проект, от простенького сайта до крупного портала.

На PHP написано огромное количество как отдельных скриптов, так и завершенных проектов — форумов, систем управления контентом и др.

Что нового в 3-м издании

Начну с самого начала — с *главы 1*. В ней появилась подробная информация об использовании комплекса программ XAMPP, представляющего собой связку Apache + PHP + MySQL (и много чего еще). Вообще-то XAMPP довольно прост и удобен в использовании и ничем не хуже других подобных проектов (вроде Denwer), но почему-то у начинающих пользователей возникает много вопросов относительно использования этой программы. Поэтому и пришлось рассмотреть ее подробно в *главе 1*.

В *главе 2* вы найдете информацию о директиве `open_short_tag` и о том, как запустить в PHP 5 старые сценарии, оптимизированные под PHP 3/4.

Современные сайты страдают от спама. Если у вас есть форма добавления комментария или гостевая книга, вам придется бороться со спаммерами. Чтобы облегчить вам жизнь, в *главе 6* находятся инструкции по установке CAPTCHA-формы из проекта reCAPTCHA (Google). Также в *главе 6* приводится код формы поиска Яндекс.

Существенные изменения были произведены в *главах 9–13*. Теперь в этих главах рассмотрены все полезные PHP-функции, которые обрабатывают массивы и строки. Это не означает, что в книге описаны все возможные функции — для этого есть руководство по PHP. Но в книгу я включил все функции, которые посчитал полезными с точки зрения практического применения. Вы найдете множество полезных примеров использования этих функций на практике.

В *главе 31* описан новый класс отправки электронных сообщений `PHPMailer` и приведен пример сценария автоматической рассылки прайс-листа клиентам абстрактной фирмы. Этот сценарий пригодится вам на практике.

Несколько небольших "багов" было исправлено в движке для сайта, разработка которого рассматривается в *главе 36*. А в *главе 35* представлен второй вариант Ajax-приложения, отправляющего данные на сервер только после нажатия кнопки отправки, что в некоторых случаях оказывается более удобным решением.

В 3-м издании появилось три новых приложения. В первом приложении рассматривается популярная программа `phpMyAdmin`. Не спору, описание `phpMyAdmin` было и в 1-м издании, но не такое подробное, как в этом. Во втором приложении рассматривается взаимодействие Flash-приложения и PHP-сценария, а в третьем описывается оптимизация PHP-кода, что пригодится любому профессиональному PHP-программисту.

Особое внимание было уделено предметному указателю. Теперь предметный указатель значительно облегчит поиск нужной вам функции.

В каталоге `Edition_1` на прилагаемом диске вы по-прежнему найдете материал, удаленный из 1-го издания. Аналогичного каталога для 2-го издания не будет, поскольку в 3-м издании из книги ничего не удалялось, а, наоборот, был добавлен новый и полезный материал.

Прилагаемый диск также был обновлен. На него были помещены новые версии программного обеспечения.

Немного истории

В 1994 г. датский программист Расмус Лердорф (Rasmus Lerdorf) создал надстройку над Perl/CGI (набор скриптов) для вывода и учета посетителей своего сайта и назвал ее `Personal Home Page` (отсюда и произошло первоначальное название PHP).

Но Perl — довольно медленный интерпретатор, и вскоре его производительности перестало хватать, поэтому разработчик написал на языке C новый интерпретатор и назвал его PHP/FI (`Personal Home Page / Form Interpreter`). В новом интерпретаторе четко прослеживались черты Perl, например символ доллара в начале имени переменной. Тогда же были заложены черты современного PHP, такие как автоматическая обработка форм, встраиваемость в HTML и др.

В 1997 г. появилась вторая версия — PHP/FI 2.0. Она была установлена примерно на 50 тыс. серверов. С одной стороны, цифра внушительная, но это всего 1% от общего числа интернет-серверов.

В 1998 г. появился PHP 3.0. Именно с этой версии PHP стал рекурсивным акронимом PHP: `Hypertext Preprocessor`. Третья версия пользовалась существенно большим успехом, чем вторая: она была установлена на 10% всех интернет-серверов, а это уже не мало!

PHP 3.0 частенько критиковали за медленное ядро, т. к. сценарии выполнялись очень долго. Поэтому разработку версии PHP 4 начали с переработки ядра. Начи-

ная с PHP 4 интерпретатор транслирует PHP-код во внутреннее представление, а потом выполняет (а не выполняет сценарий строка за строкой, как PHP 3), благодаря чему существенно повысилась производительность сценариев. Четвертая версия PHP появилась в мае 2000 г. Обновления для этой версии PHP выпускались до конца 2007 г. А с августа 2008 г. поддержка PHP 4 была прекращена.

