

Максим Кузнецов  
Игорь Симдянов



# PHP 5/6



- Новое в PHP 5/6
- Наиболее полное описание языка PHP
- Объектно-ориентированное программирование
- Работа с базами данных
- Тонкости работы с HTTP-протоколом
- Разработка динамических Web-приложений
- Динамические изображения

Наиболее  
полное  
руководство

**В ПОДЛИННИКЕ®**

**Максим Кузнецов**

**Игорь Симдянов**

# **PHP 5/6**

Санкт-Петербург

«БХВ-Петербург»

2010

УДК 681.3.068+800.92RHP5/6  
ББК 32.973.26–018.1  
К89

**Кузнецов, М. В.**

К89 РНР 5/6 / М. В. Кузнецов, И. В. Симдянов. — СПб.: БХВ-Петербург, 2010. — 1024 с.: ил. — (В подлиннике)

ISBN 978-5-9775-0304-4

Рассмотрены самые последние версии языка разработки серверных сценариев РНР — 5.3 и 6.0. Подробно описан язык РНР, в том числе вопросы объектно-ориентированного программирования на РНР, обработки исключительных ситуаций, взаимодействия с MySQL и многое другое. В книге обсуждаются все вопросы, с которыми может столкнуться Web-разработчик, начиная с создания инструментария для быстрой разработки Web-приложений и последних нововведений языка программирования РНР и заканчивая вопросами безопасности и особенностями программирования клиент-серверных приложений. Книга имеет практическую направленность, т. к. содержит множество примеров, взятых из практики разработки динамических Web-сайтов.

*Для программистов и Web-разработчиков*

УДК 681.3.068+800.92RHP5/6  
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>Григорий Добин</i>       |
| Редактор                | <i>Анна Кузьмина</i>        |
| Компьютерная верстка    | <i>Натальи Караваевой</i>   |
| Корректор               | <i>Виктория Пиотровская</i> |
| Дизайн серии            | <i>Инны Тачиной</i>         |
| Оформление обложки      | <i>Елены Беляевой</i>       |
| Зав. производством      | <i>Николай Тверских</i>     |

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.09.09.

Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 82,56.

Тираж 2000 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов  
в ГУП "Типография "Наука"  
199034, Санкт-Петербург, 9 линия, 12.

# Оглавление

|   |           |
|---|-----------|
| <b>Введение</b> .....                           | <b>1</b>  |
| Нововведения PHP 6 .....                        | 1         |
| Путеводитель по книге .....                     | 2         |
| On-line-поддержка книги.....                    | 4         |
| Благодарности .....                             | 4         |
| <br>  |           |
| <b>Глава 1. Интернет</b> .....                  | <b>5</b>  |
| 1.1. История развития Интернета.....            | 5         |
| 1.2. Принципы работы Интернета .....            | 7         |
| 1.3. Место и роль PHP в Интернете.....          | 12        |
| 1.3.1. Серверные технологии.....                | 13        |
| UNIX-подобная операционная система.....         | 13        |
| Web-сервер .....                                | 13        |
| Серверный язык .....                            | 13        |
| Файлы и базы данных.....                        | 14        |
| Электронная почта.....                          | 15        |
| 1.3.2. Клиентские технологии.....               | 15        |
| Web-браузеры, HTML .....                        | 15        |
| Каскадные таблицы стилей CSS и XML.....         | 16        |
| Flash-ролики .....                              | 16        |
| FTP-клиенты.....                                | 17        |
| Удаленный доступ к серверу. Протокол SSH .....  | 17        |
| <br>  |           |
| <b>Глава 2. Быстрый старт</b> .....             | <b>18</b> |
| 2.1. Скрипты.....                               | 18        |
| 2.2. Начальные и конечные теги .....            | 20        |
| 2.3. Использование точки с запятой .....        | 22        |
| 2.4. Составные выражения. Фигурные скобки ..... | 24        |
| 2.5. Комментарии .....                          | 25        |
| <br>  |           |
| <b>Глава 3. Переменные и типы данных</b> .....  | <b>27</b> |
| 3.1. Объявление переменной. Оператор = .....    | 27        |
| 3.2. Типы данных.....                           | 28        |
| 3.3. Целые числа .....                          | 29        |

|  |            |
|--|------------|
| 3.4. Вещественные числа .....  | 30         |
| 3.5. Строки .....  | 31         |
| 3.6. Кавычки .....   | 31         |
| 3.7. Оператор <<< .....  | 35         |
| 3.8. Обращение к неинициализированной переменной. Замечания (Notice) .....                 | 35         |
| 3.9. Специальный тип <i>NULL</i> .....   | 37         |
| 3.10. Логический тип .....   | 38         |
| 3.11. Уничтожение переменной. Конструкция <i>unset()</i> .....                             | 38         |
| 3.12. Проверка существования переменной. Конструкции <i>isset()</i> и <i>empty()</i> ..... | 39         |
| 3.13. Определение типа переменной .....  | 41         |
| 3.14. Неявное приведение типов .....   | 46         |
| 3.15. Явное приведение типов .....   | 48         |
| 3.16. Динамические переменные .....  | 52         |
| <b>Глава 4. Константы .....</b>  | <b>54</b>  |
| 4.1. Объявление константы. Функция <i>define()</i> .....                                   | 54         |
| 4.2. Функции для работы с константами .....  | 58         |
| 4.3. Константы с динамическими именами. Функция <i>constant()</i> .....                    | 58         |
| 4.4. Проверка существования константы .....  | 59         |
| 4.5. Предопределенные константы .....  | 60         |
| <b>Глава 5. Операторы и конструкции языка .....</b>  | <b>61</b>  |
| 5.1. Объединение строк. Оператор "точка" .....   | 61         |
| 5.2. Конструкция <i>echo</i> . Оператор "запятая" .....                                    | 62         |
| 5.3. Арифметические операторы .....  | 63         |
| 5.4. Поразрядные операторы .....   | 67         |
| 5.5. Операторы сравнения .....   | 72         |
| 5.6. Условный оператор <i>if</i> .....   | 75         |
| 5.7. Логические операторы .....  | 77         |
| 5.8. Условный оператор <i>x ? y : z</i> .....  | 84         |
| 5.9. Переключатель <i>switch</i> .....   | 85         |
| 5.10. Цикл <i>while</i> .....  | 90         |
| 5.11. Цикл <i>do ... while</i> .....   | 96         |
| 5.12. Цикл <i>for</i> .....  | 97         |
| 5.13. Включение файлов .....   | 101        |
| 5.14. Подавление вывода ошибок. Оператор <i>@</i> .....                                    | 106        |
| 5.15. Приоритет выполнения операторов .....  | 108        |
| <b>Глава 6. Массивы .....</b>  | <b>110</b> |
| 6.1. Создание массива .....  | 110        |
| 6.2. Ассоциативные и индексные массивы .....   | 116        |
| 6.3. Многомерные массивы .....   | 121        |
| 6.4. Интерполяция элементов массива в строки .....   | 122        |
| 6.5. Конструкция <i>list()</i> .....   | 123        |

