



МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени Н.Э. БАУМАНА

Учебное пособие

Методические указания для выполнения
домашнего задания по курсу :

«Основы конструкторско-технологической информатики»

МГТУ имени Н.Э. Баумана

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени Н.Э. БАУМАНА

Методические указания для выполнения
домашнего задания по курсу :

«Основы конструкторско-технологической информатики»

Москва
МГТУ имени Н.Э. Баумана

2012

Лабораторная работа 1

Пример 1: структура документа:

```
<html>
  <head>
    <title>Заголовок</title>
  </head>
  <body>
    Тело документа
  </body>
</html>
```

Пример 2: дополнительная информация в заголовке документа:

```
<HEAD>
<meta name="author" content="iu4.bmstu.ru">
<meta name="generator" content="Mozilla/4.08 [en] (Win95;l) [Netsacpe]">
<meta name="Description"
content="Кафедра Конструирование и производство электронно-вычислительных и
телекоммуникационных систем: История кафедры и МГТУ им. Н.Э. Баумана,
Дистанционное компьютерное образование, Компьютерная подготовка, сети,
телекоммуникации, нейронные сети и нейрокомпьютеры, вычислительная техника">
<meta name="Keywords"
content="МГТУ, дистанционное, компьютерное, обучение, подготовка , Linux , Сети,
сетевые технологии, микропроцессоры, ЭВМ, нейро, нейрокомпьютеры, курсовик,
реферат, диплом">
<title>
Department of Designing and Technology of Computers and Telecommunications Systems (IU-
4)
</title>
</HEAD>
```

Пример 3: Тело документа

```
<body background="fon.gif" bgcolor=#b0c9c0 text="#000000"
link="#000000" vlink="#000000" alink="#000000"
marginheight=0 marginwidth=0
leftmargin=0 topmargin=0>
```

тело HTML документа

```
</body>
```

Пример 4: форматирование текста:

Спецификация дескриптора определения заголовка <Hn>	
Дескриптор	Назначение

<Hn></Hn>	Заголовок уровня n, где = 1 ... 6	
	Атрибут	Значение
	align={left, right, center}	Выравнивание заголовка по горизонтали

```
<html><!-- Задание типа документа - т.е. документ html -->
<head><!-- Заголовок страницы -->
  <title>Пример 4</title>
</head>
```

```
<!-- Тело страницы -->
<body bgcolor="#FFFFFF" text="#000000" link="#FF0000"
alink="#FF00FF" vlink="#0000FF"
marginwidth="0" topmargin="0">
```

```
  <center><h1>Страница №5</h1></center>
  <hr color="#FF0000" size=2>
  <center>
    <h1><b>
<font color="#0000FF">ПРИ</font>
<font color="#FFFF00">МЕР</font>
<font color="#00FF00">ОФОР</font>
<font color="#FF8000">МЛЕ</font>
<font color="#00ff80">НИЯ</font>
<font color="#00FFFF">ТЕКС</font>
<font color="#FF0080">ТА</font>
<font color="#800080">!</font>
    </h1></b>
  </center>

  <hr color="#FF0000" size=2>
  <font color="#000000"><b>Жирный текст</b></FONT><br>
  <font color="#FF0000"><i>Наклонный текст</i></FONT><br>
  <font color="#00FF00"><u>Подчеркнутый текст</u></FONT><br>
  <font color="#0000FF" size=+1>Текст размера +1</FONT><br>
  <font color="#FF0080" size=+2>Текст размера +2</FONT><br>
  <font color="#000080" size=-1>Текст размера -1</FONT><br>
  <font color="#800000" size=-2>Текст размера -2</FONT><br>
  </body>
</html>
```

Пример 5: вставка ссылок :

Варианты задания ссылок:

- `1` - открыть файл index.htm расположенный в каталоге primer, расположенном в том же каталоге, что и файл с ссылкой, текущего сервера (или локального каталога клиента).
- `1` - открыть файл index.htm расположенный в каталоге более высокого уровня, чем файл где расположена ссылка текущего сервера.
- `1` - открыть файл index.htm расположенный в каталоге более высокого уровня (на два порядка выше), чем файл где расположена ссылка текущего сервера.
- `1` - указание полного URL адреса ресурса.
- `vlasov@chat.ru` - запуск программы отправки почтового сообщения по e-mail: vlasov@chat.ru.
- `1` - Загрузить архив (вообще то файл может быть любой).
- `` - Отображает рисунок с именем bmstu.gif, расположенный по указанному URL.
- `` - Оформление изображения как ссылки.