Пятая версия PHP появилась в июле 2004 г. Было переработано ядро Zend, что увеличило эффективность интерпретатора, добавлена поддержка XML, переработана поддержка ООП. Теперь PHP является полноценным объектно-ориентированным языком программирования, а его объектная модель во многом схожа с моделью Java. В данное время доступна версия 5.3, которая пока еще будет поддерживаться, несмотря на наличие шестой версии PHP.

PHP 6

Итак, что же нового в PHP 6? Прежде всего, в шестой версии значительно улучшена поддержка Unicode. Получить список всех функций, поддерживающих Unicode, можно по адресу http://www.php.net/~scoates/unicode/render_func_data.php.

Поддержка Unicode очень важна. Раньше, когда в знаковый генератор можно было загрузить только 256 символов, поддержка разных языков была ограничена, поэтому и были созданы разные кодировки (иногда даже несколько) для одного языка. Русский язык не стал исключением. Для него было разработано три кодировки: CP866 (DOS), Windows-1251 (или CP-1251, Windows) и KOI8-R (UNIX/Linux). При использовании разных кодировок всегда есть проблема их конвертирования. Например, символ "Б" во всех этих кодировках имеет разный код.

Первая версия Unicode, представленная в 1991 г., позволяла хранить 65 536 символов. Текущая версия Unicode позволяет хранить 2 147 483 648 символов, чего вполне достаточно для хранения символов всех языков, в частности кириллицы.

Когда все программное обеспечение будет переведено на Unicode (а это произойдет в ближайшем будущем), проблема несовместимости кодировок исчезнет. Подробно о Unicode вы можете прочитать по адресу <http://ru.wikipedia.org/wiki/Unicode>.

Второе существенное усовершенствование PHP 6 — это поддержка пространств имен, что позволяет избежать коллизий между именами функций и классов. Подробно о пространствах имен мы поговорим в *главе 24*.

Кроме того, PHP 6 обеспечивается "родная" (без использования дополнительных классов) поддержка SOAP и XML. Правда, рассмотрение SOAP и XML выходит за рамки этой книги, но все же вы должны знать, что такая поддержка есть уже на уровне PHP.

Вместе с тем некоторые функции из новой версии были удалены. В частности из состава PHP 6 исключены следующие директивы (и выполняемые ими функции):

- ☐ `magic_quotes` — управляла экранированием кавычек;
- ☐ `register_globals` — регистрировала глобальные переменные;

- ❑ `register_long_arrays` — регистрировала длинные массивы;
- ❑ `safe_mode` — безопасный режим, но на самом деле он не делал безопаснее сценарии, поэтому его и устранили.

Исключение этих директив позволило сделать PHP-сценарии более защищенными. Не спору, с включенными директивами `magic_quotes` и `register_globals` программировать было несколько удобнее, и разработчики очень часто жертвовали безопасностью ради комфорта. Сейчас же наоборот — безопасность на первом месте.

MySQL

Ни один серьезный Web-проект не обходится без базы данных, ведь хранение данных в файлах давно считается пережитком прошлого, хотя до сих пор можно встретить проекты, работающие по старинке. PHP поддерживает много разных баз данных, но стандартом де-факто является сервер баз данных MySQL. На сегодняшний день доступна уже шестая версия этого популярного сервера. Кроме MySQL в этой книге мы рассмотрим еще одну альтернативную систему управления базами данных — SQLite. Почему именно SQLite? На заре пятой версии PHP ходили слухи, что новая (т. е. пятая на тот момент) версия PHP из-за лицензионных разногласий не будет вообще поддерживать MySQL. Поэтому возникла необходимость в легком и простом сервере баз данных. Им стал SQLite. К счастью, поддержка MySQL не была удалена из PHP 5, и обе последние версии (5 и 6) полностью поддерживают MySQL.

Поддержка читателей

Если при чтении книги у вас возникнут какие-либо вопросы, вы всегда можете задать их на форуме сайта **www.dkws.org.ua**.



