

Денис Колисниченко

# **PHP и MySQL**

## **Разработка веб-приложений**

**5-е издание**

Санкт-Петербург  
«БХВ-Петербург»  
2015

УДК 004.43+004.738.5  
ББК 32.973.26-018.1  
К60

**Колисниченко Д. Н.**

К60 PHP и MySQL. Разработка веб-приложений. — 5-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2015. — 592 с.: ил. — (Профессиональное программирование)

ISBN 978-5-9775-3514-4

Даны начала программирования на PHP: установка и настройка Apache, PHP и MySQL, а также кроссплатформенной сборки XAMPP, выбор редактора PHP-кода, синтаксис языка и самые полезные функции PHP. Рассмотрено создание собственного движка сайта и ряда дополнительных модулей — фотогалереи, RSS-граббера, модуля для работы с MP3, сервиса reCAPTCHA, а также применение мощного шаблонизатора Smarty и создание простейшего собственного шаблонизатора. Показано, как с помощью технологии Ajax добиться обновления данных на странице без ее перезагрузки. В книге представлено два реальных проекта: универсальная система управления контентом, которая подойдет для построения типичного сайта, и одного из реальных проектов автора. В пятом издании рассмотрены нововведения версии 5.5 (генераторы, новые ключевые слова, Password Hashing API и др.), а также особенности создания индикатора загрузки файла и разыменовывания массивов.

На сайте издательства находятся дополнительные главы, все листинги из книги, а также необходимое программное обеспечение.

*Для веб-программистов*

УДК 004.43+004.738.5  
ББК 32.973.26-018.1

**Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Екатерина Капалыгина</i>
Редактор	<i>Григорий Добин</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Марины Дамбиевой</i>

Подписано в печать 29.08.14.

Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 47,73.

Тираж 1500 экз. Заказ №

"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Первая Академическая типография "Наука"  
199034, Санкт-Петербург, 9 линия, 12/28

ISBN 978-5-9775-3514-4

© Колисниченко Д. Н., 2015  
© Оформление, издательство "БХВ-Петербург", 2015

# Оглавление

<b>Предисловие .....</b>	<b>15</b>
Что нового в 5-м издании? .....	15
Немного истории .....	16
RНР 5.5, или что случилось с RНР 6?.....	17
Основные нововведения в RНР версии 5.4 .....	19
Типажи.....	19
Разыменовывание массива.....	19
Встроенный веб-сервер .....	19
Индикатор загрузки файла.....	19
Основные нововведения в RНР версии 5.5 .....	20
Генераторы.....	20
Ключевые слова <i>finally</i> и <i>class</i> .....	20
API хэширования паролей .....	20
Разыменовывание строк.....	20
Изменения в работе конструкции <i>foreach</i> .....	20
Поддержка графического формата WebP.....	20
Исключенные возможности.....	20
MySQL 21 .....	
Поддержка читателей .....	21
 <b><u>ЧАСТЬ I. ТЕОРИЯ</u> .....</b>	<b>23</b>
 <b>РАЗДЕЛ 1. БЫСТРЫЙ СТАРТ .....</b>	<b>25</b>
 <b>Глава 1. Установка необходимого программного обеспечения.....</b>	<b>27</b>
1.1. Нужно ли устанавливать программное обеспечение? .....	27
1.2. Выбор RНР-редактора и FTP-клиента .....	29
1.3. Установка связки Apache + RНР + MySQL в Windows.....	31
1.3.1. Для опытных пользователей: установка вручную .....	31
1.3.2. Установка для новичков (рекомендуется) .....	38
1.4. Установка Apache + RНР + MySQL в Linux .....	42
1.5. Конфликты с другим программным обеспечением .....	43
1.6. Функция <i>phpinfo()</i> : получение информации о RНР.....	44
1.7. Несколько советов .....	46

<b>Глава 2. Программа на PHP.....</b>	<b>47</b>
2.1. Ваша первая программа .....	47
2.2. Запуск PHP-программы.....	48
2.3. Вывод текста без <i>echo</i> .....	49
<b>Глава 3. Основы синтаксиса PHP .....</b>	<b>51</b>
3.1. Переменные.....	51
3.1.1. Правила объявления переменных. Имена переменных.....	51
3.1.2. Типы данных переменных .....	52
3.1.3. Булевы переменные .....	53
3.1.4. Операции над переменными.....	54
3.1.5. Ссылки.....	55
3.2. Константы .....	56
3.3. Выражения и операции .....	57
3.3.1. Что такое выражение.....	57
3.3.2. Арифметические операции .....	57
3.3.3. Логические выражения .....	58
3.3.4. Приоритеты операций.....	58
3.3.5. Операторы эквивалентности <code>==</code> и <code>===</code> .....	59
3.3.6. Операции со строками.....	60
3.4. Условный оператор .....	61
3.5. Циклы .....	62
3.5.1. Цикл со счетчиком <i>for</i> .....	62
3.5.2. Цикл <i>while</i> .....	62
3.5.3. Цикл <i>do-while</i> .....	63
3.5.4. Принудительное завершение цикла и пропуск итерации.....	63
3.6. Оператор выбора <i>switch-case</i> .....	64
<b>РАЗДЕЛ 2. ПЕРЕДАЧА ПАРАМЕТРОВ PHP-ПРОГРАММАМ.....</b>	<b>67</b>
<b>Глава 4. Методы <i>GET</i> и <i>POST</i>.....</b>	<b>69</b>
4.1. Интерфейс CGI .....	69
4.2. Метод <i>GET</i> .....	71
4.3. Метод <i>POST</i> .....	71
<b>Глава 5. Протокол HTTP и интерфейс CGI .....</b>	<b>73</b>
5.1. Специальные переменные окружения CGI.....	73
5.2. Заголовки протокола HTTP .....	74
5.3. Коды ответов протокола HTTP .....	75
<b>Глава 6. Передача параметров посредством HTML-формы .....</b>	<b>77</b>
6.1. Создание простейшей формы и ее обработка в сценарии.....	77
6.2. Создание пользовательского интерфейса с помощью формы .....	80
6.2.1. Ввод текста. Теги <i>INPUT</i> и <i>TEXTAREA</i> .....	81
6.2.2. Зависимые и независимые переключатели.....	82
6.2.3. Списки выбора .....	83
6.2.4. Форма для передачи файлов.....	85
6.2.5. Кнопки .....	85

6.3. Проверка параметров формы.....	86
6.3.1. Проверка корректности e-mail.....	87
6.3.2. Проверка правильности номера кредитной карты.....	88
6.3.3. Удаление лишних пробелов.....	90
6.4. Защита от спама с помощью формы CAPTCHA.....	90
6.5. Форма поиска в Яндексе.....	92
<b>Глава 7. Запоминание параметров с помощью Cookies и сессий .....</b>	<b>94</b>
7.1. Что такое Cookies и как с ними работать?.....	94
7.2. Механизм сессий .....	96
7.2.1. Сессии и Cookies: преимущества и недостатки .....	96
7.2.2. Для чего нужны сессии? .....	96
7.2.3. Как работает механизм сессий?.....	98
7.2.4. Обход Cookies .....	99
7.3. Массивы и Cookies .....	100
<b>Глава 8. Отдельное слово о директиве <i>register_globals</i> .....</b>	<b>101</b>
8.1. Почему опасно использовать <i>register_globals</i> ? .....	101
8.2. Если <i>register_globals</i> отключена.....	105
<b>РАЗДЕЛ 3. МАССИВЫ И СПИСКИ.....</b>	<b>107</b>
<b>Глава 9. Основные операции над массивами и списками .....</b>	<b>109</b>
9.1. Массив и список. Цикл <i>foreach</i> .....	109
9.2. Функции <i>list()</i> и <i>array()</i> .....	111
9.3. Удаление массива .....	113
9.4. Слияние массивов.....	113
9.5. Функция <i>print_r()</i> .....	115
9.6. Разыменовывание массива.....	116
<b>Глава 10. Функции сортировки массивов.....</b>	<b>118</b>
10.1. Сортировка массивов .....	118
10.2. Функция <i>sort()</i> — сортировка списка.....	118
10.3. Функция <i>asort()</i> — сортировка массива по значениям.....	119
10.4. Функция <i>ksort()</i> — сортировка по ключам.....	120
10.5. Функции <i>array_reverse()</i> и <i>shuffle()</i> .....	121
10.6. Собственная функция сортировки.....	121
10.7. Натуральная сортировка .....	122
<b>Глава 11. Особые операции над массивами.....</b>	<b>124</b>
11.1. Добавление и удаление элементов массива.....	124
11.2. Упаковка переменных в массив и их извлечение .....	125
11.3. Получение части массива.....	127
11.4. Функции автоматического заполнения массива .....	127
11.5. Сравнение массивов .....	128
11.6. Полезные операции над массивом .....	129
11.6.1. Вычисление суммы и произведения всех элементов массива .....	129
11.6.2. Проверка существования элемента в массиве.....	129
11.6.3. Получение случайного элемента из массива.....	130

