



# PHP – ЭТО ПРОСТО

## ПРОГРАММИРУЕМ ДЛЯ WEB-САЙТА

**ЧТО НУЖНО ЗНАТЬ О PHP**

**РАБОТА С PHP, APACHE, MySQL И phpMyAdmin**

**ПРАКТИЧЕСКИЕ ПРИМЕРЫ И ПОЛЕЗНЫЕ СОВЕТЫ**

**ГОТОВЫЕ РЕШЕНИЯ Mambo, phpBB, FCKEditor И CPanel**



УДК 681.3.06  
ББК 32.973.26-018.1  
Ш66

**Шкрыль А. А.**

Ш66 РНР — это просто. Программируем для Web-сайта. — СПб.: БХВ-Петербург, 2006. — 368 с.: ил.

ISBN 5-94157-905-5

Рассмотрены практические вопросы программирования на языке РНР и создания полноценных интерактивных Web-сайтов. На реальных примерах показаны особенности работы с APACHE, MySQL, phpMyAdmin и с популярными готовыми решениями — Mambo, phpBB, FCKEditor и CPanel. Материал сопровождается множеством иллюстраций, схем и полезных советов: начиная с использования общедоступных интернет-сервисов, таких как курс валют, и заканчивая настройкой Web-сервера APACHE.

*Для начинающих программистов*

УДК 681.3.06  
ББК 32.973.26-018.1

#### **Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Екатерина Капалыгина</i>
Компьютерная верстка	<i>Натальи Караваевой</i>
Корректор	<i>Зинаида Дмитриева</i>
Художник	<i>Елена Беляева</i>
Дизайн обложки	<i>Инны Тачиной</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 28.06.06.

Формат 70×100<sup>1</sup>/<sub>16</sub>. Печать офсетная. Усл. печ. л. 29,67.

Тираж 3000 экз. Заказ №  
"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию  
№ 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой  
по надзору в сфере защиты прав потребителей и благополучия человека.

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

ISBN 5-94157-905-5

© Шкрыль А. А., 2006  
© Оформление, издательство "БХВ-Петербург", 2006

# Оглавление

<b>Благодарности .....</b>	<b>1</b>
<b>Введение .....</b>	<b>3</b>
<b>Глава 1. История PHP .....</b>	<b>5</b>
<b>Глава 2. Что нужно знать о PHP .....</b>	<b>7</b>
2.1. Как работает PHP .....	7
2.2. Преимущества PHP .....	16
<b>Глава 3. Среда разработки PHP-программ .....</b>	<b>17</b>
3.1. PHP Expert Editor .....	18
3.2. PHP Designer 2006 .....	20
3.3. NOTEPAD++ .....	24
<b>Глава 4. Джентльменский набор Web-разработчика .....</b>	<b>29</b>
4.1. Установка Денвера .....	31
4.2. Работаем с Денвером .....	35
4.3. Денвер изнутри .....	39
4.4. Создаем свой сайт .....	39
4.5. Конфигурационные файлы .....	40
4.6. Информация о PHP .....	41
<b>Глава 5. Наши первые программы на PHP .....</b>	<b>45</b>
5.1. Программируем — начнем с простого .....	45
5.2. Переменные .....	54
5.3. Практикуемся в работе с переменными .....	57
5.4. Функции .....	62
5.5. Константы .....	65
5.6. Массивы .....	67
5.7. Переменные окружения .....	71



<b>Глава 6. Счетчик посещений .....</b>	<b>75</b>
6.1. Разработка программы .....	75
6.2. Счетчики, которые не надо разрабатывать .....	88
6.3. Информеры .....	91
6.3.1. Информеры от Rambler .....	91
6.3.2. Курс валют .....	93
6.3.3. Цены на российские автомобили .....	95
 <b>Глава 7. Все, что нужно знать о формах .....</b>	 <b>97</b>
7.1. Назначение форм .....	97
7.2. Создание формы .....	98
7.3. Простые элементы формы: поле ввода и кнопка .....	101
7.4. Немного практики .....	105
7.5. Методы отправки данных формы .....	108
7.5.1. Метод GET .....	109
7.5.2. Метод POST .....	110
7.5.3. Что лучше: GET или POST? .....	111
7.6. Остальные элементы формы .....	112
7.6.1. Поле для ввода пароля .....	112
7.6.2. Переключатель (Radio button) .....	113
7.6.3. Флаг (CheckBox) .....	114
7.6.4. Список .....	115
7.6.5. Поле ввода многострочного текста (TextArea) .....	118
7.6.6. Скрытое поле .....	119
 <b>Глава 8. Что скрывает браузер .....</b>	 <b>121</b>
 <b>Глава 9. Сплошная практика .....</b>	 <b>127</b>
9.1. Форма обратной связи .....	127
9.2. Гостевая книга .....	135
9.2.1. Приступаем к работе .....	136
9.2.2. Дорабатываем гостевую книгу .....	151
9.2.3. Цензура не дремлет .....	155
9.2.4. Управляем с удобством — админка .....	160
9.2.5. Использование сессий .....	168
9.3. Голосование .....	175
9.3.1. Приступаем к работе .....	175
9.3.2. Использование cookie .....	183
9.3.3. Админка для системы голосования .....	190
9.4. Загрузка файлов .....	198
9.4.1. Основы .....	198
9.4.2. Познаем тонкости .....	208
9.4.3. Полноценный скрипт для загрузки файла .....	209
9.5. Определяем быстродействие скрипта .....	212