ЧАСТЬ I

ТЕОРИЯ



I. РАЗДЕЛ 1

Быстрый старт

Глава 1.	Установка необходимого программного обеспечения
Глава 2.	Первая PHP-программа
Глава 3.	Основы синтаксиса PHP

ГЛАВА 1



Установка необходимого программного обеспечения

1.1. Нужно ли устанавливать программное обеспечение

При написании этой книги я вообще сомневался в целесообразности данной главы. Почему? Если вы купили эту книгу, значит, со временем собираетесь профессионально заниматься PHP (разрабатывать сайты, сценарии) или же планируете создать собственный сайт на PHP. Следовательно, вам понадобится хостинг с поддержкой PHP. Учитывая то, что PHP-хостинг сейчас стоит очень дешево, как и безлимитный Интернет, то все, что нужно для PHP-разработки, — это какой-нибудь текстовый редактор с поддержкой PHP-синтаксиса.

Все, что вам нужно, — это написать сценарий. В свою очередь, сценарий — это обычный текстовый файл. Как только сценарий будет готов, вам его нужно загрузить по FTP на ваш хостинг и запустить через браузер. При этом совсем не имеет значения, какую операционную систему вы используете: Windows, Linux или даже MacOS.

Ранее имело смысл устанавливать на свой компьютер Apache + PHP + сервер баз данных MySQL. Сейчас можно производить разработку и отладку сценариев непосредственно на сервере. Так даже правильнее, поскольку конфигурации Apache и PHP на локальном и удаленном сервере в 99% случаев будут различны, поэтому при переносе сценариев могут возникнуть нюансы. Какие именно? Начиная с самых безобидных вроде отличающихся параметров базы данных MySQL до разницы в версии PHP. С параметрами базы данных все просто: при переносе сценария с одного сервера на другой не забывайте проверять параметры доступа, иначе ваши сценарии откажутся работать. Чуть позже мы продемонстрируем пример, позволяющий легко избавиться от этой проблемы. А вот с версией PHP и установленными PHP-расширениями вообще могут возникнуть глобальные проблемы. Помню, я столкнулся со следующей проблемой. У меня был отлично работающий проект, отправляющий прайс-листы клиентам одной из поддерживаемых мною организаций. Сценарий был предельно простым: пользователь загружал новый прайс-лист, а сценарий отправлял его всем клиентам, адреса которых были в базе данных. Для

отправки сообщений использовался класс `PHPMailer`. После переноса на другой сервер сценарий перестал запускаться, причем даже не выдавал сообщение об ошибке. Первым делом я запустил функцию `phpinfo()`:

```
<?php
phpinfo();
?>
```

Результат меня немного поразил. На сервере была установлена четвертая (!) версия PHP. Учитывая, что уже есть шестая версия, PHP 4 выглядит антиквариатом. Решить проблему удалось путем загрузки класса `PHPMailer` для PHP 4. Хорошо, что такой до сих пор сохранился на серверах разработчиков!

Иная распространенная проблема — отсутствие на другом сервере необходимых вам расширений. Например, вы создали галерею картинок, использующую функции из библиотеки GD. Но ваша галерея не будет работать, если на сервере не установлено расширение GD (PHP-библиотека GD). Поэтому, прежде чем размещать сценарий на сервере, вызовите функцию `phpinfo()` и убедитесь, что на сервере установлена нужная версия PHP и нужные расширения.

Как узнать, какая версия PHP вам нужна? Все зависит от того, как вы писали свой код. Если вы не использовали функции PHP 5, например `scandir()`, `file_get_contents()` и др., то вам, по сути, хватит PHP 4. А вот если вы использовали пространства имен и другие возможности PHP 6, тогда вам нужна только шестая версия PHP. Все новые версии PHP поддерживают все функции предыдущих версий. Однако не нужно забывать об особенностях PHP 6. Если вы ранее программировали на PHP, то могу поспорить, что вы привыкли работать с включенной директивой `register_globals`. В PHP 6 ее нет, поэтому большую часть ваших сценариев придется переписать с учетом этой особенности. Если вы не программировали до этого на PHP, то не беспокойтесь по этому поводу: все сценарии в этой книге будут написаны с учетом особенностей PHP 6.

Итак, если вы не собираетесь устанавливать Web-сервер Apache, интерпретатор PHP и сервер баз данных на свой компьютер, вам понадобятся две программы — текстовый редактор и FTP-клиент.

1.2. Выбор PHP-редактора и FTP-клиента

Из всех PHP-редакторов, которыми я пользовался, мне больше всего понравились два редактора — PHP Expert Editor и Zend Studio (вы без проблем найдете сайты разработчиков этих редакторов). Оба редактора коммерческие, но они стоят своих денег. PHP Expert Editor (рис. 1.1) обеспечивает не только подсветку синтаксиса, но и умеет проверять синтаксис сценария. Правда, для проверки синтаксиса вам нужно установить PHP (об этом позже) и указать его в настройках программы.