11.6.4. Удаление дубликатов из массива .....	130
11.6.5. Получение значений и ключей массива .....	131
11.6.6. Замена местами значений и ключей .....	131
11.6.7. Подсчет значений в массиве .....	131
11.6.8. Замена в массиве .....	132
11.6.9. Поиск в массиве .....	132
11.6.10. Прогулка по массиву .....	133
<b>РАЗДЕЛ 4. ФУНКЦИИ В PHP .....</b>	<b>135</b>
<b>Глава 12. Полезные стандартные функции .....</b>	<b>137</b>
12.1. Генератор случайных чисел .....	137
12.2. Дата и время .....	138
12.2.1. Кратко о timestamp .....	138
12.2.2. Функции <i>strtotime()</i> и <i>checkdate()</i> .....	138
12.2.3. Вывод даты .....	139
12.3. Математические функции .....	141
<b>Глава 13. Функции для работы со строками .....</b>	<b>142</b>
13.1. Основные строковые функции .....	142
13.2. Специальные функции замены .....	145
13.3. Преобразование строки .....	147
13.4. Функции преобразования кодировок .....	149
13.5. Функции для работы с отдельными символами строки. Разыменовывание строки .....	150
13.6. Функция <i>md5()</i> и другие функции шифрования/хэширования. API хэширования паролей .....	151
13.7. Функция <i>explode()</i> : выделение подстрок .....	152
13.8. Статистические функции .....	153
13.9. Функции вывода текста .....	154
13.10. Установка локали .....	156
13.11. Форматирование чисел и денежных величин .....	157
13.12. Преобразование систем счисления .....	157
<b>Глава 14. Работаем с файлами и каталогами .....</b>	<b>158</b>
14.1. Права доступа в UNIX .....	158
14.2. Чтение файла .....	160
14.2.1. Функции <i>fopen()</i> и <i>fread()</i> .....	161
14.2.2. Функция <i>file()</i> : построчное чтение файла .....	163
14.2.3. Чтение всего файла: функция <i>file_get_contents()</i> .....	164
14.3. Запись файла .....	164
14.4. Создание временных файлов .....	165
14.5. Работа с CSV-файлами .....	165
14.6. Специальные функции для работы с файлами .....	167
14.6.1. Функции для работы с именами файлов .....	168
14.6.2. Работа с правами доступа .....	168
14.6.3. Копирование, переименование и удаление файлов .....	169
14.6.4. Время доступа к файлу .....	170
14.6.5. Другие полезные функции .....	170
14.7. Совместный доступ к файлу .....	171
14.8. Функции для работы с каталогами .....	172

<b>Глава 15. Вывод графических изображений средствами PHP .....</b>	<b>174</b>
15.1. Библиотека GD.....	174
15.1.1. Получение информации об изображении .....	174
15.1.2. Конвертирование графических форматов .....	177
15.1.3. Вывод текста поверх картинки. Задание цвета .....	179
15.1.4. Прозрачность .....	182
15.1.5. Рисование графических примитивов .....	183
15.1.6. Поворот изображения.....	185
15.2. Изменение размера изображения .....	185
15.3. Создание водяных знаков .....	187
15.4. Поддержка графического формата WebP в PHP 5.5.....	189
<b>Глава 16. Работа с сетевыми сокетами в PHP. Сетевые функции .....</b>	<b>191</b>
16.1. Еще раз о том, что такое сокет .....	191
16.2. Использование сокетов .....	191
16.3. Пример использования сокетов.....	192
16.4. Блокирующий и неблокирующий режимы сокета.....	195
16.5. DNS-функции .....	196
<b>Глава 17. Собственные функции.....</b>	<b>197</b>
17.1. Зачем нужны собственные функции? .....	197
17.2. Особенности функций в PHP .....	197
17.3. Объявление функции .....	198
17.4. Области видимости функции.....	199
17.5. Вложенность функций.....	199
17.6. Переменное число аргументов .....	201
17.7. Передача массивов в качестве параметров.....	202
17.8. Генераторы.....	202
17.9. Особенности использования функции <i>empty()</i> .....	203
<b>РАЗДЕЛ 5. СЕРВЕРЫ БАЗ ДАННЫХ MYSQL 6 И SQLITE .....</b>	<b>205</b>
<b>Глава 18. Основы SQL .....</b>	<b>207</b>
18.1. Немного истории .....	207
18.2. Преимущества SQL .....	207
18.3. Как выглядят запросы?.....	208
18.4. Что такое база данных?.....	209
18.5. Создание таблиц .....	210
18.6. Добавление записей в таблицу .....	214
18.7. Обновление записей .....	214
18.8. Выборка записей.....	215
18.9. Удаление записей.....	216
18.10. Встроенные функции.....	217
18.11. Группировка записей. Сложные запросы .....	218
18.12. Копирование записей из одной таблицы в другую .....	221
18.13. Кэширование запросов.....	222
<b>Глава 19. Функции для работы с MySQL.....</b>	<b>224</b>
19.1. Подключение к серверу MySQL.....	224
19.2. Несколько MySQL-соединений .....	226

19.3. Передача запросов серверу .....	227
19.4. Работа с базой данных. Создание базы данных .....	230
19.5. Функция <i>mysql_real_escape_string(\$content)</i> .....	231
19.6. Получение информации о результате запроса .....	231

## **Глава 20. Альтернативная база данных SQLite .....** 233

20.1. MySQL vs SQLite: что лучше? .....	233
20.2. Открытие базы данных .....	235
20.3. Передача запросов .....	235
20.4. Работа с результатом запроса .....	236
20.5. Список PHP-функций для работы с SQLite .....	237

## **Глава 21. Полезные приемы при работе с MySQL 6 .....** 239

21.1. Выбор кодировки .....	239
21.2. Сортировка: вывод новинок. Вывод случайных записей .....	240
21.3. Постраничный вывод таблицы .....	241

## **РАЗДЕЛ 6. ИНСТРУМЕНТЫ ДЛЯ СОЗДАНИЯ СЛОЖНЫХ ПРОЕКТОВ ...** 245

### **Глава 22. Разработка собственного шаблонизатора .....** 247

22.1. Организация файлов и каталогов проекта .....	247
22.2. Выносим параметры в отдельный файл .....	250
22.3. Подключение дополнительных файлов .....	251
22.3.1. Инструкции <i>include</i> и <i>require</i> .....	251
22.3.2. Альтернативный способ подключения сценариев .....	251
22.3.3. Инструкции <i>include_once</i> и <i>require_once</i> .....	253
22.4. Шаблоны .....	253

