

Денис Колисниченко

PHP 5/6 и MySQL 6

Разработка Web-приложений 2-е издание

Санкт-Петербург

«БХВ-Петербург»

2010

УДК 681.3.06
ББК 32.973.26-018.2
К60

Колисниченко Д. Н.

К60 PHP 5/6 и MySQL 6. Разработка Web-приложений. —
2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2010. — 560 с.: ил.
+ CD-ROM — (Профессиональное программирование)

ISBN 978-5-9775-0581-9

На практических примерах описана разработка Web-приложений на языке PHP версий 5 и 6. Большая часть кода примеров совместима с обеими версиями PHP, но особое внимание уделено новым функциям PHP 6. Даны начала программирования на PHP: установка и настройка PHP и MySQL, выбор редактора PHP-кода, основы синтаксиса и самые полезные функции PHP. Рассмотрено создание собственного движка сайта и ряда дополнительных модулей — фотогалереи, RSS-граббера, модуля для работы с MP3, модуля продажи недвижимости, гостевой книги, а также применение мощного шаблонизатора Smarty и создание простейшего собственного шаблонизатора. 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. Во втором издании описаны новые инструменты для создания сложных проектов PEAR, CVS и RHPUnit. 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>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 01.04.10.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 45,15.

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 978-5-9775-0581-9

© Колисниченко Д. Н., 2010
© Оформление, издательство "БХВ-Петербург", 2010

Оглавление

ВВЕДЕНИЕ.....	1
Что нового во 2-м издании.....	1
Немного истории	2
PHP 6.....	3
MySQL	4
Поддержка читателей.....	5
ЧАСТЬ I. ТЕОРИЯ	7
РАЗДЕЛ 1. БЫСТРЫЙ СТАРТ	9
ГЛАВА 1. УСТАНОВКА НЕОБХОДИМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	11
1.1. Нужно ли устанавливать программное обеспечение	11
1.2. Выбор PHP-редактора и FTP-клиента	13
1.3. Установка Apache + PHP + MySQL в Windows	16
1.4. Установка Apache + PHP + MySQL в Linux	21
1.5. Несколько советов	22
ГЛАВА 2. ПЕРВАЯ PHP-ПРОГРАММА	23
2.1. Ваша первая программа	23
2.2. Запуск PHP-программы.....	24
2.3. Вывод текста без <i>echo</i>	25
ГЛАВА 3. ОСНОВЫ СИНТАКСИСА PHP.....	28
3.1. Переменные.....	28
3.1.1. Правила объявления переменных. Имена переменных	28
3.1.2. Типы данных переменных	29
3.1.3. Булевы переменные	31
3.1.4. Операции над переменными.....	32
3.1.5. Ссылки	33
3.2. Константы.....	34

3.3. Выражения и операции	35
3.3.1. Что такое выражение	35
3.3.2. Арифметические операции	36
3.3.3. Логические выражения.....	36
3.3.4. Приоритеты операций	37
3.3.5. Операторы эквивалентности == и ===	38
3.3.6. Операции со строками.....	39
3.4. Условный оператор	40
3.5. Циклы.....	41
3.5.1. Цикл со счетчиком.....	41
3.5.2. Цикл <i>while</i>	42
3.5.3. Цикл <i>do-while</i>	42
3.5.4. Принудительное завершение цикла и пропуск итерации	43
3.6. Оператор выбора <i>switch-case</i>	43

РАЗДЕЛ 2. ПЕРЕДАЧА ПАРАМЕТРОВ PHP-ПРОГРАММАМ..... 45

ГЛАВА 4. МЕТОДЫ GET и POST.....47

4.1. Интерфейс CGI.....	47
4.2. Метод GET	49
4.3. Метод POST	50

ГЛАВА 5. ПРОТОКОЛ HTTP И ИНТЕРФЕЙС CGI.....51

5.1. Специальные переменные окружения CGI	51
5.2. Заголовки протокола HTTP	52
5.3. Коды ответов протокола HTTP	54

ГЛАВА 6. ПЕРЕДАЧА ПАРАМЕТРОВ ПОСРЕДСТВОМ HTML-ФОРМЫ55

6.1. Создание простейшей формы и ее обработка в сценарии	55
6.2. Создание пользовательского интерфейса с помощью формы	58
6.2.1. Ввод текста. Теги <i>INPUT</i> и <i>TEXTAREA</i>	60
6.2.2. Зависимые и независимые переключатели	61
6.2.3. Списки выбора	63
6.2.4. Форма для передачи файлов	65
6.2.5. Кнопки	65
6.3. Проверка параметров формы.....	66
6.3.1. Проверка корректности e-mail	67
6.3.2. Проверка правильности номера кредитной карты	68
6.3.3. Удаление лишних пробелов.....	70