<b>Глава 10. Базы данных и работа с MySQL</b> .....	<b>221</b>
10.1. Основные понятия .....	221
10.2. phpMyAdmin — первое знакомство .....	224
10.3. Разрабатываем структуру будущей базы данных .....	224
10.4. Создаем БД или работаем в phpMyAdmin .....	229
10.5. Разрабатываем план кодирования .....	241
10.6. SQL-запросы .....	242
10.7. Кодирование .....	245
10.7.1. Подключаемся к базе .....	246
10.7.2. Модуль авторизации .....	248
10.7.3. Модуль logout.php .....	258
10.7.4. Основной файл форума index.php .....	258
10.7.5. Модуль вывода информации show.php .....	260
10.7.6. Модуль действий action.php .....	276
10.8. В заключение главы .....	283
<b>Глава 11. FCKEditor</b> .....	<b>285</b>
11.1. Установка .....	285
11.2. Первое знакомство .....	285
11.3. Простой пример .....	287
11.4. Настраиваем панели инструментов .....	289
11.5. Получаем данные из редактора .....	291
11.6. Настраиваем инструмент по загрузке файлов .....	293
<b>Глава 12. phpBB</b> .....	<b>297</b>
12.1. Установка .....	297
12.2. Работаем с форумом .....	300
<b>Глава 13. Знакомство с Mambo</b> .....	<b>305</b>
13.1. Установка .....	306
13.2. Основные принципы работы с Mambo .....	311
13.3. Разрабатываем свой сайт .....	313
13.4. Устанавливаем модуль для Mambo .....	329
13.5. Устанавливаем компонент для Mambo .....	330
13.6. Устанавливаем шаблоны для Mambo .....	334
13.7. В заключение .....	334
<b>Глава 14. Закачиваем сайт на хостинг</b> .....	<b>335</b>
14.1. Бесплатный хостинг от Holm.ru .....	335
14.2. cPanel .....	340
<b>Заключение</b> .....	<b>347</b>
<b>Приложение 1. Настройка PHP</b> .....	<b>349</b>
<b>Приложение 2. Список сайтов, связанных с PHP</b> .....	<b>353</b>
<b>Предметный указатель</b> .....	<b>355</b>

## Глава 2

# Что нужно знать о РНР

Пожалуй, РНР — один из немногих языков, который удобнее изучать, реализуя конкретные примеры. Именно так мы и будем поступать на протяжении всей книги. На мой взгляд, это один из самых удобных и практичных способов освоить РНР, потому что данный язык тем и хорош, что позволяет достаточно быстро решать поставленные перед ним задачи, даже если человек, который этим занимается, обладает не высокой квалификацией.

Как я уже говорил ранее, при изучении РНР достаточно обладать хотя бы базовыми знаниями HTML. Если их у вас нет, это тоже не проблема, т. к. достаточно зайти в Интернет и ввести в любом поисковике фразу "Учебник по HTML". Могу сказать точно, вам будет из чего выбрать.

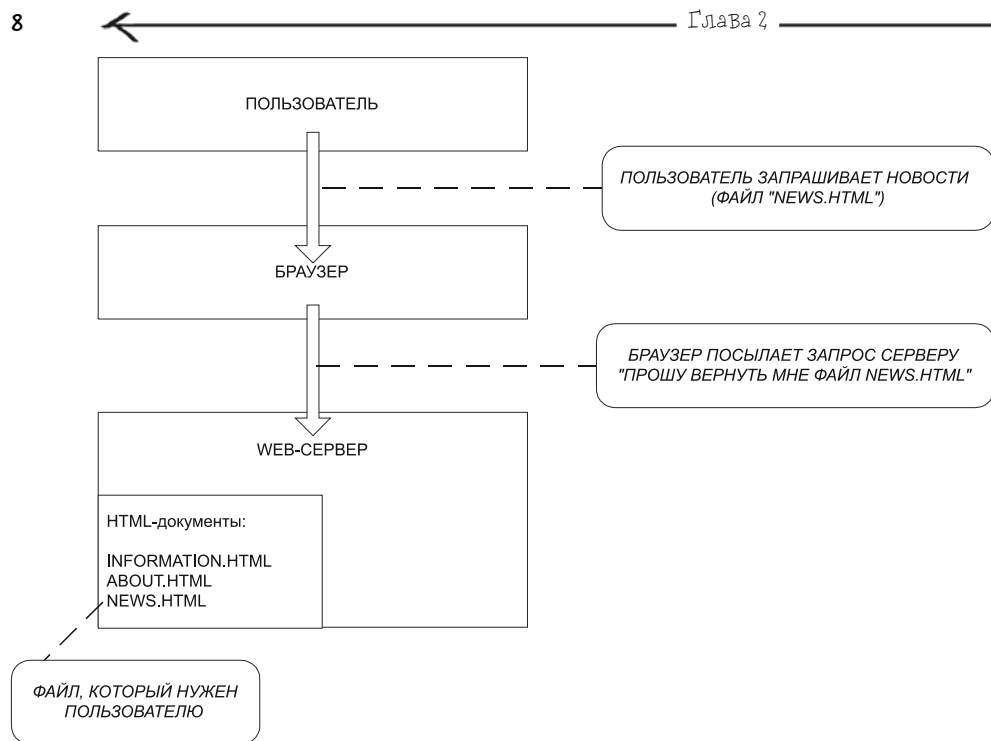
## 2.1. Как работает РНР

Для того чтобы понять, каким образом работает РНР, предлагаю обратиться к самому простейшему примеру, когда пользователь запрашивает обычную HTML-страничку (рис. 2.1).