|  |     |
|--|-----|
| 6.6. Обход массива .....   | 125 |
| 6.7. Цикл <i>foreach</i> .....                                       | 129 |
| 6.8. Сечения массива .....   | 131 |
| 6.9. Слияние массивов .....  | 133 |
| 6.10. Сравнение массивов.....  | 136 |
| 6.11. Проверка существования элементов массива.....                  | 140 |
| 6.12. Строки как массивы .....                                       | 144 |
| 6.13. Количество элементов в массиве.....                            | 145 |
| 6.14. Сумма элементов массива .....                                  | 147 |
| 6.15. Случайные элементы массива.....                                | 147 |
| 6.16. Сортировка массивов.....                                       | 150 |
| 6.17. Добавление/удаление элементов массива.....                     | 158 |
| 6.18. Работа с ключами массива .....                                 | 165 |
| 6.19. Преобразование массивов .....                                  | 170 |
| 6.20. Суперглобальные массивы. Массив <code>\$_SERVER</code> .....   | 177 |
| 6.20.1. Элемент <code>\$_SERVER['DOCUMENT_ROOT']</code> .....        | 178 |
| 6.20.2. Элемент <code>\$_SERVER['HTTP_ACCEPT']</code> .....          | 178 |
| 6.20.3. Элемент <code>\$_SERVER['HTTP_ACCEPT_LANGUAGE']</code> ..... | 179 |
| 6.20.4. Элемент <code>\$_SERVER['HTTP_HOST']</code> .....            | 180 |
| 6.20.5. Элемент <code>\$_SERVER['HTTP_REFERER']</code> .....         | 180 |
| 6.20.6. Элемент <code>\$_SERVER['HTTP_USER_AGENT']</code> .....      | 181 |
| 6.20.7. Элемент <code>\$_SERVER['REMOTE_ADDR']</code> .....          | 182 |
| 6.20.8. Элемент <code>\$_SERVER['SCRIPT_FILENAME']</code> .....      | 182 |
| 6.20.9. Элемент <code>\$_SERVER['SERVER_NAME']</code> .....          | 182 |
| 6.20.10. Элемент <code>\$_SERVER['REQUEST_METHOD']</code> .....      | 183 |
| 6.20.11. Элемент <code>\$_SERVER['QUERY_STRING']</code> .....        | 183 |
| 6.20.12. Элемент <code>\$_SERVER['PHP_SELF']</code> .....            | 184 |
| 6.20.13. Элемент <code>\$_SERVER['REQUEST_URI']</code> .....         | 184 |

## **Глава 7. Функции .....** 185

|   |     |
|---|-----|
| 7.1. Объявление и вызов функции .....                             | 185 |
| 7.2. Параметры функции.....                                       | 188 |
| 7.3. Передача параметров по значению и ссылке .....               | 189 |
| 7.4. Необязательные параметры.....                                | 190 |
| 7.5. Переменное количество параметров .....                       | 191 |
| 7.6. Глобальные переменные.....                                   | 193 |
| 7.7. Статические переменные.....                                  | 194 |
| 7.8. Возврат массива функцией.....                                | 195 |
| 7.9. Рекурсивные функции.....                                     | 196 |
| 7.10. Вложенные функции .....                                     | 197 |
| 7.11. Динамическое имя функции.....                               | 198 |
| 7.12. Анонимные функции .....                                     | 199 |
| 7.13. Проверка существования функции.....                         | 201 |
| 7.14. Неявное выполнение функций. Оператор <i>declare()</i> ..... | 206 |
| 7.15. Вспомогательные функции .....                               | 211 |

|  |            |
|--|------------|
| <b>Глава 8. Взаимодействие PHP с HTML .....</b>          | <b>214</b> |
| 8.1. Передача параметров методом <i>GET</i> .....        | 214        |
| 8.2. HTML-форма и ее обработчик .....                    | 218        |
| 8.3. Текстовое поле .....                                | 223        |
| 8.4. Поле для приема пароля .....                        | 224        |
| 8.5. Текстовая область .....                             | 225        |
| 8.6. Скрытое поле .....                                  | 226        |
| 8.7. Флажок .....  | 227        |
| 8.8. Список .....  | 229        |
| 8.9. Переключатель .....                                 | 231        |
| 8.10. Загрузка файла на сервер .....                     | 232        |
| <b>Глава 9. Строковые функции .....</b>                  | <b>237</b> |
| 9.1. Функции для работы с символами .....                | 237        |
| 9.2. Поиск в строке .....                                | 243        |
| 9.3. Замена в тексте .....                               | 248        |
| 9.4. Преобразование регистра .....                       | 253        |
| 9.5. Работа с HTML-кодом .....                           | 255        |
| 9.6. Экранирование .....                                 | 264        |
| 9.7. Локальные настройки (локаль) .....                  | 266        |
| 9.8. Форматный вывод .....                               | 273        |
| 9.9. Преобразование кодировок .....                      | 282        |
| 9.10. Сравнение строк .....                              | 292        |
| 9.11. Хранение данных .....                              | 298        |
| 9.12. Работа с путями к файлам и каталогам .....         | 301        |
| 9.13. Объединение и разбиение строк .....                | 303        |
| <b>Глава 10. Регулярные выражения .....</b>              | <b>313</b> |
| 10.1. Как изучать регулярные выражения? .....            | 313        |
| 10.2. Синтаксис регулярных выражений .....               | 314        |
| 10.3. Функции для работы с регулярными выражениями ..... | 317        |
| 10.4. Функция <i>preg_match()</i> .....                  | 318        |
| 10.5. Функция <i>preg_match_all()</i> .....              | 323        |
| 10.6. Функция <i>preg_replace()</i> .....                | 326        |
| 10.7. Функция <i>preg_replace_callback()</i> .....       | 330        |
| 10.8. Функция <i>preg_split()</i> .....                  | 332        |
| 10.9. Функция <i>preg_quote()</i> .....                  | 334        |
| <b>Глава 11. Дата и время .....</b>                      | <b>336</b> |
| 11.1. Формирование даты и времени .....                  | 336        |
| 11.2. Географическая привязка .....                      | 342        |
| 11.3. Форматирование даты и времени .....                | 348        |

