

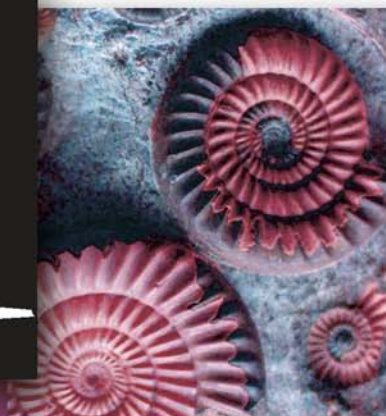
ДЕНИС КОЛИСНИЧЕНКО



PHP и MySQL

Разработка веб-приложений

6-е издание



**СИНТАКСИС PHP.
ОСОБЕННОСТИ PHP 7.x**

**ОБЪЕКТНО-
ОРИЕНТИРОВАННОЕ
ПРОГРАММИРОВАНИЕ
НА PHP 7.x**

**MySQL и SQLite.
РАСШИРЕНИЕ PDO**

ОСНОВЫ Ajax

**РАБОТА С RSS, MP3,
reCAPTCHA, Curl, MobileDetect**

**ВВЕДЕНИЕ ВО ФРЕЙМВОРК
Symfony**

**УСТРАНЕНИЕ ТИПИЧНЫХ
SEO-ОШИБОК, ДОПУСКАЕМЫХ
ПРОГРАММИСТАМИ**

PRO

**ПРОФЕССИОНАЛЬНОЕ
ПРОГРАММИРОВАНИЕ**

Денис Колисниченко

PHP и MySQL

Разработка веб-приложений

6-е издание

Санкт-Петербург

«БХВ-Петербург»

2017

УДК 004.43+004.738.5
ББК 32.973.26-018.1
К60

Колисниченко Д. Н.

К60 РНР и MySQL. Разработка веб-приложений. — 6-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2017. — 640 с.: ил. — (Профессиональное программирование)

ISBN 978-5-9775-3835-0

Даны начала программирования на РНР: установка и настройка Apache, РНР и MySQL, а также кроссплатформенной сборки ХАМРР, выбор редактора РНР-кода, синтаксис языка и самые полезные функции РНР. Рассмотрено создание собственного движка сайта и ряда дополнительных модулей — фотогалереи, RSS-граббера, модуля для работы с MP3, сервиса reCAPTCHA, 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 шестом издании рассмотрены нововведения версии РНР 7.x, особенности создания индикатора загрузки файла и разыменовывания массивов, приведены примеры устранения типичных SEO-ошибок, допускаемых программистами, описана работа с PDO, Curl и MobileDetect, дано введение во фреймворк Symfony.

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 31.03.17.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 51,6.

Тираж 1400 экз. Заказ №

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

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

ISBN 978-5-9775-3835-0

© ООО "БХВ", 2017
© Оформление. ООО "БХВ-Петербург", 2017

Оглавление

- Предисловие 1**
 - Что нового в 6-м издании? 1
 - Немного истории 2
 - PHP 7, или что случилось с PHP 6?..... 3
 - Основные нововведения в PHP версии 7..... 4
 - Улучшена производительность 4
 - Удалены старые расширения и SAPI 4
 - Изменения в синтаксисе языка 4
 - Новые возможности 5
 - Измененные функции..... 6
 - Новые функции..... 6
 - Поддержка читателей 6
- 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 14
 - 1.3. Установка связки Apache + PHP + MySQL в Windows..... 17
 - 1.3.1. Для опытных пользователей: установка вручную 17
 - Установка веб-сервера Apache 17
 - Установка PHP..... 19
 - 1.3.2. Установка для новичков (рекомендуется) 26
 - 1.4. Установка Apache + PHP + MySQL в Linux 29
 - 1.5. Конфликты с другим программным обеспечением 30
 - 1.6. Несколько советов 31
- Глава 2. Программа на PHP..... 33**
 - 2.1. Ваша первая программа 33
 - 2.2. Запуск PHP-программы..... 35
 - 2.3. Вывод текста без *echo* 35

Глава 3. Основы синтаксиса PHP	38
3.1. Переменные.....	38
3.1.1. Правила объявления переменных. Имена переменных.....	38
3.1.2. Типы данных переменных	39
3.1.3. Булевы переменные	40
3.1.4. Операции над переменными.....	41
3.1.5. Ссылки.....	42
3.2. Константы	43
3.3. Выражения и операции	47
3.3.1. Что такое <i>выражение</i> ?	47
3.3.2. Арифметические операции	47
3.3.3. Логические выражения	48
3.3.4. Битовые операции.....	48
3.3.5. Приоритеты операций.....	49
3.3.6. Операторы эквивалентности == и ===	50
3.3.7. Оператор ?? — сокращенная форма тернарной условной операции.....	51
3.3.8. Операции со строками.....	51
3.4. Условный оператор	52
3.5. Циклы	53
3.5.1. Цикл со счетчиком <i>for</i>	53
3.5.2. Цикл <i>while</i>	54
3.5.3. Цикл <i>do-while</i>	54
3.5.4. Принудительное завершение цикла и пропуск итерации.....	55
3.6. Оператор выбора <i>switch-case</i>	55
 РАЗДЕЛ 2. ПЕРЕДАЧА ПАРАМЕТРОВ PHP-ПРОГРАММАМ.....	57
 Глава 4. Методы <i>GET</i> и <i>POST</i>.....	59
4.1. Интерфейс CGI	59
4.2. Метод <i>GET</i>	61
4.3. Метод <i>POST</i>	61
 Глава 5. Протокол HTTP и интерфейс CGI	63
5.1. Специальные переменные окружения CGI.....	63
5.2. Заголовки протокола HTTP	64
5.3. Коды ответов протокола HTTP	65
 Глава 6. Передача параметров посредством HTML-формы	67
6.1. Создание простейшей формы и ее обработка в сценарии.....	67
6.2. Создание пользовательского интерфейса с помощью формы.....	70
6.2.1. Ввод текста. Теги <i>INPUT</i> и <i>TEXTAREA</i>	71
6.2.2. Зависимые и независимые переключатели.....	72
6.2.3. Списки выбора.....	73
6.2.4. Форма для передачи файлов.....	75
6.2.5. Кнопки	75
6.3. Проверка параметров формы.....	76
6.3.1. Проверка корректности e-mail.....	77
6.3.2. Проверка правильности номера банковской карты	78
6.3.3. Удаление лишних пробелов.....	79

6.4. Защита от спама с помощью формы CAPTCHA.....	80
6.5. Форма поиска в Яндексе	82
Глава 7. Запоминание параметров с помощью Cookies и сессий	84
7.1. Что такое Cookies и как с ними работать?.....	84
7.2. Механизм сессий	86
7.2.1. Сессии и Cookies: преимущества и недостатки	86
7.2.2. Для чего нужны сессии?	86
7.2.3. Как работает механизм сессий?.....	88
7.2.4. Обход Cookies	89
7.3. Массивы и Cookies	90
Глава 8. Не забываем о поисковой оптимизации	91
8.1. "Дружественные" интернет-адреса	91
8.2. Идентификаторы сеансов.....	94
8.3. Производительность сценария	94
РАЗДЕЛ 3. МАССИВЫ И СПИСКИ.....	97
Глава 9. Основные операции над массивами и списками	99
9.1. Массив и список. Цикл <i>foreach</i>	99
9.2. Функции <i>list()</i> и <i>array()</i>	102
9.3. Удаление массива	104
9.4. Слияние массивов.....	104
9.5. Функция <i>print_r()</i>	106
9.6. Разыменовывание массива.....	107
Глава 10. Функции сортировки массивов.....	109
10.1. Сортировка массивов	109
10.2. Функция <i>sort()</i> — сортировка списка.....	109
10.3. Функция <i>asort()</i> — сортировка массива по значениям.....	110
10.4. Функция <i>ksort()</i> — сортировка по ключам	111
10.5. Функции <i>array_reverse()</i> и <i>shuffle()</i>	112
10.6. Собственная функция сортировки.....	112
10.7. Натуральная сортировка	113
Глава 11. Особые операции над массивами.....	115
11.1. Добавление и удаление элементов массива.....	115
11.2. Упаковка переменных в массив и их извлечение	116
11.3. Получение части массива.....	118
11.4. Функции автоматического заполнения массива	118
11.5. Сравнение массивов	119
11.6. Полезные операции над массивом	120
11.6.1. Вычисление суммы и произведения всех элементов массива	120
11.6.2. Проверка существования элемента в массиве.....	120
11.6.3. Получение случайного элемента из массива.....	121
11.6.4. Удаление дубликатов из массива	121
11.6.5. Получение значений и ключей массива.....	122
11.6.6. Замена местами значений и ключей.....	122

11.6.7. Подсчет значений в массиве.....	122
11.6.8. Замена в массиве.....	123
11.6.9. Поиск в массиве.....	123
11.6.10. Прогулка по массиву.....	124
РАЗДЕЛ 4. ФУНКЦИИ В PHP	127
Глава 12. Полезные стандартные функции	129
12.1. Генератор случайных чисел.....	129
12.2. Дата и время.....	130
12.2.1. Кратко о timestamp.....	130
12.2.2. Функции <i>strtotime()</i> и <i>checkdate()</i>	130
12.2.3. Вывод даты.....	131
12.3. Математические функции.....	133
Глава 13. Функции для работы со строками	135
13.1. Основные строковые функции	135
13.2. Специальные функции замены	138
13.3. Функции преобразования строки	140
13.4. Функции преобразования кодировок.....	142
13.5. Функции для работы с отдельными символами строки. Разыменовывание строки.....	143
13.6. Функция <i>md5()</i> и другие функции шифрования/хэширования. API хэширования паролей	144
13.7. Функции <i>explode()</i> и <i>implode()</i> : работа с подстроками	146
13.8. Статистические функции	146
13.9. Функции вывода текста.....	147
13.10. Установка локали.....	149
13.11. Форматирование чисел и денежных величин.....	150
13.12. Преобразование систем счисления	151
13.13. Строки в PHP 7	151
Глава 14. Работа с файлами и каталогами	152
14.1. Права доступа в UNIX.....	152
14.2. Чтение файла.....	154
14.2.1. Функции <i>fopen()</i> и <i>fread()</i>	155
14.2.2. Функция <i>file()</i> : построчное чтение файла	157
14.2.3. Чтение всего файла: функция <i>file_get_contents()</i>	158
14.3. Запись файла	158
14.4. Создание временных файлов.....	159
14.5. Работа с CSV-файлами.....	159
14.6. Специальные функции для работы с файлами.....	162
14.6.1. Функции для работы с именами файлов.....	163
14.6.2. Работа с правами доступа	163
14.6.3. Копирование, переименование и удаление файлов	164
14.6.4. Время доступа к файлу.....	165
14.6.5. Другие полезные функции	166
14.7. Совместный доступ к файлу	166
14.8. Функции для работы с каталогами.....	167

Глава 15. Вывод графических изображений средствами PHP	169
15.1. Библиотека GD.....	169
15.1.1. Получение информации об изображении	169
15.1.2. Конвертирование графических форматов	172
15.1.3. Вывод текста поверх картинки. Задание цвета	174
15.1.4. Прозрачность	177
15.1.5. Рисование графических примитивов	178
15.1.6. Поворот изображения.....	180
15.2. Изменение размера изображения	180
15.3. Создание водяных знаков	182
15.4. Поддержка графического формата WebP в PHP 5.5.....	184
Глава 16. Работа с сетевыми сокетами в PHP. Сетевые функции	186
16.1. Еще раз о том, что такое сокет	186
16.2. Использование сокетов	186
16.3. Пример использования сокетов.....	187
16.4. Блокирующий и неблокирующий режимы сокета.....	190
16.5. DNS-функции	191
Глава 17. Собственные функции.....	192
17.1. Зачем нужны собственные функции?	192
17.2. Особенности функций в PHP.....	192
17.3. Объявление функции.....	193
17.4. Области видимости функции.....	194
17.5. Вложенность функций.....	195
17.6. Переменное число аргументов	196
17.7. Передача массивов в качестве параметров.....	198
17.8. Генераторы.....	200
17.9. Особенности использования функции <i>empty()</i>	202
РАЗДЕЛ 5. СЕРВЕРЫ БАЗ ДАННЫХ MYSQL 6 И SQLITE	205
Глава 18. Основы SQL	207
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