То есть сначала пользователь запрашивает определенный HTML-документ, предположим, он хочет получить файл NEWS.HTML, чтобы просмотреть последние новости. Для этого, как правило, используется специальная программа под названием Web-браузер (например, Internet Explorer, Opera, Mozilla и др.), наш пользователь не исключение, он тоже использует эту программу. Соответственно он запрашивает файл



*В мире Web много различной терминологии, что поначалу может запутать человека, который попадает в этот мир впервые. Если вы предполагаете писать только на РНР, вы все равно столкнетесь со многими терминами, напрямую к РНР не относящимися, но являющимися неотъемлемой частью мира Web. Поэтому если на страницах книги вы столкнулись с каким-то новым, неизвестным для вас понятием, то советую воспользоваться любым поисковым сервисом, введя в него фразу "Толковый словарь Web-терминов", что позволит более быстро и точно понять материал.*



**Рис. 2.1.** Схема обработки запроса пользователя по получению файла NEWS.HTML

NEWS.HTML у определенного сервера, назовем его просто СЕРВЕР, с помощью своего Web-браузера.

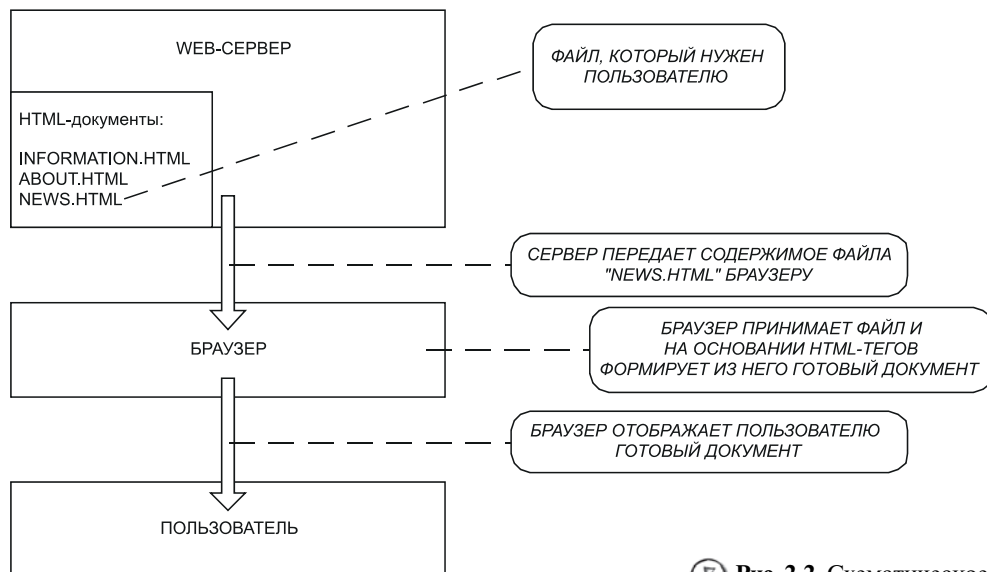
Web-браузер, получив команду от пользователя, посылает запрос на сервер, в нашем случае это будет приблизительно следующее: "ПРОШУ ВЕРНУТЬ МНЕ ФАЙЛ NEWS.HTML".

На рис. 2.2 в виде схемы изображен процесс передачи сервером браузеру запрошенного файла и соответственно получение той информации пользователем, которую он хотел увидеть.

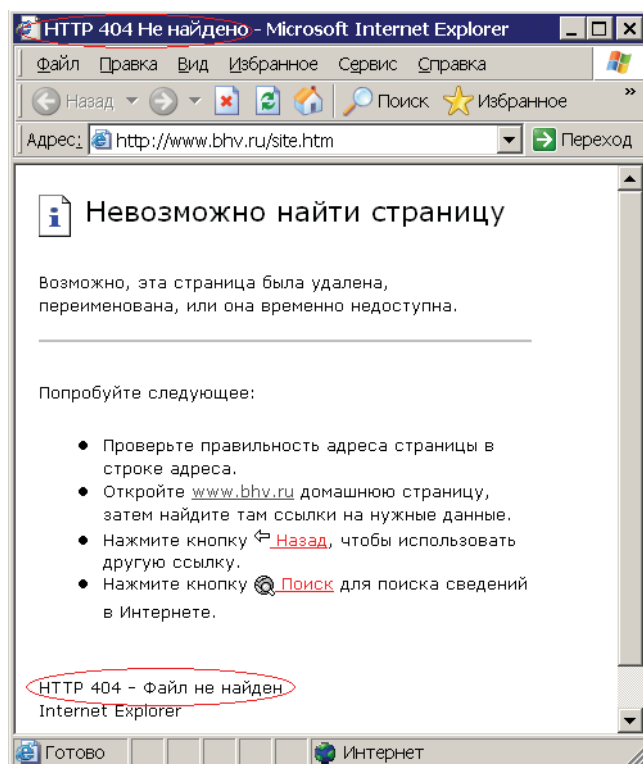
Сервер, получив запрос от браузера, начинает его обрабатывать. Сначала он проверяет — есть ли у него такой файл, если да (мы рассматриваем именно этот случай), то передает его браузеру, если нет, то возвращает *ошибку 404*, которая свидетельствует о том, что файл не найден. На рис. 2.3 вы можете увидеть окно **Internet Explorer**, когда при попытке обратиться к файлу обнаружилось, что запрашиваемого файла на сервере нет.

Браузер, получив файл, на основании HTML-тегов, которые содержатся в нем, формирует готовый документ, удобный для просмотра пользователя.