Пример 6: Списки

```

<html
<head>
  <title>шестой пример</title>
</head>
<body bgcolor="#ffffff" text="#000000" link="#ff0000"
  alink="#ff00ff" vlink="#0000ff" marginwidth="0" topmargin="0">

  <center><h1>страница №6</h1></center>

  <hr color="#ff0000" size=2>
  <center><h1><b>
<font color="#0000ff">при</font>
<font color="#ffff00">мер</font>
<font color="#00ff00">офор</font>
<font color="#ff8000">мле</font>
<font color="#00ff80">ния</font>
<font color="#00ffff">спис</font>
<font color="#ff0080">ков</font>
<font color="#800080">!</font>
</h1></b></center>

  <hr color="#ff0000" size=2>

  <h2>неупорядоченный список</h2>
  <ul>
<li><font color="#000000"><b>жирный текст</b></font><br>
<li><font color="#ff0000"><i>наклонный текст</i></font><br>
<li><font color="#00ff00"><u>подчеркнутый текст</u></font><br>
</ul>

  <hr color="#ff0000" size=2>

```

```

<h2>нумерованный список</h2>
<ol>
<li><font color="#000000"><b>жирный текст</b></font><br>
<li><font color="#ff0000"><i>наклонный текст</i></font><br>
<li><font color="#00ff00"><u>подчеркнутый текст</u></font><br>
</ol>

<hr color="#ff0000" size=2>
<h2>меню</h2>
<menu>
  <li><font color="#000000"><b>жирный текст</b></font><br>
<li><font color="#ff0000"><i>наклонный текст</i></font><br>
<li><font color="#00ff00"><u>подчеркнутый текст</u></font><br>
</menu>

<hr color="#ff0000" size=2>

<h2>каталог</h2>
<dir>
<li><font color="#000000"><b>жирный текст</b></font><br>
<li><font color="#ff0000"><i>наклонный текст</i></font><br>
<li><font color="#00ff00"><u>подчеркнутый текст</u></font><br>
</dir>

</body>
</html>

```

Пример 7: Таблицы

```

<html>
  <head>
<title>пример</title>
  </head>

  <body bgcolor="#ffffff" text="#000000" link="#ff0000"
    alink="#ff00ff" vlink="#0000ff"
    marginwidth="0" topmargin="0">

  <!--создаем таблицу с размером во весь экран броузера с прозрачными границами-->
  <table width=100%>
    <tr><!-- определяем первую строку таблицы-->
      <td width=40%><!-- определяем первый столбец таблицы -->
        <center> 
        </center>
      </td>
      <td align="center" width=60%>
        <!-- определяем второй столбец таблицы-->
        <h1>страница №6</h1>
      </td>
    </tr>
  </table>

  <!--создаем таблицу с размером во весь экран броузера с прозрачными границами-->

