Максим Кузнецов
Игорь Симдянов

ГОЛОВОЛОМКИ на PHP для ХАКЕРА

2-е издание

Санкт-Петербург
«БХВ-Петербург»
2008

УДК 681.3.06
ББК 32.973.26-018.2
К89

Кузнецов, М. В.

К89 Головоломки на PHP для хакера / М. В. Кузнецов, И. В. Симдянов. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2008. — 544 с.: ил. + CD-ROM

ISBN 978-5-9775-0204-7

Книга представляет собой задачник по Web-технологиям с уклоном в защиту Web-приложений от злоумышленников. Цель книги — помочь Web-разработчику научиться самостоятельно обнаруживать и устранять уязвимости в своем коде.

Главы второго издания существенно обновлены, кроме этого написаны две новые главы, посвященные динамическому формированию изображений и объектно-ориентированному программированию.

На компакт-диске, поставляемом вместе с книгой, приведены скрипты, являющиеся ответами на предлагаемые задачи.

Для программистов и Web-разработчиков

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Евгений Рыбаков*
Зав. редакцией	*Григорий Добин*
Редактор	*Ирина Иноземцева*
Компьютерная верстка	*Ольги Сергиенко*
Корректор	*Зинаида Дмитриева*
Оформление обложки	*Елены Беляевой*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 19.12.07.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 43,86.
Тираж 2500 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 978-5-9775-0204-7

© Кузнецов М. В., Симдянов И. В., 2008
© Оформление, издательство "БХВ-Петербург", 2008

Оглавление

Введение

Предлагаемая книга является сборником задач по PHP с уклоном в защиту сайта и Web-приложений от злоумышленников.

Основная проблема создателей сайтов заключается в том, что они мыслят совсем другими категориями, нежели злоумышленники. Кроме того, Web-разработчики редко прибегают к тестированию своей продукции на предмет уязвимости, так как им подсознательно не хочется ломать свои собственные Web-приложения. Снять такой настрой поможет эта книга, где, наряду с задачами по защите Web-приложений, будет предложено большое количество задач по взлому сайта с применением самых различных технологий: от межсайтового скриптинга и SQL-инъекций до подбора паролей при помощи словаря. Это позволит читателю убедиться в том, как легко может быть нарушена работа Web-сайта и как дорого может обернуться беспечность при его разработке.

Наряду с деструктивными задачами будет предложено большое количество заданий, направленных на построение обороны сайта. Выполнив задания, вы получите в руки мощную систему защиты собственного сайта, которая будет отличаться от коммерческих и свободных аналогов тем, что вы будете знать в ней каждый винтик и сможете легко модернизировать ее, быстро устранять последствия взлома и находить уязвимости.

Книга разбита на две части: непосредственно задачник и ответы на задачи. Вы можете решать все задачи последовательно или, если вам необходимо срочно защитить свой сайт, можете воспользоваться готовыми кодами, находящимися на прилагаемом к книге компакт-диске.

Все главы книги подверглись переработке по сравнению с первым изданием, кроме того, в книге появились две дополнительные главы, посвященные динамическому формированию изображений при помощи библиотеки GDLib и объектно-ориентированному программированию.

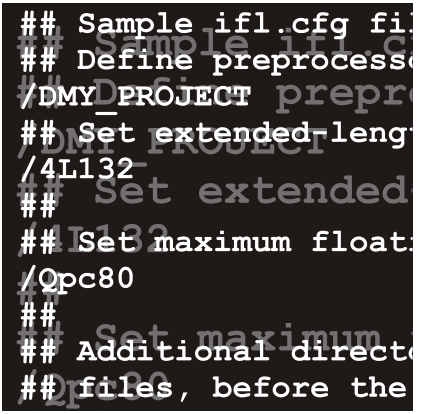
Дополнительные материалы можно также найти на группе сайтов IT-студии SoftTime, сотрудниками которой являются авторы книги:

- **http://www.softtime.ru** — главный сайт;
- **http://www.softtime.org** — проекты студии;
- **http://www.softtime.biz** — услуги студии;
- **http://www.softtime.mobi** — вариант портала для мобильных устройств.