ГЛАВА 7. ЗАПОМИНАНИЕ ПАРАМЕТРОВ С ПОМОЩЬЮ СООКIES И СЕССИЙ	71
7.1. Что такое Cookies и как с ними работать	71
7.2. Механизм сессий	74
7.2.1. Сессии и Cookies: преимущества и недостатки	74
7.2.2. Для чего нужны сессии.....	74
7.2.3. Как работает механизм сессий	76
7.2.4. Обход Cookies	78
7.3. Массивы и Cookies.....	78
ГЛАВА 8. ОТДЕЛЬНОЕ СЛОВО О ДИРЕКТИВЕ REGISTER_GLOBALS	79
8.1. Почему опасно использовать <i>register_globals</i>	79
8.2. Если <i>register_globals</i> отключена.....	84
8.3. PHP 6 и <i>register_globals</i>	86
РАЗДЕЛ 3. МАССИВЫ И СПИСКИ	87
ГЛАВА 9. ОСНОВНЫЕ ОПЕРАЦИИ НАД МАССИВАМИ И СПИСКАМИ.....	89
9.1. Массив и список. Цикл <i>foreach</i>	89
9.2. Функции <i>list()</i> и <i>array()</i>	91
9.3. Удаление массива	92
9.4. Слияние массивов.....	93
9.5. Функция <i>print_r()</i>	95
ГЛАВА 10. ФУНКЦИИ СОРТИРОВКИ МАССИВОВ.....	97
10.1. Функции для сортировки массивов	97
10.2. Функция <i>sort()</i> — сортировка списка	97
10.3. Функция <i>asort()</i> — сортировка массива по значениям	98
10.4. Функция <i>ksort()</i> — сортировка по ключам.....	99
10.5. Функции <i>array_reverse()</i> и <i>shuffle()</i>	100
10.6. Собственная функция сортировки	101
ГЛАВА 11. ОСОБЫЕ ОПЕРАЦИИ НАД МАССИВАМИ	103
11.1. Добавление и удаление элементов массива	103
11.2. Упаковка переменных в массив и их извлечение.....	105
11.3. Получение части массива.....	106

РАЗДЕЛ 4. ФУНКЦИИ В PHP	107
ГЛАВА 12. ПОЛЕЗНЫЕ СТАНДАРТНЫЕ ФУНКЦИИ	109
12.1. Генератор случайных чисел.....	109
12.2. Дата и время	110
12.2.1. Кратко о timestamp.....	110
12.2.2. Функции <i>strtotime()</i> и <i>checkdate()</i>	111
12.2.3. Вывод даты.....	112
12.3. Математические функции.....	114
ГЛАВА 13. ФУНКЦИИ ДЛЯ РАБОТЫ СО СТРОКАМИ	115
13.1. Основные строковые функции	115
13.2. Специальные функции замены.....	117
13.3. Преобразование строки	117
13.4. Функции преобразования кодировок.....	119
13.5. Функции для работы с отдельными символами строки.....	120
13.6. Функция <i>md5()</i>	120
13.7. Функция <i>explode()</i> : выделение подстрок	121
ГЛАВА 14. РАБОТАЕМ С ФАЙЛАМИ И КАТАЛОГАМИ	122
14.1. Права доступа в UNIX.....	122
14.2. Чтение файла.....	125
14.2.1. Использование функций <i>fopen()</i> и <i>fread()</i>	125
14.2.2. Использование функции <i>file()</i> : построчное чтение файла	128
14.2.3. Чтение всего файла: функция <i>file_get_contents()</i>	130
14.3. Запись файла	130
14.4. Создание временных файлов.....	131
14.5. Работа с CSV-файлами	131
14.6. Специальные функции для работы с файлами	134
14.6.1. Функции для работы с именами файлов.....	134
14.6.2. Работа с правами доступа	135
14.6.3. Копирование, переименование и удаление файлов.....	136
14.6.4. Время доступа к файлу.....	137
14.6.5. Другие полезные функции	138
14.7. Совместный доступ к файлу	139
14.8. Функции для работы с каталогами.....	140

ГЛАВА 15. ВЫВОД ГРАФИЧЕСКИХ ИЗОБРАЖЕНИЙ СРЕДСТВАМИ PHP142

15.1. Библиотека GD.....	142
15.1.1. Получение информации об изображении.....	143
15.1.2. Конвертирование графических форматов.....	146
15.1.3. Вывод текста поверх картинки.....	149
15.1.4. Прозрачность.....	152
15.2. Изменение размера изображения.....	152
15.3. Создание водяных знаков.....	154

ГЛАВА 16. РАБОТА С СЕТЕВЫМИ СОКЕТАМИ В PHP.**СЕТЕВЫЕ ФУНКЦИИ157**

16.1. Еще раз о том, что такое сокет.....	157
16.2. Использование сокетов.....	158
16.3. Пример использования сокетов.....	159
16.4. Блокирующий и неблокирующий режимы сокета.....	163
16.5. DNS-функции.....	163

ГЛАВА 17. СОБСТВЕННЫЕ ФУНКЦИИ.....165

17.1. Зачем нужны собственные функции.....	165
17.2. Особенности функций в PHP.....	166
17.3. Объявление функции.....	166
17.4. Области видимости функции.....	167
17.5. Вложенность функций.....	168
17.6. Переменное число аргументов.....	170
17.7. Передача массивов в качестве параметров.....	171

РАЗДЕЛ 5. СЕРВЕРЫ БАЗ ДАННЫХ MYSQL 6 И SQLITE173**ГЛАВА 18. ОСНОВЫ SQL175**

18.1. Немного истории.....	175
18.2. Преимущества SQL.....	176
18.3. Как выглядят запросы.....	177
18.4. Что такое база данных.....	177
18.5. Создание таблиц.....	179
18.6. Добавление записей в таблицу.....	183
18.7. Обновление записей.....	184
18.8. Выборка записей.....	185
18.9. Удаление записей.....	186