ПРИМЕЧАНИЕ

Условно-бесплатную версию программы PHP Expert Editor вы найдете на прилагаемом диске в каталоге software.

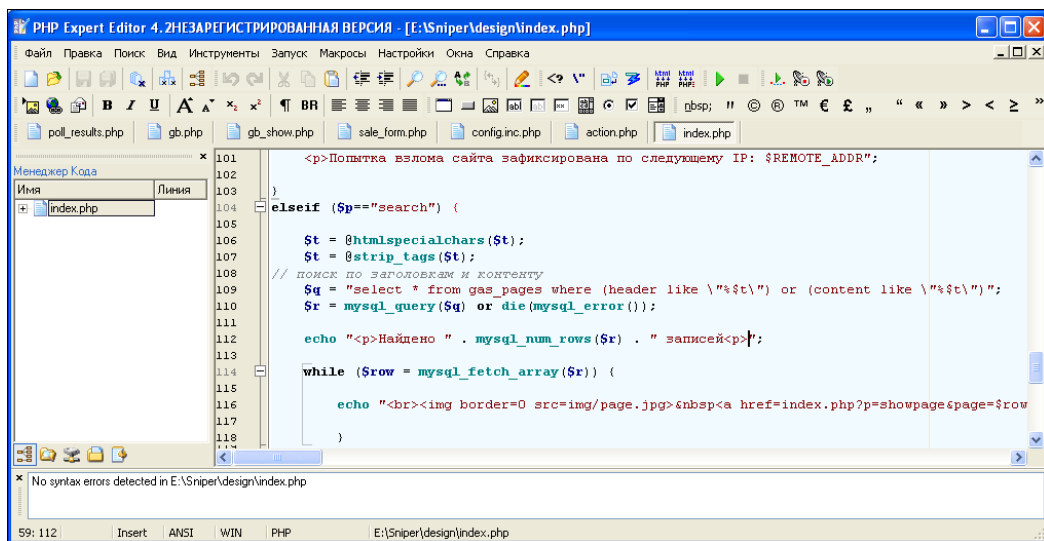


Рис. 1.1. Программа PHP Expert Editor

Для этого выполните команду **Запуск | Настройки**, в появившемся окне перейдите на вкладку **Интерпретаторы скриптов** (рис. 1.2) и укажите путь к интерпретатору PHP. Для проверки синтаксиса нажмите клавишу <F7>.

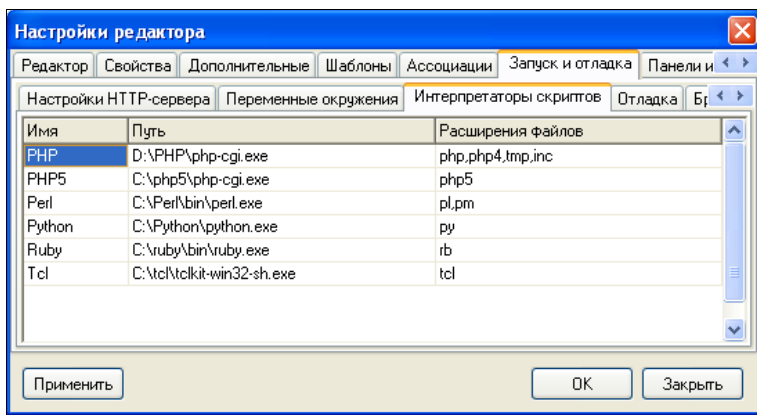


Рис. 1.2. Путь к интерпретатору PHP

Возможности Zend Studio куда шире (рис. 1.3). Чего стоит только автодополнение кода, когда программа "читает" ваши мысли и дописывает за вас PHP-код. Если вы работали с Delphi и другими продуктами Borland, то поймете, о чем я говорю. Поверьте, очень удобно.

В качестве FTP-клиента могу порекомендовать бесплатную программу FileZilla (рис. 1.4), которая доступна как для Windows, так и для Linux. Версия 3.3.5 этого FTP-клиента находится на прилагаемом диске в каталоге software.