|  |            |
|--|------------|
| <b>Глава 12. Математические функции</b> .....                            | <b>362</b> |
| 12.1. Предопределенные константы .....                                   | 362        |
| 12.2. Поиск максимума и минимума .....                                   | 363        |
| 12.3. Генерация случайных чисел .....                                    | 365        |
| 12.4. Преобразование значений между различными системами счисления ..... | 367        |
| 12.5. Округление чисел .....   | 370        |
| 12.6. Логарифмические и степенные функции .....                          | 373        |
| 12.7. Тригонометрические функции .....                                   | 378        |
| 12.8. Информационные функции .....                                       | 383        |
| 12.9. Вычисления с повышенной точностью .....                            | 389        |
| <b>Глава 13. Файлы и каталоги</b> .....                                  | <b>392</b> |
| 13.1. Создание файлов .....  | 392        |
| 13.2. Манипулирование файлами .....                                      | 398        |
| 13.3. Чтение и запись файлов .....                                       | 401        |
| 13.3.1. Чтение файлов .....  | 404        |
| 13.3.2. Запись файлов .....  | 410        |
| 13.3.3. Обязательно ли закрывать файлы? .....                            | 414        |
| 13.3.4. Дозапись файлов .....  | 415        |
| 13.3.5. Уменьшение размера файла .....                                   | 416        |
| 13.3.6. Блокировка файлов .....  | 417        |
| 13.3.7. Файлы CSV .....  | 422        |
| 13.3.8. Чтение и разбор ini-файла .....                                  | 424        |
| 13.3.9. Прямое манипулирование файловым указателем .....                 | 427        |
| 13.4. Создание и работа со ссылками .....                                | 430        |
| 13.5. Атрибуты файла .....   | 432        |
| 13.6. Права доступа .....  | 438        |
| 13.7. Состояние файловой системы .....                                   | 442        |
| 13.8. Каталоги .....   | 443        |
| 13.9. Архивные файлы .....   | 450        |
| <b>Глава 14. HTTP-заголовки</b> .....                                    | <b>454</b> |
| 14.1. Функции для управления HTTP-заголовками .....                      | 455        |
| 14.2. Кодировка страницы .....   | 457        |
| 14.3. HTTP-коды состояния .....  | 457        |
| 14.4. Список HTTP-заголовков .....                                       | 461        |
| 14.5. Подавление кэширования .....                                       | 464        |
| <b>Глава 15. Cookie</b> .....  | <b>468</b> |
| <b>Глава 16. Сессии</b> .....  | <b>473</b> |
| 16.1. Функции обработки сессии .....                                     | 473        |
| 16.2. Перегрузка механизма сессии .....                                  | 477        |



|  |            |
|--|------------|
| <b>Глава 17. Управление выводом</b> .....                    | <b>481</b> |
| 17.1. Функции управления выводом.....                        | 481        |
| 17.2. Размер и тип документа.....                            | 483        |
| 17.3. Подсветка ссылок.....                                  | 486        |
| 17.4. Вложенная буферизация.....                             | 487        |
| 17.5. Обработчики буферов.....                               | 488        |
| 17.6. Список применяемых обработчиков.....                   | 489        |
| 17.7. Сжатие страницы.....                                   | 491        |
| 17.8. Преобразование кодировки страницы.....                 | 491        |
| 17.9. Преобразование адресов ссылок.....                     | 492        |
| <br>   |            |
| <b>Глава 18. Средства шифрования</b> .....                   | <b>494</b> |
| 18.1. Необратимое шифрование.....                            | 494        |
| 18.2. Система регистрации и аутентификации.....              | 497        |
| 18.3. Обратимое шифрование.....                              | 501        |
| <br>   |            |
| <b>Глава 19. Сетевое взаимодействие. Сокеты и CURL</b> ..... | <b>502</b> |
| 19.1. Обращение к удаленным страницам.....                   | 502        |
| 19.2. Сокеты.....  | 504        |
| 19.3. Библиотека CURL.....                                   | 507        |
| 19.4. Получение HTTP-заголовков с сервера.....               | 515        |
| 19.5. Определение размера файла на удаленном хосте.....      | 519        |
| 19.6. Отправка данных методом POST.....                      | 519        |
| 19.7. Фальсификация реферера.....                            | 522        |
| 19.8. Фальсификация пользовательского агента.....            | 526        |
| 19.9. Фальсификация cookie.....                              | 527        |
| 19.10. Работа с доменами и IP-адресами.....                  | 530        |
| 19.11. Определение IP-адреса по сетевому адресу.....         | 530        |
| 19.12. Определение сетевого адреса по IP-адресу.....         | 531        |
| 19.13. Следование реферальному серверу.....                  | 532        |
| <br>   |            |
| <b>Глава 20. Взаимодействие с интерпретатором PHP</b> .....  | <b>535</b> |
| 20.1. Настройки конфигурационного файла php.ini.....         | 535        |
| 20.1.1. Параметры языка PHP.....                             | 536        |
| 20.1.2. Директивы безопасности.....                          | 539        |
| 20.1.3. Настройка подсветки PHP-кода.....                    | 545        |
| 20.1.4. Кэш файловой системы.....                            | 547        |
| 20.1.5. Взаимодействие с клиентом.....                       | 547        |
| 20.1.6. Ограничение ресурсов.....                            | 548        |
| 20.1.7. Обработка ошибок.....                                | 549        |
| 20.1.8. Обработка данных.....                                | 552        |
| 20.1.9. Загрузка файлов.....                                 | 558        |
| 20.1.10. Сетевой доступ.....                                 | 558        |

|   |     |
|---|-----|
| 20.1.11. Подключение расширений .....                                 | 558 |
| 20.1.12. Настройка сессии .....                                       | 559 |
| 20.1.13. Настройка даты и времени .....                               | 561 |
| 20.2. Изменение настроек <code>php.ini</code> средствами Apache ..... | 561 |
| 20.3. Функции управления интерпретатором PHP .....                    | 564 |