18.10. Встроенные функции.....	187
18.11. Группировка записей. Сложные запросы.....	188
ГЛАВА 19. ФУНКЦИИ ДЛЯ РАБОТЫ С MYSQL.....	192
19.1. Подключение к серверу MySQL.....	192
19.2. Несколько MySQL-соединений.....	194
19.3. Передача запросов серверу.....	195
19.4. Работа с базой данных. Создание базы данных.....	200
19.5. Функция <i>mysql_real_escape_string(\$content)</i>	200
ГЛАВА 20. АЛЬТЕРНАТИВНАЯ БАЗА ДАННЫХ SQLITE.....	202
20.1. MySQL vs SQLite: что лучше.....	202
20.2. Открытие базы данных.....	204
20.3. Передача запросов.....	205
20.4. Работа с результатом запроса.....	205
20.5. Список PHP-функций для работы с SQLite.....	207
ГЛАВА 21. ПОЛЕЗНЫЕ ПРИЕМЫ ПРИ РАБОТЕ С MYSQL 6.....	210
21.1. Выбор кодировки.....	210
21.2. Сортировка: вывод новинок. Вывод случайных записей.....	211
21.3. Постраничный вывод таблицы.....	212
РАЗДЕЛ 6. ИНСТРУМЕНТЫ ДЛЯ СОЗДАНИЯ СЛОЖНЫХ ПРОЕКТОВ.....	219
ГЛАВА 22. РАЗРАБОТКА СОБСТВЕННОГО ШАБЛОНИЗАТОРА.....	221
22.1. Организация файлов и каталогов проекта.....	221
22.2. Выносим параметры в отдельный файл.....	224
22.3. Подключение дополнительных файлов.....	225
22.3.1. Инструкции <i>include</i> и <i>require</i>	225
22.3.2. Альтернативный способ подключения сценариев.....	226
22.3.3. Инструкции <i>include_once</i> и <i>require_once</i>	228
22.4. Шаблоны.....	228
ГЛАВА 23. ШАБЛОНИЗАТОР SMARTY.....	233
23.1. Что такое Smarty.....	233
23.2. Установка Smarty.....	234

23.3. Создание <code>setup.php</code>	237
23.4. Разработка шаблонов Smarty	238
23.4.1. Комментарии в шаблонах	238
23.4.2. Переменные в Smarty	239
23.4.3. Файлы конфигурации шаблонов	240
23.4.4. Служебная переменная <code>{<i>\$smarty</i>}</code>	241
23.4.5. Модификаторы переменных	243
23.4.6. Стандартные (встроенные) функции Smarty	246
Функции <code>{include}</code> и <code>{insert}</code>	247
Функция <code>{foreach}</code>	247
Функции <code>{if}</code> , <code>{elseif}</code> , <code>{else}</code>	249
Функция <code>{capture}</code>	251
Функция <code>{php}</code>	251
Функция <code>{strip}</code>	251
23.4.7. Пользовательские функции Smarty	252
Функция <code>{assign}</code>	252
Функция <code>{cycle}</code>	252
Функция <code>{fetch}</code>	253
Функции <code>{html_checkboxes}</code> и <code>{html_radios}</code>	253
Функция <code>{html_image}</code>	254
Функция <code>{html_select_date}</code>	254
Функция <code>{html_select_time}</code>	257
Функция <code>{html_table}</code>	257
23.5. Smarty для программиста	258
23.5.1. Специальные переменные	258
23.5.2. Полезные методы класса <i>Smarty</i>	261

ГЛАВА 24. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ262

24.1. Основы ООП	262
24.2. Классы и объекты	263
24.3. Конструкторы и деструкторы класса	266
24.4. Наследование классов. Полиморфизм	267
24.5. Новые возможности PHP 5/6	268
24.5.1. Область видимости членов класса	268
24.5.2. Абстрактные классы и методы	269
24.5.3. Служебное слово <i>final</i>	270
24.5.4. Клонирование объектов	271
24.5.5. Обработка исключительных ситуаций	272
24.5.6. Константы-члены класса	273
24.5.7. Статические члены класса	274

24.5.8. Оператор <i>instanceof</i>	274
24.5.9. Итераторы.....	275
24.5.10. Пространства имен.....	275
ГЛАВА 25. МЕХАНИЗМ СЕССИЙ.....	276
25.1. Для чего нужны сессии.....	276
25.2. Как работает механизм сессий.....	278
25.3. Обход Cookies.....	279
25.4. Сценарий аутентификации.....	280
ГЛАВА 26. ВВЕДЕНИЕ В PEAR.....	285
26.1. Серьезные проекты и PEAR.....	285
26.2. Пример использования класса <i>DB</i>	287
ГЛАВА 27. КОНТРОЛЬ ВЕРСИЙ.....	291
27.1. Выбор системы контроля версий.....	291
27.2. Практическое использование TortoiseHG (Mercurial).....	293
27.3. Просмотр внесенных изменений.....	295
ГЛАВА 28. ТЕСТИРОВАНИЕ РНР-СЦЕНАРИЕВ.....	298
28.1. Программа работает, но не так, как нам нужно.....	298
28.2. "Самодельные" точки останова.....	300
28.3. Система автоматического тестирования.....	300
28.4. Директива <i>error_reporting</i>	305
ЧАСТЬ II. ПРАКТИКА.....	307
РАЗДЕЛ 7. РАЗРАБОТКА ОСНОВНЫХ ЭЛЕМЕНТОВ САЙТА.....	309
ГЛАВА 29. ЗАГРУЗКА ФАЙЛОВ НА СЕРВЕР.....	311
29.1. Что нужно знать о загрузке файлов на сервер.....	311
29.2. Реализация загрузки файла.....	314
29.3. Загрузка нескольких файлов.....	317
29.4. Проблемы при загрузке файлов.....	319