Глава 19. Функции для работы с MySQL	224
19.1. Способы работы с базой данных	224
19.2. Расширение mysql (для PHP 5)	225
19.2.1. Подключение к серверу MySQL	225
19.2.2. Несколько MySQL-соединений	227
19.2.3. Передача запросов серверу	227
19.2.4. Работа с базой данных. Создание базы данных	231
19.2.5. Функция <i>mysql_real_escape_string(\$content)</i>	231
19.2.6. Получение информации о результате запроса	232
19.3. Расширение PDO	233
19.3.1. Соединение с базой данных	233
19.3.2. Выполнение запросов и чтение результата	234
19.3.3. Получение данных	236
19.3.4. Особенности использования операторов <i>LIKE</i> , <i>LIMIT</i> и <i>IN</i>	238
19.3.5. Имена таблиц и полей при работе с PDO	238
19.3.6. Запросы вставки и обновление	239
Глава 20. Альтернативная база данных SQLite	240
20.1. MySQL vs SQLite: что лучше?	240
20.2. Открытие базы данных	242
20.3. Передача запросов	242
20.4. Работа с результатом запроса	243
20.5. Список PHP-функций для работы с SQLite	244
Глава 21. Полезные приемы при работе с MySQL 6	246
21.1. Выбор кодировки	246
21.2. Сортировка: вывод новинок. Вывод случайных записей	247
21.3. Постраничный вывод таблицы	248
РАЗДЕЛ 6. ИНСТРУМЕНТЫ ДЛЯ СОЗДАНИЯ СЛОЖНЫХ ПРОЕКТОВ ...	253
Глава 22. Разработка собственного шаблонизатора	255
22.1. Организация файлов и каталогов проекта	255
22.2. Выносим параметры в отдельный файл	258
22.3. Подключение дополнительных файлов	259
22.3.1. Инструкции <i>include</i> и <i>require</i>	259
22.3.2. Альтернативный способ подключения сценариев	259
22.3.3. Инструкции <i>include_once</i> и <i>require_once</i>	261
22.4. Шаблоны	261
Глава 23. Шаблонизатор Smarty	266
23.1. Что такое Smarty?	266
23.2. Установка Smarty	266
23.3. Создание файла <i>setup.php</i>	269
23.4. Разработка шаблонов Smarty	270
23.4.1. Комментарии в шаблонах	270
23.4.2. Переменные в Smarty	271
23.4.3. Файлы конфигурации шаблонов	272
23.4.4. Служебная переменная <i>{Smarty}</i>	273

23.4.5. Модификаторы переменных.....	274
23.4.6. Стандартные (встроенные) функции Smarty	277
Функции <code>{include}</code> и <code>{insert}</code>	277
Функция <code>{foreach}</code>	277
Функции <code>{if}</code> , <code>{elseif}</code> , <code>{else}</code>	279
Функция <code>{capture}</code>	280
Функция <code>{php}</code>	280
Функция <code>{strip}</code>	280
23.4.7. Пользовательские функции Smarty	281
Функция <code>{assign}</code>	281
Функция <code>{cycle}</code>	281
Функция <code>{fetch}</code>	282
Функции <code>{html_checkboxes}</code> и <code>{html_radios}</code>	282
Функция <code>{html_image}</code>	283
Функция <code>{html_select_date}</code>	283
Функция <code>{html_select_time}</code>	284
Функция <code>{html_table}</code>	285
23.5. Smarty для программиста	286
23.5.1. Специальные переменные	286
23.5.2. Полезные методы класса <i>Smarty</i>	288
Глава 24. Объектно-ориентированное программирование	289
24.1. Основы ООП	289
24.2. Классы и объекты	290
24.3. Конструкторы и деструкторы класса	292
24.4. Наследование классов. Полиморфизм	293
24.5. Новые возможности PHP 5.4/5.5 и PHP 7	294
24.5.1. Область видимости членов класса	294
24.5.2. Абстрактные классы и методы	295
24.5.3. Служебное слово <i>final</i>	296
24.5.4. Клонирование объектов	296
24.5.5. Обработка исключительных ситуаций	297
Служебное слово <i>finally</i>	298
Обработка фатальных ошибок	300
24.5.6. Константы-члены класса	301
24.5.7. Статические члены класса	302
24.5.8. Оператор <i>instanceof</i>	302
24.5.9. Итераторы	303
24.5.10. Пространства имен	303
24.5.11. Типажи (traits)	304
24.5.12. Вызов метода или свойства класса выражением	305
24.6. Особенности обработки переменных в PHP 7	305
Глава 25. Механизм сессий	306
25.1. Для чего нужны сессии?	306
25.2. Как работает механизм сессий?	307
25.3. Обход Cookies	309
25.4. Сценарий аутентификации	309

Глава 26. Введение в PEAR.....	314
26.1. Серьезные проекты и PEAR.....	314
26.2. Пример использования класса <i>DB</i>	315
Глава 27. Контроль версий.....	319
27.1. Выбор системы контроля версий	319
27.2. Практическое использование TortoiseHG (Mercurial)	320
27.3. Просмотр внесенных изменений.....	322
Глава 28. Тестирование PHP-сценариев.....	324
28.1. Программа работает, но не так, как нам нужно	324
28.2. "Самодельные" точки останова.....	325
28.3. Система автоматического тестирования.....	326
28.4. Директива <i>error_reporting</i>	330
<u>ЧАСТЬ II. ПРАКТИКА</u>.....	331
РАЗДЕЛ 7. РАЗРАБОТКА ОСНОВНЫХ ЭЛЕМЕНТОВ САЙТА	333
Глава 29. Загрузка файлов на сервер	335
29.1. Что нужно знать о загрузке файлов на сервер?.....	335
29.2. Реализация загрузки файла	338
29.3. Загрузка нескольких файлов.....	340
29.4. Индикатор загрузки файла на PHP 5.4/5.5.....	342
29.4.1. Некоторые теоретические предпосылки.....	342
29.4.2. Пример практической реализации	345
29.5. Проблемы при загрузке файлов.....	352
Глава 30. Использование FTP-функций	353
30.1. Функции для работы с FTP	353
30.2. Примеры использования FTP-функций	356
Глава 31. Отправка и прием почты.....	359
31.1. Отправка почты средствами PHP — функция <i>mail()</i>	359
31.2. Отправка писем с вложениями — класс <i>HtmlMimeMail</i>	360
31.2.1. Отправка сообщения	360
31.2.2. Проблемы при отправке сообщения	364
31.2.3. MIME-типы	365
31.3. Класс <i>PHPMailer</i> . Разработка сценария автоматической рассылки прайс-листа	366
31.4. Получение писем по протоколу POP3	370
Глава 32. Работа с RSS: получаем новости на сайт.....	374
32.1. Краткие сведения о RSS	374
32.2. Формат RSS-файла	374
32.3. Написание сценария импорта новостей.....	377
32.4. Подключение файла <i>import.php</i> к сайту.....	379
32.5. Создание граббера новостей.....	380

Глава 33. Поиск с использованием регулярных выражений	384
33.1. Нужно что-то найти.....	384
33.2. Язык регулярных выражений RegExp.....	385
33.2.1. Регулярные выражения	385
33.2.2. Управляющие символы	386
33.2.3. Управляющие конструкции	387
Квантификаторы.....	387
Альтернативный оператор 	387
Скобки	388
33.2.4. Псевдосимволы.....	388
33.2.5. Практическое использование функций языка RegExp	388
33.3. Фильтры, или конец эры регулярных выражений	392
Глава 34. Работаем с MP3.....	394
34.1. Формат MP3	394
34.2. Библиотека PEAR	395
34.3. Вывод ID3-тегов	396
34.4. Редактирование ID3-тегов	398
34.5. Удаление тега.....	398
Глава 35. Технология AJAX.....	400
35.1. Что такое AJAX?.....	400
35.2. Ваше первое AJAX-приложение	401
РАЗДЕЛ 8. РАЗРАБОТКА ТИПИЧНОГО САЙТА.....	407
Глава 36. Создание простейшего движка сайта	409
36.1. Планирование движка	409
36.1.1. Зачем нужно разрабатывать собственный движок?	409
36.1.2. Необходимые нам функции движка.....	410
36.1.3. Принципиальная схема движка	412
36.2. Основные функции движка.....	413
36.2.1. Разработка TPL-шаблона	413
36.2.2. Файл настроек.....	414
36.2.3. Основной файл CMS — index.php.....	415
36.2.4. Проектирование базы данных	417
Таблица <i>static</i>	417
Таблица <i>cats</i>	418
Таблица <i>pages</i>	419
36.2.5. Иерархическая структура сайта.....	420
Алгоритм работы меню	420
Сценарий menu.php	422
Вывод содержимого раздела и страницы.....	427
36.3. Дополнительные функции движка	431
36.3.1. Вывод информации из таблицы <i>static</i>	431
36.3.2. Функция вывода содержимого HTML-файла.....	432
36.3.3. Версия для печати.....	433
36.4. Где взять листинги этой главы?	434

Глава 37. Создание фотогалереи	435
37.1. Постановка задачи	435
37.2. Загрузка изображений на сервер	435
37.3. Вывод галереи.....	439
Глава 38. Мобильная версия сайта	442
38.1. Важность мобильной версии	442
38.2. Рекомендации для мобильной версии.....	442
38.3. Класс <i>MobileDetect</i>	443
38.4. Способы реализации мобильной версии	444
38.4.1. Реализация адаптивного дизайна	445
38.4.2. Реализация динамического показа	446
38.4.3. Создаем отдельный сайт	446
38.5. О производительности мобильной версии	447
Глава 39. Импорт новостей с чужого сайта в базу данных WordPress.....	448
39.1. Постановка задачи	448
39.2. Вспомогательные функции	449
39.3. Вкратце о сURL	451
39.4. Основной сценарий	451
Глава 40. Создание счетчика сайта	456
40.1. Постановка задачи	456
40.2. Файл конфигурации.....	456
40.3. Разработка таблиц <i>counter</i> и <i>ipaddr</i>	457
40.4. Сценарий <i>counter.php</i>	458
40.5. Сценарий <i>reset_counter.php</i>	462
Глава 41. Статистика сайта	464
41.1. Методы сбора статистики	464
41.2. Программы-анализаторы журналов веб-сервера	465
41.3. Системы статистики	467
Глава 42. Голосования и комментарии для сайта	470
42.1. Разработка собственной системы голосования.....	470
42.1.1. Общая структура системы	470
42.1.2. Разработка сценария <i>poll_form.php</i>	471
42.1.3. Сценарий <i>poll_process.php</i>	473
42.1.4. Сценарий <i>poll_results.php</i>	474
42.2. Комментарии DISQUS	477
42.2.1. Описание платформы	477
42.2.2. Установка комментариев на свой сайт	478
РАЗДЕЛ 9. СЛОЖНЫЙ ПРОЕКТ	485
Глава 43. Постановка задачи	487
43.1. Концепция проекта	487
43.2. Алгоритм работы	488
43.3. Структура базы данных.....	489

Глава 44. Разработка основной части сайта	493
44.1. Структура проекта	493
44.2. Основной сценарий index.php	494
44.2.1. Подключение механизмов WordPress	494
44.2.2. Аутентификация пользователя	495
44.2.3. Форма добавления цели	496
44.2.4. Добавление цели	497
44.2.5. Просмотр всех целей	501
44.3. Сценарий v.php — просмотр цели	505
44.4. Сценарии отправки уведомлений. Сценарии yes.php и no.php	506
Глава 45. Панель управления WordPress.....	514
45.1. О чем эта глава?	514
45.2. Вход в панель управления	514
45.3. Просмотр целей	514
45.4. Не забывайте о UTF-8	516
РАЗДЕЛ 10. ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ САЙТА.....	517
Глава 46. SSL-соединения.....	519
46.1. Защищаем передаваемые данные	519
46.2. Настройка SSL в DirectAdmin	520
46.3. SSL-переменные	522
Глава 47. Защита PHP с помощью конфигурационного файла.....	524
47.1. Конфигурационный файл php.ini	524
47.2. Отключение потенциально опасных функций	526
47.3. Рекомендованные значения некоторых конфигурационных директив	526
Глава 48. Защита сайта от атак.....	527
48.1. Сайт в опасности	527
48.2. Два самых распространенных метода взлома	528
48.3. Межсайтовый скриптинг	528
48.4. SQL-инъекции	530
48.5. Флуд	534
48.6. Защита форума PHPBB2 от спамеров	535
РАЗДЕЛ 11. ПОЛЕЗНЫЕ СВЕДЕНИЯ.....	537
Глава 49. Визуальный редактор SPAW2 для вашего сайта	539
49.1. Знакомство с визуальным редактором и его базовая настройка	539
49.2. Адаптация редактора к вашему сайту	543
49.2.1. Загрузка в редактор текста статьи	543
49.2.2. Вывод редактора в TPL-шаблоне	543
49.2.3. Запрет изменения размера редактора	544
49.2.4. Изменение размеров редактора "на лету" — в процессе выполнения сценария	545
49.2.5. Панель инструментов редактора	545
49.2.6. Ошибка Error 105 (net::ERR_NAME_NOT_RESOLVED)	548

Глава 50. WAP-сервер средствами PHP	549
50.1. Есть ли необходимость в WAP?	549
50.2. Настройка веб-сервера Apache	550
50.3. PHP-сценарий	551
Глава 51. Пишем коммерческий сценарий	552
51.1. Zend Guard: за и против	552
51.2. Установка Zend Guard	554
51.3. Быстрая защита сценария	554
51.4. Ошибка Unable to locate XML file	558
51.5. Лицензии	558
51.6. Zend Guard — панацея?	562
Глава 52. Расширение cURL: практические примеры	563
52.1. Этот загадочный cURL	563
52.2. Авторизация на сайте и загрузка файла после нее	565
52.3. Замена функции <i>file_get_contents()</i> с помощью cURL	568
52.4. Загрузка файла через FTP	568
52.5. Проверка доступности сайта	569
Глава 53. Фреймворк или чистый PHP-код?	571
53.1. Что такое фреймворк?	571
53.2. Разработка простой гостевой книги на PHP	572
53.2.1. Исходная точка	572
53.2.2. Отделяем код от представления	573
53.2.3. Изолируем логику приложения	574
53.2.4. Изолируем разметку	575
53.2.5. Добавление еще одной страницы	577
53.2.6. Создаем фронт-контроллер	578
53.3. Простая гостевая книга на Symfony	580
53.3.1. Модифицируем имеющийся код	580
53.3.2. Symfony-приложение с нуля	582
ПРИЛОЖЕНИЯ	587
Приложение 1. Программа phpMyAdmin	589
П1.1. Действия над таблицами	590
П1.2. Создание новой таблицы. Изменение структуры таблицы	592
П1.3. Вставка записей	593
П1.4. Обзор таблицы	593
П1.5. Выполнение произвольного SQL-кода	593
П1.6. Резервное копирование БД	595
Приложение 2. Flash и PHP	596
Приложение 3. Оптимизация PHP-кода	600
П3.1. Правильное использование <i>echo</i>	600
П3.2. Длина переменных	601
П3.3. Правильное обращение к элементам массива	601

ПЗ.4. Циклы.....