### **Глава 23. Шаблонизатор Smarty .....** 258

23.1. Что такое Smarty? .....	258
23.2. Установка Smarty .....	258
23.3. Создание файла <i>setup.php</i> .....	261
23.4. Разработка шаблонов Smarty .....	262
23.4.1. Комментарии в шаблонах .....	262
23.4.2. Переменные в Smarty .....	263
23.4.3. Файлы конфигурации шаблонов .....	264
23.4.4. Служебная переменная <i>{Smarty}</i> .....	265
23.4.5. Модификаторы переменных .....	266
23.4.6. Стандартные (встроенные) функции Smarty .....	269
Функции <i>{include}</i> и <i>{insert}</i> .....	269
Функция <i>{foreach}</i> .....	269
Функции <i>{if}</i> , <i>{elseif}</i> , <i>{else}</i> .....	271
Функция <i>{capture}</i> .....	272
Функция <i>{php}</i> .....	272
Функция <i>{strip}</i> .....	272
23.4.7. Пользовательские функции Smarty .....	273
Функция <i>{assign}</i> .....	273
Функция <i>{cycle}</i> .....	273
Функция <i>{fetch}</i> .....	274



Функции <code>{html_checkboxes}</code> и <code>{html_radios}</code> .....	274
Функция <code>{html_image}</code> .....	275
Функция <code>{html_select_date}</code> .....	275
Функция <code>{html_select_time}</code> .....	276
Функция <code>{html_table}</code> .....	277
23.5. Smarty для программиста.....	278
23.5.1. Специальные переменные.....	278
23.5.2. Полезные методы класса <i>Smarty</i> .....	280
<b>Глава 24. Объектно-ориентированное программирование.....</b>	<b>281</b>
24.1. Основы ООП.....	281
24.2. Классы и объекты.....	282
24.3. Конструкторы и деструкторы класса.....	284
24.4. Наследование классов. Полиморфизм.....	285
24.5. Новые возможности PHP 5 и 5.4/5.5.....	286
24.5.1. Область видимости членов класса.....	286
24.5.2. Абстрактные классы и методы.....	287
24.5.3. Служебное слово <i>final</i> .....	288
24.5.4. Клонирование объектов.....	288
24.5.5. Обработка исключительных ситуаций.....	289
24.5.6. Константы-члены класса.....	292
24.5.7. Статические члены класса.....	293
24.5.8. Оператор <i>instanceof</i> .....	293
24.5.9. Итераторы.....	293
24.5.10. Пространства имен.....	294
24.5.11. Типажи (traits).....	294
24.5.12. Вызов метода или свойства класса выражением.....	295
<b>Глава 25. Механизм сессий.....</b>	<b>297</b>
25.1. Для чего нужны сессии?.....	297
25.2. Как работает механизм сессий?.....	298
25.3. Обход Cookies.....	300
25.4. Сценарий аутентификации.....	300
<b>Глава 26. Введение в PEAR.....</b>	<b>305</b>
26.1. Серьезные проекты и PEAR.....	305
26.2. Пример использования класса <i>DB</i> .....	306
<b>Глава 27. Контроль версий.....</b>	<b>310</b>
27.1. Выбор системы контроля версий.....	310
27.2. Практическое использование TortoiseHG (Mercurial).....	311
27.3. Просмотр внесенных изменений.....	313
<b>Глава 28. Тестирование PHP-сценариев.....</b>	<b>315</b>
28.1. Программа работает, но не так, как нам нужно.....	315
28.2. "Самодельные" точки останова.....	316
28.3. Система автоматического тестирования.....	317
28.4. Директива <i>error_reporting</i> .....	321

<b>ЧАСТЬ II. ПРАКТИКА.....</b>	<b>323</b>
<b>РАЗДЕЛ 7. РАЗРАБОТКА ОСНОВНЫХ ЭЛЕМЕНТОВ САЙТА .....</b>	<b>325</b>
<b>Глава 29. Загрузка файлов на сервер .....</b>	<b>327</b>
29.1. Что нужно знать о загрузке файлов на сервер?.....	327
29.2. Реализация загрузки файла .....	330
29.3. Загрузка нескольких файлов .....	332
29.4. Индикатор загрузки файла на PHP 5.4/5.5.....	334
29.4.1. Некоторые теоретические предпосылки.....	334
29.4.2. Пример практической реализации .....	337
29.5. Проблемы при загрузке файлов.....	344
<b>Глава 30. Использование FTP-функций .....</b>	<b>345</b>
30.1. Функции для работы с FTP .....	345
30.2. Примеры использования FTP-функций .....	348
<b>Глава 31. Отправка и прием почты.....</b>	<b>351</b>
31.1. Отправка почты средствами PHP — функция <i>mail()</i> .....	351
31.2. Отправка писем с вложениями — класс <i>HtmlMimeMail</i> .....	352
31.2.1. Отправка сообщения .....	352
31.2.2. Проблемы при отправке сообщения .....	356
31.2.3. MIME-типы .....	357
31.3. Класс <i>PHPMailer</i> . Разработка сценария автоматической рассылки прайс-листа .....	358
31.4. Получение писем по протоколу POP3 .....	362
<b>Глава 32. Работа с RSS: получаем новости на сайт.....</b>	<b>366</b>
32.1. Краткие сведения о RSS .....	366
32.2. Формат RSS-файла .....	366
32.3. Написание сценария импорта новостей.....	369
32.4. Подключение файла <i>import.php</i> к сайту.....	371
32.5. Создание граббера новостей.....	372
<b>Глава 33. Поиск с использованием регулярных выражений .....</b>	<b>376</b>
33.1. Нужно что-то найти.....	376
33.2. Язык регулярных выражений RegExp.....	377
33.3. Управляющие конструкции .....	379
33.3.1. Квантификаторы .....	379
33.3.2. Альтернативный оператор   .....	379
33.3.3. Скобки .....	379
33.4. Псевдосимволы.....	380
33.5. Практическое использование RegExp-функций.....	380
33.6. Фильтры, или Конец эры регулярных выражений.....	384
<b>Глава 34. Работаем с MP3.....</b>	<b>386</b>
34.1. Формат MP3 .....	386
34.2. Библиотека PEAR .....	387
34.3. Вывод ID3-тегов .....	388
34.4. Редактирование ID3-тегов .....	390
34.5. Удаление тега.....	390

<b>Глава 35. Технология AJAX.....</b>	<b>392</b>
35.1. Что такое AJAX?.....	392
35.2. Ваше первое AJAX-приложение .....	393
 <b>РАЗДЕЛ 8. РАЗРАБОТКА ТИПИЧНОГО САЙТА.....</b>	<b>399</b>
 <b>Глава 36. Создание простейшего движка сайта .....</b>	<b>401</b>
36.1. Планирование движка .....	401
36.1.1. Зачем нужно разрабатывать собственный движок? .....	401
36.1.2. Необходимые нам функции движка.....	402
36.1.3. Принципиальная схема движка .....	404
36.2. Основные функции движка.....	405
36.2.1. Разработка TPL-шаблона .....	405
36.2.2. Файл настроек .....	406
36.2.3. Основной файл CMS — index.php.....	407
36.2.4. Проектирование базы данных .....	409
Таблица <i>static</i> .....	409
Таблица <i>cats</i> .....	410
Таблица <i>pages</i> .....	411
36.2.5. Иерархическая структура сайта.....	412
Алгоритм работы меню .....	412
Сценарий menu.php .....	414
Вывод содержимого раздела и страницы.....	419
36.3. Дополнительные функции движка .....	423
36.3.1. Вывод информации из таблицы <i>static</i> .....	423
36.3.2. Функция вывода содержимого HTML-файла.....	424
36.3.3. Версия для печати.....	425
36.4. Где взять листинги этой главы? .....	426
 <b>Глава 37. Создание фотогалереи .....</b>	<b>427</b>
37.1. Постановка задачи .....	427
37.2. Загрузка изображений на сервер .....	427
37.3. Вывод галереи.....	431
 <b>Глава 38. Гостевая книга.....</b>	<b>434</b>
38.1. Пережиток прошлого?.....	434
38.2. Разработка базы данных и структура гостевой книги .....	435
38.3. Вывод гостевой книги .....	435
38.4. Добавление записей в гостевую книгу.....	438
38.5. Сервисный сценарий gb_service.php .....	441
 <b>Глава 39. Импорт новостей с чужого сайта в базу данных WordPress.....</b>	<b>443</b>
39.1. Постановка задачи .....	443
39.2. Вспомогательные функции .....	444
39.3. Вкратце о cURL .....	446
39.4. Основной сценарий .....	446
 <b>Глава 40. Создание счетчика сайта .....</b>	<b>451</b>
40.1. Постановка задачи .....	451
40.2. Файл конфигурации.....	451