В частности, исходные коды к книге можно найти по адресу **http://www.softtime.ru/security/**, обсудить вопросы, которые могут возникнуть по мере чтения материала книги, можно на форуме авторов **http://www.softtime.ru/forum/**. На странице **http://www.softtime.ru/info/task.php** доступен постоянно пополняющийся раздел задач (с ответами), не вошедших в состав книги.

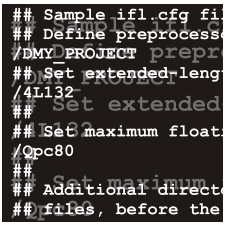
Благодарности

Авторы выражают признательность сотрудникам издательства "БХВ-Петербург", благодаря которым эта рукопись увидела свет, а также посетителям форума **http://www.softtime.ru/forum/** за интересные вопросы и конструктивное обсуждение.



ЧАСТЬ I

Задачи



ГЛАВА I.1

Строки и числа

Работа со строками составляет основу любого программирования. Виртуозное манипулирование строками позволит программисту создавать более короткие и эффективные программы. Исследования показали, что плотность ошибок в программах не зависит от языка программирования, а зависит только от квалификации программиста. Чем короче будут программы, тем меньше ошибок и уязвимостей в них будет. Хорошее знание особенностей строк позволяет безошибочно определять возможные проблемные с точки зрения безопасности места в коде. Данная глава содержит задачи на знание строковых функций PHP и умение обращаться с ними.

Замечание

Все примеры из данной главы можно найти в каталоге scripts\1 компакт-диска, поставляемого вместе с книгой.

I.1.1. Количество и имена файлов в произвольном каталоге

Определите количество и имена файлов в каталоге, не прибегая к функциям работы с каталогами (`opendir()`, `readdir()`, `closedir()` и т. п.). Решение задачи основано на том факте, что в PHP существует несколько видов кавычек, каждый из которых обладает своими свойствами.

I.1.2. Вывод случайного количества символов

Создайте скрипт, который выводит случайное количество символов * от 0 до 10.

I.1.3. Выравнивание по правому краю

Пусть есть список файлов в массиве (листинг I.1.1). У имен файлов может быть различная длина, и необходимо выровнять их по правому краю так, как это изображено на рис. I.1.1. При решении задачи нельзя прибегать к атрибуту `align`, HTML-таблицам и каскадным таблицам стилей CSS, можно использовать только HTML-теги `<pre>` и `</pre>` и средства PHP.

Листинг I.1.1. Массив `$filename` с именами файлов

```
<?php
  $filename = array("all.php", "auth.php",
                    "auth.txt", "base.txt",
                    "chat.html", "config.php",
                    "count.txt", "count_new.txt",
                    "counter.dat", "counter.php",
                    "create.php", "dat.db");
?>
```

Замечание

Файл с массивом можно найти на прилагаемом к книге компакт-диске (scripts\1\1.3\1.php).