Здесь очень удобно провести аналогию с книгой. У книги есть определенные правила оформления, такие как наличие оглавления, названия книги и издательства, которое выпустило книгу, имя автора, нумерация страниц, что позволяет читателю удобно работать с каждой книгой. Так же и HTML (язык гиперразметки документов) позволяет задавать структуру документа на основании общих принципов,



**Рис. 2.2.** Схематическое представление процесса, когда после запроса пользователя сервер возвращает нужный ему файл



**Рис. 2.3.** Ошибка 404 возникает, когда запрашиваемый файл не найден. Фрагменты, отображающие код ошибки, обведены овалами



положенных в его основу. Если правила оформления книги важны для удобства работы читателя с книгой, то HTML-разметка в первую очередь рассчитана на браузер, который в соответствии с тегами формирует готовый документ, который уже в свою очередь получается удобным для просмотра пользователем. Сам по себе HTML-файл со всеми своими тегами будет очень не удобен для работы с ним, а вот когда браузер в соответствии с этими тегами оформляет документ, то мы получаем информацию в удобном для работы виде. На рис. 2.4 изображен HTML-файл, открытый в Блокноте, на рис. 2.5 HTML-файл, открытый в браузере Internet Explorer (на самом деле открытый в браузере и обработанный согласно тегам, размещенным в файле).

Теперь рассмотрим описанный ранее процесс еще раз, но предположим, что пользователь запрашивает файл, который является программой, написанной на PHP.

PHP является языком серверных скриптов (или сценариев), который можно встраивать в обычный HTML-документ. В главе 1 уже упоминалось, что полное название PHP — это "PHP: Hypertext Preprocessor" (PHP: Препроцессор гипертекста). Так вот именно этот препроцессор в лице файла php.exe и обрабатывает программы (скрипты или сценарии), написанные на PHP. Предлагаю обратить внимание на рис. 2.6 и 2.7.



*Вы можете получить аналогичный рис. 2.4 результат, если в Internet Explorer откроете любой сайт, а затем выберете пункт меню Вид | Просмотр HTML-кода.*

```
www.php[1] - Блокнот
Файл  Правка  Формат  Вид  Справка

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>PHP: Hypertext Preprocessor</title>
<link rel="stylesheet" href="http://static.php.net/www.php.net/style.css" />
<link rel="stylesheet" href="http://static.php.net/www.php.net/styles/phpnet.css" />
<link rel="shortcut icon" href="http://static.php.net/www.php.net/favicon.ico" />
<link rel="alternate" type="application/rss+xml" title="PHP: Hypertext Preprocessor"
href="http://www.php.net/news.rss" />

<script language="JavaScript">
<!--
function SymError()
{
    return true;
}

window.onerror = SymError;

var SymRealWinOpen = window.open;
```

**Рис. 2.4.** HTML-файл, открытый в Блокноте (главная страница сайта [www.php.net](http://www.php.net))

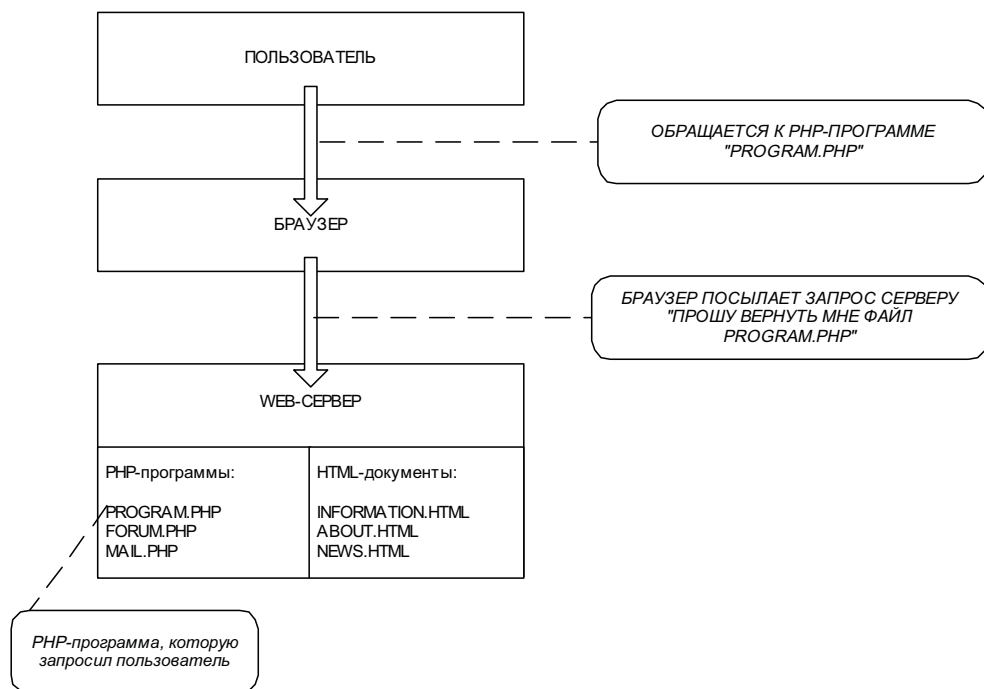


Рис. 2.5. HTML-файл, открытый в браузере Internet Explorer. По сравнению с предыдущим рисунком, он подвергся обработке браузером и поэтому представлен в удобном для восприятия виде

На рис. 2.6 представлена ситуация, аналогичная запросу HTML-файла, которую мы рассматривали ранее (см. рис. 2.1), главное отличие заключается в том, что сейчас пользователь запрашивает не HTML-файл, а PHP-программу и то, что на сервере теперь хранятся эти самые PHP-программы. Давайте теперь посмотрим, что происходит, после того как сервер получил запрос от пользователя (рис. 2.7).