40.3. Разработка таблиц <i>counter</i> и <i>ipaddr</i> .....	452
40.4. Сценарий <i>counter.php</i> .....	453
40.5. Сценарий <i>reset_counter.php</i> .....	457
<b>Глава 41. Статистика сайта .....</b>	<b>459</b>
41.1. Методы сбора статистики .....	459
41.2. Программы-анализаторы журналов веб-сервера .....	460
41.3. Системы статистики .....	462
<b>Глава 42. Голосования и комментарии для сайта .....</b>	<b>465</b>
42.1. Разработка собственной системы голосования.....	465
42.1.1. Общая структура системы .....	465
42.1.2. Разработка сценария <i>poll_form.php</i> .....	466
42.1.3. Сценарий <i>poll_process.php</i> .....	468
42.1.4. Сценарий <i>poll_results.php</i> .....	469
42.2. Комментарии DISQUS .....	472
42.2.1. Описание платформы .....	472
42.2.2. Установка комментариев на свой сайт .....	473
<b>РАЗДЕЛ 9. СЛОЖНЫЙ ПРОЕКТ .....</b>	<b>479</b>
<b>Глава 43. Постановка задачи .....</b>	<b>481</b>
43.1. Концепция проекта .....	481
43.2. Алгоритм работы .....	482
43.3. Структура базы данных.....	483
<b>Глава 44. Разработка основной части сайта .....</b>	<b>487</b>
44.1. Структура проекта .....	487
44.2. Основной сценарий <i>index.php</i> .....	488
44.2.1. Подключение механизмов WordPress .....	488
44.2.2. Аутентификация пользователя .....	489
44.2.3. Форма добавления цели .....	490
44.2.4. Добавление цели.....	491
44.2.5. Просмотр всех целей.....	495
44.3. Сценарий <i>v.php</i> — просмотр цели.....	499
44.4. Сценарии отправки уведомлений. Сценарии <i>yes.php</i> и <i>no.php</i> .....	500
<b>Глава 45. Панель управления WordPress.....</b>	<b>508</b>
45.1. О чем эта глава?.....	508
45.2. Вход в панель управления.....	508
45.3. Просмотр целей .....	508
45.4. Не забывайте о UTF-8 .....	510
<b>РАЗДЕЛ 10. ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ САЙТА.....</b>	<b>511</b>
<b>Глава 46. SSL-соединения.....</b>	<b>513</b>
46.1. Защищаем передаваемые данные.....	513
46.2. Настройка SSL в DirectAdmin.....	514
46.3. SSL-переменные .....	516

<b>Глава 47. Защита PHP с помощью конфигурационного файла .....</b>	<b>518</b>
47.1. Конфигурационный файл <code>php.ini</code> .....	518
47.2. Отключение потенциально опасных функций .....	520
47.3. Рекомендованные значения некоторых конфигурационных директив .....	520
<b>Глава 48. Защита сайта от атак .....</b>	<b>521</b>
48.1. Сайт в опасности .....	521
48.2. Два самых распространенных метода взлома .....	522
48.3. Межсайтовый скриптинг .....	522
48.4. SQL-инъекции .....	524
48.5. Флуд .....	528
48.6. Защита форума PHPBB2 от спамеров .....	529
<b>РАЗДЕЛ 11. ПОЛЕЗНЫЕ СВЕДЕНИЯ .....</b>	<b>531</b>
<b>Глава 49. Визуальный редактор SPAW2 для вашего сайта .....</b>	<b>533</b>
49.1. Знакомство с визуальным редактором и его базовая настройка .....	533
49.2. Адаптация редактора к вашему сайту .....	537
49.2.1. Загрузка в редактор текста статьи .....	537
49.2.2. Вывод редактора в TPL-шаблоне .....	537
49.2.3. Запрет изменения размера редактора .....	538
49.2.4. Изменение размеров редактора "на лету" — в процессе выполнения сценария .....	539
49.2.5. Панель инструментов редактора .....	539
49.2.6. Ошибка Error 105 (net::ERR_NAME_NOT_RESOLVED) .....	542
<b>Глава 50. WAP-сервер средствами PHP .....</b>	<b>543</b>
50.1. Есть ли необходимость в WAP? .....	543
50.2. Настройка веб-сервера Apache .....	544
50.3. PHP-сценарий .....	545
<b>Глава 51. Пишем коммерческий сценарий .....</b>	<b>546</b>
51.1. Zend Guard: за и против .....	546
51.2. Установка Zend Guard .....	548
51.3. Быстрая защита сценария .....	548
51.4. Ошибка Unable to locate XML file .....	552
51.5. Лицензии .....	552
51.6. Zend Guard — панацея? .....	556
<b>ПРИЛОЖЕНИЯ .....</b>	<b>557</b>
<b>Приложение 1. Программа phpMyAdmin .....</b>	<b>559</b>
П1.1. Действия над таблицами .....	560
П1.2. Создание новой таблицы. Изменение структуры таблицы .....	562
П1.3. Вставка записей .....	563
П1.4. Обзор таблицы .....	563
П1.5. Выполнение произвольного SQL-кода .....	563
П1.6. Резервное копирование БД .....	565
<b>Приложение 2. Flash и PHP .....</b>	<b>566</b>

<b>Приложение 3. Оптимизация PHP-кода .....</b>	<b>570</b>
ПЗ.1. Правильное использование <i>echo</i> .....	570
ПЗ.2. Длина переменных.....	571
ПЗ.3. Правильное обращение к элементам массива .....	571
ПЗ.4. Циклы.....	572
ПЗ.5. Чтение файлов.....	573
ПЗ.6. Вывод текста: <i>echo</i> против <i>printf</i> .....	575
ПЗ.7. "Лишние" вызовы функций.....	577
ПЗ.8. Контрольные замеры .....	578
ПЗ.8.1. Чтение файла.....	578
ПЗ.8.2. Вывод содержимого файла .....	579
ПЗ.9. Выводы .....	581
<b>Приложение 4. Описание электронного архива.....</b>	<b>582</b>
<b>Предметный указатель .....</b>	<b>584</b>

# Предисловие

PHP (Hypertext Preprocessor) — один из самых популярных языков программирования, используемый для разработки веб-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 5-м издании?