602

ПЗ.5. Чтение файлов.....

603

ПЗ.6. Вывод текста: *echo* против *printf*.....

605

ПЗ.7. "Лишние" вызовы функций.....

607

ПЗ.8. Контрольные замеры

608

ПЗ.8.1. Чтение файла.....

608

ПЗ.8.2. Вывод содержимого файла

609

ПЗ.9. Выводы

611

Приложение 4. Описание электронного архива.....

612

Предметный указатель

615

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 6-м издании?

Прежде всего, с учетом нововведений, появившихся в PHP 7, актуализирован весь материал предыдущих изданий книги.

По традиции, ее обновление началось с *первой главы*. При этом в рассказе о PHP версии 7.0 банальной заменой изображений интерфейса уже не обойтись. Основная проблема новой версии PHP в том, что она более не поддерживает старые версии HTTP-сервера Apache, — нужна только его версия 2.4, а у этой версии нет даже инсталлятора, поэтому устанавливать ее придется вручную. Конечно, можно сэкономить кучу времени, установив современную версию XAMPP¹. Но это решение PHP оставляет на волю пользователей.

В числе обновленных и *глава 19* — в ней будет показано, как работать с базой данных по-новому, — когда отсутствует расширение PHP для работы с сервером MySQL и нужно использовать для подключения к базе данных расширение PDO (PHP Data Objects).

Полностью обновлены в этом издании книги и четыре следующие главы:

- ❑ в *главе 8* вместо устаревшей директивы `register_globals` рассматриваются некоторые вопросы оптимизации сайта, написанного на PHP. В ней мы разберемся

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

с рядом проблем, с которым приходится сталкиваться после разработки сайта, — при его оптимизации. Ведь проще устранить все эти проблемы при разработке сайта, чем тратить время на их исправление, когда он полностью завершен;

- ❑ современные поисковые машины много внимания уделяют мобильному поиску. Именно поэтому в *главе 38* рассмотрен класс `MobileDetect`, с помощью которого вы сможете создать мобильную версию сайта. Да, в *главе 38* более не рассматривается такой пережиток прошлого, как гостевая книга. Ее время вышло;
- ❑ в *главе 52* содержится описание `cURL` — расширения PHP, с помощью которого вы сможете регистрироваться на сайтах, как из обычного браузера, загружать файлы с удаленных сайтов и делать еще многое другое.
- ❑ последняя глава книги, *глава 53*, показывает преимущество использования фреймворков PHP над написанием кода с нуля. Будущее — за фреймворками, и вам уже сейчас пора начинать неспешно переносить свои проекты на подходящие для них (и для вас) фреймворки.

Электронный архив с информацией, расширяющей и дополняющей материал "бумажной" книги, можно скачать с FTP-сервера издательства по ссылке: **<ftp://ftp.bhv.ru/9785977538350.zip>**, со страницы книги на сайте **www.bhv.ru**, а также с сайта автора: **http://dkws.org.ua/mybooks/php_adds.zip**. Подробная информация об архиве находится в *приложении 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 язык PHP и ведет свое название).

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 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 посвятили переработке ядра. И теперь, 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 нее 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 приблизилась к модели Java. Далеко не все хостинг-провайдеры успели к настоящему моменту перейти на версию PHP 5.4, выпущенную в 2012 году, поэтому как минимум еще год-два на хостингах будет по-прежнему встречаться версия 5.3 (2009 г.).

Начиная с 2005 года, велась разработка PHP 6 (что случилось с этой версией, вы узнаете в следующем разделе), но вместо нее в июне 2013 года вышла версия 5.5. В августе 2014 года была выпущена версия 5.6, но она не получила особого распространения.

Версия PHP 7 вышла 3 декабря 2015 года, а версия 7.1 должна была появиться в ноябре 2016 года, но на момент написания этих строк ее выход еще официально не анонсировали.

PHP 7, или что случилось с PHP 6?

В июне 2015 года (если быть предельно точным, то 11 июня) миру была показана следующая версия PHP — PHP 7. Но, как все мы помним, официального релиза PHP 6 так и не состоялось. Почему же разработчики не продолжили существующую нумерацию и не выпустили PHP 6, а "перешагнули" через шестую версию и явили миру сразу PHP 7.

Причин тому две. Во-первых, как все мы помним, релиз PHP 6 должен был состояться еще в 2005 году. Но из-за проблем с реализацией Unicode¹ разработчики из года в год откладывали выпуск шестой версии. Вместо нее все новинки включались в ветку PHP 5.x — появлялись версии 5.3, 5.4, 5.5 и 5.6 (самая последняя из ветки 5.x). Очевидно, разработчики не хотели, чтобы новая версия языка ассоциировалась с неудачами.

Во-вторых, разработчики хотели показать революционность новой версии, подчеркивая таким образом, что в ней очень много важных изменений. Вспомним, Microsoft поступила так же с Windows — все ждали Windows 9, а вышла сразу "десятка".

¹ Получить список всех функций, поддерживающих Unicode, можно по адресу http://www.php.net/~scoates/unicode/render_func_data.php.

Основные нововведения в PHP версии 7

Улучшена производительность

PHP 7, прежде всего, отличается от предшествующих версий улучшенной производительностью — сценарии, выполняемые PHP 7, работают в 2 (!) раза быстрее, чем под управлением PHP 5.6. Примерно такой же прирост производительности в свое время обеспечила версия 5.3, работающая существенно быстрее, чем 5.2.

Удалены старые расширения и SAPI

PHP 7, наконец, избавился от всего старого и тянущего его ко дну. Это как раз и есть одна из причин улучшения его производительности. Были удалены все старые расширения и SAPI, а именно: sapi/aolserver, sapi/apache, sapi/apache_hooks, sapi/apache2filter, sapi/caudium, sapi/continuity, sapi/isapi, sapi/milter, sapi/nsapi, sapi/phptpd, sapi/pi3web, sapi/roxen, sapi/thttpd, sapi/tux, sapi/webjames, ext/mssql, ext/sybase_ct.

Об удалении из PHP 7 поддержки расширения ext/mysql для работы с сервером MySQL, пожалуй, нужно рассказать отдельно, а не просто включить его в список удаленных. Дело в том, что, начиная с версии 5.5, разработчики начали считать это расширение устаревшим. Чем оно им не угодило, непонятно, но из версии 7.0 его исключили совсем. И если в версии 5.5 PHP просто выдавал предупреждения об устаревшем расширении, то в версии 7.0 его нет вообще, — предлагается использовать расширение mysqli или PDO (см. главу 19).

Подробно о том, что было при подготовке версии 7.0 удалено, вы можете прочитать по адресу: https://wiki.php.net/rfc/removal_of_dead_sapis_and_exts.

Из файла php.ini была удалена поддержка следующих директив:

- ☐ asp_tags;
- ☐ always_populate_raw_post_data и связанная с ней переменная \$HTTP_RAW_POST_DATA — вместо нее теперь нужно использовать дескриптор входного потока php://input.

Вот что еще считается устаревшим:

- ☐ синтаксис конструкторов в стиле PHP 4 (когда имя метода конструктора совпадает с именем класса);
- ☐ статичные вызовы (::) нестатичных методов.

Удалена также поддержка модификатора /e в PCRE. Аналогичная возможность может быть реализована функцией preg_replace_callback().

Изменения в синтаксисе языка

Изменений по части обработки переменных, массивов и функций очень много, настолько много, что написать о них в предисловии было бы неправильно, и каждое из них будет описано в соответствующей главе книги. Некоторые из нововведений

не являются обратно совместимыми, поэтому приготовьтесь к тому, что код, написанный для PHP 7, не будет работать в более ранних версиях PHP. Однако все сценарии, написанные для 5-й версии, будут работать в PHP 7. Почти все...

Тем не менее, некоторые изменения имеет смысл анонсировать и здесь.

Новые возможности

- Добавлена поддержка сокращенной формы `use`:

```
use Symfony\Component\Console\{
    Helper\Table,
    Question\ChoiceQuestion as Choice,
    Question\ConfirmationQuestion,
};
```

- Добавлен оператор `??` (Null coalescing operator), позволяющий проверить переменную на существование и вернуть ее значение либо значение по умолчанию. Например, следующую конструкцию:

```
$action = isset($_POST['action']) ? $_POST['action'] : 'index';
```

теперь можно записать так:

```
$action = $_POST['action'] ?? 'index';
```

- Добавлена поддержка строк длиной больше 2^{31} байт в 64-битных сборках.
- Добавлен метод `Closure::call(object $to[, mixed $parameters])`, позволяющий вызвать анонимную функцию с произвольным `$this`.
- Добавлен синтаксис `\u{xxxxxxx}` для строк, позволяющий указывать произвольные Unicode-символы в строках.
- В качестве значения констант, объявляемых через `define()`, теперь можно указывать массивы.
- Добавлен новый оператор сравнения `<=>`, так же известный как *spaceship operator*. Конструкция `$a <=> $b` возвращает -1, 0 или +1, если `$a` соответственно меньше, равно или больше `$b`. Этот оператор удобно использовать в `callback` для `usort()`.
- Резервированные ключевые слова теперь можно использовать в качестве имен методов: `$object::new('foo', 'bar')->forEach(function($index, $item) {});`.
- Добавлена константа `PHP_INT_MIN`.
- Строки, содержащие шестнадцатеричные числа теперь всегда обрабатываются как строки и не обрабатываются как числа: `is_numeric("0xFF")` теперь `false`.
- Целые числа в 64-битных сборках для Windows теперь представляются в виде 64-битных, а не как раньше, в виде 32-битных, что делало бессмысленным занятием использование 64-битных сборок на Windows, когда нужны были манипуляции с большими числами.
- Добавлена поддержка `type-hint`'ов для скалярных типов. Ранее контроль типов был возможен только для классов, интерфейсов, массивов и типа `callable`.

- ❑ Итерация по массиву при помощи `foreach()` больше не сдвигает внутренний указатель массива, который можно получать и изменять при помощи функций `current()`, `next()`, `reset()` и им подобных. Так же `foreach` по значению теперь всегда работает с копией массива.
- ❑ Оператор левого побитового сдвига (`<<`) на количество битов, превышающее количество битов в `integer`, теперь всегда возвращает 0. До этого результат зависел от архитектуры процессора. Аналогичный правый сдвиг всегда дает 0 или -1 в зависимости от знака исходного числа.

Измененные функции

Были модифицированы следующие функции:

- ❑ `parse_ini_file()`;
- ❑ `parse_ini_string()`;
- ❑ `unserialize()`;
- ❑ `proc_open()`.

Новые функции

Появились новые функции:

- ❑ `preg_replace_callback_array()`;
- ❑ `intdiv()`;
- ❑ `error_clear_last()`;
- ❑ `gmp_random_seed()`;
- ❑ `deflate_init()`;
- ❑ `deflate_add()`;
- ❑ `inflate_init()`;
- ❑ `inflate_add()`.

Конечно, это далеко не все измененные или новые функции. С полным списком изменений вы можете ознакомиться по адресу: <https://github.com/php/php-src/blob/php-7.0.0alpha1/UPGRADING>.

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.	Программа на PHP
Глава 3.	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 проекта на другой сервер сценарий пере-

стал запускаться, 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 установлены 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 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 так, 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 варианты версии 6.0, которая так и не вышла, но без поддержки Unicode.

Что же касается версии 7.0, то хостеры пока не спешат ее устанавливать. Дело связано даже не с ленью сотрудников хостера, а с отсутствием поддержки расширения PHP для работы с сервером MySQL и еще некоторых (см. *предисловие*). Да, теперь в версии 7.0 расширение `mysql` не поддерживается, и огромное число сценариев, которые сейчас используют функции `mysql*`, должно быть переписано с использованием расширения PDO (см. *главу 19*). Увы, но это так.

Самыми распространенными версиями PHP, установленными на хостингах сейчас (по состоянию на ноябрь 2016 года), являются версии 5.2.x–5.4.x. Некоторые хостеры даже предоставляют возможность выбора версии PHP в панели управления хостингом (рис. 1.1).

ПРИМЕЧАНИЕ

Не хочется делать никому рекламу, именно поэтому я не раскрываю, панель управления каким хостингом здесь показана. Но если вы ее узнали, то мое мнение об этом

хостинге скорее отрицательное. Панель управления хорошая, а вот производительность сервера оставляет желать лучшего.