То есть когда сервер получает запрос, он также проверяет, есть ли такой файл? Если да, то далее он обращается к файлу настроек Web-сервера, где сказано, что все файлы с расширением PHP являются программами и их нужно обрабатывать препроцессором гипертекста, прежде чем возвращать пользователю. Расширение PHP является самым распространенным, но есть также и другие (альтернативные) варианты расширений для программ, написанных на PHP:

- PHP2 — программа, написанная на PHP второй версии (например, PROGRAM.PHP2);
- PHP3 — программа, написанная на PHP третьей версии (например, PROGRAM.PHP3);
- PHP4 — программа, написанная на PHP четвертой версии (например, PROGRAM.PHP4);



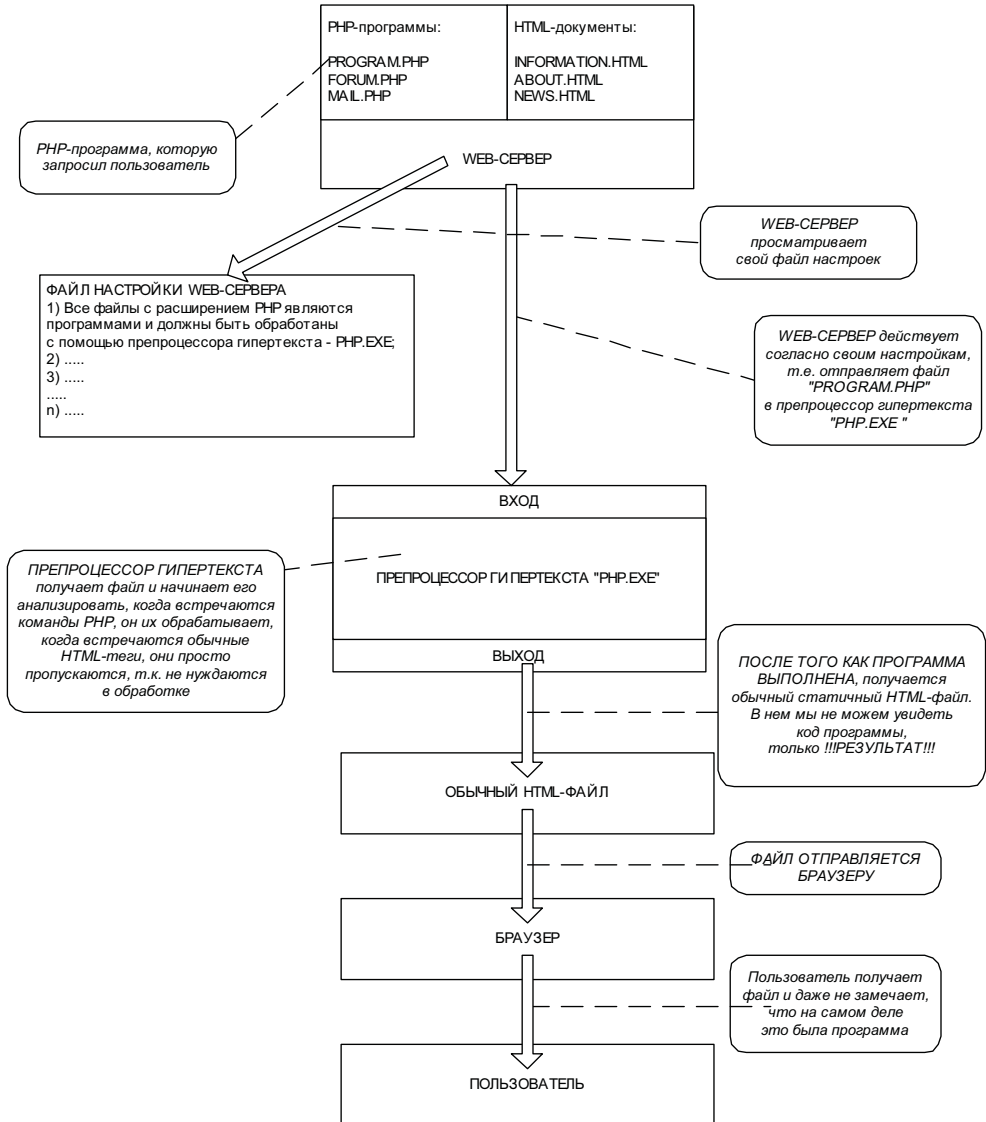
**Рис. 2.6.** Схематическое изображение процесса обращения пользователя к PHP-программе



Вообще вы можете настроить свой Web-сервер так, чтобы файлы с самыми разнообразными расширениями, например TXT, обрабатывались как PHP-программы — только в этом нет особого смысла.

- PHP5 — программа, написанная на PHP пятой версии (например, PROGRAM.PHP5);
- PHTML и PHTM — менее распространенный вариант (например, PROGRAM.PHTML или PROGRAM.PHTM).

Таким образом, файл PROGRAM.PHP передается препроцессору, на выходе которого получается обычный HTML-файл, он передается браузеру и пользователь получает статический HTML-документ — то есть результат выполнения PHP-программы.



**Рис. 2.7.** Схематическое изображение процесса получения пользователем результата выполнения PHP-программы



Я специально упростил модель обработки PHP-программы и назвал элемент, который ее обрабатывает, препроцессором. Во многих изданиях PHP называют интерпретатором или компилятором, но на самом деле все немного сложнее. Он полностью не является ни тем, ни другим. Это может достаточно сильно запутать человека, который только знакомится с PHP. Таким образом, намного удобнее называть его препроцессором гипертекста, который обрабатывает PHP-программы и на выходе имеет обычный HTML-документ.



Рекомендуется использовать `<?php...?>`, этот способ наиболее распространен и предпочтителен. Далее я буду предполагать, что вы используете первый вариант оформления.