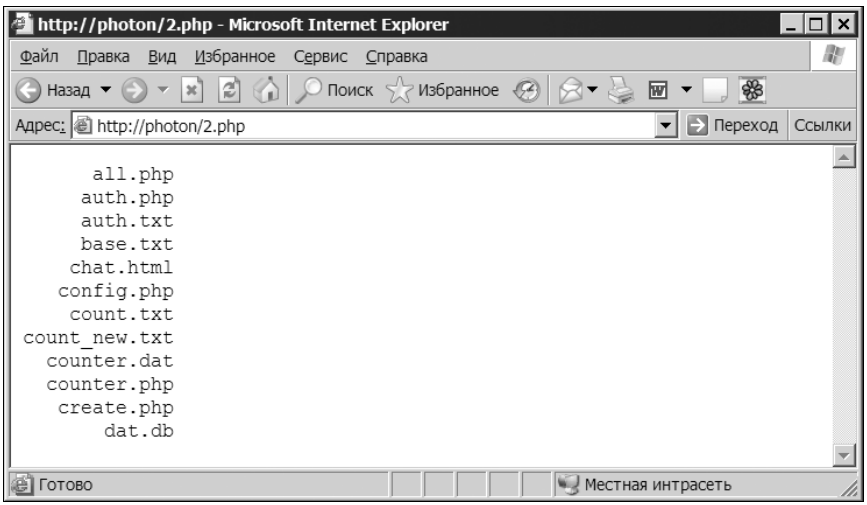


Рис. I.1.1. Выравнивание имен файлов по правому краю

I.1.4. Выравнивание по левому и правому краям

Необходимо разбить массив `$filename` (листинг I.1.1) на две части и вывести в виде двух колонок так, как это представлено на рис. I.1.2. При этом левая колонка должна быть выровнена по левому краю, а правая — по правому. При решении задачи нельзя прибегать к атрибуту `align`, HTML-таблицам и каскадным таблицам стилей CSS, можно использовать только HTML-теги `<pre>` и `</pre>` и средства PHP.

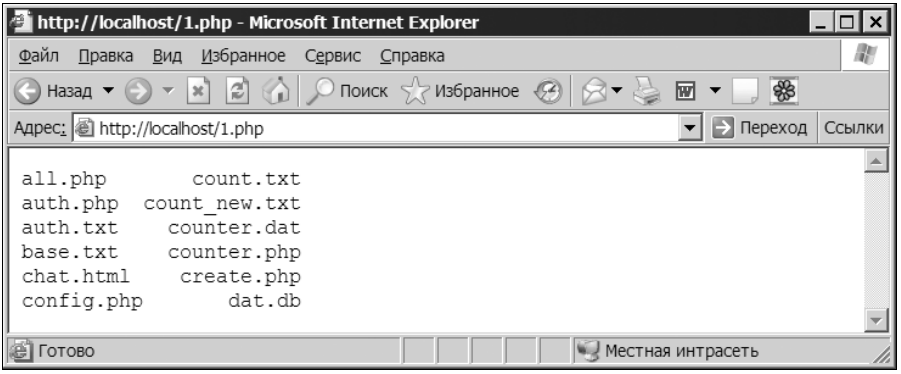


Рис. I.1.2. Выравнивание имен файлов по левому и правому краям

I.1.5. Вывод данных в три столбца

Часто перед Web-разработчиками встает задача вывода таблицы, содержащей несколько столбцов. Выведите имена файлов из массива `$filename` (см. листинг I.1.1) двумя способами:

- ❒ по строкам — сначала заполняется первая строка, затем вторая и т. д. (рис. I.1.3);

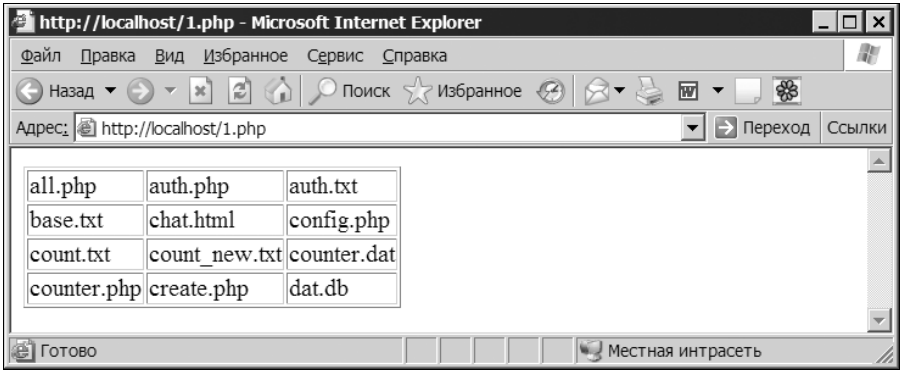


Рис. I.1.3. Первый вариант вывода массива в три столбца

- по столбцам — сначала заполняется первый столбец, затем второй и т. д. (рис. I.1.4).

При решении этой задачи необходимо динамически сформировать HTML-таблицу.