Как можно видеть, здесь предоставляется выбор как старых версий: 5.2–5.4, так и новых — от 5.5, однако с предупреждением, что некоторые сценарии могут быть несовместимы с новыми версиями. При этом текущей версией определена самая старая версия — 5.2. Используется она вынужденно — из-за одного коммерческого сценария, который почему-то не хочет работать в версии 5.3, а разработчики не спешат вносить изменения в зашифрованный с помощью Zend код.

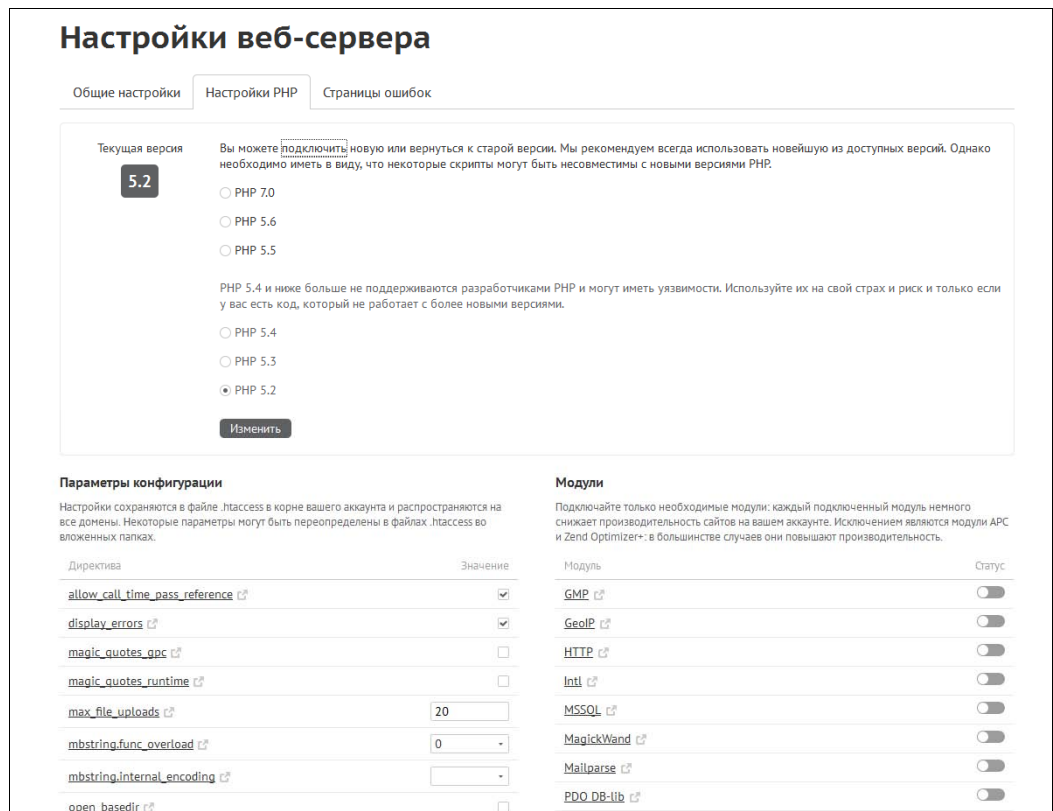


Рис. 1.1. Панель управления хостингом: выбор версии PHP

Однако нужно отметить, что версия 5.3 гораздо быстрее, чем 5.2, и поэтому, если у вас есть возможность перейти на более новую версию (5.3 или 5.4), — сделайте это. Сценарии станут работать быстрее, и уменьшится нагрузка на сервер. Разработчики PHP прилагают массу усилий для оптимизации своего продукта, и каждая новая версия работает, как правило, быстрее предыдущей.

Чтобы узнать, какая версия PHP используется на том или ином хостинге:

- ❑ если хостинг уже куплен, напишите сценарий, вызывающий функцию `phpinfo()`, и запустите его. Кроме версии PHP, вы увидите массу другой полезной информации, в том числе список доступных расширений;

- ❑ если вы только планируете покупать хостинг, сведения о версии PHP имеются на веб-сайте хостера. Можно также обратиться в его службу поддержки — там подскажут, какие версии PHP доступны на их серверах.

Если вам нужен хостинг именно с поддержкой PHP 7, воспользуйтесь поиском Google, — множество компаний уже начали предоставлять такой хостинг. Также уточните наличие версии 7 у вашего хостера — вполне возможно, что новая версия уже поселилась на одном из его серверов, и хостер сможет переместить ваш сайт на такой сервер.

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, больше всех мне понравились два редактора: PHP Expert Editor и Zend Studio (вы без проблем найдете в Интернете сайты их разработчиков). Оба редактора коммерческие, но они стоят своих денег. PHP Expert Editor (рис. 1.2) обеспечивает не только подсветку синтаксиса, но и умеет проверять синтаксис сценария. Правда, для проверки синтаксиса вам придется все же 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/>.

Чтобы указать PHP в настройках программы PHP Expert Editor, выполните команду **Запуск | Настройки**, в открывшемся окне перейдите на вкладку **Интерпретаторы скриптов** (рис. 1.3) и укажите путь к интерпретатору PHP. Для проверки синтаксиса нажмите клавишу <F7>.

Возможности Zend Studio (рис. 1.4) куда шире — это не просто редактор, а настоящая IDE (среда разработки). Чего стоит только автодополнение кода, когда программа "читает" ваши мысли и дописывает за вас PHP-код. Если вы работали с IDE — например, с Visual Studio 2015, то поймете, о чем я говорю. Поверьте, это очень удобно.

В качестве FTP-клиента могу порекомендовать бесплатную программу FileZilla (рис. 1.5), которая доступна как для Windows, так и для Linux.

ПРИМЕЧАНИЕ

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

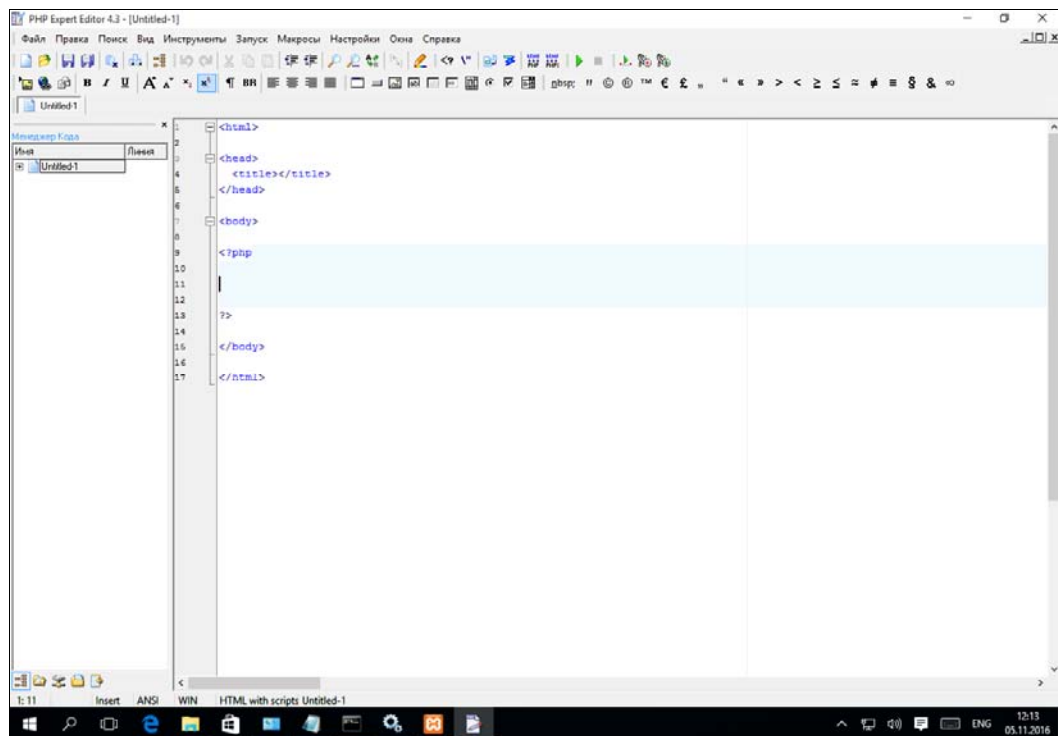


Рис. 1.2. Программа PHP Expert Editor

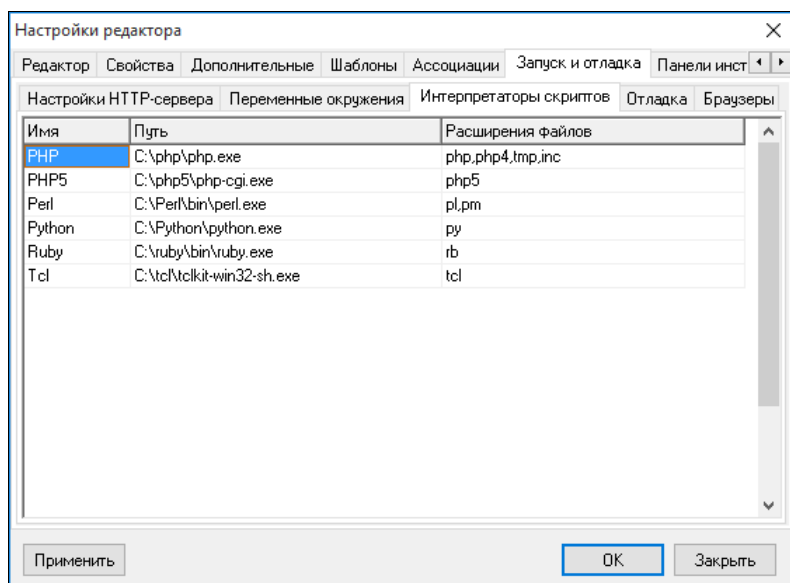


Рис. 1.3. Путь к интерпретатору PHP

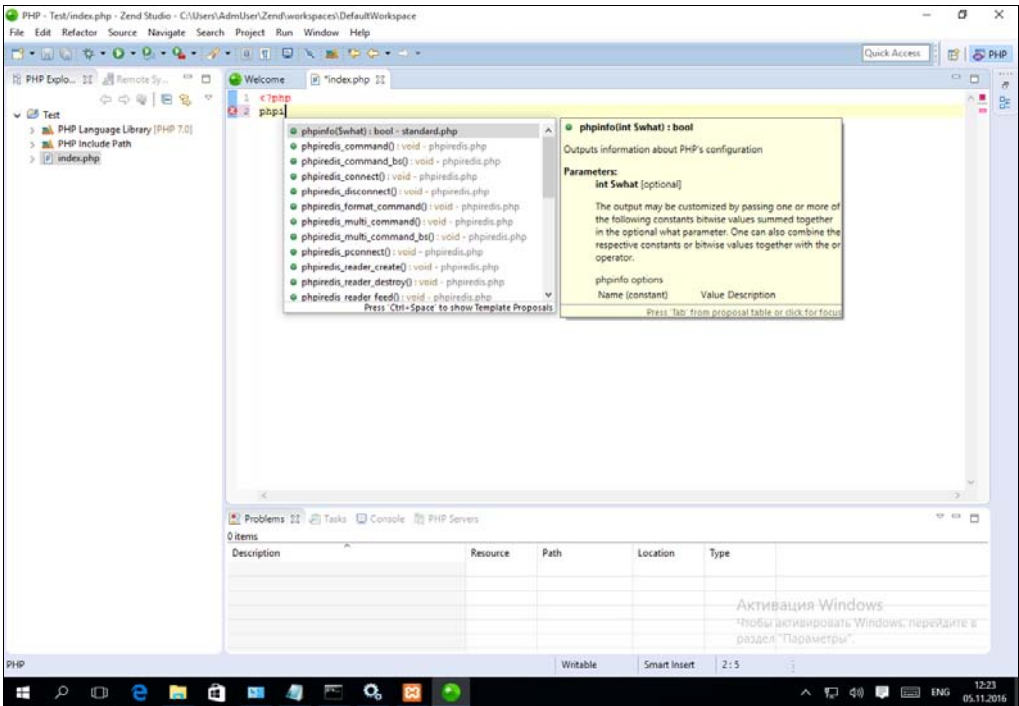


Рис. 1.4. Zend выводит не только имена функций, но и краткую справку по каждой функции