По традиции, обновление книги началось с *первой главы*. Однако в рассказе о PHP версии 5.5 банальной заменой иллюстраций уже не обойтись. Теперь у программиста есть выбор: или использовать уже готовые пакеты программного обеспечения типа XAMPP<sup>1</sup> или Денвер<sup>2</sup>, или же настраивать систему самостоятельно. Если раньше инсталлятор PHP делал всю необходимую работу, а пользователю нужно было лишь выбрать каталог, в который он устанавливал Apache, то теперь редактировать конфигурационные файлы веб-сервера придется вручную, потому что инсталлятор PHP для Windows упразднен и, похоже, что навсегда. Отсутствие инсталлятора обнаружилось в PHP версии 5.4, и это сразу усложнило процесс установки программного обеспечения. И если во время подготовки 4-го издания книги, когда версия 5.4 только-только вышла, казалось, что разработчики PHP просто не успели создать инсталлятор для Windows, то после выхода версии 5.5 стало понятно, что простого инсталлятора больше не будет. Именно поэтому в *первой главе* описан процесс установки PHP версии 5.5 во всех подробностях.

---

<sup>1</sup> XAMPP — кроссплатформенная сборка веб-сервера, содержащая Apache, MySQL, интерпретатор скриптов PHP, язык программирования Perl и большое количество дополнительных библиотек, позволяющих запустить полноценный веб-сервер.

<sup>2</sup> Денвер (от сокр. Д.н.в.р или ДНВР — джентльменский набор Web(веб)-разработчика) — набор дистрибутивов и программная оболочка, предназначенные для создания и отладки сайтов (веб-приложений, прочего динамического содержимого интернет-страниц) на локальном ПК (без необходимости подключения к сети Интернет) под управлением ОС Windows.

В *главе 8* читатели найдут продолжение эпопеи с директивой `register_globals`. Если мне не изменяет память, то, начиная с PHP версии 4.3, она выключена, а в версии 6 (которая до сих пор не вышла) ее собирались упразднить. Однако на дворе 2014 год, вышла версия PHP 5.5, а директива `register_globals` все еще присутствует в конфигурационном файле `php.ini`. (Честно говоря, я думал, что в версии 5.5 от нее наконец-то избавятся, и в *главе 8* информацию о ней можно будет заменить чем-то другим.)

Из *главы 9* вы узнаете о том, что такое *разыменовывание массива*, как использовать `list()` внутри `foreach`, будет также приведено сравнение функций `var_dump()` и `print_r()` для анализа содержимого массива.

В новой *главе 13* рассмотрено *разыменовывание строки* (впрочем, эта процедура нам не очень полезна и вызывает разве что академический интерес), а также API хэширования паролей (Password Hashing API).

О поддержке нового графического формата, разработанного компанией Google, мы поговорим в *главе 15* — при рассмотрении библиотеки GD.

В *главе 17* будут рассмотрены функции-генераторы и приведены особенности использования функции `empty()`.

В *главе 24* вы познакомитесь с новыми служебными словами: `finally` и `class`. Первое служит для обработки исключений, а второе — для вывода имени класса.

Существенные изменения внесены в *главу 29*. Дело в том, что в PHP 5.4 появился механизм контроля процесса загрузки файлов, позволяющий создавать индикаторы загрузки практически средствами PHP, не прибегая к сторонним средствам. Конечно, "не прибегая к сторонним средствам" — это сказано сильно. Вам все равно придется использовать jQuery для обращения к этому механизму, но не будем забегать вперед — обо всем этом вы и узнаете из *главы 29*. Надеюсь, я вас заинтриговал...

Полностью заменена *глава 39*. Теперь вместо безнадежно устаревшего и неактуального примера интеграции в свой сайт галереи LiveJournal она посвящена импорту новостей с чужого сайта в базу данных WordPress.

Остальные главы книги также подверглись основательному пересмотру, как с точки зрения повышения их удобочитаемости, так и в плане приведения всей терминологии к современному звучанию.

Электронный архив с информацией, расширяющей и дополняющей материал "бумажной" книги, можно скачать с FTP-сервера издательства по ссылке: <ftp://ftp.bhv.ru/9785977535144.zip>, со страницы книги на сайте [www.bhv.ru](http://www.bhv.ru) или с сайта автора: [http://dkws.org.ua/mybooks/php\\_adds.zip](http://dkws.org.ua/mybooks/php_adds.zip). Архив этот также обновлен и теперь содержит дополнительные материалы к 4-му и 5-му изданиям книги, включая новое программное обеспечение (см. *приложение 4*).

## Немного истории

В 1994 году датский программист Расмус Лердорф (Rasmus Lerdorf) создал надстройку (набор скриптов) над Perl/CGI 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 С 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 году появилась вторая версия интерпретатора — PHP/FI 2.0. Она была установлена примерно на 50 тыс. серверов. С одной стороны, цифра внушительная, но это всего лишь 1 % от общего числа интернет-серверов на тот момент.

В 1998 году появился PHP 3.0. Именно с этой версии аббревиатура "PHP" стала 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, и действительно — сценарии выполнялись медленно. Поэтому разработку версии PHP 4 начали с переработки ядра. Начиная с PHP 4, интерпретатор сначала транслирует PHP-код во внутреннее представление, а потом выполняет (а не выполняет сценарий строка за строкой, как в PHP 3), благодаря чему существенно повысилась производительность сценариев. Четвертая версия PHP появилась в мае 2000 года. Обновления для этой версии PHP выпускались до конца 2007 года, но с августа 2008 года поддержка PHP 4 была прекращена.

Пятая версия PHP появилась в июле 2004 года. Она отличалась переработанным ядром Zend, что увеличило эффективность интерпретатора, добавлением поддержки XML, переработанной поддержкой объектно-ориентированного программирования (ООП). С этого момента PHP стал полноценным объектно-ориентированным языком программирования, а его объектная модель во многом схожа с моделью Java. Несмотря на наличие версии 5.4, все еще актуальной остается версия 5.3. Далеко не все хостинг-провайдеры успели перейти на новую версию PHP, поэтому как минимум еще год на хостингах версия 5.3 будет по-прежнему встречаться.

## PHP 5.5, или что случилось с PHP 6?

Разработка шестой версии PHP началась в октябре 2006 года. В ней было сделано множество нововведений.

Прежде всего, в шестой версии значительно улучшена поддержка Unicode<sup>1</sup>, что чрезвычайно важно. Раньше, когда в знаковый генератор можно было загрузить только 256 символов, поддержка разных языков оказывалась существенно ограниченной, вследствие чего и появилось множество разных кодировок — иногда даже по несколько для одного и того же языка. Русский язык не стал исключением. Для

---

<sup>1</sup> Получить список всех функций, поддерживающих Unicode, можно по адресу [http://www.php.net/~scoates/unicode/render\\_func\\_data.php](http://www.php.net/~scoates/unicode/render_func_data.php).

него было разработано три кодировки: CP866 (DOS), Windows-1251 (или CP-1251, Windows) и KOI8-R (UNIX/Linux). Использование разных кодировок всегда поднимало проблему их конвертирования — например, символ "Б" во всех этих кодировках имеет разные коды.

Первая версия Unicode, представленная в 1991 году, позволяла хранить 65 536 символов. Текущая версия Unicode позволяет хранить 2 147 483 648 символов, чего вполне достаточно для хранения символов всех языков и, в частности, кириллицы. Так что, когда все программное обеспечение будет переведено на Unicode (а это произойдет в ближайшем будущем), проблема несовместимости кодировок исчезнет<sup>1</sup>.

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, в 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.

Но шестой версии PHP так и не суждено было увидеть свет. Камнем преткновения стала как раз поддержка Unicode — от нее решили отказаться, а все наработки вынесли в отдельную ветку, которая вскоре превратилась в версию 5.4. И даже сейчас, когда у нас уже вовсю течет 2014 год, версия 6 так и не появилась на горизонте. Вместо нее 20 июля 2013 года вышла версия 5.5, которая и стала поводом для пятого издания этой книги.