Рассмотрим работу препроцессора. Обратите внимание на рис. 2.8, на котором изображен процесс обработки препроцессором программы, написанной на языке PHP.

Как же препроцессор узнает о том, что это именно код на языке PHP, который нужно обработать, а не обычные теги HTML. Условие того, что файл имеет расширение PHP, является не достаточным. Все очень просто — код на PHP оформляется специальными тегами, вариантов оформления несколько:

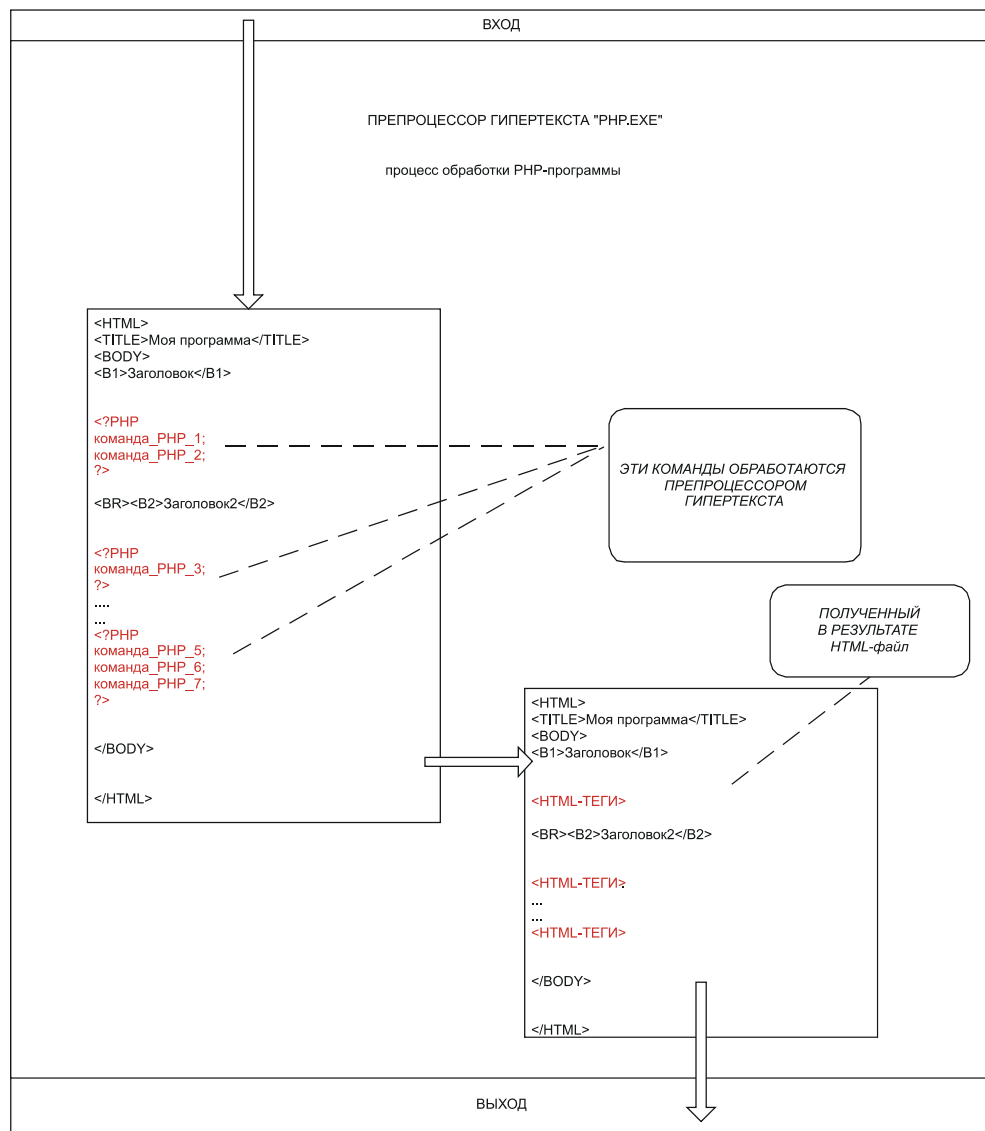
- `<?php...?>`
- `<?...?>`
- `<%...%>`
- `<script language="php">...</script>`

Рассмотрим работу препроцессора более подробно:

- ① После того как на вход препроцессора поступает файл, он начинает "просматривать" (или обрабатывать) его содержимое.
- ② Как только ему встречается открывающий тег, обозначающий код PHP, тут препроцессор переключается в активный режим, т. е. начинается выполнение команд, которые заключены в эти специальные теги. Некоторые команды после обработки будут заменены на HTML-теги, некоторые (например, осуществляющие подключение к базе данных) будут просто выполнены.
- ③ Как только встречается закрывающий тег, обозначающий окончание кода на PHP, препроцессор опять переключается в режим просмотра кода, пока ему не встретится очередной участок кода на PHP или пока не будет достигнут конец файла.

Как частный случай, можно рассмотреть пример, когда файл полностью состоит из команд PHP, тогда препроцессор от начала до конца обрабатывает файл. Но такой подход применяется очень редко, чаще всего PHP-команды совместно используются с HTML-тегами, а можно сказать наоборот, HTML-теги используются с PHP-командами.

Положительным моментом обработки PHP-программы препроцессором является то, что пользователь не видит исходный код программы, он только получает результат ее выполнения.



**Рис. 2.8.** Схема обработки PHP-программы препроцессором гипертекста

## 2.2. Преимущества PHP

Фрагменты PHP-кода можно внедрять прямо в HTML-документ и наоборот. Главное при этом не забыть установить для файла расширение PHP.