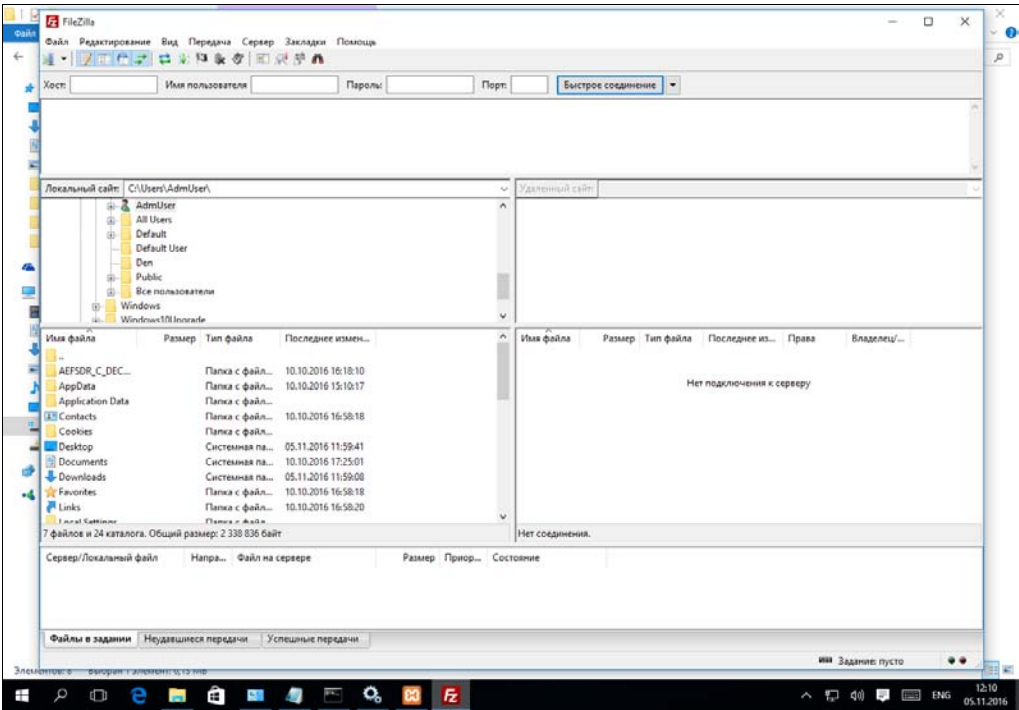


Рис. 1.5. Программа FileZilla

1.3. Установка связки Apache + PHP + MySQL в Windows

1.3.1. Для опытных пользователей: установка вручную

Сразу хочу вас предупредить: этот способ требует затрат времени, и его описание приводится только в образовательных целях, — на тот случай, если вы желаете знать, как все устроено. Гораздо проще установить пакет XAMPP, который содержит в себе уже все необходимое (Apache, MySQL, PHP 7), и наслаждаться сэкономленным временем, поскольку сразу же после установки XAMPP вы сможете приступить к созданию сценариев. Но если у вас есть время и желание, тогда предлагаю продолжить чтение этого раздела. Будет интересно!

Установка веб-сервера Apache

Начнем с установки веб-сервера Apache. В настоящий момент имеются две его актуальные версии: 2.2 и 2.4. Грубо говоря: старая и новая. Так вот, первые версии PHP 7.0 прекрасно поддерживали версию Apache 2.2. Однако в последних сборках оказалось, что файл `php7apache2_2.dll`, позволяющий связать Apache 2.2 и PHP 7.0, просто отсутствует. Найти в Интернете мне его тоже не удалось. Отсюда можно сделать вывод, что поддержку Apache 2.2 из PHP 7.0 исключили. И хотя в файле `install.txt`, входящем в состав PHP 7.0, сказано, что файл `php7apache2_2.dll` можно найти по адресу <http://snaps.php.net/>, однако по состоянию на 5 ноября 2016 года этого файла на указанном сайте не было, — уж поверьте.

Именно поэтому, если вы планируете использовать PHP 5, то можете установить Apache 2.2, оснащенный удобным инсталлятором. Если же вы хотите попробовать PHP 7, тогда вам придется устанавливать Apache 2.4 вручную, — инсталлятора для него не предусмотрено.

СОВЕТ

Учитывая, что версию Apache 2.4 поддерживают как последние версии PHP 5, так и PHP 7, я рекомендую использовать именно ее.

Обратите внимание, что для версии 2.4 нужна, как минимум, Windows 7 SP 1. Если ваша Windows 7 не обновлена до этого сервис-пака, то запустить из-под нее Apache 2.4 не получится. Также перед установкой версии 2.4 убедитесь, что у вас установлена последняя версия C++ Redistributable Visual Studio 2015, скачать которую можно по адресу: <https://www.microsoft.com/en-us/download/details.aspx?id=53840>.

ПРИМЕЧАНИЕ

Навигация на сайте Apache оставляет желать лучшего. Чтобы упростить вашу задачу, подсказываю ссылку, откуда можно скачать версию 2.4 для Windows: <http://www.apachelounge.com/download/> (именно с этого сайта предлагается скачать Windows-версию на официальном сайте Apache).

В каталоге *software* сопровождающего книгу электронного архива (см. *приложение 4*) вы найдете версии 2.2 и 2.4 для Windows. Linux-версии в архиве нет, поскольку Apache входит в состав любого дистрибутива Linux, и вам нужно будет только его установить.

Установить Apache 2.4 довольно просто. Прежде всего распакуйте архив `httpd-2.4.23-win32-VC14.zip` или `httpd-2.4.23-win64-VC14.zip` (в зависимости от разрядности Windows) в корневой каталог диска C:\ (или любого другого, но далее я буду предполагать, что вы распаковали архив на диск C:).

ОСОБЕННОСТИ УСТАНОВКИ В WINDOWS 7 И WINDOWS 10

Установку Apache, PHP и MySQL в Windows 7 и в Windows 10 желательно производить от имени администратора. Для этого щелкните на установочном файле правой кнопкой мыши и выберите команду **Запуск от имени администратора**.

Теперь запустите командную строку *от имени администратора*, перейдите в каталог `c:\Apache24\bin` и введите команду:

```
httpd.exe -k install
```

После этого вы увидите сообщение о том, что сервис Apache2.4 успешно установлен, а также окно, предлагающее настроить правила брандмауэра (рис. 1.6).

Управлять сервисом Apache можно или через приложение Apache Monitor (рис. 1.7, а), или через оснастку **Службы** (рис. 1.7, б) — как вам больше нравится.

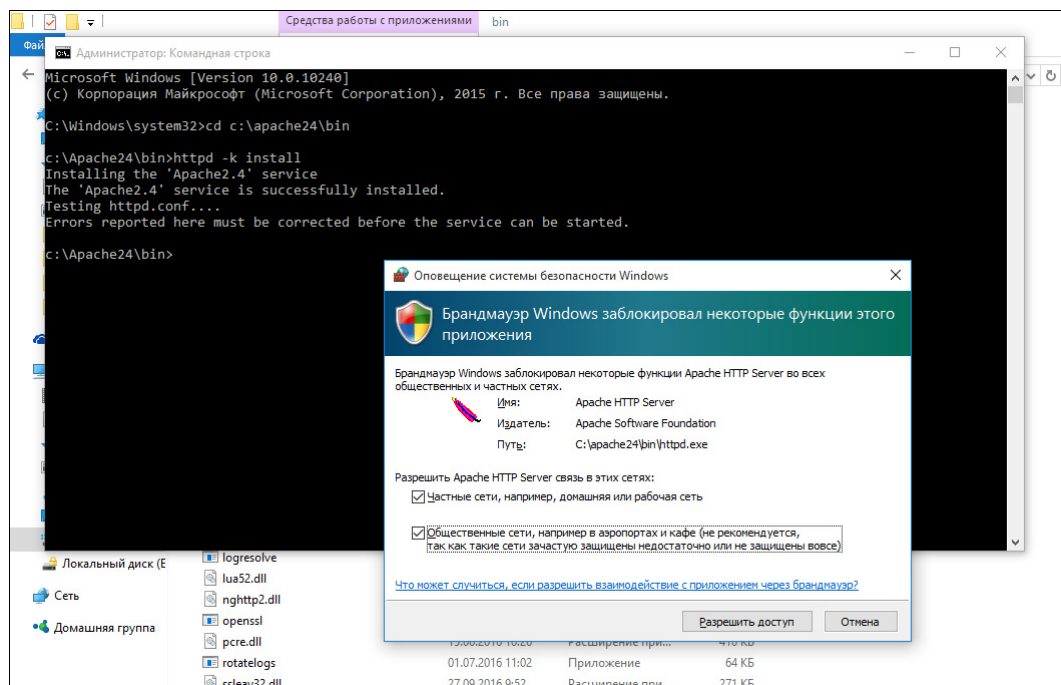
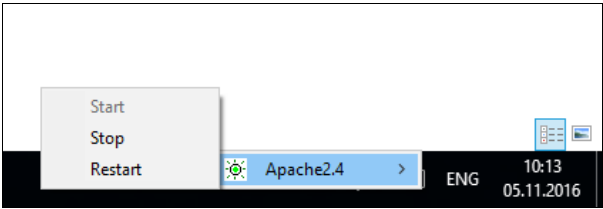
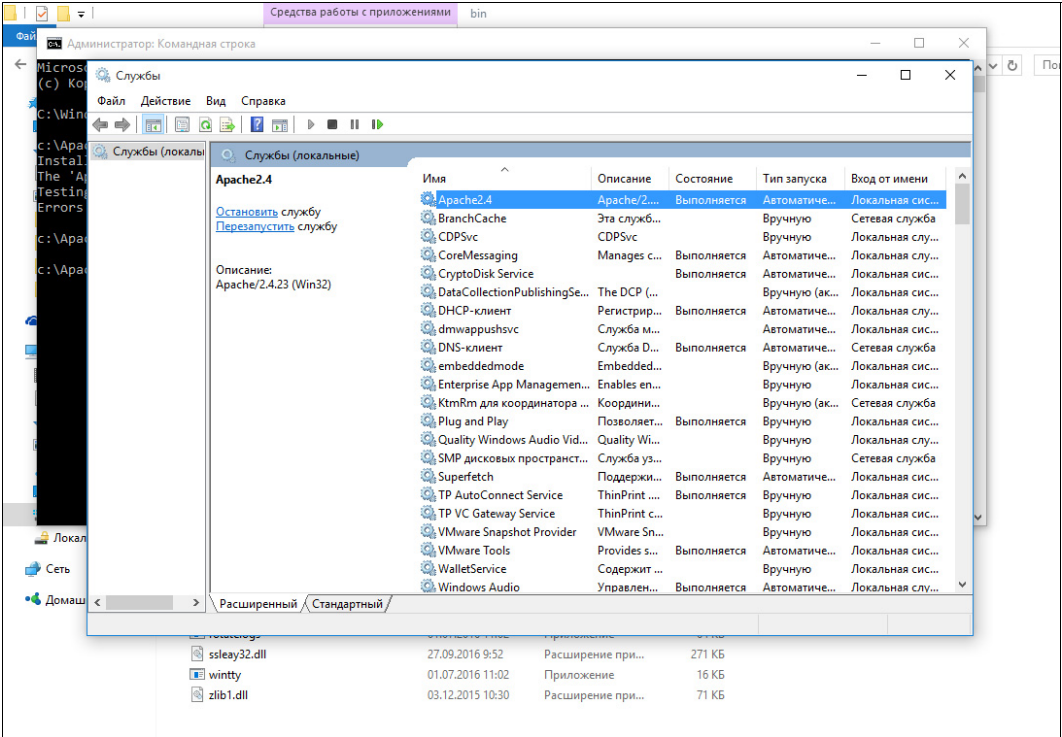


Рис. 1.6. Установка Apache 2.4



а



б

Рис. 1.7. Приложение Apache Monitor (а); оснастка Службы (б)

После установки сервиса Apache его нужно запустить, как уже было сказано, или через Apache Monitor, или с помощью оснастки **Службы**. Затем запустите браузер и введите следующий URL:

```
http://localhost
```

Если в ответ вы увидите сообщение **It works!** (рис. 1.8), значит, установка Apache прошла успешно.

Установка PHP

По умолчанию сервер Apache не поддерживает PHP, поэтому PHP придется установить отдельно. Ранее (до версии 5.4) в PHP был доступен инсталлятор, который существенно упрощал процесс установки, помогал внести все необходимые изме-

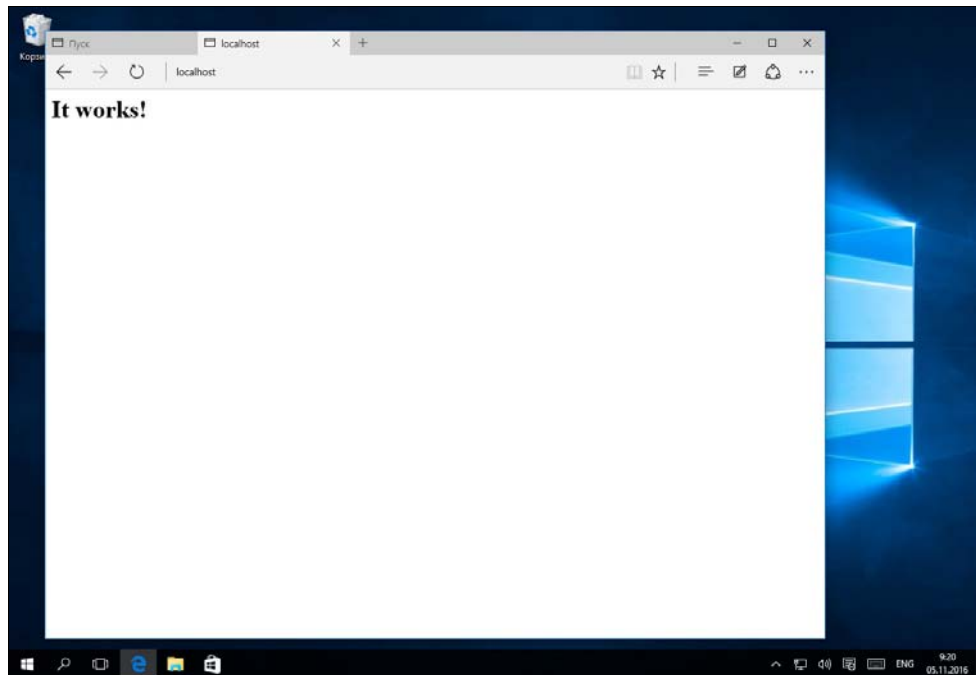


Рис. 1.8. Веб-сервер работает!