Далее представлен список нововведений в версиях 5.4 (хотя с момента выхода версии 5.5 прошло уже более полугода, версия 5.4 остается актуальной, и в большинстве случаев на серверах хостинг-провайдеров установлена именно она) и 5.5.

---

<sup>1</sup> Подробно о Unicode вы можете прочитать по адресу <http://ru.wikipedia.org/wiki/Unicode>.

## Основные нововведения в PHP версии 5.4

### Типажи

Существенно упрощают мультинаследование *типажи* (traits). Сейчас мы не будем подробно их рассматривать, поскольку без знания объектно-ориентированного программирования читатель вряд ли что поймет. Скажу только, что это нововведение описано в *главе 24*. Там же описаны дополнительные новшества, появившиеся в PHP 5.4 и связанные с ООП.

### Разыменовывание массива

Поддерживается *разыменовывание массива*. Представим, что у нас есть метод, возвращающий массив, например:

```
function foo() {  
    return array(1, 2, 3);  
}
```

Функция возвращает массив из трех элементов. Как обратиться, скажем, к нулевому элементу? Раньше это делалось так:

```
$res = foo();  
echo $res[0];
```

Теперь можно сделать так:

```
echo foo()[0];
```

### Встроенный веб-сервер

Фактически теперь PHP-разработчикам не нужен Apache, поскольку в PHP 5.4 встроен собственный веб-сервер, запустить который можно из консоли:

```
php -S localhost:8080 -t /htdocs
```

В данном случае мы "повесили" сервер на порт 8080, а каталог /htdocs станет корневым каталогом документов, — именно в этом каталоге наш сервер будет искать документы. Конечно, этот сервер очень простой, но он лучше, чем ничего, когда нет возможности установить другой сервер.

### Индикатор загрузки файла

Наверное, все вы видели индикаторы загрузки файла на сервер. Как такой сделать? В PHP 5.4 это очень просто. Теперь информация о процессе загрузки файла хранится в сессии пользователя — например, в ключе `upload_progress`. Далее дело техники — можно получить значение этой переменной и отобразить его пользователю (или преобразовать в графический индикатор). В этом вам поможет Ajax (чтобы изменять отображаемое значение "на лету" без перезагрузки страницы).

Конечно, это не все новшества версии 5.4, об остальных вы узнаете по ходу чтения книги.

## Основные нововведения в PHP версии 5.5

### Генераторы

Генератор — это функция, генерирующая последовательность значений. О том, как использовать генераторы, мы поговорим в *главе 17*.

### Ключевые слова *finally* и *class*

Теперь в исключениях (блоках `try`) можно использовать слово `finally`, которого так не хватало многим PHP-программистам. О том, как использовать `finally`, мы поговорим, когда будем рассматривать исключения, а именно в *главе 24*. В этой главе мы рассмотрим и другое новое ключевое слово — `class`.

### API хэширования паролей

В версии 5.5 появился механизм хэширования паролей — Password Hashing API. Его возможности будут рассмотрены в *главе 13*.

### Разыменовывание строк

В версии 5.4, как уже отмечалось, появилась возможность разыменовывания массивов, а в версии 5.5 — возможность разыменовывания строк. Если вы читали предыдущее издание, то уже догадываетесь, как будет работать новый механизм. Если же нет, тогда загляните в *главу 13*.

### Изменения в работе конструкции *foreach*

Теперь конструкция `foreach` может работать с ключами не скалярного типа. В `foreach` появилась также возможность использовать конструкцию `list()`. Эти нововведения будут рассмотрены в *главе 9*.

### Поддержка графического формата WebP

Формат WebP — это формат сжатия изображений с потерями и без потерь качества, предложенный компанией Google в 2010 году. Теперь поддержка этого формата появилась и в PHP, а рассмотрена она будет в *главе 11*.

### Исключенные возможности

Из PHP 5.5 исключены следующие возможности:

- ❑ удалены функции, касающиеся PHP Logo: `php_logo_guid()`, `php_egg_logo_guid()`, `php_real_logo_guid()`, `zend_logo_guid()`;
- ❑ в функции `preg_replace()` больше не поддерживается модификатор `/e`;

- ❑ больше не поддерживаются операционные системы Windows XP и Windows 2003. Чтобы установить PHP 5.5, вам придется обновить свою операционную систему до Windows 7 или 8;
- ❑ расширение `mysql` использовать не рекомендуется, вместо него лучше обратиться к `mysqli`.

На мой взгляд, некоторые изменения в версии 5.5 вызывают вопросы. Как бы и изменения есть, но пользы от них нет. Конечно, есть и полезные нововведения — например, генераторы, ключевое слово `finally`, но что касается разыменовывания строк и массивов, а также использования функции `empty()` над другой функцией (см. главу 17), то пользы от таких новшеств мало. Но обо всем по порядку. В соответствующих главах будут рассмотрены те или иные нововведения и рассказано, почему одни из них представляются полезными, а другие — нет.

## MySQL

Ни один серьезный веб-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. В настоящее время доступна уже шестая версия этого популярного сервера. Кроме MySQL мы здесь рассмотрим еще одну альтернативную систему управления базами данных — SQLite. Почему именно SQLite? На заре 5-й версии PHP ходили слухи, что новая (т. е. 5-я 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 и стал SQLite. К счастью, поддержка MySQL из PHP 5 удалена не была, и последние версии PHP 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](http://www.dkws.org.ua)**.



**ЧАСТЬ I**

---

**ТЕОРИЯ**



# **I. РАЗДЕЛ 1**

## **Быстрый старт**

<b>Глава 1.</b>	Установка необходимого программного обеспечения
<b>Глава 2.</b>	Программа на PHP
<b>Глава 3.</b>	Основы синтаксиса PHP



# ГЛАВА 1

## Установка необходимого программного обеспечения

### 1.1. Нужно ли устанавливать программное обеспечение?

Подготавливая книгу к печати, я вообще сомневался в целесообразности включения в нее такой 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, что PHP-хостинг сейчас стоит очень дешево, как и безлимитный Интернет, то все, что требуется для PHP-разработки, — это какой-нибудь текстовый редактор с поддержкой PHP-синтаксиса.

Ведь вам всего-то необходимо написать сценарий. В свою очередь, сценарий — это обычный текстовый файл. Как только сценарий будет готов, 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 Mac OS.

Ранее имело смысл устанавливать на свой компьютер связку Apache + PHP + сервер баз данных MySQL. Сейчас же 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 окажутся различными, поэтому при переносе сценариев могут возникнуть проблемы. 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 проблемы могут возникнуть глобальные. Помню, у меня был отлично работающий проект, отправляющий прайс-листы клиентам одной из поддерживаемых мною организаций. Сценарий был предельно прост — пользователь загружал новый прайс-лист, а сценарий отправлял его всем клиентам, адреса которых были в базе данных. Для отправки сообщений использовался класс `PHPMailer`. Но после переноса на другой сервер сценарий перестал за-



пускаться, причем даже не выдавал сообщение об ошибке. Для выявления причины такого нештатного поведения сервера первым делом я запустил функцию `phpinfo()`:

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

Результат меня просто поразил — оказалось, что на сервере установлена четвертая (!) версия PHP. Учитывая существование версии 5.5, было странно, что кто-то еще пользуется таким антиквариатом, как PHP 4. Решить проблему удалось путем загрузки класса `PHPMailer` для PHP 4. Хорошо, что таковой до сих пор еще 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 эта 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, под какую версию PHP готовить сценарии? Раньше на серверах могла быть установлена или одна из ранних версий PHP 5, или проверенная временем четвертая версия. Но сейчас PHP 4 на сервере за редким исключением уже не встретишь — все перешли на PHP 5.