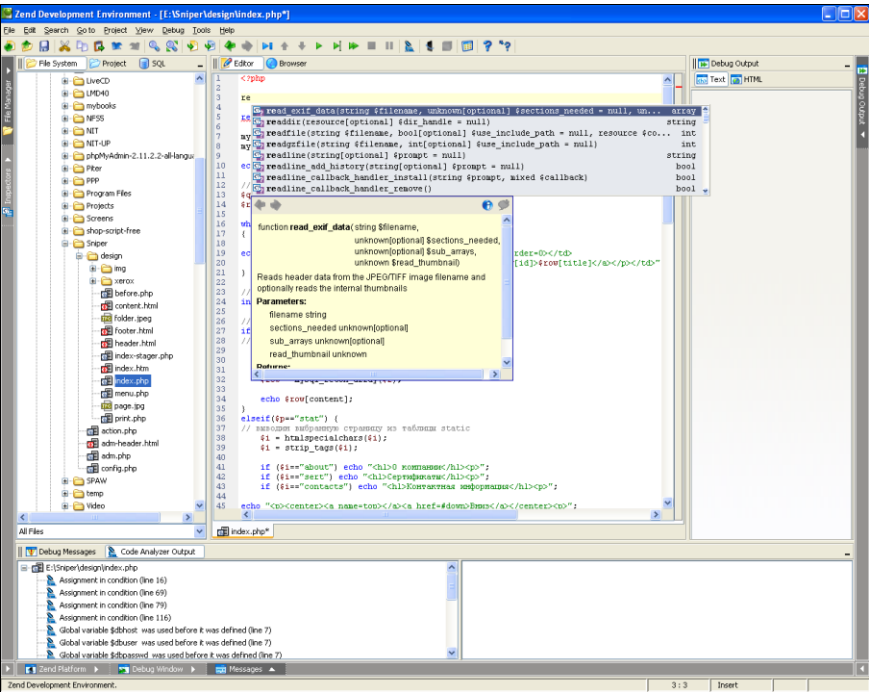


Рис. 1.3. Zend выводит не только имена функций, но и краткую справку по каждой функции

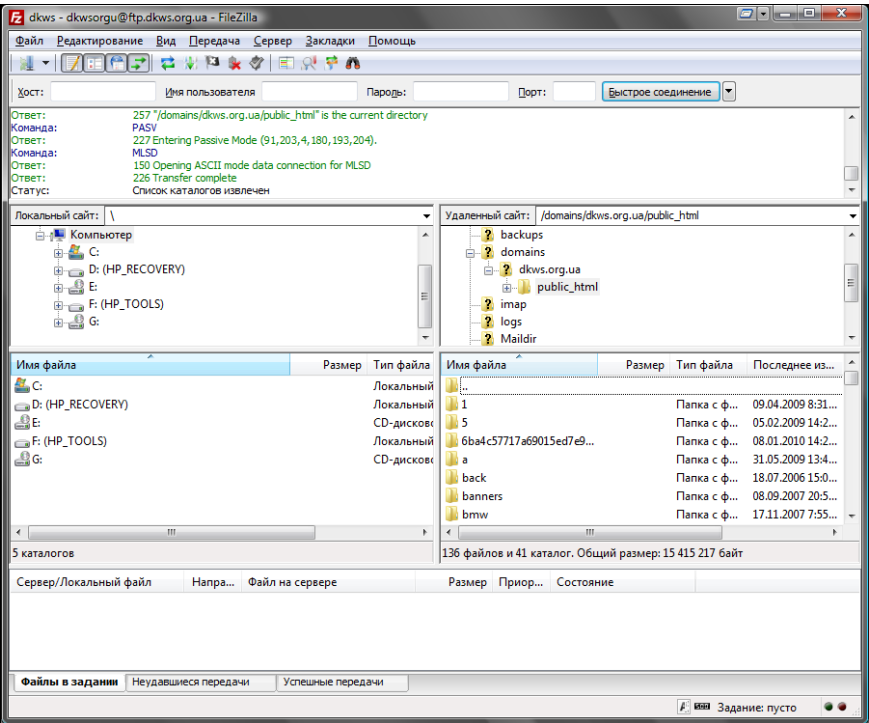


Рис. 1.4. Программа FileZilla

1.3. Установка Apache + PHP + MySQL в Windows

Если вы все-таки решили настроить собственный сервер, то приступим к установке. Первым делом вам нужно скачать с сайта www.apache.org Web-сервер Apache. Вы скачаете файл `httpd_2.2.X-win32-x86-no_ssl.msi` (на момент написания этих строк уже была доступна альфа-версия 2.3, но пока ее не стоит использовать). Запустите его. Процесс установки обычно проходит без осложнений, но мы все же рассмотрим его.