нения в конфигурационные файлы Apache и изменить параметры Windows. Сейчас все это придется делать вручную. Если вам этим заниматься не с руки, удалите Apache и установите пакет XAMPP, где в одном флаконе содержится все необходимое: Apache, PHP и MySQL, и вам вручную ничего настраивать не придется.

ПРИМЕЧАНИЕ

В каталоге *software* сопровождающего книгу электронного архива (см. приложение 4) вы найдете различные версии PHP: 5.5, 5.6, 7.0 для 32- и 64-разрядных операционных систем Windows.

Итак, распакуйте архив с PHP в каталог `C:\php7` (рис. 1.9).

После этого добавьте каталог `C:\php` в путь поиска приложений. Для этого выполните следующие действия:

1. Откройте окно **Система** (рис. 1.10).
2. В окне **Система** на панели слева выберите команду **Дополнительные параметры системы**.
3. Нажмите кнопку **Переменные среды**.
4. В окне **Переменные среды** выберите переменную **Path** и нажмите кнопку **Изменить** (рис. 1.11).
5. Добавьте каталог `C:\php7` в путь поиска программы (рис. 1.12).
6. Закройте последние три окна, последовательно нажимая кнопки **ОК**.
7. Закройте окно **Система**.

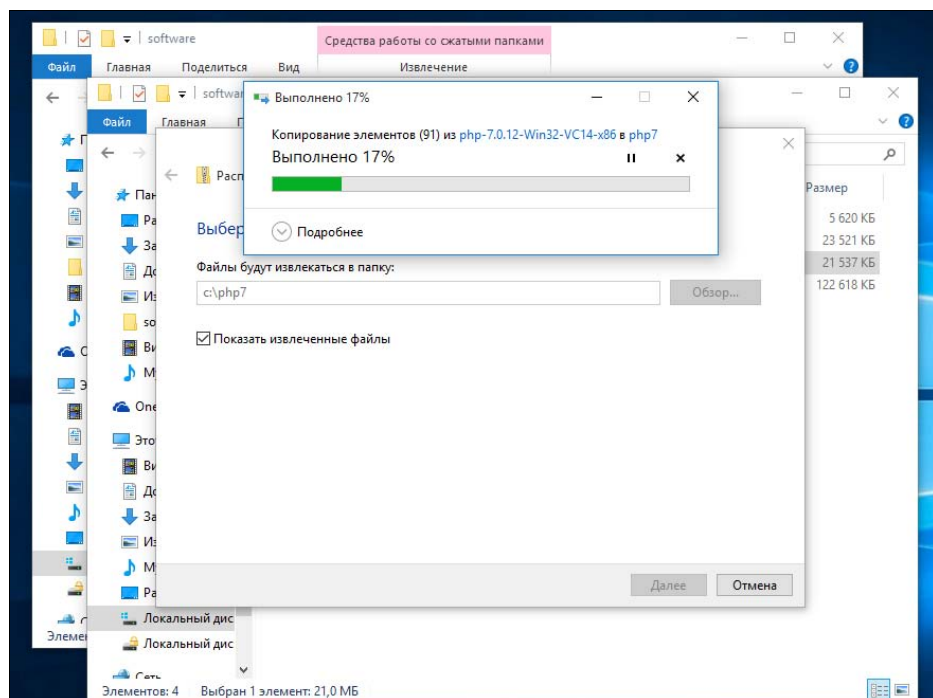


Рис. 1.9. Распаковка PHP

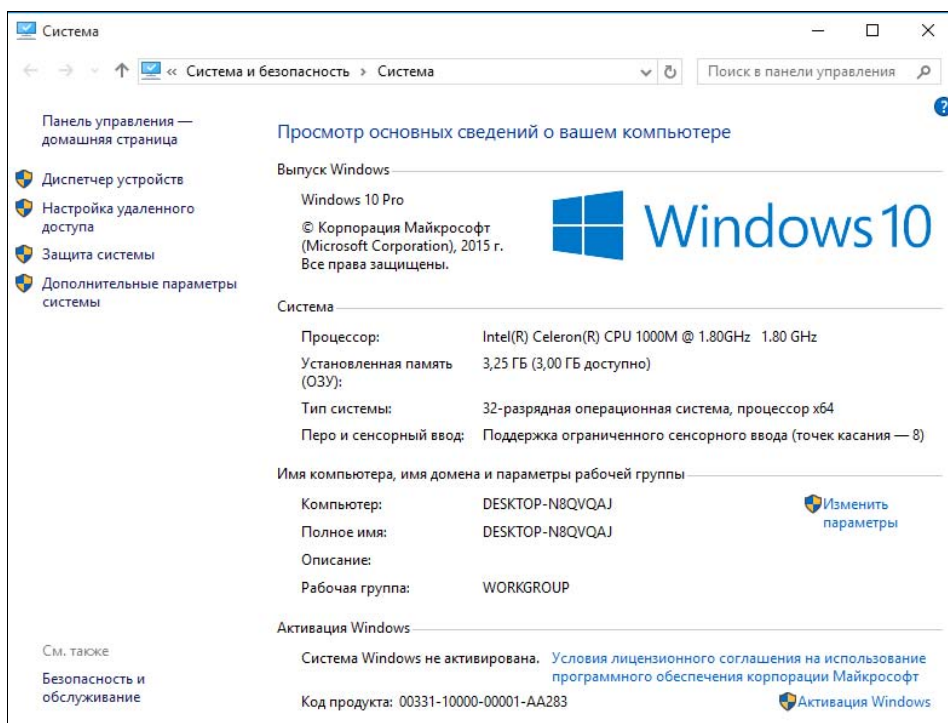


Рис. 1.10. Окно Система

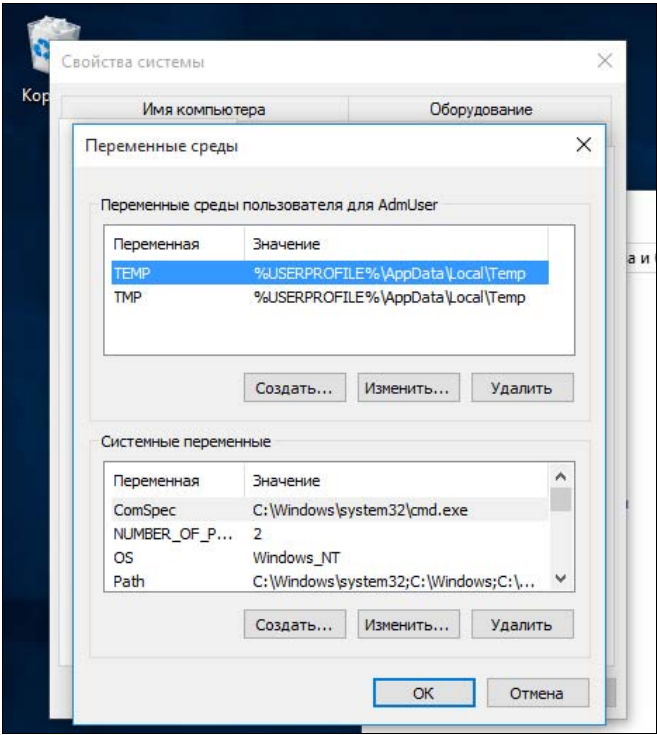


Рис. 1.11. Переменные среды

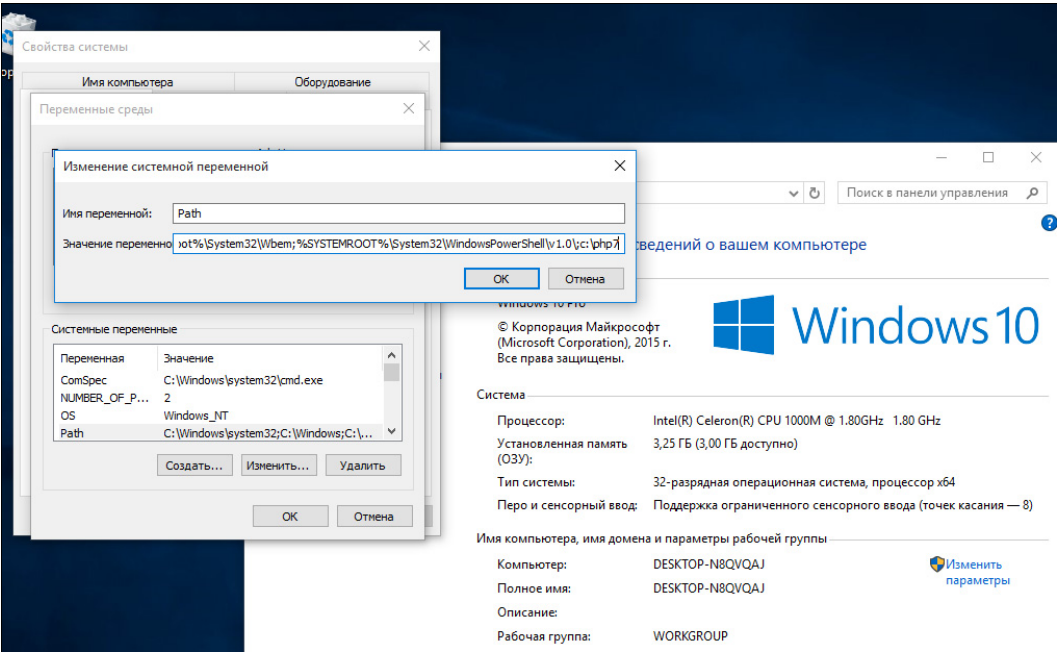


Рис. 1.12. Редактирование переменной Path

Откройте любой файловый менеджер (подойдет и Проводник, но удобнее использовать приложение вроде Total Commander). Перейдите в каталог `C:\php7` и сохраните файл `php.ini-development` как `php.ini`. Откройте файл `php.ini` и найдите в нем строку:

```
; extension_dir = "ext"
```

Эту строку нужно раскомментировать, чтобы получилось так:

```
extension_dir = "ext"
```

Сохраните файл `php.ini` с произведенным изменением.

ПРИМЕЧАНИЕ

В дистрибутиве вы найдете две заготовки файла `php.ini`: `php.ini-development` и `php.ini-production`. Для разработки и отладки кода лучше использовать настройки для разработчика, т. е. файл `php.ini-development`. Файл `php.ini-production` содержит так называемые *производственные параметры*. Разработчику от них мало пользы, т. к. сообщения об ошибках выводиться не будут.

Теперь проверим, все ли вы сделали правильно. Откройте командную строку и введите команду `php -v`. Если вы увидите сообщение с номером версии PHP и другой информацией (рис. 1.13), тогда вы все сделали как надо — т. е. добавили каталог с PHP в путь поиска программ и создали файл `php.ini` с основными настройками.

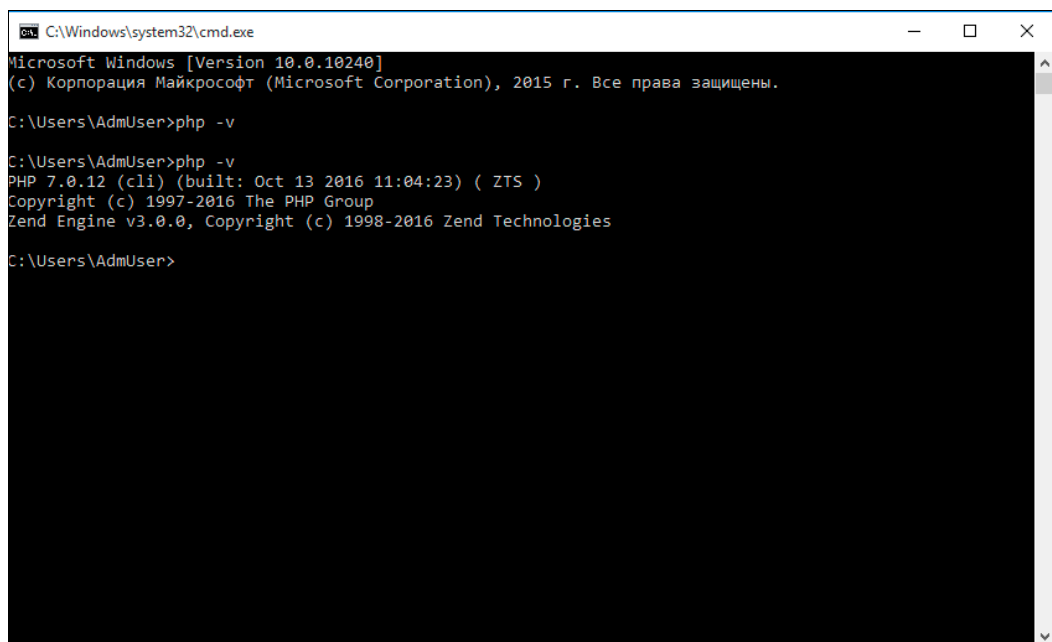


Рис. 1.13. PHP 7 установлен и работает

Однако если на вашем компьютере не установлен распространяемый пакет Visual C++ для Visual Studio 2015, вы увидите совсем другое сообщение (рис. 1.14). К счастью, исправить ситуацию достаточно просто — нужно лишь загрузить

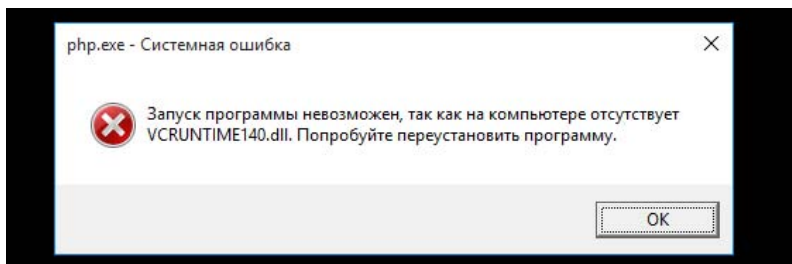


Рис. 1.14. Для запуска PHP 7 не хватает нужного DLL-файла

(с адреса <http://www.microsoft.com/ru-RU/download/details.aspx?id=48145>) и установить недостающее ПО.