## **Глава 21. Объекты и классы..... 567**

|  |     |
|--|-----|
| 21.1. Введение в объектно-ориентированное программирование.....                                | 567 |
| 21.2. Создание класса.....   | 568 |
| 21.3. Создание объекта .....   | 569 |
| 21.4. Инкапсуляция. Спецификаторы доступа .....  | 571 |
| 21.5. Методы класса. Член <code>\$this</code> .....  | 572 |
| 21.6. Дамп объекта .....   | 576 |
| 21.7. Специальные методы класса .....  | 577 |
| 21.8. Функции для работы с методами и классами .....   | 578 |
| 21.9. Конструктор. Метод <code>__construct()</code> .....                                      | 580 |
| 21.10. Параметры конструктора.....   | 582 |
| 21.11. Деструктор. Метод <code>__destruct()</code> .....                                       | 584 |
| 21.12. Автозагрузка классов. Функция <code>__autoload()</code> .....                           | 586 |
| 21.13. Аксессуары. Методы <code>__set()</code> и <code>__get()</code> .....                    | 586 |
| 21.14. Проверка существования члена класса. Метод <code>__isset()</code> .....                 | 588 |
| 21.15. Уничтожение члена класса. Метод <code>__unset()</code> .....                            | 589 |
| 21.16. Динамические методы. Метод <code>__call()</code> .....                                  | 591 |
| 21.17. Интерполяция объекта. Метод <code>__toString()</code> .....                             | 593 |
| 21.18. Экспорт объектов. Метод <code>__set_state()</code> .....                                | 595 |
| 21.19. Статические члены класса.....   | 600 |
| 21.20. Статические методы класса.....  | 603 |
| 21.21. Константы класса .....  | 603 |
| 21.22. Предопределенные константы .....  | 605 |
| 21.23. Клонирование объекта.....   | 607 |
| 21.24. Управление процессом клонирования. Метод <code>__clone()</code> .....                   | 608 |
| 21.25. Сериализация объектов .....   | 610 |
| 21.26. Управление сериализацией. Методы <code>__sleep()</code> и <code>__wakeup()</code> ..... | 612 |

## **Глава 22. Наследование .....**

|  |     |
|--|-----|
| 22.1. Наследование .....                         | 621 |
| 22.2. Спецификаторы доступа и наследование ..... | 624 |
| 22.3. Перегрузка методов.....                    | 627 |
| 22.4. Полиморфизм .....                          | 628 |
| 22.5. Абстрактные классы .....                   | 630 |
| 22.6. Абстрактные методы.....                    | 631 |
| 22.7. <i>Final</i> -методы класса .....          | 632 |
| 22.8. <i>Final</i> -классы .....                 | 634 |

|   |            |
|---|------------|
| <b>Глава 23. Интерфейсы</b> .....   | <b>635</b> |
| 23.1. Создание интерфейса .....   | 635        |
| 23.2. Интерфейсы и наследование классов .....                                     | 637        |
| 23.3. Реализация нескольких интерфейсов .....                                     | 639        |
| 23.4. Проверка существования интерфейса .....                                     | 640        |
| 23.5. Наследование интерфейсов .....  | 641        |
| 23.6. Реализует ли объект интерфейс? .....  | 643        |
| <b>Глава 24. Обработка ошибок и исключительных ситуаций</b> .....                 | <b>647</b> |
| 24.1. Обработка ошибок .....  | 647        |
| 24.2. Обработка ошибок и исключения .....   | 660        |
| 24.2.1. Синтаксис исключений .....  | 660        |
| 24.2.2. Интерфейс класса <i>Exception</i> .....                                   | 662        |
| 24.2.3. Генерация исключений в функциях .....                                     | 666        |
| 24.2.4. Стек обработки исключительной ситуации .....                              | 669        |
| 24.2.5. Генерация исключений в классах .....                                      | 671        |
| 24.2.6. Генерация исключений в иерархиях классов .....                            | 675        |
| 24.2.7. Использование объекта класса <i>Exception</i> в строковом контексте ..... | 678        |
| 24.2.8. Создание собственных исключений .....                                     | 679        |
| 24.2.9. Создание новых типов исключений .....                                     | 683        |
| 24.2.10. Перехват исключений производных классов .....                            | 685        |
| 24.2.11. Что происходит, когда исключения не перехватываются? .....               | 687        |
| 24.2.12. Вложенные контролируемые блоки .....                                     | 689        |
| 24.2.13. Повторная генерация исключений .....                                     | 691        |
| <b>Глава 25. Отражения</b> .....  | <b>694</b> |
| 25.1. Иерархия классов отражения .....  | 694        |
| 25.2. Отражение функции. Класс <i>ReflectionFunction</i> .....                    | 695        |
| 25.3. Отражение параметра функции. Класс <i>ReflectionParameter</i> .....         | 700        |
| 25.4. Отражение класса. Класс <i>ReflectionClass</i> .....                        | 702        |
| 25.5. Отражение объекта. Класс <i>ReflectionObject</i> .....                      | 709        |
| 25.6. Отражение метода класса. Класс <i>ReflectionMethod</i> .....                | 710        |
| 25.7. Отражение члена класса. Класс <i>ReflectionProperty</i> .....               | 712        |
| 25.8. Исключения механизма отражения .....  | 715        |
| 25.9. Отражение расширения. Класс <i>ReflectionExtension</i> .....                | 716        |
| 25.10. Вспомогательный класс <i>Reflection</i> .....                              | 719        |
| <b>Глава 26. Предопределенные классы</b> .....                                    | <b>721</b> |
| 26.1. Класс <i>Dir</i> .....  | 721        |
| 26.2. Классы времени .....  | 723        |
| 26.3. Библиотека SPL .....  | 730        |
| 26.3.1. Итераторы .....   | 731        |
| 26.3.2. Интерфейс <i>Iterator</i> .....   | 734        |
| 26.3.3. Класс <i>DirectoryIterator</i> .....                                      | 736        |

|   |            |
|---|------------|
| 26.3.4. Класс <i>FilterIterator</i> .....         | 738        |
| 26.3.5. Класс <i>LimitIterator</i> .....          | 739        |
| 26.3.6. Рекурсивные итераторы .....               | 740        |
| <b>Глава 27. Работа с СУБД MySQL.....</b>         | <b>742</b> |
| 27.1. Введение в СУБД и SQL.....                  | 743        |
| 27.2. Первичные ключи.....                        | 746        |
| 27.3. Создание и удаление базы данных .....       | 747        |
| 27.4. Выбор базы данных.....                      | 749        |
| 27.5. Типы данных.....                            | 751        |
| 27.6. Создание и удаление таблиц .....            | 756        |
| 27.7. Вставка числовых значений в таблицу.....    | 763        |
| 27.8. Вставка строковых значений в таблицу.....   | 764        |
| 27.9. Вставка календарных значений .....          | 766        |
| 27.10. Вставка уникальных значений .....          | 769        |
| 27.11. Механизм <i>AUTO_INCREMENT</i> .....       | 769        |
| 27.12. Многострочный оператор <i>INSERT</i> ..... | 770        |
| 27.13. Удаление данных.....                       | 771        |
| 27.14. Обновление записей.....                    | 772        |
| 27.15. Выборка данных .....                       | 775        |
| 27.16. Условная выборка .....                     | 776        |
| 27.17. Псевдонимы столбцов .....                  | 782        |
| 27.18. Сортировка записей .....                   | 783        |
| 27.19. Вывод записей в случайном порядке .....    | 786        |
| 27.20. Ограничение выборки.....                   | 786        |
| 27.21. Вывод уникальных значений .....            | 787        |
| 27.22. Объединение таблиц .....                   | 789        |
| 27.23. Функции MySQL .....                        | 791        |
| 27.23.1. Математические функции.....              | 791        |
| Вычисление площади треугольников .....            | 793        |
| Округление результатов вычисления.....            | 795        |
| 27.23.2. Функции даты и времени.....              | 796        |
| Форматирование календарных значений .....         | 801        |
| Вычисление возраста человека.....                 | 802        |
| Преобразование даты в UNIXSTAMP-формат .....      | 804        |
| 27.23.3. Строковые функции .....                  | 805        |
| Изменение кодировки строки.....                   | 809        |
| Первые несколько символов строки .....            | 810        |
| Извлечение инициалов .....                        | 810        |
| Изменение регистра строки .....                   | 811        |
| 27.23.4. Функции шифрования .....                 | 812        |
| Обратимое шифрование .....                        | 814        |
| Необратимое шифрование .....                      | 815        |
| 27.23.5. Агрегатные функции .....                 | 816        |
| Среднее значение.....                             | 817        |
| Сортировка агрегатных значений .....              | 820        |