ПРИМЕЧАНИЕ

На прилагаемом диске в каталоге `software` вы найдете обычную (no-SSL) и SSL-версию Apache для Windows. Linux-версии на диске нет, поскольку Apache входит в состав любого дистрибутива Linux, вам нужно только установить его. Несмотря на наличие в каталоге `software` установочных файлов Apache, не забудьте посетить страницу <http://archive.apache.org/dist/httpd/binaries/win32/> — вполне возможно, что к моменту покупки книги вышла новая версия Apache. На диске находится версия 2.2.17, а также модуль `mod_ftp` для Windows.

УСТАНОВКА В WINDOWS VISTA И WINDOWS 7

Установку Apache, PHP и MySQL в Windows Vista и в Windows 7 желательно производить от имени администратора. Для этого щелкните на установочном файле правой кнопкой мыши и выберите команду **Запуск от имени администратора**.

Важно правильно указать информацию о сервере: доменное имя сети, доменное имя компьютера (если вы не планируете использовать сервер во всей локальной сети, лучше установить `localhost`) и адрес администратора (рис. 1.5). Также вам нужно выбрать способ запуска сервера:

- ☐ **for All Users** — сервер будет установлен как Windows-сервис и будет доступен всем пользователям;
- ☐ **only for the Current User** — сервер нужно будет запускать вручную и он будет использовать порт 8080, т. е. при обращении к серверу нужно будет указывать порт в URL, например **`http://localhost:8080/`**.

Затем выберите Custom-тип установки и измените каталог установки Apache (рис. 1.6). По умолчанию используется каталог `C:\Program Files\Apache Software Foundation\Apache2.2\`, но намного проще установить сервер в каталог `C:\WWW` или `D:\WWW`. Нажав кнопку **Space**, вы можете посмотреть, сколько свободного места осталось на ваших дисках.

После установки Apache запустите браузер и введите следующий URL:

`http://localhost`

Если вы увидите "It works!" (рис. 1.7), значит, установка Apache прошла успешно.

По умолчанию сервер Apache не поддерживает PHP. Поэтому скачайте с сайта www.php.net инсталлятор PHP (он тоже выполнен в виде MSI-пакета) и установите его. Обратите внимание на версию PHP! Если вы хотите использовать последние возможности PHP, нужно скачать шестую версию PHP.

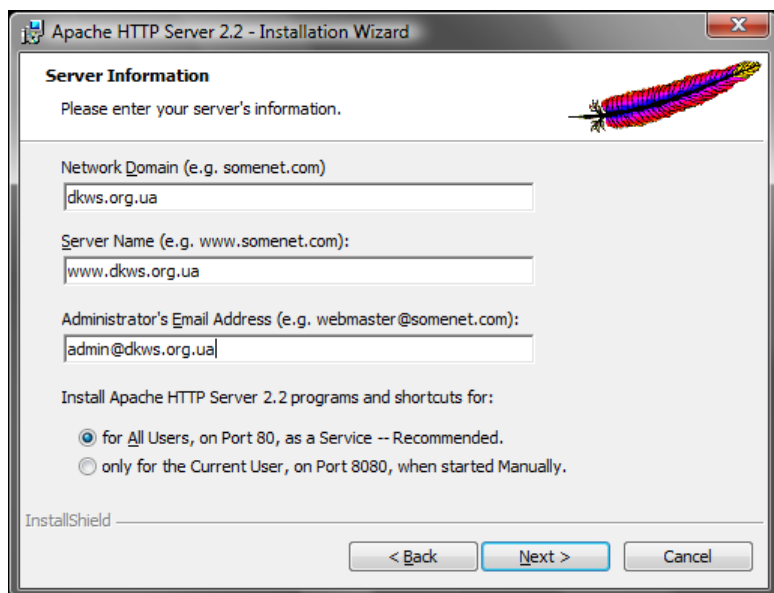


Рис. 1.5. Параметры сервера

PHP 5 и PHP 6

В каталоге software прилагаемого диска вы найдете установочный файл PHP 5.2. А в каталоге php6 вы найдете архивы как с откомпилированной Windows-версией, так и исходный код PHP 6.