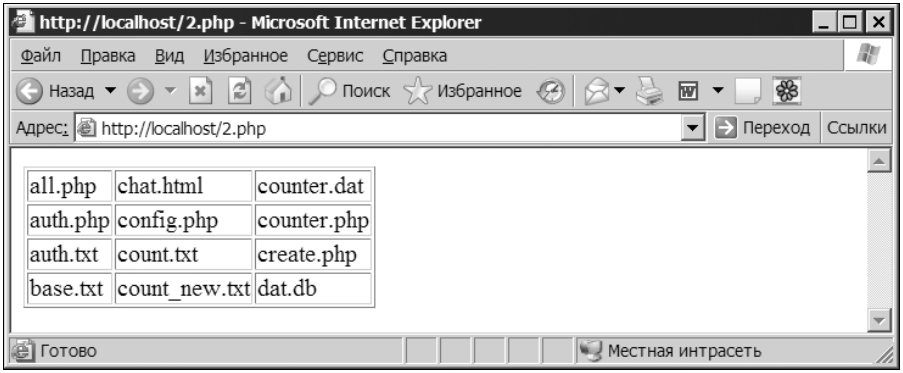


Рис. I.1.4. Второй вариант вывода массива в три столбца

I.1.6. Передача массива между двумя страницами

Пусть массив `$filename`, представленный в листинге I.1.1, определен на странице first.php. Отобразите его на странице second.php, используя инструкцию `include`.

I.1.7. Передача массива методом GET

Пусть массив `$filename`, представленный в листинге I.1.1, определен на странице first.php. Отобразите его на странице second.php, используя для передачи метод GET.

I.1.8. Передача массива методом POST

Пусть массив `$filename`, представленный в листинге I.1.1, определен на странице first.php. Отобразите его на странице second.php, используя для передачи метод POST.

I.1.9. Передача массива через сессии

Пусть массив `$filename`, представленный в листинге I.1.1, определен на странице first.php. Отобразите его на странице second.php, используя сессии.

I.1.10. Передача массива через cookie

Пусть массив `$filename`, представленный в листинге I.1.1, определен на странице first.php. Отобразите его на странице second.php, используя cookie.

I.1.11. Вертикальный вывод строки

Выведите строку "Hello world!" вертикально, так, как это представлено на рис. I.1.5.

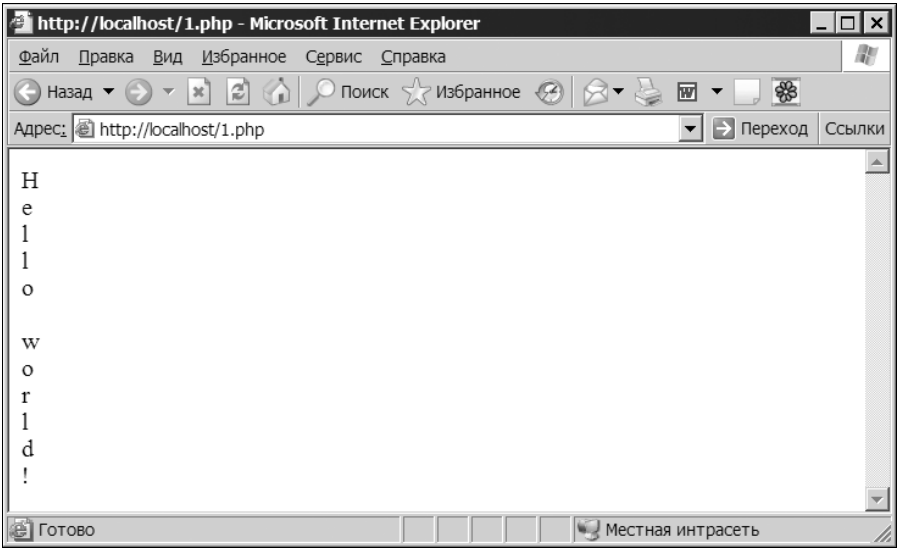


Рис. I.1.5. Вертикальный вывод строки

I.1.12. Число в денежном формате

Пусть имеется число 18439529234.5678, его необходимо представить в денежном формате, т. е. чтобы после запятой осталось только два знака, а триады были бы разделены пробелом — 18 439 529 234.57.

I.1.13. Упаковка IP-адреса

Язык PHP обладает огромным количеством самых разнообразных функций. Среди них имеются функция для упаковки IP-адреса в целое число — `ip2long()` и функция обратного преобразования полученного целого числа в IP-адрес — `long2ip()`.

В листинге I.1.2 демонстрируется использование функций `ip2long()` и `long2ip()`.