|   |            |
|---|------------|
| Подсчет количества записей в таблице .....                        | 820        |
| Объединение значений группы .....                                 | 823        |
| Поиск минимального и максимального значений .....                 | 825        |
| Сумма столбца .....   | 826        |
| 27.23.6. Разное .....   | 827        |
| Преобразование IP-адреса .....                                    | 828        |
| <b>Глава 28. Взаимодействие MySQL и PHP .....</b>                 | <b>830</b> |
| 28.1. Библиотека <code>php_mysql</code> .....                     | 830        |
| 28.1.1. Установка соединения с базой данных .....                 | 830        |
| 28.1.2. Выбор базы данных .....                                   | 833        |
| 28.1.3. Выполнение SQL-запросов .....                             | 835        |
| 28.1.4. Получение результатов запроса .....                       | 836        |
| 28.1.5. Количество строк в таблице .....                          | 844        |
| 28.1.6. Экранирование данных. SQL-инъекции .....                  | 846        |
| 28.2. Библиотека <code>php_mysqli</code> .....                    | 853        |
| 28.2.1. Создание базы данных .....                                | 860        |
| 28.2.2. Создание и заполнение таблицы .....                       | 861        |
| 28.2.3. Заполнение связанных таблиц .....                         | 862        |
| 28.2.4. Вывод данных .....  | 865        |
| 28.2.5. Повторное чтение результирующей таблицы .....             | 867        |
| 28.2.6. Количество строк в таблице .....                          | 868        |
| 28.2.7. Удаление данных .....                                     | 869        |
| 28.2.8. Сортировка .....  | 871        |
| 28.2.9. Параметризация SQL-запросов .....                         | 873        |
| <b>Глава 29. FTP-менеджер .....</b>                               | <b>875</b> |
| 29.1. Функции для работы с FTP-сервером .....                     | 875        |
| 29.2. Какой объем жесткого диска занимает сайт? .....             | 879        |
| 29.3. Перенос сайта с одного хоста на другой .....                | 881        |
| <b>Глава 30. Электронная почта .....</b>                          | <b>885</b> |
| 30.1. Отправка почтового сообщения .....                          | 885        |
| 30.2. Рассылка писем .....  | 887        |
| 30.3. Отправка писем с вложением .....                            | 888        |
| 30.4. Отправка писем со встроенными изображениями .....           | 891        |
| <b>Глава 31. Динамические изображения. Библиотека GDLib .....</b> | <b>894</b> |
| 31.1. Информационные функции .....                                | 894        |
| 31.1.1. Текущая версия библиотеки GDLib .....                     | 895        |
| 31.1.2. Формат файла .....  | 897        |
| 31.1.3. Размер файла .....  | 898        |
| 31.1.4. Получение MIME-типа файла .....                           | 899        |
| 31.2. Функции создания изображений .....                          | 901        |

|   |            |
|---|------------|
| 31.3. Функции сохранения и вывода изображений.....          | 903        |
| 31.4. Функции преобразования изображений.....               | 905        |
| 31.4.1. Создание уменьшенной копии изображения.....         | 906        |
| 31.4.2. Поворот изображения.....                            | 908        |
| 31.5. Функции для работы с цветом.....                      | 909        |
| 31.5.1. Заливка изображения цветом.....                     | 911        |
| 31.5.2. Получение цвета заданного пиксела.....              | 913        |
| 31.5.3. Уменьшение количества цветов в изображении.....     | 914        |
| 31.5.4. Замена одного цвета другим.....                     | 915        |
| 31.6. Функции рисования.....                                | 916        |
| 31.6.1. Кривая Безье.....                                   | 918        |
| 31.6.2. Построение гистограммы средствами GDLib.....        | 922        |
| 31.6.3. Построение круговой диаграммы средствами GDLib..... | 924        |
| 31.7. Функции настройки рисования.....                      | 925        |
| 31.7.1. Пунктирная линия.....                               | 926        |
| 31.8. Функции для работы с текстом.....                     | 928        |
| 31.8.1. Защитное изображение для HTML-формы.....            | 930        |
| <b>Заключение.....</b>                                      | <b>935</b> |

## **ПРИЛОЖЕНИЯ..... 937**

### **Приложение 1. Установка и настройка PHP, Web-сервера Apache и MySQL-сервера..... 939**

|  |     |
|--|-----|
| П1.1. Где взять дистрибутивы?.....                             | 939 |
| П1.1.1. Дистрибутив PHP.....                                   | 940 |
| П1.1.2. Дистрибутив Apache.....                                | 940 |
| П1.1.3. Дистрибутив MySQL.....                                 | 941 |
| П1.2. Установка Web-сервера Apache под Windows.....            | 942 |
| П1.3. Установка Web-сервера Apache под Linux.....              | 944 |
| П1.4. Настройка виртуальных хостов.....                        | 945 |
| П1.5. Настройка кодировки по умолчанию.....                    | 949 |
| П1.6. Управление запуском и остановкой Web-сервера Apache..... | 949 |
| П1.7. Управление Apache из командной строки.....               | 950 |
| П1.8. Установка PHP под Windows.....                           | 951 |
| П1.8.1. Установка PHP в качестве модуля.....                   | 951 |
| П1.8.2. Установка PHP как CGI-приложения.....                  | 952 |
| П1.9. Установка PHP под Linux.....                             | 954 |