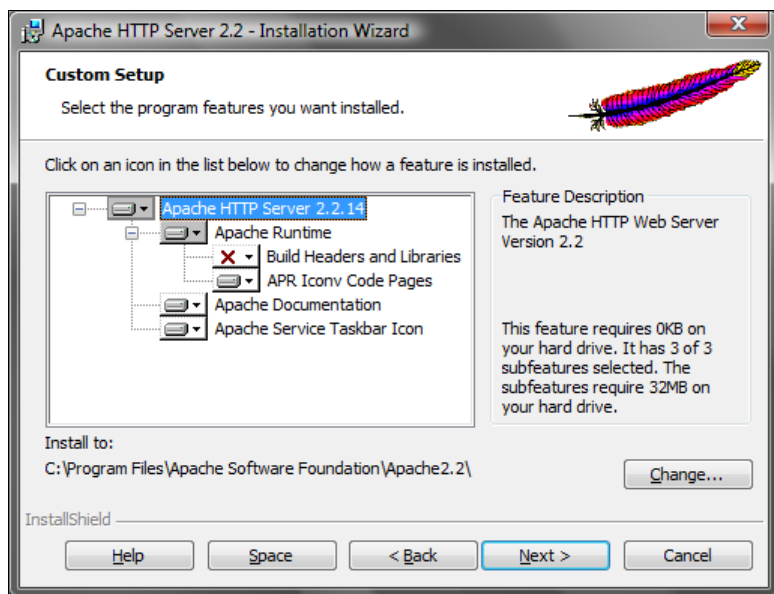


Рис. 1.6. Каталог для установки Apache

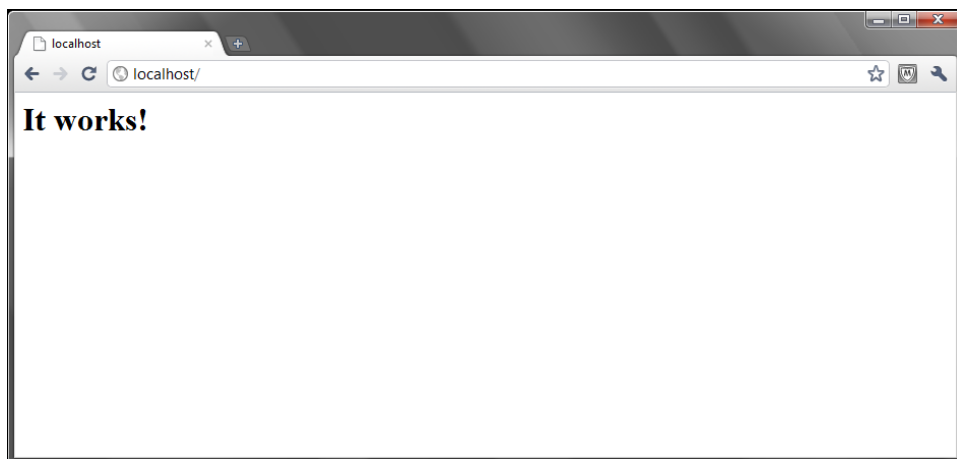


Рис. 1.7. Web-сервер работает!

При установке PHP обязательно выберите версию вашего Web-сервера. Мы установили версию 2.2, поэтому ее нужно выбрать при установке PHP (рис. 1.8). Далее нужно указать путь к конфигурационному каталогу Apache, чтобы инсталлятор PHP внес необходимые изменения в файлы конфигурации Apache. Если вы установили Apache в D:\WWW, то вам нужно указать следующий каталог: D:\WWW\conf.

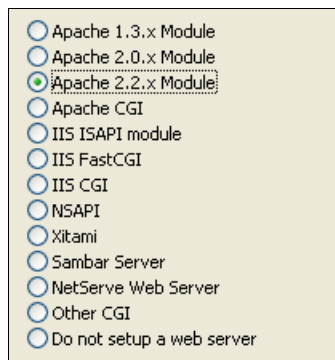


Рис. 1.8. Выбор версии Web-сервера

Теперь нужно перезагрузить Apache. Для этого выполните команду **Пуск | Выполнить** и введите команду `services.msc`. Отыщите в списке служб Apache 2.2 и выберите команду **Перезапустить службу** (рис. 1.9). В случае возникновения ошибки перезапустите компьютер.

Теперь вам осталось скачать и установить сервер баз данных MySQL (<http://mysql.com/>). Его установка проходит вообще без каких-либо нюансов, поэтому мы не будем ее подробно рассматривать. В каталоге software вы найдете версию 5.1 (на данный момент это рекомендуемая, не экспериментальная версия MySQL) для 32- и 64-битных систем. В *главе 2* мы напишем сценарий `test.php`, тестирующий созданную вами конфигурацию. Данный сценарий нужно поместить в подкаталог D:\WWW\htdocs.