ГЛАВА 30. ИСПОЛЬЗОВАНИЕ FTP-ФУНКЦИЙ.....	321
30.1. Функции для работы с FTP	321
30.2. Примеры использования FTP-функций	325
ГЛАВА 31. ОТПРАВКА И ПРИЕМ ПОЧТЫ	328
31.1. Отправка почты средствами PHP — функция <i>mail()</i>	328
31.2. Отправка писем с вложениями — класс <i>HtmlMimeMail</i>	330
31.2.1. Отправка сообщения.....	330
31.2.2. Проблемы при отправке сообщения	334
31.2.3. MIME-типы.....	336
31.3. Получение писем по протоколу POP3	337
ГЛАВА 32. РАБОТА С RSS: ПОЛУЧАЕМ НОВОСТИ НА САЙТ	342
32.1. Краткие сведения о RSS	342
32.2. Формат RSS-файла	343
32.3. Написание сценария импорта новостей	345
32.4. Подключение файла <i>import.php</i> к сайту.....	349
32.5. Создание граббера новостей.....	349
ГЛАВА 33. ПОИСК С ИСПОЛЬЗОВАНИЕМ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ.....	354
33.1. Нужно что-то найти... ..	354
33.2. Язык регулярных выражений <i>RegExp</i>	355
33.3. Управляющие конструкции	357
33.3.1. Квантификаторы	357
33.3.2. Альтернативный оператор <i> </i>	358
33.3.3. Скобки.....	358
33.4. Псевдосимволы.....	359
33.5. Практическое использование <i>RegExp</i> -функций	359
ГЛАВА 34. РАБОТАЕМ С MP3	364
34.1. Формат MP3	364
34.2. Библиотека <i>PEAR</i>	365
34.3. Вывод ID3-тегов.....	366
34.4. Редактирование ID3-тегов	368
34.5. Удаление тега	369
ГЛАВА 35. ТЕХНОЛОГИЯ AJAX.....	370
35.1. Что такое Ajax	370
35.2. Ваше первое Ajax-приложение.....	371

РАЗДЕЛ 8. ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ САЙТА377**ГЛАВА 36. СОЗДАНИЕ ПРОСТЕЙШЕГО ДВИЖКА САЙТА379**

36.1. Планирование движка	379
36.1.1. Зачем нужно разрабатывать собственный движок	379
36.1.2. Необходимые нам функции движка.....	381
36.1.3. "Принципиальная схема" движка.....	383
36.2. Основные функции движка	385
36.2.1. Разработка TPL-шаблона	385
36.2.2. Файл настроек	386
36.2.3. Основной файл CMS — index.php.....	386
36.2.4. Проектирование базы данных	389
Таблица static	390
Таблица cats	390
Таблица pages	392
36.2.5. Иерархическая структура сайта.....	393
Алгоритм работы меню	393
Сценарий menu.php	395
Вывод содержимого раздела и страницы	402
36.3. Дополнительные функции движка.....	406
36.3.1. Вывод информации из таблицы static	406
36.3.2. Функция вывода содержимого HTML-файла	408
36.3.3. Версия для печати	409
36.4. Где взять листинги этой главы	410

ГЛАВА 37. СОЗДАНИЕ ФОТОГАЛЕРЕИ412

37.1. Постановка задачи	412
37.2. Загрузка изображений на сервер	412
37.3. Вывод галереи	417

ГЛАВА 38. ГОСТЕВАЯ КНИГА.....420

38.1. Пережиток прошлого?.....	420
38.2. Разработка базы данных и структура гостевой книги.....	421
38.3. Вывод гостевой книги	421
38.4. Добавление записей в гостевую книгу	425
38.5. Сервисный сценарий gb_service.php	429

ГЛАВА 39. ИНТЕГРАЦИЯ ГАЛЕРЕИ LIVEJOURNAL И ВАШЕГО САЙТА.....431

39.1. Что такое Живой журнал	431
39.2. Интеграция фотогалереи LiveJournal и сайта.....	433
39.3. Настройка внешнего вида галереи.....	437