Однако не забывайте, что в каждой версии PHP: 5.1, 5.2, 5.3, 5.4 и 5.5 — имеются свои изменения. Например, новые параметры у функций и новые функции. Для изучения PHP, по большому счету, все равно, какую взять версию. Главное, чтобы вы знали, какая она именно, дабы не возникало потом вопросов, почему та или иная функция не работает, как следует.

Впрочем, на самом деле все не так страшно. Большинство функций будут работать во всех версиях PHP 5, так что вы не испытаете никаких проблем. Исключение могут составить разве что версии 5.4/5.5. По сути, это версия 6.0 (которая так и не вышла), но без поддержки Юникода.

Пока хостинг-провайдеры не спешат устанавливать версию 5.4 (не говоря уже о 5.5). Так, на одном из моих хостингов стоит версия 5.3, а на другом — вообще 5.2, хотя версии 5.4/5.5 уже давно стабильны.

Так как же быть с самыми последними версиями: 5.4 и 5.5? Поскольку есть релиз версии 5.5, то изучать ее стоит. Однако будьте пока готовы, что сейчас далеко не все хостеры используют 5.5, но со временем переход обязательно состоится. В настоящее время вы можете загрузить PHP 5.5 на свой компьютер и изучить новые возможности этой версии.

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

Мы уже отмечали, что, начиная с версии 5.4, у PHP имеется свой встроенный веб-сервер, но я рекомендую все же использовать Apache, поскольку в большинстве

случаев на сервере хостера будет установлен именно этот сервер, так что условия окажутся ближе к "боевым".

## 1.2. Выбор PHP-редактора и FTP-клиента

Из всех PHP-редакторов, которыми я пользовался, мне более всех 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 в каталоге *software* сопровождающего книгу электронного архива (см. *приложение 4*). Последнюю версию программы также всегда можно скачать с сайта разработчиков: <http://phpexperteditor.com/>.

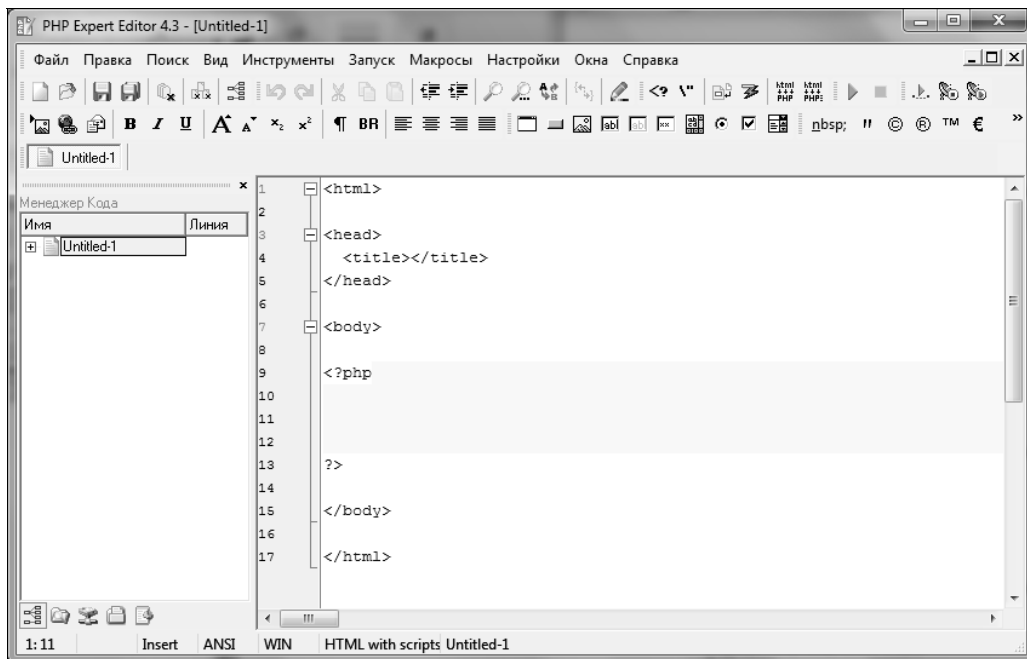


Рис. 1.1. Программа PHP Expert Editor

Чтобы указать PHP в настройках программы PHP Expert Editor, выполните команду **Запуск | Настройки**, в открывшемся окне перейдите на вкладку **Интерпретаторы скриптов** (рис. 1.2) и укажите путь к интерпретатору PHP. Для проверки синтаксиса нажмите клавишу <F7>.

Возможности Zend Studio (рис. 1.3) куда шире. Чего стоит только автодополнение кода, когда программа "читает" ваши мысли и дописывает за вас PHP-код. Если вы