ПРИМЕЧАНИЕ

На момент написания этих строк MySQL 6 все еще находится на стадии тестирования. Windows-версию MySQL 6 вы найдете на прилагаемом диске в каталоге `php6`.

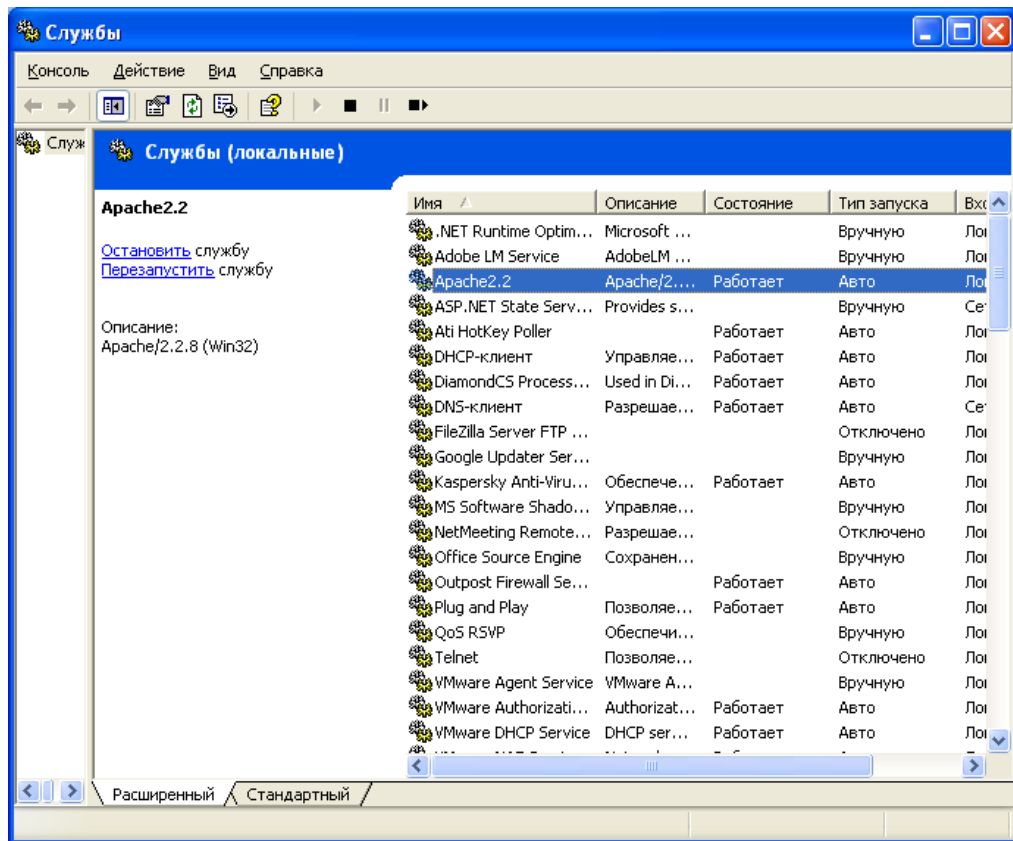


Рис. 1.9. Управление службами Windows

Если вам лень устанавливать все отдельно, вы можете установить пакет XAMPP, сочетающий Apache, PHP и MySQL "в одном флаконе". Скачать последнюю версию XAMPP можно по адресу <http://www.apachefriends.org/en/xampp.html>. На прилагаемом диске в каталоге software вы найдете последнюю (на момент написания этих строк) версию XAMPP — 1.7.3.

Основное преимущество XAMPP — простота установки и управления всеми сервисами, входящими в его состав (Web-сервер, FTP-сервер и MySQL). Но у XAMPP есть и недостаток — версии программного обеспечения, входящие в его состав, отстают от текущих. Например, в состав последней на момент написания этих строк версии XAMPP входила версия Apache 2.2.14, а на www.apache.org можно было скачать версию 2.2.17. Для начинающего программиста особой разницы нет, просто, чтобы вы знали, что в комплект "все в одном" редко входит все самое "свежее".

Установка XAMPP проблем не вызывает. Программа нормально работает в актуальных версиях Windows — XP, Vista и Windows 7. В Windows Vista и Windows 7 программу установки XAMPP нужно запускать от имени администратора.

Желательно устанавливать XAMPP в корневой каталог диска C:\, а не в каталог C:\Program Files, чтобы не было различных проблем с правами доступа — так ре-