Установив РНР, нужно связать его с Apache. Раньше все эти операции делал инсталлятор, сейчас, повторяю, приходится выполнять их вручную.

Перейдите в каталог, куда вы установили Apache (в нашем случае — это C:\Apache24\), затем — в подкаталог conf. Откройте файл конфигурации Apache — httpd.conf. Найдите в этом файле секцию LoadModule. Каждая строка LoadModule загружает какой-то модуль Apache. Проллистайте файл, пока не закончатся строки LoadModule. После последней строки LoadModule добавьте следующие строки:

```
LoadModule php7_module "c:/php7/php7apache2_4.dll"
AddType application/x-httpd-php .php
```

Обратите внимание на пути — возможно, они у вас будут другими, если вы распаковали РНР 7 в иной каталог.

После этого найдите в файле httpd.conf следующие строки:

```
<IfModule dir_module>
DirectoryIndex index.html
</IfModule>
```

Добавьте сюда поддержку файла index.php:

```
<IfModule dir_module>
DirectoryIndex index.php index.html
</IfModule>
```

На этом все... Файл конфигурации Apache можно сохранить. Далее с помощью приложения Apache Monitor, которое устанавливается вместе с Apache, перезапустите Apache, выбрав команду **Restart**.

В каталоге C:\Apache24\htdocs создайте файл info.php со следующим содержимым:

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

Откройте браузер и введите URL:

```
http://localhost/info.php
```

Если вы все сделали правильно, то увидите информационную страницу, сообщающую о конфигурации РНР (рис. 1.15).

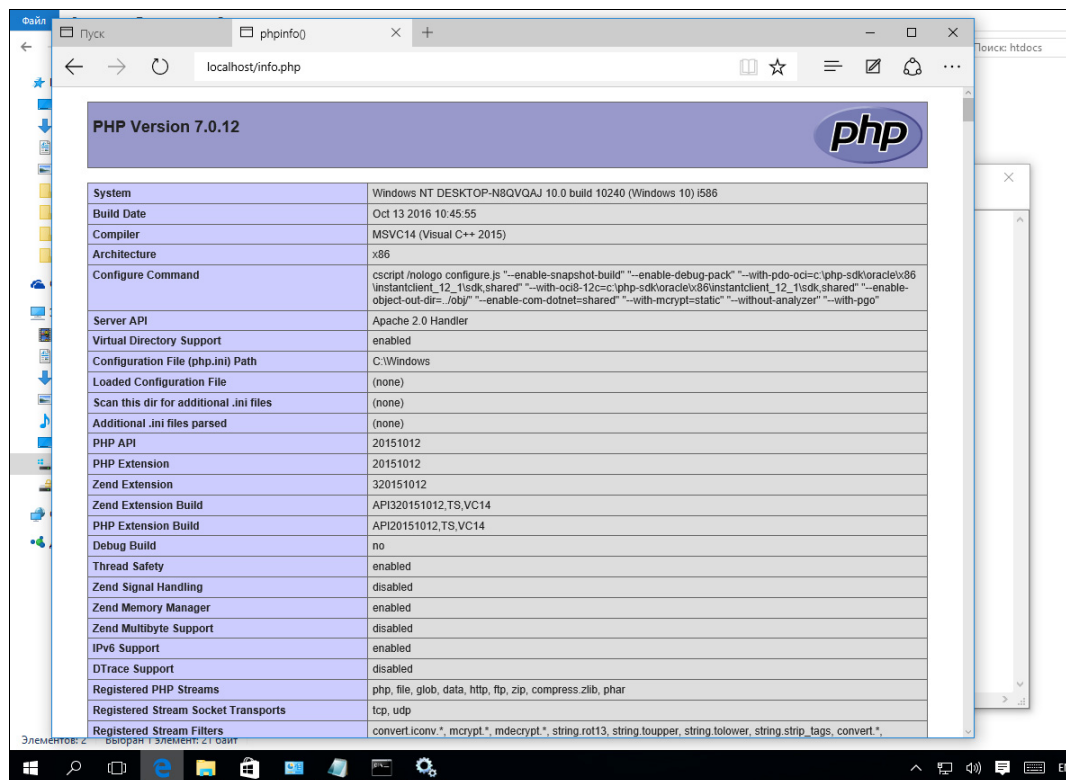


Рис. 1.15. Информационная страница

Если же вы вместо информационной таблицы увидели содержимое файла `info.php`, то PHP не работает вместе с Apache. Чтобы исправить это, выполните следующие действия:

1. Перезапустите Apache (ранее было показано, как это сделать) и повторите попытку. Возможно, вы просто забыли перезапустить веб-сервер. Если же перезапуск сервера не дал результатов, перейдите к следующему шагу.
2. Ранее было показано, какие строки нужно было добавить в файл `httpd.conf`. Убедитесь, что вы все сделали без ошибок, и снова перезапустите веб-сервер. Если и сейчас у вас ничего не получилось, скачайте и установите XAMPP (см. далее) — в 99.99% у вас не будет никаких проблем, и вы сразу получите все необходимое программное обеспечение, готовое к использованию.

Теперь вам осталось скачать и установить сервер баз данных MySQL (<http://mysql.com/>). Его установка проходит вообще без каких-либо нюансов, поэтому ее мы здесь подробно рассматривать не станем.

ПРИМЕЧАНИЕ

В каталоге *software* сопровождающего книгу электронного архива (см. приложение 4) вы найдете MySQL версии 5.5 (файл `mysql-5.5.22-win32.msi`). Это не самая новая версия MySQL, но зато проверенная и стабильная, а самая новая (сейчас это 5.7) нам не нужна (хотя на всякий случай ее инсталлятор также в комплекте имеется).

На мой взгляд, установка в Windows всех компонентов, необходимых для организации связки Apache + PHP + MySQL, в настоящее время напоминает установку такой связки в Linux лет эдак 5 или 7 лет назад. Сейчас же в Linux, наоборот, все стало гораздо проще — там достаточно только установить необходимые пакеты (apache, php, mysql), о чем далее.

1.3.2. Установка для новичков (рекомендуется)

Если вам не с руки устанавливать все компоненты связки отдельно, вы можете установить пакет XAMPP, содержащий Apache, PHP и MySQL, как говорится, "в одном флаконе". Скачать последнюю версию XAMPP можно по адресу <http://www.apachefriends.org/en/xampp.html>.

ПРИМЕЧАНИЕ

В каталоге *software* сопровождающего книгу электронного архива (см. приложение 4) вы найдете последнюю (на момент написания этих строк) версию XAMPP (файл *xampp-win32-7.0.9-1-VC14-installer.exe*). Это версия, содержащая самое новое ПО, включая PHP 7.

Основное преимущество XAMPP — простота установки и управления всеми сервисами, входящими в его состав (веб-сервером, FTP-сервером и MySQL).

Установка XAMPP проблем не вызывает. Программа нормально работает в актуальных версиях Windows: 7, 8 и 10. 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 возникло каких-либо проблем с правами доступа, — так рекомендуют разработчики XAMPP. Да и вам добираться до своих файлов будет проще (рис. 1.16).

При установке Apache и MySQL вас спросят, как вы хотите запускать эти программы: в качестве сервисов (служб) Windows или вручную. Если вы собираетесь программировать на PHP каждый день, тогда удобнее запускать Apache и MySQL как сервисы. Но помните, что даже, когда они вам не нужны, они все равно занимают память и потребляют ресурсы процессора. Поэтому, если вы собираетесь программировать на PHP время от времени, лучше запускать эти программы вручную. Для этого запустите приложение XAMPP Control Panel — откроется панель управления XAMPP. Она позволяет запустить, остановить и настроить подконтрольные сервисы (рис. 1.17):

- ❑ **Apache** — это и есть веб-сервер;
- ❑ **MySQL** — сервер баз данных;
- ❑ **FileZilla** — простейший FTP-сервер для Windows;
- ❑ **Mercury** — простейший почтовый сервер (точнее, агент доставки почты — MTA) для Windows;
- ❑ **Tomcat** — программа-контейнер сервлетов, написанная на языке Java и реализующая спецификацию сервлетов и спецификацию JavaServer Pages (JSP).

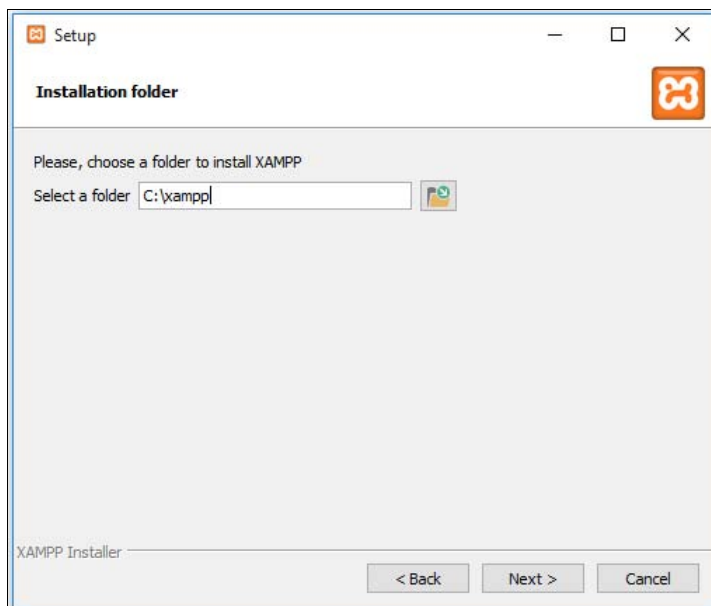


Рис. 1.16. Установка XAMPP

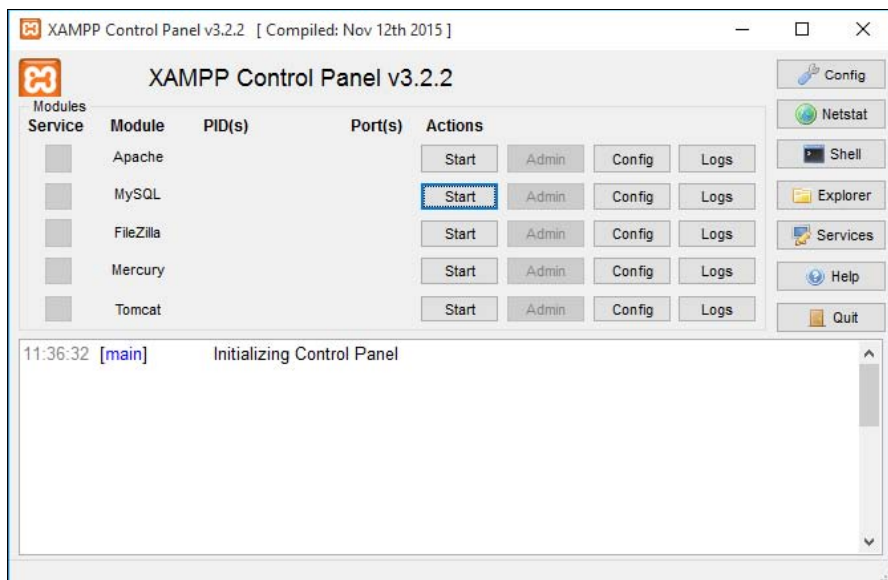


Рис. 1.17. Панель управления XAMPP

Все сервисы запускать не надо — вам хватит первых двух. Третий сервис нужно запустить, если вы хотите написать PHP-сценарий, взаимодействующий с FTP-сервером, — вам не придется искать какой-нибудь FTP-сервер, все будет у вас под рукой. Система **Mercury** подойдет для проектов, подразумевающих отправку почты, — без запуска Mercury ваши проекты не смогут отправлять почту стандартной

функцией `mail()`, или же вам придется использовать дополнительные классы вроде `PHPMailer` для отправки почты через сторонние SMTP-серверы.

Для запуска сервиса нажмите кнопку **Start** напротив его имени. В случае успешного запуска кнопка **Start** превратится в кнопку **Stop** (служит для останова сервиса), а между названием сервиса и кнопкой **Stop** появится надпись **Running**.