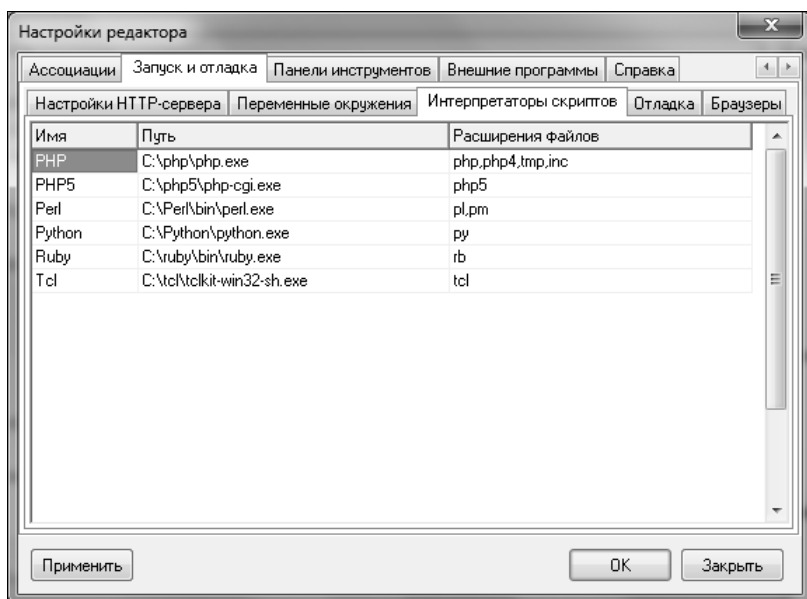


Рис. 1.2. Путь к интерпретатору PHP

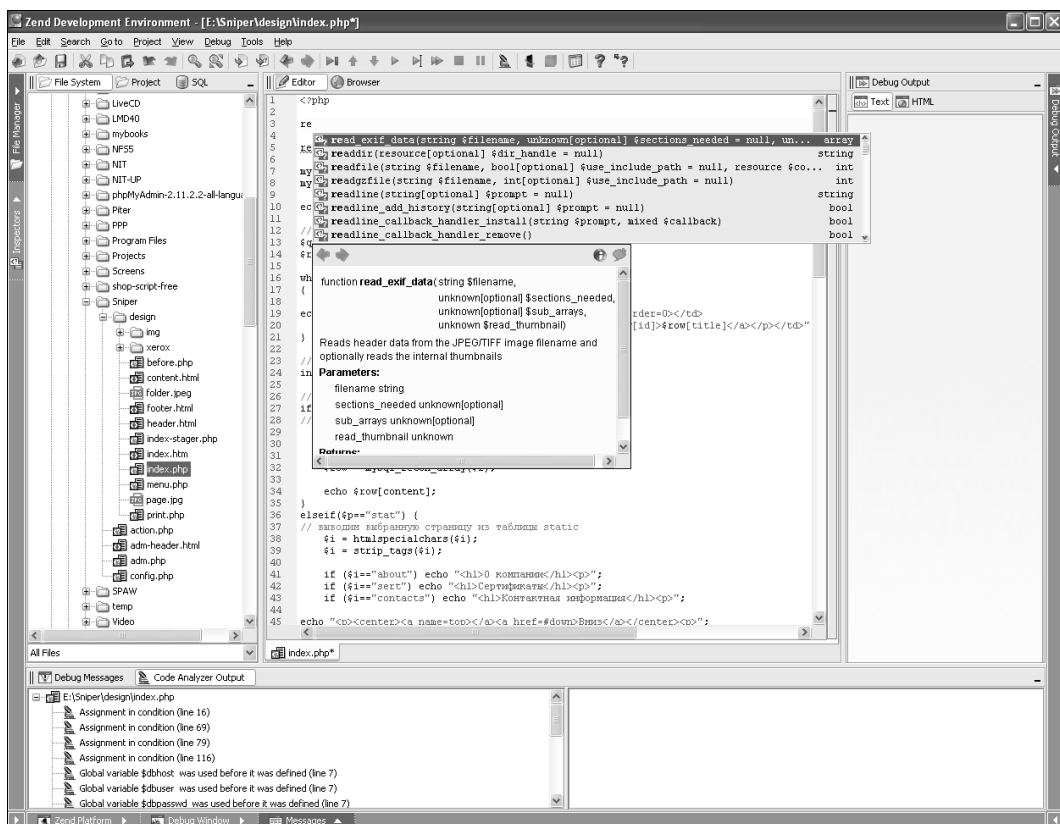


Рис. 1.3. Zend выводит не только имена функций, но и краткую справку по каждой функции

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.

### ПРИМЕЧАНИЕ

Версия 3.5.3 FTP-клиента FileZilla находится в каталоге *software* сопровождающего книгу электронного архива (см. приложение 4).

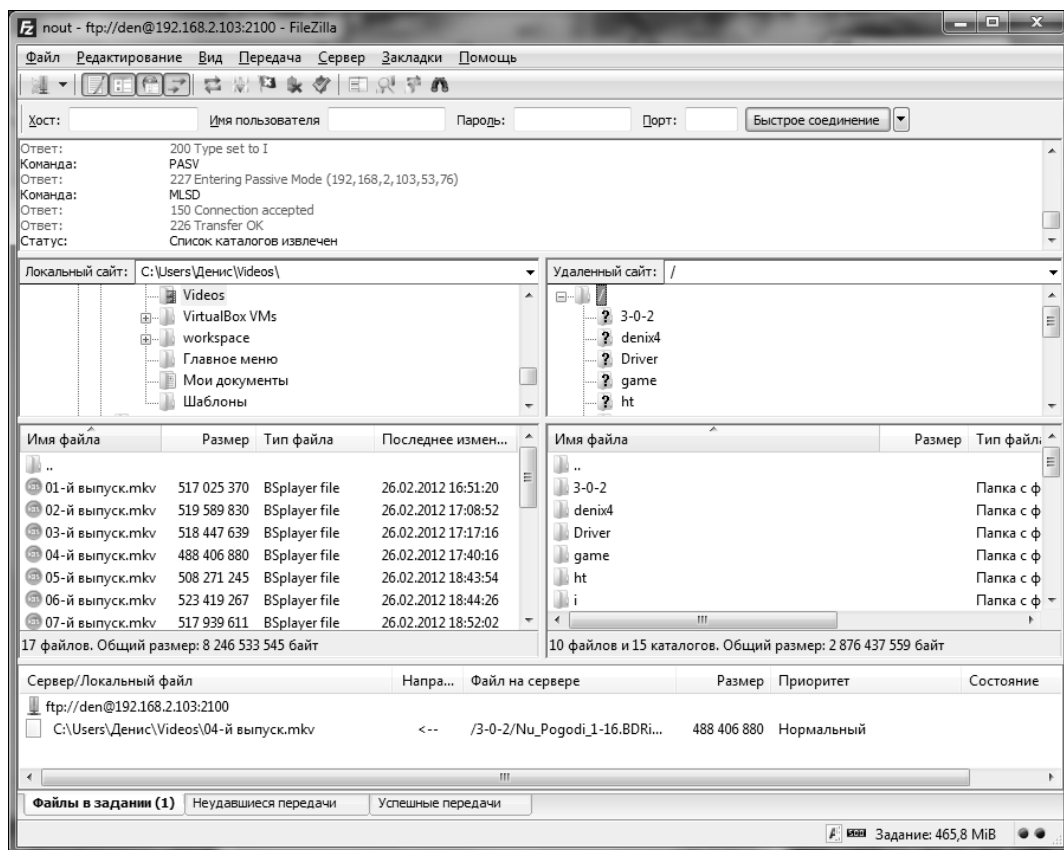


Рис. 1.4. Программа FileZilla

## 1.3. Установка связки Apache + PHP + MySQL в Windows

### 1.3.1. Для опытных пользователей: установка вручную

Если вы все-таки решили настроить собственный сервер, приступим к его установке. Первым делом вам нужно скачать с сайта [www.apache.org](http://www.apache.org) веб-сервер Apache —

это файл `httpd_2.2.X-win32-x86-no_ssl.msi`. Запустите скачанный файл. Процесс установки обычно проходит без осложнений, но мы все же его рассмотрим.

### ПРИМЕЧАНИЯ

Версия 2.4 доступна пока только для пользователей UNIX, поэтому Windows-пользователям придется довольствоваться версией 2.2.

В каталоге *software* сопровождающего книгу электронного архива (см. приложение 4) вы найдете последнюю версию Apache для Windows. Linux-версии в архиве нет, поскольку Apache входит в состав любого дистрибутива Linux, и вам нужно будет только установить его. В архив включены версия Apache 2.2.25 (файл `httpd_2.2.X-win32-x86-no_ssl.msi`), а также модуль `mod_ftp` для Windows (файл `mod_ftp-0.9.6-beta-win32-x86.zip`). Несмотря на наличие в архиве установочного файла Apache, не забудьте посетить страницу <http://archive.apache.org/dist/httpd/binaries/win32/> — вполне возможно, что к моменту покупки книги вышла новая версия Apache.

### ОСОБЕННОСТИ УСТАНОВКИ 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 станет доступен всем пользователям;
- ☐ **only for the Current User** — сервер нужно будет запускать вручную, и он станет использовать порт 8080, т. е. при обращении к серверу придется указывать этот порт в URL — например: **`http://localhost:8080/`**.

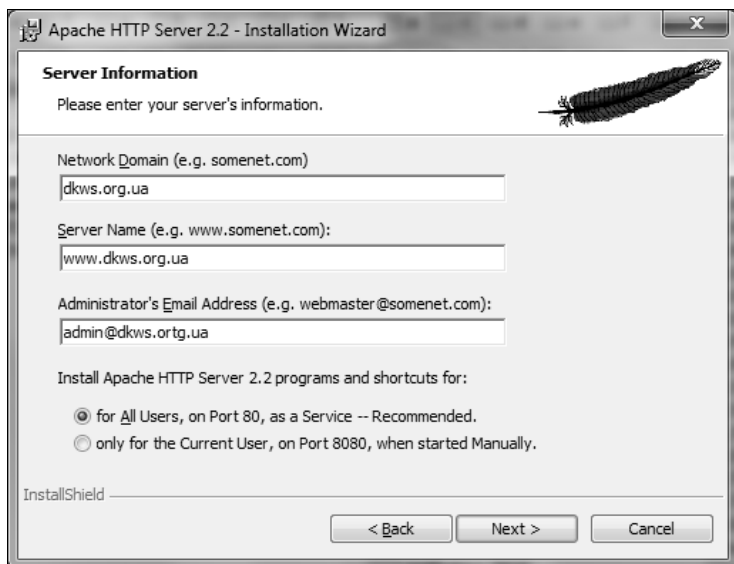


Рис. 1.5. Параметры сервера