### **Приложение 2. Установка MySQL..... 955**

|   |     |
|---|-----|
| П2.1. Установка MySQL под Windows.....        | 955 |
| П2.1.1. Процесс установки.....                | 955 |
| П2.1.2. Постинсталляционная настройка.....    | 960 |
| П2.1.3. Проверка работоспособности MySQL..... | 966 |

|   |            |
|---|------------|
| П2.2. Установка MySQL под Linux.....                      | 968        |
| П2.3. Конфигурационный файл .....                         | 971        |
| П2.4. Утилита mysql .....                                 | 974        |
| П2.4.1. Командная строка .....                            | 974        |
| П2.4.2. Работа с утилитой mysql в диалоговом режиме.....  | 978        |
| П2.5. Перенос баз данных с одного сервера на другой ..... | 982        |
| П2.5.1. Копирование бинарных файлов .....                 | 983        |
| П2.5.2. Создание SQL-дампа .....                          | 984        |
| <b>Рекомендуемая литература .....</b>                     | <b>987</b> |
| HTML, XML, CSS, JavaScript и Flash .....                  | 989        |
| PHP и Perl .....  | 991        |
| СУБД MySQL .....  | 993        |
| Интернет и Web-сервер Apache .....                        | 994        |
| Регулярные выражения .....                                | 995        |
| UNIX-подобные операционные системы.....                   | 995        |
| Методология программирования.....                         | 997        |
| <b>Предметный указатель.....</b>                          | <b>999</b> |



## Глава 2

# Быстрый старт

Несмотря на то, что PHP является универсальным языком программирования и может применяться для разработки практически любого программного обеспечения, основная его специализация — Web-разработка. Именно с этой целью он и проектировался, поэтому содержит множество инструментов для работы в Интернете.

### **ЗАМЕЧАНИЕ**

В данной и последующих главах будем исходить из факта, что интерпретатор PHP и Web-сервер установлены у вас на локальной машине и связаны таким образом, что скрипты, помещенные в файл с расширением `php`, обрабатываются интерпретатором, и по запросу к Web-серверу в браузере выводится результат работы. Если у вас не настроена связка Web-сервера и PHP, вы можете обратиться к *приложениям 1 и 2*, в которых детально описывается процесс развертывания Web-среды в операционных системах Windows и UNIX.

## 2.1. Скрипты

Язык программирования PHP считается скриптовым языком, поэтому программы, написанные на нем, называют *скриптами*. Главное отличие традиционных программ от скриптов заключается в том, что скрипты работают только в определенной среде и используют ресурсы данной среды. Программы, будь то компилируемые или интерпретируемые, не зависят от какого-то стороннего программного обеспечения. Например, скриптовый язык программирования JavaScript работает только в Web-браузерах, Visual Basic for Applications — только в среде Microsoft Office. С использованием этих языков программирования невозможно создать программу, работающую без соответствующей среды. В случае PHP в качестве такой среды выступает Web-окружение (Web-сервер, сервер базы данных, почтовый сервер и т. п.).

### **ЗАМЕЧАНИЕ**

Впрочем, язык программирования PHP допускает создание программ, работающих независимо от Web-сервера, однако в такой форме он не получил сколько бы то ни было широкого распространения.

Одной из главных особенностей языка программирования PHP является тот факт, что его код может располагаться вперемешку с HTML-кодом. Для того чтобы интерпретатор PHP различал HTML- и PHP-код, последний заключается в специальные теги `<?php` и `?>`, между которыми располагаются конструкции и операторы языка программирования



PHP. В листинге 2.1 приводится классический пример, выводящий в окно браузера при помощи конструкции `echo` фразу "Hello, world!" ("Привет, мир!"). Содержимое листинга 2.1 следует поместить в файл с расширением `php`, например в файл `index.php`. По умолчанию, файлы с другими расширениями не обрабатываются PHP-интерпретатором, и PHP-код остается необработанным. Впрочем, такое поведение Web-сервера можно изменить (см. приложение 1).

### ЗАМЕЧАНИЕ

`echo` — это конструкция языка программирования, позволяющая вывести одну или несколько строк в окно браузера. Более подробно конструкция `echo` обсуждается в разд. 5.2, посвященном строковым функциям.

#### Листинг 2.1. Простейший PHP-скрипт, `index.php`

```
<html>
<head>
  <title>Пример</title>
</head>
<body>
  <?php
    echo "Hello, world!";
  ?>
</body>
</html>
```

Результатом работы скрипта из листинга 2.1 в окно браузера будет выведена фраза "Hello, world!" (рис. 2.1).

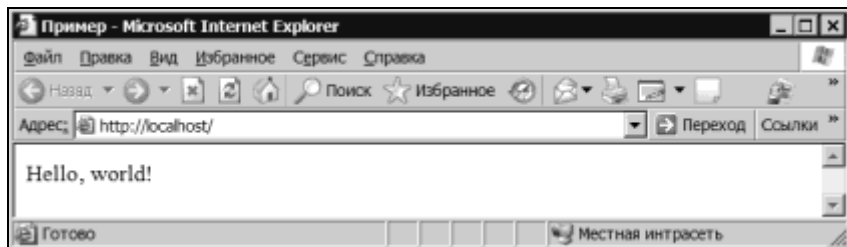


Рис. 2.1. Результат работы скрипта из листинга 2.1

При работе с серверными языками программирования, такими как PHP, следует помнить, что скрипты, расположенные между тегами `<?php` и `?>`, выполняются на сервере, клиенту приходит лишь результат работы PHP-кода, в чем можно легко убедиться, просмотрев исходный код HTML-страницы. В листинге 2.2 приводится результат работы PHP-скрипта из листинга 2.1, который браузер получает от сервера.

### ЗАМЕЧАНИЕ

Для просмотра исходного текста необходимо щелкнуть правой кнопкой мыши по странице и выбрать в контекстном меню соответствующий пункт, название которого зависит от браузера. В Internet Explorer необходимо выбрать пункт **Просмотр HTML-кода**, в Opera — **Исходный текст**, в FireFox — **View Page Source**.

#### Листинг 2.2. Исходный код HTML-страницы, которую получает браузер

```
<html>
  <head>
    <title>Пример</title>
  </head>
  <body>
    Hello, world! </body>
</html>
```

Если связка PHP и Web-сервера настроена неправильно или PHP-скрипт был размещен в файле с расширением, отличным от php, например, в файле index.html, то в исходном коде HTML-страницы вместо содержимого листинга 2.2 можно увидеть содержимое листинга 2.1. В этом случае все, что расположено между тегами `<?php` и `?>`, будет рассматриваться браузером как один большой неизвестный ему HTML-тег, а окно браузера будет пустым (рис. 2.2).