ГЛАВА 40. СОЗДАНИЕ СЧЕТЧИКА САЙТА	439
40.1. Постановка задачи	439
40.2. Файл конфигурации	440
40.3. Разработка таблиц counter и ipaddr	440
40.4. Сценарий counter.php	442
40.5. Сценарий reset_counter.php	447
ГЛАВА 41. СТАТИСТИКА САЙТА.....	448
41.1. Методы сбора статистики	448
41.2. Программы-анализаторы журналов Web-сервера	450
41.3. Системы статистики	451
ГЛАВА 42. ГОЛОСОВАНИЯ (ОПРОСЫ).....	455
42.1. Разработка собственной системы голосования	455
42.2. Разработка сценария poll_form.php	457
42.3. Сценарий poll_process.php	459
42.4. Сценарий poll_results.php	461
РАЗДЕЛ 9. СЛОЖНЫЙ ПРОЕКТ: САЙТ ПО ПРОДАЖЕ НЕДВИЖИМОСТИ	465
ГЛАВА 43. ПОСТАНОВКА ЗАДАЧИ	467
43.1. Функции будущего проекта	467
43.2. Разработка базы данных.....	468
ГЛАВА 44. РАЗРАБОТКА ОСНОВНОЙ ЧАСТИ САЙТА	479
44.1. С чего начать	479
44.2. Реализация основных функций системы	481
44.2.1. Аутентификация пользователей	481
44.2.2. Вывод VIP-объявлений	485
44.2.3. Вывод объявлений об услугах	488
44.2.4. Вывод рекламных баннеров.....	489
44.2.5. Постраничный вывод объявлений	489
44.3. На что обратить внимание	493
ГЛАВА 45. РАЗРАБОТКА ПАНЕЛИ АДМИНИСТРИРОВАНИЯ	501
45.1. Функции панели управления	501
45.2. Управление VIP-объявлениями.....	502

45.3. Массовая отправка электронного сообщения.....	503
45.4. Общие операции с базой данных	504
45.5. Редактирование статей, новостей и контактов	505
РАЗДЕЛ 10. ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ САЙТА.....	507
ГЛАВА 46. SSL-СОЕДИНЕНИЯ.....	509
46.1. Защищаем передаваемые данные.....	509
46.2. Настройка SSL в DirectAdmin.....	510
46.3. SSL-переменные.....	512
ГЛАВА 47. ЗАЩИТА PHP С ПОМОЩЬЮ КОНФИГУРАЦИОННОГО ФАЙЛА	515
47.1. Конфигурационный файл php.ini	515
47.2. Отключение потенциально опасных функций.....	517
47.3. Рекомендованные значения некоторых конфигурационных директив	517
ГЛАВА 48. ЗАЩИТА САЙТА ОТ АТАК.....	519
48.1. Сайт в опасности.....	519
48.2. Два самых распространенных метода взлома.....	520
48.3. Межсайтовый скриптинг	521
48.4. SQL-инъекции	523
48.5. Флуд	527
48.6. Защита форума PHPBB2 от спаммеров	528
ПРИЛОЖЕНИЕ. ОПИСАНИЕ CD.....	531
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	533

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 во 2-м издании

Во 2-м издании довольно много изменений, даже не знаю, с чего и начать. Начну с прилагаемого диска. В 1-м издании диска не было, а листинги из книги нужно было скачивать с моего сайта (www.dkws.org.ua). Во 2-м издании диск есть. Кроме листингов на диске вы найдете все необходимое для программирования на PHP программное обеспечение — от Web-сервера и самого PHP до текстового редактора. Подробное описание диска приводится в *приложении*, поэтому не будем сейчас тратить время на его описание.

Раздел 9, в котором рассматривалась интеграция сайта с форумом phpBB (версий 2 и 3), полностью переработан. Теперь в нем рассматривается разработка системы продажи недвижимости — полностью реальный проект, разработанный мною в 2009 г. Полный исходный код системы доступен на прилагаемом диске. Такая перестановка "слагаемых" не означает, что читатели 2-го издания не узнают, как "прикрутить" сайт к форуму или форум к сайту. В каталоге Edition_1 прилагаемого диска вы найдете все главы 1-го издания, которые были удалены из 2-го издания. Также на диск "переехало" описание программы phpMyAdmin (ранее это было единственное приложение).

В 2-м издании книги появился абсолютно новый материал. Кроме мелких изменений в различных главах (вроде создания механизма генерирования уникальных имен при загрузке файлов) было добавлено три новых главы (26–28). В новых главах описывается репозиторий PEAR, система контроля версий РНР-файлов, а также тестирование РНР-сценариев.

И еще хочется сделать одно очень важное замечание. Так сказать, о наболвшем. В 2009 г. на сайте www.bhv.ru одним из читателей был размещен не очень лестный отзыв, в котором говорилось, что в книге очень много опечаток. Отзыв сразу мне показался странным, поскольку все листинги тщательно тестировались как мной лично, так и редактором. Ради "спортивного интереса" при подготовке 2-го издания я проверил еще раз все листинги — ни в одном из них не было найдено ошибок. Если у вас не запускается какой-нибудь сценарий, то это связано либо с неправильными настройками РНР/Apache, либо с тем, что *вы* допустили ошибку в самом сценарии. В любом случае, добро пожаловать на мой форум www.dkws.org.ua, где можно обсудить любые вопросы, связанные с этой книгой, а также с программированием на РНР в целом. Если интересно, то отзывы о моих книгах можно прочитать (и, конечно же, разместить собственный), можно по адресам:

- <http://www.dkws.org.ua/phpbb2/viewtopic.php?t=3333>;
- <http://www.dkws.org.ua/phpbb2/viewtopic.php?t=4425>.

Немного истории

В 1994 г. датский программист Расмус Лердорф 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 РНР).