При первом запуске каждой из служб вы увидите окно брандмауэра, предлагающее разрешить или запретить приложению доступ в сеть (рис. 1.18), — установите флажок **Общественные сети** и нажмите кнопку **Разрешить доступ**.

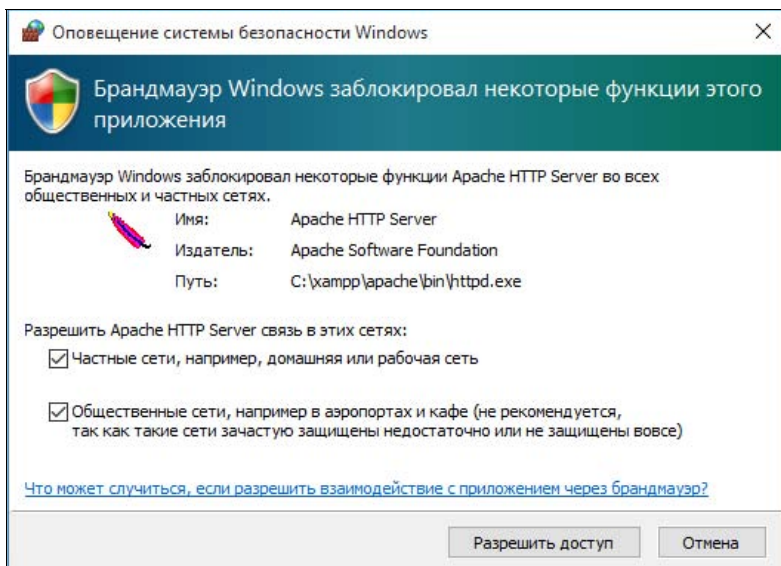


Рис. 1.18. Оповещение системы безопасности Windows

Кнопка **Admin**, которая становится активной после запуска сервиса, позволяет настраивать выбранный сервис. Так, нажатие кнопки **Admin** для сервиса **MySQL** приводит к вызову программы `phpMyAdmin`, которая предоставляет удобный веб-интерфейс для управления вашими базами данных. Подробно программа описана в *приложении 1*.

Для выхода из панели управления XAMPP служит кнопка **Exit** (см. рис. 1.17). Помните, что при выходе из панели запущенные сервисы не будут остановлены. Если вы хотите завершить работу сервисов, нажмите кнопку **Stop** каждого запущенного сервиса, а затем уже нажимайте кнопку **Exit**.

Свои PHP-сценарии при использовании XAMPP вам нужно поместить в подкаталог `htdocs` каталога установки XAMPP. Например, если вы установили XAMPP в каталог `C:\xampp`, то PHP-сценарии следует поместить в каталог `C:\xampp\htdocs`.

При написании сценариев, использующих MySQL, вы можете применить следующие параметры доступа:

- ☐ сервер MySQL — `localhost`;
- ☐ имя пользователя — `root`;

- ❑ пароль — отсутствует;
- ❑ база данных по умолчанию — test (но вы можете создать другую БД с помощью программы phpMyAdmin).

1.4. Установка Apache + PHP + MySQL в Linux

Процесс установки связки Apache + PHP + MySQL в Linux зависит от вашего дистрибутива. В общем случае вам нужно установить следующие пакеты:

- ❑ `apache2` — веб-сервер Apache;
- ❑ `php5-cli/php-7.0-cli` — интерпретатор PHP (версии 5 или 7), работающий в режиме командной строки (command-line interpreter);
- ❑ `php5-imap/php7.0-imap` — поддержка протоколов POP/IMAP для PHP;
- ❑ `php5-gd/php7.0-gd` — поддержка графических функций PHP;
- ❑ `php5-mysql` — поддержка функций для работы с базой данных MySQL. Учтите, что расширение `mysql` для PHP 7 больше не поддерживается, вы не сможете найти аналогичного пакета для версии 7.0. Вместо расширения `mysql` нужно использовать расширение PDO — в том числе и для работы с СУБД MySQL (см. главу 19);
- ❑ `mysql-server-5` — MySQL-сервер;
- ❑ `mysql-client-5` — MySQL-клиент.

Например, в Fedora для установки пакета служит команда `dnf install`:

```
dnf install apache2
```

Сразу после установки MySQL нужно задать пароль администратора:

```
# mysqladmin -u root password ваш_пароль
```

Для управления веб-сервером Apache можно применять следующие команды:

- ❑ `# service httpd start` — запуск сервера;
- ❑ `# service httpd stop` — останов сервера;
- ❑ `# service httpd restart` — перезапуск сервера.

Если в вашем дистрибутиве нет команды `service`, используйте следующие команды:

```
sudo /etc/init.d/apache2 start
sudo /etc/init.d/apache2 stop
sudo /etc/init.d/apache2 restart
```

Аналогично, для управления MySQL служат следующие команды:

```
# service mysql start
# service mysql stop
# service mysql restart
```

1.5. Конфликты с другим программным обеспечением

Дело в том, что стандартные порты Apache (80 и 443) могут использоваться другими приложениями, в частности — популярной программой Skype. Чтобы на вашем компьютере работали и Skype, и веб-сервер Apache, откройте Skype и выполните команду меню **Инструменты | Настройки**. Затем перейдите в раздел **Дополнительно | Соединение** и снимите флажок **Для дополнительных входящих соединений следует использовать порты 80 и 443** (рис. 1.19). Нажмите кнопку **Сохранить**.

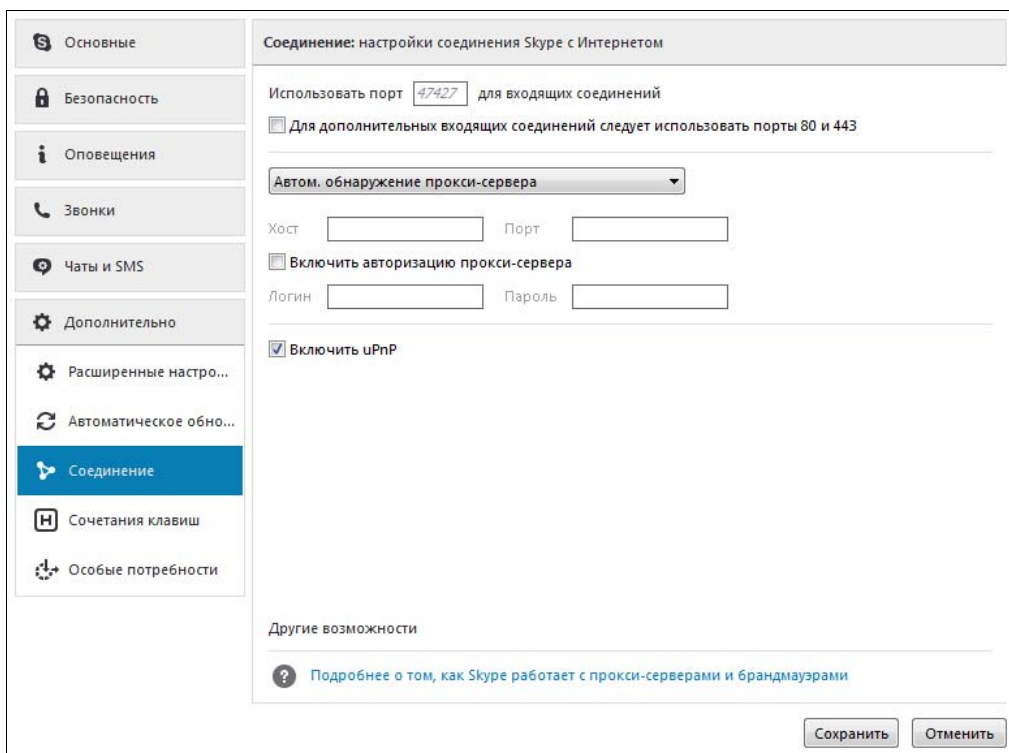


Рис. 1.19. Настройка Skype

Аналогично, программа uTorrent использует порт 8080 для своего веб-интерфейса — и к этому же порту обращается программа Tomcat, входящая в состав XAMPP. Чтобы можно было использовать обе эти программы одновременно, я рекомендую перенести веб-интерфейс uTorrent на другой порт — например, на 8088 (рис. 1.20).

Интересно, что даже при выключенном веб-интерфейсе uTorrent все равно обращается к порту 8080, поэтому предлагаемую настройку придется выполнить, даже если вы вообще не используете веб-интерфейс uTorrent.

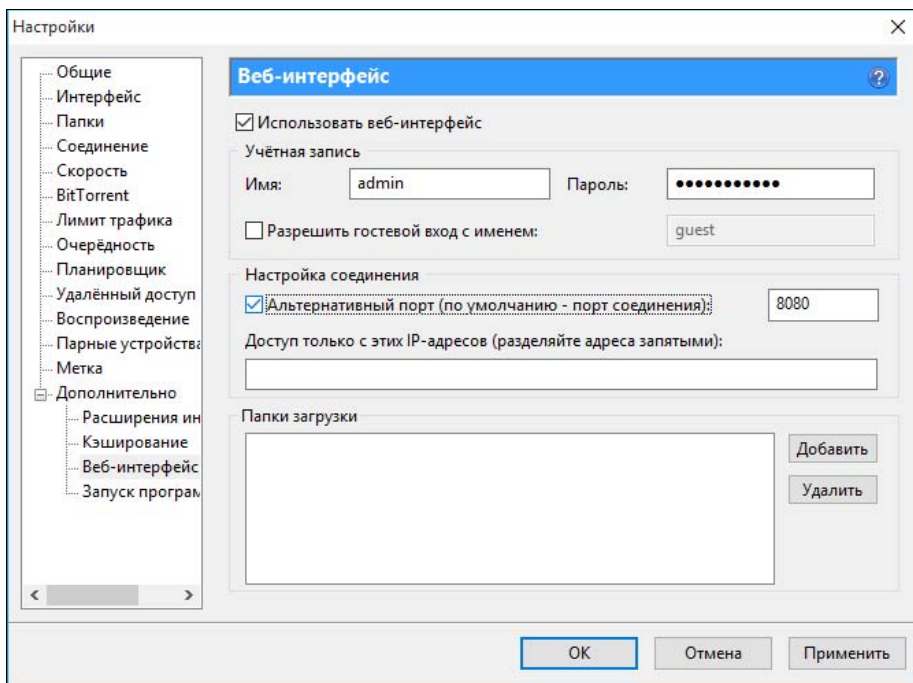


Рис. 1.20. Настройка uTorrent

1.6. Несколько советов

Перед тем как приступить к написанию сценариев, примите от меня еще несколько советов. Итак, если вы при запуске сценария ничего не видите, хотя сценарий должен осуществлять вывод, — нет даже сообщений об ошибках, то добавьте в начало сценария следующую строку:

```
error_reporting(E_ALL);
```

Например:

```
<?php
error_reporting(E_ALL);
...
```

Скорее всего, у вас выключена директива `error_reporting`, и имеется синтаксическая ошибка, поэтому интерпретатор не выводит ее в браузер. Добавив эту директиву, вы научите PHP "говорить".

Представим, что мы подключаемся к серверу баз данных MySQL:

```
// имя узла — localhost, пользователь — root, без пароля
mysql_connect("localhost", "root", "");
mysql_select_db("test");
```

или выполняем запрос:

```
mysql_query($q);
```


Но от сервера нет никакой реакции, и вы не можете понять, в чем ошибка. В этом случае рекомендуется использовать следующую конструкцию:

```
mysql_connect("localhost","root","") or die(mysql_error());  
mysql_query($q) or die(mysql_error());
```

Функция `die()` завершит сценарий и выведет сообщение об ошибке, полученное от функции `mysql_error()`.

Вот теперь можно перейти ко *второй главе*, в которой вы напишете свою первую программу.

ГЛАВА 2



Программа на PHP

2.1. Ваша первая программа

Традиционно многие книги по программированию начинаются с создания простейшей программы, выводящей строку **Hello world!**. В этой книге такой программы не будет. Вместо этого мы напишем программу, тестирующую наше MySQL-соединение. Этим мы сразу "убьем двух зайцев": напишем первую PHP-программу и проверим созданную в *главе 1* конфигурацию.

Логика нашей будущей программы проста — если удалось подключиться к MySQL-серверу, то все настроено правильно, иначе будет выведено сообщение об ошибке. В последующих главах книги будут приведены более сложные примеры работы с MySQL.

Итак, приступим к написанию нашего первого PHP-сценария (листинг 2.1).

ЕЩЕ РАЗ О РАСШИРЕНИИ MYSQL

Как уже отмечалось ранее, в PHP 7 отсутствует поддержка расширения *mysql*. Поэтому далее приводятся два аналогичных сценария: один (листинг 2.1, а) будет работать в старых версиях PHP, второй, использующий расширение PDO (листинг 2.1, б), — в новой, 7-й версии.

Листинг 2.1, а. Первая PHP-программа (PHP 5)

```
<?php

// параметры доступа к MySQL, возможно, придется изменить:
if (!mysql_connect('localhost','root','')) die('error');

/* это пример многострочного
   комментария */

# пример комментария в стиле UNIX bash:
/* если программа не остановлена die(), тогда выводим
   сообщение об успешном подключении */
```