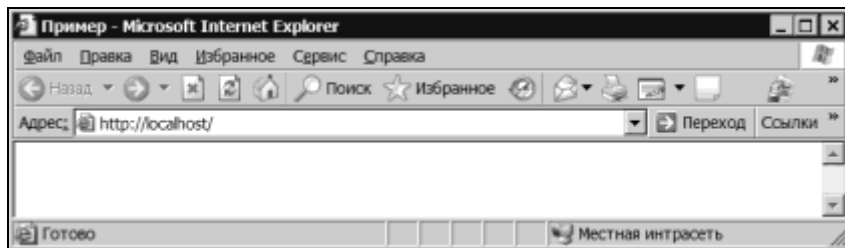


Рис. 2.2. PHP-код не подвергся интерпретации

## 2.2. Начальные и конечные теги

Как было указано в предыдущем разделе, PHP-скрипт должен быть размещен между начальным тегом `<?php` и конечным тегом `?>` для того, чтобы интерпретатор мог разделить HTML- и PHP-коды. Даже если HTML-код не используется, указание PHP-тегов является обязательным, в противном случае PHP-код будет выведен в окно браузера как есть, без интерпретации. Помимо тегов `<?php` и `?>` PHP поддерживает еще ряд альтернативных тегов, представленных в табл. 2.1.

**ЗАМЕЧАНИЕ**

До версии PHP 6.0 поддерживались начальные и конечные теги `<%` и `%>`, характерные для ASP-скриптов, включить которые можно было в конфигурационном файле `php.ini`, назначив директиве `asp_tags` значение `On` (см. главу 17). В версии PHP 6.0 данный тип тегов был исключен.

**Таблица 2.1. Варианты PHP-тегов**

| Вариант тегов  | Описание  |
|--|---|
| <code>&lt;?php ... ?&gt;</code>                                | Классический вариант PHP-тегов, именно его рекомендуется использовать в повседневной практике   |
| <code>&lt;? ... ?&gt;</code>                                   | Краткий вариант PHP-тегов, работает только в том случае, если в конфигурационном файле <code>php.ini</code> включена (т. е. принимает значение <code>On</code> ) директива <code>short_open_tag</code> . По умолчанию эта директива включена, однако все-таки лучше ориентироваться на полный вариант PHP-тегов <code>&lt;?php ... ?&gt;</code> . Последнее особенно актуально при использовании PHP совместно с XML-кодом, который использует теги <code>&lt;?xml ... ?&gt;</code> . При использовании коротких тегов может возникнуть неоднозначность в интерпретации |
| <code>&lt;script language="php"&gt; ... &lt;/script&gt;</code> | Расширенный вариант PHP-тегов, так же как и вариант <code>&lt;?php ... ?&gt;</code> доступный в любой момент без дополнительных настроек конфигурационного файла <code>php.ini</code> . Данный вариант несколько громоздок и распознается не всеми PHP-редакторами, тем не менее, следует быть готовыми встретить и такой вариант в PHP-скриптах  |
| <code>&lt;?= ... ?&gt;</code>                                  | Специальный вид тегов, предназначенных для вывода простого выражения в окно браузера. Конструкция <code>&lt;?= ... ?&gt;</code> эквивалентна <code>&lt;?php echo ... ?&gt;</code>   |

Отдельно следует отметить конструкцию `<?= ... ?>`, которую удобно использовать, когда скрипт состоит из одного выражения. Например, скрипт из листинга 2.1 можно было бы переписать так, как это продемонстрировано в листинге 2.3.

**Листинг 2.3. Использование тегов `<?= ... ?>`**

```
<html>
  <head>
    <title>Пример</title>
  </head>
  <body>
    <?= "Hello, world!" ?>
  </body>
</html>
```

Следует отметить, что HTML-страница может содержать более одной PHP-вставки. В листинге 2.4 приводится пример, который содержит две вставки: одна задает назва-

ние страницы (в HTML-теге `<title>`), а вторая определяет содержимое страницы (в HTML-теге `<body>`).

### ЗАМЕЧАНИЕ

В листинге 2.4 используется функция `date()`, которая возвращает текущую дату и время. Более подробно эта функция обсуждается в *главе 11*, посвященной функциям даты и времени.

#### Листинг 2.4. Допускается несколько PHP-вставок в HTML-код

```
<html>
  <head>
    <title><?php echo "Вывод текущей даты" ?></title>
  </head>
  <body>
    <?php
      echo "Текущая дата:<br>";
      echo date (DATE_RSS);
    ?>
  </body>
</html>
```

Результат работы скрипта из листинга 2.4 представлен на рис. 2.3.

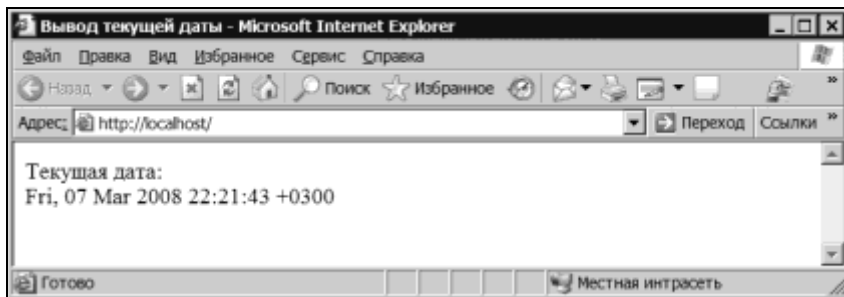


Рис. 2.3. Результат работы скрипта из листинга 2.4

## 2.3. Использование точки с запятой

Совокупность конструкций языка программирования, завершающуюся точкой с запятой, будем называть *выражением*.

Как видно из листинга 2.4, после строки "Вывод текущей даты" не указывается точка с запятой. Выражение одно, и надобности разделять его между другими выражениями

нет. Однако, как можно видеть во второй вставке, в конце каждой из конструкций `echo` имеется точка с запятой. Если забыть указать этот разделитель, интерпретатор языка программирования PHP посчитает выражение на новой строке продолжением предыдущего и не сможет корректно разобрать скрипт. В результате будет сгенерировано сообщение об ошибке "Parse error: syntax error, unexpected T\_ECHO, expecting ',' or ';'". ("Ошибка разбора: синтаксическая ошибка, неожиданно встречена конструкция `echo`, ожидается либо запятая ',', либо точка с запятой ';'").