Но Perl — довольно медленный интерпретатор, и вскоре его производительности перестало хватать, поэтому разработчик написал на языке С новый интерпретатор и назвал его РНР/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 РНР, такие как автоматическая обработка форм, встраиваемость в HTML и др.

В 1997 г. появилась вторая версия — РНР/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 бóльшим успехом, чем вторая: она была установлена на 10% всех интернет-серверов, а это уже не мало!

PHP 3.0 частенько критиковали за медленное ядро, так как сценарии выполнялись медленно. Поэтому разработку версии PHP 4 начали с переработки ядра. Начиная 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 спорю, 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 (то есть 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



РАЗДЕЛ 1

Быстрый старт



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. Буквально вчера (на момент написания этих строк, понятно) 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.

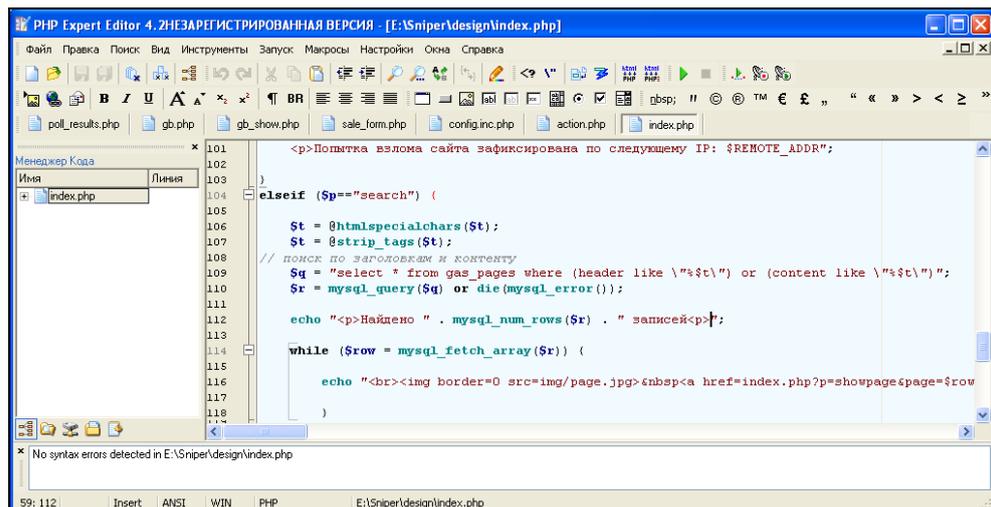


Рис. 1.1. Программа PHP Expert Editor

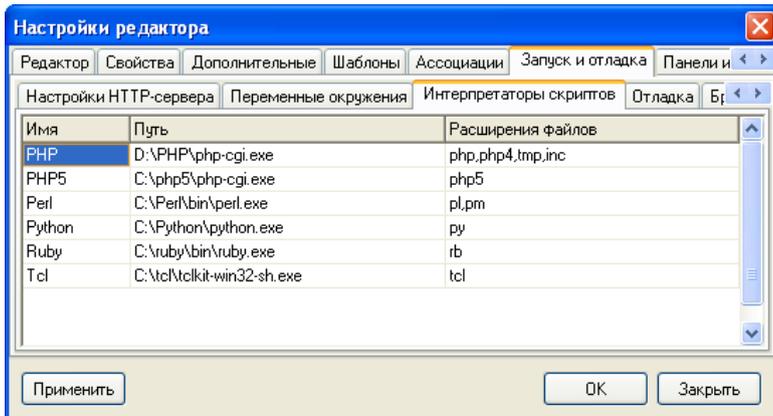


Рис. 1.2. Путь к интерпретатору PHP

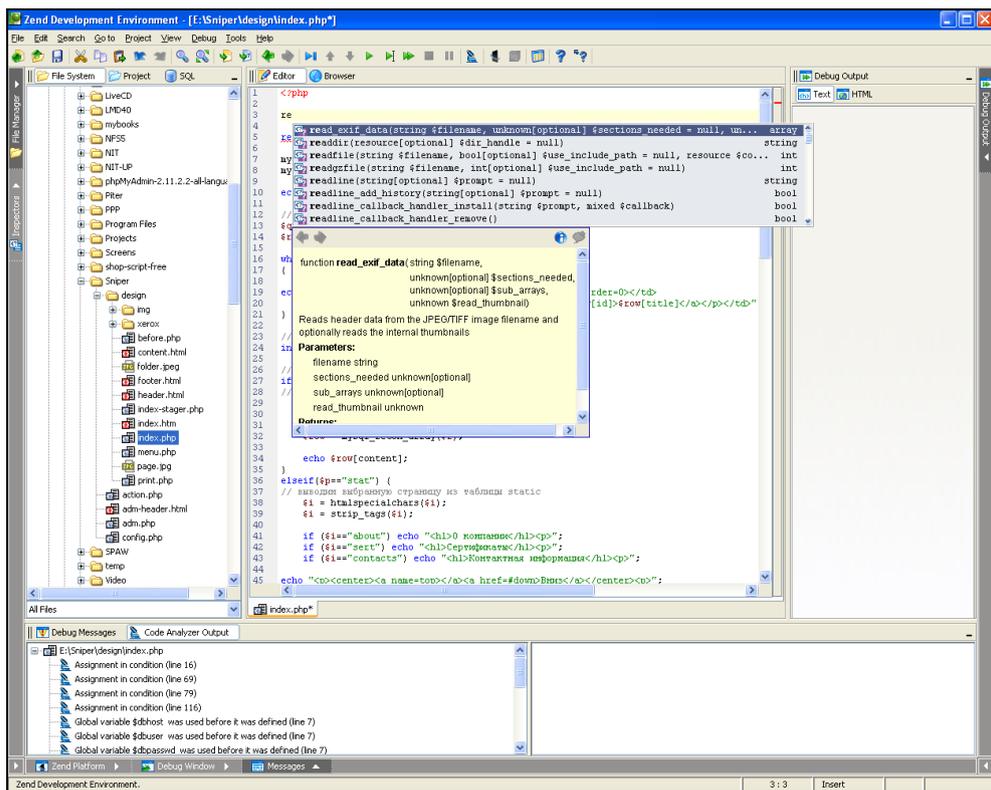


Рис. 1.3. Zend выводит не только имена функций, но и краткую справку по каждой функции

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>.

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, вы 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 3.3.1 этого FTP-клиента находится на прилагаемом диске в каталоге software.

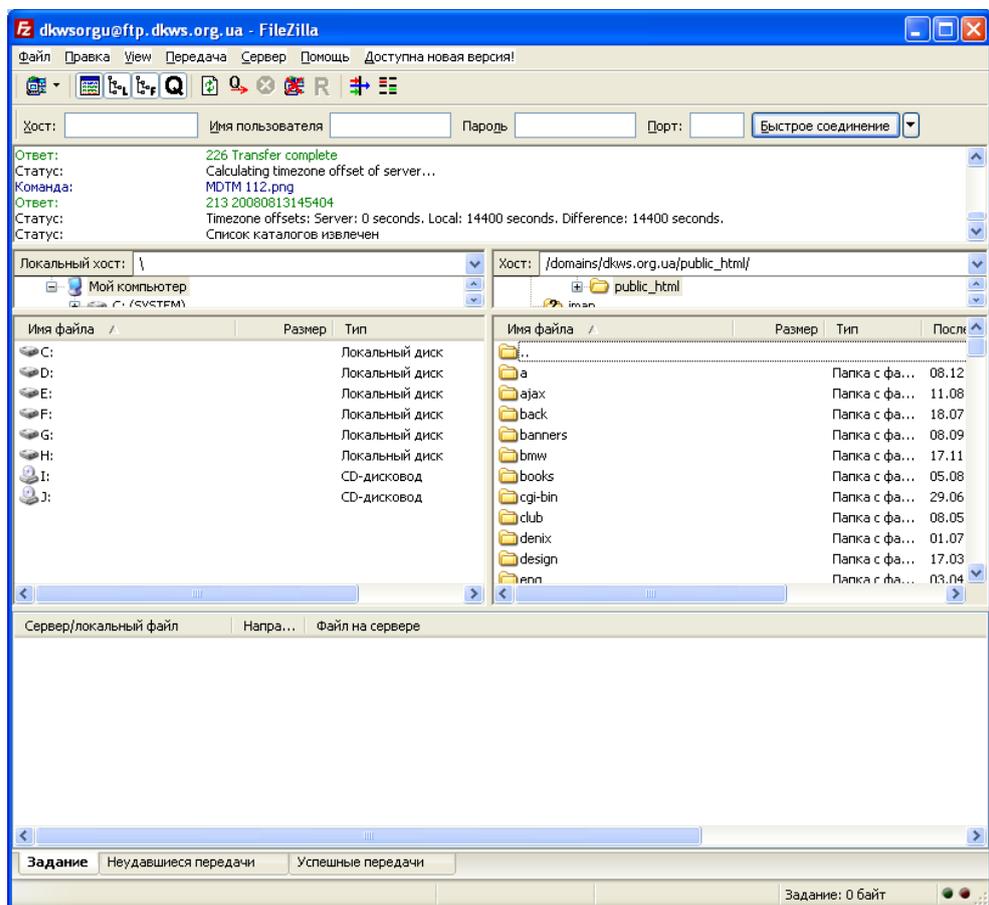


Рис. 1.4. Программа FileZilla

1.3. Установка Apache + PHP + MySQL в Windows

Если вы-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 `apache_2.2.X-win32-x86-no_ssl.msi`. 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 2.2.14, 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, то есть 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.