Листинг I.1.2. Использование функций `ip2long()` и `long2ip()`

```
<?php
  // Получаем IP-адрес сайта http://www.softtime.ru/
  $ip = gethostbyname('www.softtime.ru');
  echo "$ip<br>";
  // Преобразуем IP-адрес в число
  $long = ip2long($ip);
  echo "$long<br>";
  // Преобразуем число в IP-адрес
  $ip = long2ip($long);
  echo "$ip<br>";
?>
```

Результатом работы скрипта из листинга I.1.2 будут следующие строки:

```
82.208.89.164
1389386148
82.208.89.164
```

Не прибегая к библиотечным функциям, создайте аналоги для функций `ip2long()` и `long2ip()`.

I.1.14. Календарь

Создайте календарь на текущий месяц в двух форматах: американском (рис. I.1.6) и российском (рис. I.1.7).

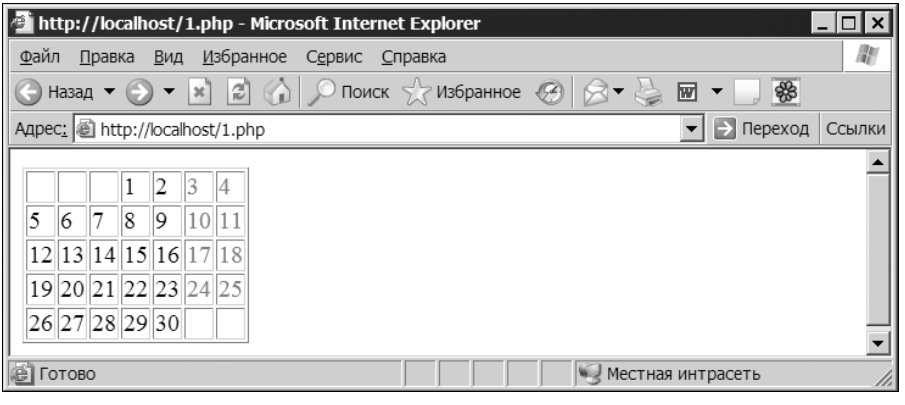


Рис. I.1.6. Календарь на текущий месяц в американском формате

	5	12	19	26
	6	13	20	27
	7	14	21	28
1	8	15	22	29
2	9	16	23	30
3	10	17	24	
4	11	18	25	

Рис. I.1.7. Календарь на текущий месяц в российском формате

Замечание

Данная задача перекликается с задачей I.1.5. Здесь также требуется вывод календаря по строкам и по столбцам.

Дни, соответствующие субботе и воскресенью, необходимо подсветить красным цветом.

I.1.15. Замена символов bbCode

Замените в тексте "Очень [b]жирный[/b], жирный [b]текст" символы bbCode [b] и [/b] на их HTML-эквиваленты и , не прибегая к регулярным выражениям. То есть для решения задачи должны быть использованы только строковые функции.

I.1.16. Преобразование десятичного числа в двоичное и обратно

Создайте два небольших Web-приложения, содержащих текстовую область и кнопку. Первое приложение при вводе десятичного числа должно выводить его двоичное представление, например, если пользователь ввел 482, то скрипт должен вывести 111100010. А второй скрипт должен решать обратную задачу — двоичное число переводить в десятичное.

Для решения задачи нельзя использовать никакие функции. Допустимы только арифметические операторы, циклы, операторы ветвления и конструкция `echo`.

Замечание

Использовать `print()` нельзя, т. к., в отличие от конструкции `echo`, это функция.

I.1.17. Возведение числа в степень

Создайте программу, которая запрашивает у пользователя два целых числа и возводит первое число в степень второго, не используя функцию `pow()`.

Замечание

Следует воспользоваться битовыми операторами.

I.1.18. Изменение регистра строки

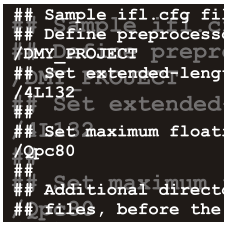
Используя лишь поразрядные операторы, измените регистр каждого символа строки, введенной пользователем на строчный (прописной). Считается, что пользователь вводит только символы английского алфавита.