Последнее выражение перед завершающим тегом `?>` может не снабжаться точкой с запятой не только в тегах `<?= ... ?>`, но и во всех остальных вариантах, рассмотренных в табл. 2.1. Например, в листинге 2.4 после выражения `echo date (DATE_RSS)` точку с запятой можно не указывать. Однако настоятельно рекомендуется не пользоваться этой особенностью и помещать точку с запятой после каждого выражения, т. к. добавление новых операторов может привести к появлению трудно улавливаемых ошибок.

Переводы строк никак не влияют на интерпретацию скрипта, выражение может быть разбито на несколько строк — интерпретатор PHP будет считать, что выражение закончено лишь после того, как обнаружит точку с запятой или завершающий тег `?>`. В листингах 2.5 и 2.6 представлены два скрипта, аналогичные по своей функциональности.

#### Листинг 2.5. Использование точки с запятой

```
<?php
    echo 5 + 5;
    echo 5 - 2;
    echo "Hello, world!";
?>
```

#### Листинг 2.6. Альтернативная запись скрипта из листинга 2.5

```
<?php
echo (5
    +
    5); echo (5-
    2); echo ("Hello, world!"
    );
?>
```

#### **ЗАМЕЧАНИЕ**

Следует избегать конструкций, подобной той, которая приведена в листинге 2.6. Понятно написанный код — залог успеха, т. к. такой код намного проще и приятнее отлаживать. А искусство программиста, как известно, состоит не в безошибочном написании кода (ошибки делают все), а в умении его отлаживать.

## 2.4. Составные выражения. Фигурные скобки

Фигурные скобки позволяют объединить несколько выражений в группу, которую обычно называют *составным выражением* (листинг 2.7).

### Листинг 2.7. Составное выражение

```
<?php
{
    echo 5 + 5;
    echo 5 - 2;
    echo "Hello, world!";
}
?>
```

Само по себе составное выражение практически никогда не используется, основное его предназначение — работа совместно с условными операторами, операторами цикла и т. п., которые подробно рассматриваются в *главе 5*.

Составное выражение может быть расположено в нескольких РНР-вставках. В листинге 2.8 приводится пример двух составных выражений, которые разбиты несколькими РНР-вставками. Задача скрипта сводится к случайному выводу в окно браузера либо слова "Истина" зеленого цвета, либо слова "Ложь" красного цвета. Без использования фигурных скобок оператор `if` распространял бы свое действие только на одно выражение, использование составного выражения позволяет распространить его действие на несколько простых выражений.

### ЗАМЕЧАНИЕ

Логический оператор `if` рассматривается в *главе 5*, а функция `rand()`, генерирующая случайное число из заданного интервала, — в *главе 12*.

### Листинг 2.8. Составное выражение может располагаться в нескольких РНР-вставках

```
<?php
if (rand(0,1))
{
    ?>
    <div style='color:green'>
    <?php
        echo "Истина";
    ?>
    </div>
    <?php
}
else
{
```

```

?>
<div style='color:red'>
<?php
    echo "Ложь";
?>
</div>
<?php
}
?>

```

Как видно из листинга 2.8, составное выражение в любой момент может быть прервано тегами `<?php` и `?>`, а затем продолжено. Впрочем, существуют исключения, например, составное выражение, применяемое для формирования класса, нельзя разбивать тегами `<?php` и `?>`. Возможно, в будущих версиях языка это ограничение будет устранено.

## 2.5. Комментарии

Код современных языков программирования является достаточно удобным для восприятия человеком по сравнению с машинными кодами, ассемблером или первыми языками программирования высокого уровня. Тем не менее, конструкции языка продиктованы архитектурой компьютера и, создавая программы, разработчик волевым образом использует компьютерную, а не человеческую логику. Это зачастую приводит к созданию достаточно сложных построений, которые нуждаются в объяснении на обычном языке. Для вставки таких пояснений в код предназначены *комментарии*.

PHP предоставляет несколько способов для вставки комментариев, варианты которых представлены в табл. 2.2.

**Таблица 2.2.** Варианты комментариев

| Комментарий                    | Описание  |
|--------------------------------|---|
| <code>// ...</code>            | Комментарий в стиле языка C++, начинающийся с символа двух слешей <code>//</code> и заканчивающийся переводом строки                        |
| <code># ...</code>             | Комментарий в стиле скриптовых языков UNIX, начинающийся с символа диэза <code>#</code> и заканчивающийся переводом строки                  |
| <code>/*<br/>...<br/>*/</code> | Если два предыдущих комментария ограничены лишь одной строкой, то комментарий в стиле языка C <code>/* ... */</code> является многострочным |

В листинге 2.9 демонстрируется использование всех трех видов комментариев из табл. 2.2.

### Листинг 2.9. Комментарии

```

<?php
    echo "Hello"; // это комментарий
    echo "Hello"; # и это комментарий

```

```

/*
    а это многострочный
    комментарий
*/
?>

```

Естественно, что комментарии PHP действуют только внутри тегов ограничителей `<?php ... ?>`. То есть если символы комментариев будут находиться вне тегов-ограничителей, то они, как и любой текст, будут отображены браузером (листинг 2.10).

#### Листинг 2.10. Комментарии действуют только внутри тегов `<?php` и `?>`

```

<?php
    echo "Hello<br>"; // нормальный комментарий
?>
// этот текст отобразится браузером.
<!-- Этот текст не будет отображен браузером, поскольку заключен между
символами, являющимися комментариями HTML. Однако он может быть просмотрен
в исходном коде HTML-страницы -->

```

Результат работы скрипта из листинга 2.10 представлен на рис. 2.4.

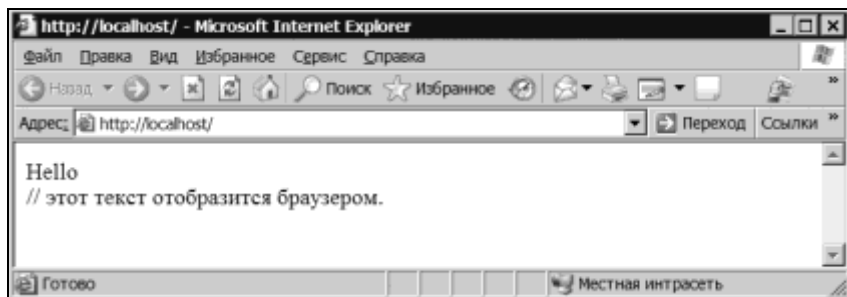


Рис. 2.4. Результат работы скрипта из листинга 2.10

Комментарии можно вставлять не только в конце выражения после точки с запятой, но и посередине. Так скрипт, представленный в листинге 2.11, является вполне корректным.

#### Листинг 2.11. Комментарий в списке аргументов функции

```

<?php
    echo strstr( // эту функцию мы рассмотрим позднее
                "Hello, world", "H");
?>

```