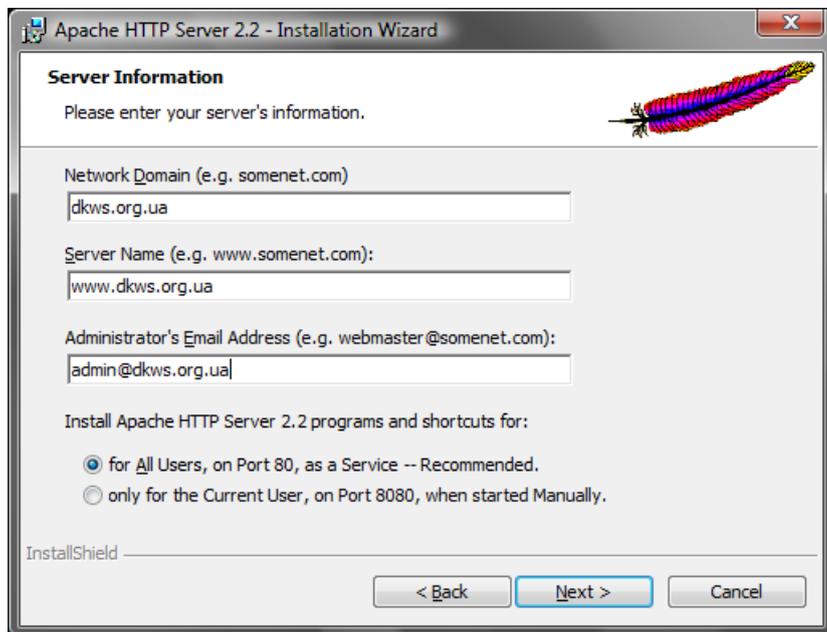


Рис. 1.5. Параметры сервера

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.

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.

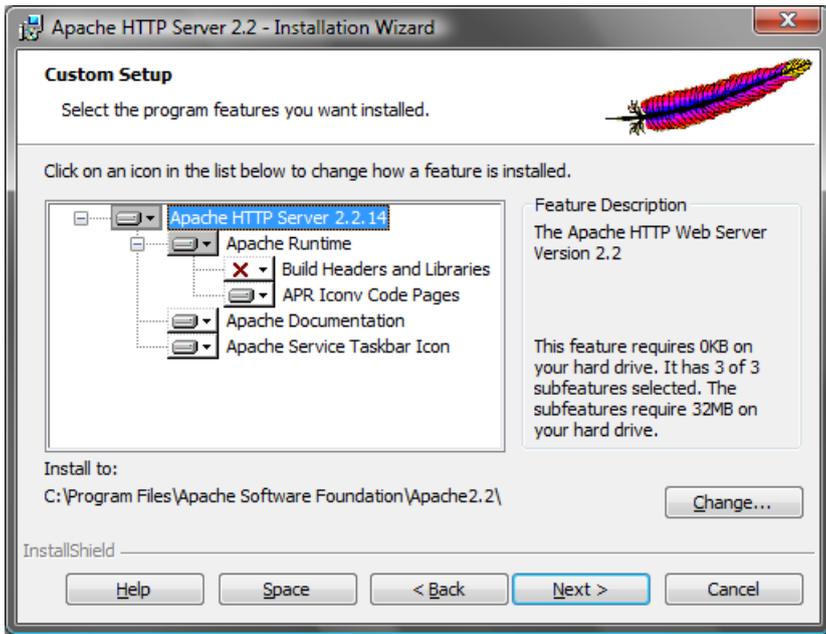


Рис. 1.6. Каталог для установки Apache

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`.

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`.

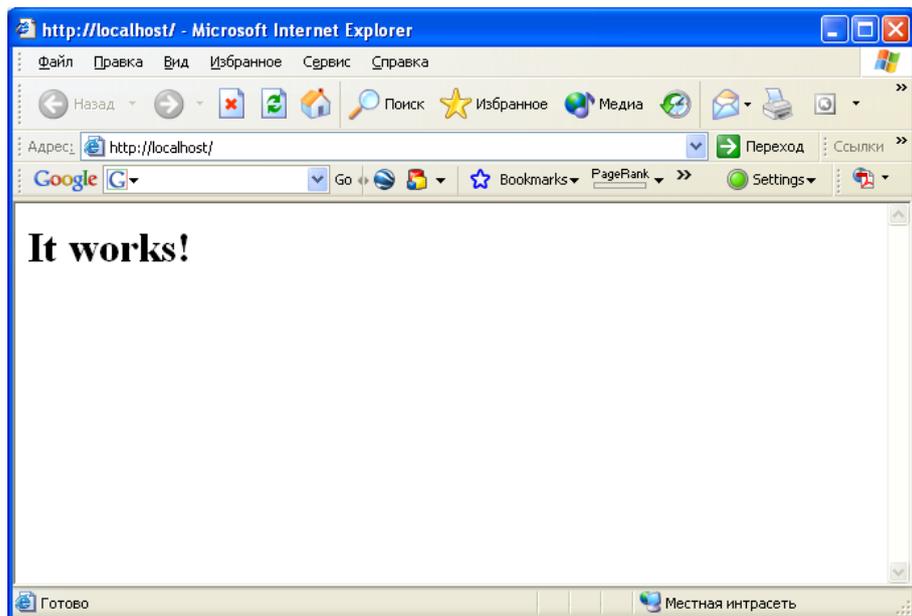


Рис. 1.7. Web-сервер работает!

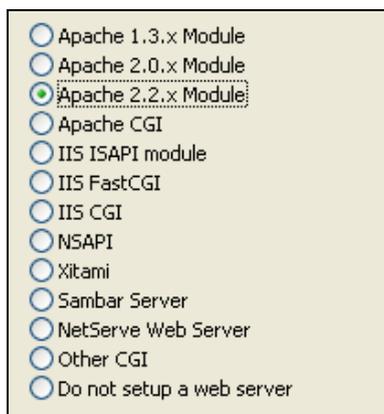


Рис. 1.8. Выбор версии Web-сервера

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.