I.1.19. Преобразование арабского числа в римское

Создайте скрипт, преобразующий число в арабской нотации (от 1 до 2000) в римское.

Замечание

Арабские числа соотносятся с римскими следующим образом: 1 — I, 5 — V, 10 — X, 50 — L, 100 — C, 500 — D, 1000 — M. Например, 116 == CXVI, 199 == CXCIX, 14 == XIV.



ГЛАВА I.2

Регулярные выражения

Регулярные выражения являются специализированным мини-языком, используемым совместно со многими языками программирования. Сложную задачу можно решить двумя способами: либо создав сложное решение, используя простые технологии, либо создав простое решение, используя сложную технологию. Точно так же и с регулярными выражениями — изучить их достаточно сложно, но, поняв их один раз, далее в одну строку можно решать задачи, для решения которых при помощи строковых функций может понадобиться сотня строк. В *главе I.1* было сказано, что плотность ошибок тем меньше, чем короче программа — регулярные выражения позволяют создавать не просто короткие программы, а очень короткие.

Замечание

Синтаксис регулярных выражений, а также обзор функций для работы с ними представлены в *приложении 1*.

Сами по себе регулярные выражения являются достаточно специфическим языком программирования, аналога которому нет среди языков общего назначения. Поэтому их достаточно сложно изучить — знание других языков программирования фактически никак не облегчает задачу. Поэтому важно закреплять навыки при помощи практики.

Замечание

Все примеры из данной главы можно найти в каталоге scripts\2 компакт-диска, поставляемого вместе с книгой.

I.2.1. Удаление всех тегов из HTML-страницы

На компакт-диске найдите HTML-страницу scripts\2\index.htm. Прочитайте содержимое страницы и удалите все HTML-теги, оставив только полезный текст. Текст необходимо вывести в окно браузера (рис. I.2.1).

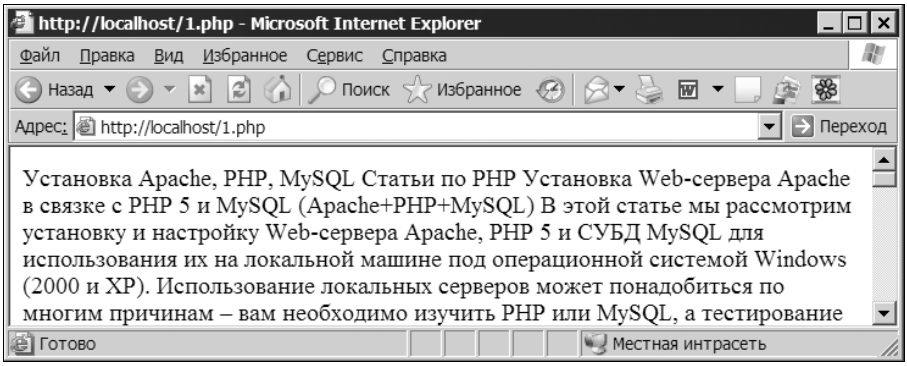


Рис. I.2.1. Чистый текст, извлеченный из HTML-страницы index.htm

I.2.2. Удаление изображений из HTML-страницы

На компакт-диске найдите HTML-страницу scripts\2\index.htm. Прочитайте содержимое страницы и удалите HTML-теги `<img>`.

I.2.3. Преобразование нескольких пробельных символов в один

Пусть имеется текст, который содержит между словами от одного до нескольких пробельных символов. Необходимо таким образом преобразовать текст, чтобы между словами остался только один пробел.

I.2.4. Извлечение названия HTML-страницы

На компакт-диске найдите HTML-страницу scripts\2\index.htm. Извлеките название страницы, которое помещается между тегами `<title>` и `</title>`.

I.2.5. Конвертация даты из MySQL-формата в календарный формат

Используя регулярные выражения, переконвертируйте дату из формата 2003-03-21 в формат 21.03.2003.

I.2.6. Проверка корректности ввода адреса электронной почты

Разработайте HTML-форму, обработчик которой будет проверять корректность ввода адреса электронной почты (рис. I.2.2).