Так как PHP специально разработан для Web, многие вещи реализуются намного проще, чем если бы вы то же самое выполняли с помощью других языков.

PHP очень прост в освоении и позволяет достаточно быстро решать поставленные перед ним задачи. То есть уже в течение нескольких минут вы сможете написать свою первую программу, а через несколько часов вы сможете сделать уже что-то более-менее серьезное, например, разработать собственную гостевую книгу.

Если вы знакомы хотя бы с одним языком программирования, то освоить PHP вам будет очень легко. Если нет, то это тоже будет сделать несложно, просто на это уйдет немного больше времени.

PHP бесплатен.

PHP — кроссплатформенный язык, что не привязывает его к определенной операционной системе или определенному Web-серверу. В результате мы получаем огромную свободу в выборе программного обеспечения при работе с PHP.

Совместимость программ, разработанных в ранних версиях PHP, с его более поздними версиями.

PHP очень гибок. Он не зависит от браузера, т. к. программа выполняется на сервере, а результат возвращается в виде HTML-документа.

Простота работы с базами данных: PHP может работать с различными СУБД, например, MySQL, MS SQL Server, ORACLE, InterBase, Firebird и др. Причем работа может быть организована с использованием различных технологий доступа к данным.

PHP является объектно-ориентированным языком, что позволяет вам воспользоваться всеми преимуществами этой технологии.

PHP располагает огромным количеством стандартных функций, которые позволяют вам решить множество задач за минимальное количество времени.

PHP многогранен, вам предоставляется множество готовых инструментов для вашего творчества, например, вы запросто можете работать с файлами формата PDF, электронной почтой или даже разработать интернет-магазин, который будет использовать средства электронной коммерции, такие как WebMoney или PayPal.

PHP может работать с XML.

В настоящее время в Интернете есть огромное количество готовых решений, написанных на PHP, начиная от простых счетчиков на вашу страницу и заканчивая системами управления и построения сайтов, которые позволяют до максимума упростить процесс создания и сопровождения сайта, поэтому, даже не будучи гуру в PHP, вы сможете создать качественный интерактивный сайт.

Наконец, вы можете принять участие в совершенствовании PHP, т. к. его код открыт для всех.

## Глава 3

# Среда разработки PHP-программ

Любую программу удобно разрабатывать в специальной среде, конечно, можно писать и в Блокноте, но недостатки этого выбора сразу налицо: во-первых, синтаксис языка не подсвечивается, во-вторых, нет ни удобной справки, ни подсказок, ни возможностей автоматически вставлять наиболее распространенные команды или фрагменты кода, ни многих других, которые не только помогают сократить время разработки программы, но и сделать этот процесс максимально удобным и комфортным.

В настоящее время существует огромное количество редакторов, в которых вы можете создавать и редактировать файлы PHP. В эту главу вошли три программы, которыми можно пользоваться сразу после элементарного процесса установки, и самое главное они бесплатные, правда, одну из них — PHP Expert Editor — нужно зарегистрировать, но это абсолютно бесплатно, лишь сделать пару дополнительных действий. Данная глава не претендует на исчерпывающее руководство по их использованию, она призвана помочь читателю комфортно разрабатывать свои первые программы на PHP.

Очень сложно дать ответ на вопрос, какой из редакторов лучше всего использовать, т. к. это личное предпочтение каждого. Наподобие того, как каждому человеку нравятся разные машины, так же и редакторы: у различных людей разные предпочтения. Для того чтобы убедиться в этом, достаточно зайти на любой более менее серьезный форум, посвященный PHP, и посмотреть топики, посвященные обсуждению редакторов. Такая тема встречается практически на каждом форуме, если же у вас не получается найти ее,



*Блокнот рекомендуется использовать лишь в крайнем случае, например, когда нужно внести изменения в код очень срочно, а других средств под рукой нет.*



*Часто можно встретить синоним термина "редактор" — IDE (Integrated Development Environment, интегрированная среда разработки).*





то всегда можно создать свою собственную. Вы удивитесь, насколько разные предпочтения у людей.

К тому же вы можете посетить следующие два сайта, на которых вы найдете информацию по самым разнообразным редакторам:

☛ <http://www.thelinuxconsultancy.co.uk/phpeditors.php>

☛ <http://www.php-editors.com/review/>

### 3.1. PHP Expert Editor



**Разработчик:** Ankord Development Group.

**Официальный сайт:** <http://www.ankord.com/> (русскаяязычный вариант доступен по адресу <http://www.ankord.com/ru>).

**Размер дистрибутива:** 3,11 Мбайт.

**Платформа:** Windows 98/ME/NT/2000/XP.

**Минимальные требования к компьютеру:** процессор — Pentium100, оперативная память — 64 Мбайт, свободного пространства на винчестере — 4 Мбайт.

**Дополнительная информация:** программа нуждается в регистрации (незарегистрированную копию можно использовать в течение 30 дней). Для регистрации достаточно зайти по адресу [http://www.ankord.com/ru/phpxedit\\_reg.html](http://www.ankord.com/ru/phpxedit_reg.html) и заполнить регистрационную форму, в течение нескольких дней вы получите по электронной почте регистрационный ключ.

PHP Expert Editor — удобный в использовании PHP-редактор или IDE (Integrated Development Environment, интегрированная среда разработки), созданный специально для PHP-мастеров. Будет удобен как начинающим, так и профессиональным PHP-программистам. Имеет встроенный HTTP-сервер и отладчик для запуска и отладки PHP-скриптов (вы можете использовать также внешний HTTP-сервер, на мой взгляд, это даже удобнее). Предоставляет возможность проверки синтаксиса PHP, имеет встроенный браузер, FTP-клиент, файл-эксплорер, настраиваемые шаблоны кода, три режима подсветки кода (PHP & HTML, HTML only, PHP only), существует подсветка Java Script и CSS-файлов, функции быстрой навигации в PHP-коде. Поддерживает следующие форматы файлов: Windows, UNIX, Mac.

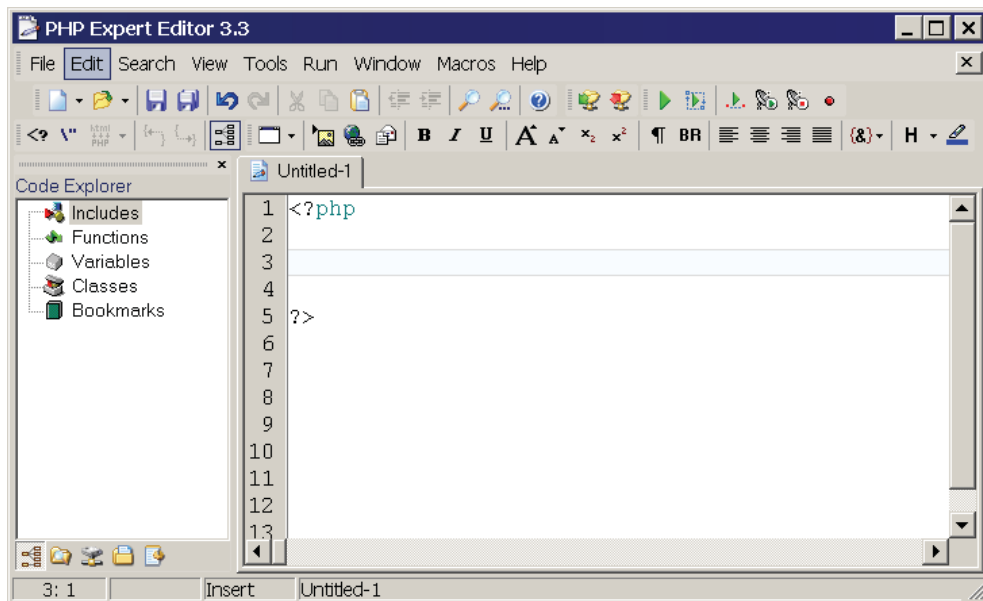


Рис. 3.1. Окно PHP Expert Editor 3.3



Рис. 3.2. Окно About PHP Expert Editor

К дополнительным возможностям можно отнести быструю вставку всех функций PHP с отображением подсказки и их параметров; подсветку парных символов (скобок, кавычек); два стиля интерфейса — Classic и Office XP.

На рис. 3.1 и 3.2 вы можете увидеть окно программы **PHP Expert Editor 3.3** и окно **About PHP Expert Editor**.



### Несколько советов, которые помогут при работе с PHP Expert Editor.

☛ Очень удобно использовать **Code Templates** (Шаблоны кода). Есть набор predefined шаблонов, но вы также можете создавать свои собственные. Предположим, вам надо будет вставить условную конструкцию **IF-Else**, вы просто нажимаете **<Ctrl>+<J>** и выбираете ее из появившегося окошка. Для того чтобы создать свой собственный шаблон, нужно выбрать пункт меню **Tools | Code Templates**.

☛ Для того чтобы вызвать окно настройки программы — **Properties**, необходимо выбрать пункт меню **View | Editor Options**.

☛ Для того чтобы включить функцию автосохранения, необходимо в окне **Properties** перейти на вкладку **Advanced** и установить флаг **Auto save every**, после этого в поле, расположенном напротив, указать в минутах значение, которое будет указывать, через какой промежуток времени будет выполняться автоматическое сохранение файла, с которым вы работаете.

☛ Для смены стиля интерфейса достаточно выбрать пункт меню **View | Interface Style**, а затем выбрать необходимый вам стиль.

## 3.2. PHP Designer 2006



Разработчик: MP SOFTWARE.

Официальный сайт: <http://www.mpsoftware.dk/phpdesigner.php>.

Размер дистрибутива: 1,7 Мбайт.

Платформа: Windows 98/ME/NT/2000/XP.

Минимальные требования к компьютеру: оперативная память — 32 Мбайт; свободного пространства на винчестере — 3,5 Мбайт.

Не важно, являетесь ли вы профессионалом или новичком в PHP. Бесплатный редактор PHP Designer 2006 обеспечит вас полным набором инструментов, которые необходимы для разработки программ на PHP.

PHP Designer 2006 позволяет подсвечивать код для PHP, HTML, XHTML, CSS, Perl, C#, JavaScript, VB, Java и SQL (Ingres, InterBase, MSSQL, MySQL, Oracle, SyBase и Standard SQL), что очень удобно, т. к. вы можете использовать его не только при разработке PHP-программ, но и в других ситуациях. Например, при построении SQL-запросов к различным СУБД. Этот редактор очень универсален, так сказать "все в одном".

Он содержит библиотеку, разделенную по категориям, для большинства команд, относящихся к PHP, HTML/XHTML, CSS, SQL. Имеет встроенный FTP-клиент, браузер и файл-эксплорер. Содержит встроенную систему напоминания ToDo и менеджер проектов. Обладает настраиваемым интерфейсом и содержит более 10 встроенных тем оформления. Оболочка PHP Designer очень удобна и практична.

На рис. 3.3 и 3.4 вы можете увидеть окно программы **PHP Designer 2006** и окно **About**.

Рис. 3.3. Окно PHP Designer 2006