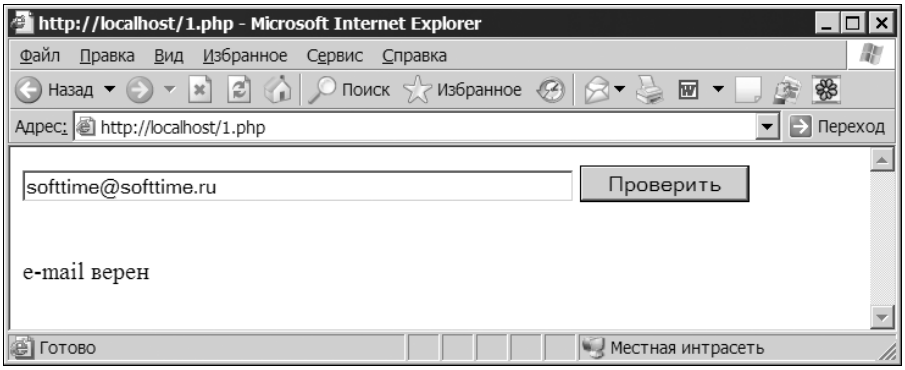


Рис. I.2.2. HTML-форма проверки адреса электронной почты

I.2.7. Проверка корректности ввода URL

Разработайте HTML-форму, обработчик которой будет проверять корректность ввода адреса Web-сайта. Допускается ввод как с указанием протокола, например, **http://www.softtime.ru**, так и без него, например, **www.softtime.ru**. Следует учитывать, что адрес может содержать путь после доменного имени, а также параметры, например,
http://www.softtime.ru/php5/index.php?id_article=43.

I.2.8. Подсветка URL

Часто возникает задача превращения текстовой ссылки в гиперссылку. На компакт-диске найдите текстовый файл scripts\2\text.txt и выведите его содержимое в окно браузера, преобразовав все URL в гиперссылки.

I.2.9. Проверка корректности ввода чисел

Создайте HTML-форму, состоящую из двух текстовых полей, в первом из которых вводится количество товарных позиций, а во втором их цена в формате ###.##. Обработчик формы должен проверить, является ли введенная в первом поле информация целым числом, а во втором — удовлетворяющим денежному формату. Если все верно, необходимо вывести произведение этих двух чисел.

I.2.10. Изменение регистра

Пусть имеется фраза "**ПРОГРАММИРОВАНИЕ — это ИСКУССТВО. Ему и ЖИЗНЬ посвятить не жалко**". Создайте скрипт и регулярное выражение,

которое заменит все слова в верхнем регистре на слова, начинающиеся с заглавной буквы: "**Программирование — это Искусство. Ему и Жизнь посвятить не жалко**".

I.2.11. Разбивка длинной строки

При построении различных Web-приложений, главным образом гостевых книг, форумов и чатов, часто возникает необходимость защиты дизайна страниц от длинных последовательностей символов, которые могут исказить дизайн. Создайте функцию, разбивающую на части все слова, длина которых превышает 25 символов.

I.2.12. Разбивка HTML-страницы на предложения

На компакт-диске найдите HTML-страницу scripts\2\index.htm. Прочитайте содержимое страницы и поместите каждое предложение текста в элементы массива `$text` так, чтобы первое предложение оказалось в элементе с индексом 0 — `$text[0]`, второе в элементе с индексом 1 — `$text[1]` и т. д. После чего в цикле преобразуйте массив `$text` в двумерный массив таким образом, чтобы в элементе `$text[0][0]` хранилось первое слово первого предложения, в элементе `$text[0][1]` хранилось второе слово первого предложения и т. д. Проконтролируйте результаты работы, отправив дамп массива в окно браузера при помощи функции `print_r()`.

I.2.13. Количество слов в тексте

На компакт-диске найдите HTML-страницу scripts\2\index.htm. Прочитайте содержимое страницы и сосчитайте, сколько в нем содержится одно-, двух-, ..., десятибуквенных слов.

I.2.14. Интерпретация тегов bbCode

В Интернете большое распространение получили теги в квадратных скобках, именуемые так же, как теги в стиле phpBB (известного и широко распространенного форума). Удобство использования таких тегов заключается в том, что все теги HTML можно запретить, преобразуя их при помощи функции `htmlspecialchars()` в безопасную форму, и в то же время разрешить посетителям использовать их эквиваленты. Например, `[i]` вместо `<i>` и `[code]` вместо `<code>`. Теги в квадратных скобках можно заменить на теги в угловых скобках уже после преобразования текста при помощи функции

`htmlspecialchars()`. Чаще всего прибегают к тегам `[url]`, которые имеют следующий синтаксис:

```
[url = ссылка] имя ссылки [/url]
```

При выводе на страницу этот шаблон следует преобразовать в

```
<a href=ссылка>имя ссылки</a>
```

Если используется форма тега

```
[url]ссылка[/url]
```

то на страницу выводится гиперссылка вида:

```
<a href=ссылка>ссылка</a>
```

На компакт-диске найдите HTML-страницу scripts\2\bb.txt, содержимое этой страницы представлено на рис. I.2.3.

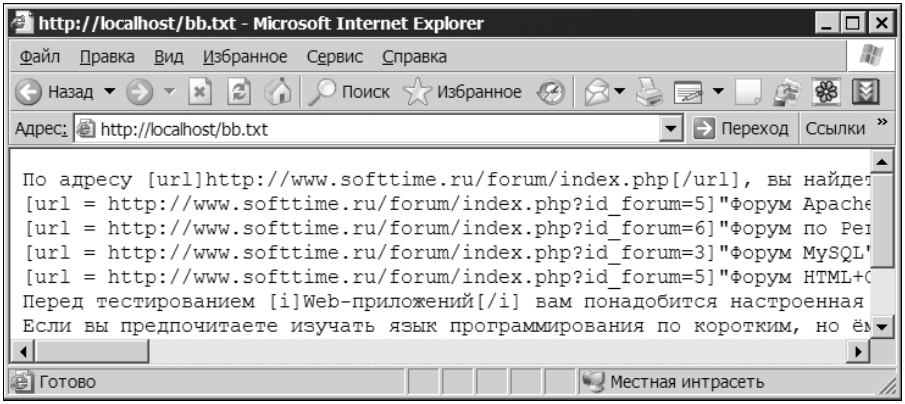


Рис. I.2.3. Содержимое файла bb.txt

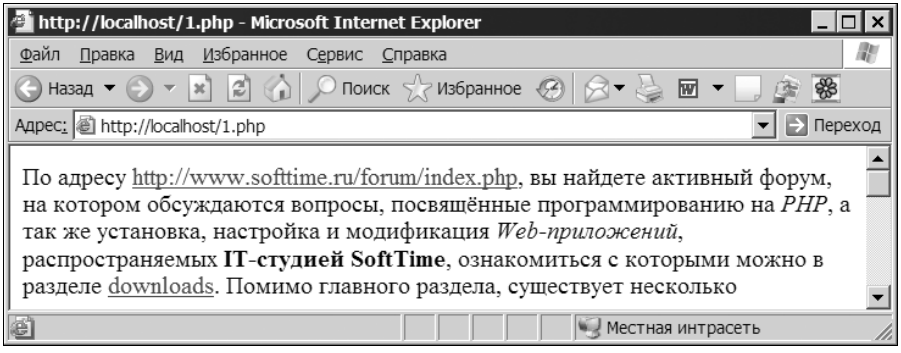


Рис. I.2.4. Преобразованное содержимое файла bb.txt

Необходимо преобразовать все имеющиеся на странице теги в их HTML-эквиваленты (рис. I.2.4).

I.2.15. Подсветка PHP-кода

В PHP есть две стандартные функции для подсветки кода: `highlight_string()` и `highlight_file()`. Данные функции имеют два серьезных недостатка: поддерживается только подсветка PHP-кода и только кода, размещенного между тегами `<?php` и `?>` (а также `<?` и `?>`). Создайте собственную функцию подсветки синтаксиса, лишенную этого недостатка.

I.2.16. Замена подстроки с условием

Создайте регулярное выражение, которое заменит в тексте все символы точки "." на троеточие "...", но только в том случае, если точка не стоит в конце сокращений "г.", "рис. " и "табл.".