```

```

<table width=100% border=3>
  <tr><!-- определяем первую строку таблицы-->
    <td bgcolor="#80ff00" width=50>
<h2>неупорядоченный список</h2>
<ul>
  <li><font color="#000000"><b>жирный текст</b></font>
  <li><font color="#0000ff"><i>наклонный текст</i></font>
  <li><font color="#ff0000"><u>подчеркнутый текст</u></font>
</ul>
    </td>
<td bgcolor="#ff0080" width=50%>
<h2>нумерованный список</h2>
<ol>
  <li><font color="#000000"><b>жирный текст</b></font>
  <li><font color="#0000ff"><i>наклонный текст</i></font>
  <li><font color="#00ff00"><u>подчеркнутый текст</u></font>
</ol>
</td>
</tr>
<tr><!-- определяем вторую строку таблицы-->
<td bgcolor="#00ffff" width=50%>
<h2>меню</h2>
<menu>
  <li><font color="#000000"><b>жирный текст</b></font>
  </li>
  <li><font color="#0000ff"><i>наклонный текст</i></font>
  </li>
  <li><font color="#ff0000"><u>подчеркнутый текст</u></font>
  </li>
</menu>
</td>
<td bgcolor="#00ff80" width=50%>
<h2>каталог</h2>
<dir>
  <li><font color="#000000"><b>жирный текст</b></font>
  </li>
  <li><font color="#0000ff"><i>наклонный текст</i></font>
  </li>
  <li><font color="#ff0000"><u>подчеркнутый текст</u></font>
  </li>
</dir>
</td>
</tr>
</table>
</body>
</html>

```

Пример 9: Стили

```

<html>
<head>
<title>Пример использования стилей</title>
<style type="text/css">
S.green_caps {color: green; font-style: small-caps}

```



```

S.yellow_helv {font-family: Helvetica, sans-serif; color:
yellow}
</style>
</head>
<body>
<s class="green_caps">Красный заголовок</p>
<ol class="yellow_helv">
<li>Желтый элемент 1</li>
<li>Желтый элемент 2</li>
</ol>
</body>
</html>

```

Пример 10: Дескриптор span

```

<html>
<head>
  <title>Пример использования дескриптора span</title>
  <style type="text/css">
    span {font: 62 pt Courier; color: red}
  </style>
</head>
<body>
  <p><span>Пример использования дескриптора span</span>
  в документе</P>
</body>
</html>

```

Пример11: замена стилей

```

<html>
<head>
  <title>Замена стилей</title>
  <style></style>
</head>
<body>
  <p style="text-align: center">Абзац по центру</p>
  <em style="color: red">Красный абзац</em>
  <ol style="list-style: lower roman">
    <li> Каждый элемент списка
    <li> нумеруется
    <li> строчными римскими цифрами
  </ol>
</body>
</html>

```

Пример 12: Использование внешнего файла стилей

Файл: index.htm

```

<html>
<head>
  <title>Страница использующая внешний файл стилей</title>
  <link href="common.css" rel="stylesheet">
</head>

```

</html>

Файл *common.css*

BODY.back {background: yellow}

H1 {font-style: bold; color: red; text-align: center}

H2.ital {font-style: italic}

H3 {text-align: center}

p.dblue {color: darkblue}

Лабораторная работа 2

Пример 1: Варианты объявления PHP скрипта:

```
<b>Варианты объявления PHP скрипта.</b>
<br><? echo "Пример объявления 1"; ?></br>
<br><?php echo "Пример объявления 2"; ?></br>
<br><script language="php">
echo "Пример объявления 3";
</script></br>
```

Пример 2: Объявления переменных:

```
<?php
// объявление переменных
$a = 1;
$b = 2;

/* тело скрипта*/
$c = $a + $b ;
echo ( "результат сложения a=1 и b=2 равен" );
echo $c;
?>
```

Пример 3: Передача значения из формы:

```
<form action="02.php" method="post">
    Name: <input type="text" name="name"><br>
    <input type="submit">
</form>

<?php
echo ("name=");
echo $_POST["name"]; //используется глобальный массив $_POST["name"]
?>
```

Пример 4: работа с массивами:

```
$languages = array ("English", "Russian", "Spanish");
echo("My native language is $languages[1] \n");
```

```
//ассоциативные массивы
```

```
<?
$ship = array(
    "Passenger ship" => array("Yacht", "Liner", "Ferry"),
    "War ship" => array("Battle-wagon", "Submarine", "Cruiser"),
    "Fright ship" => array("Tank vessel", "Dry-cargo ship", "Container
    cargo ship")
);
foreach($ship as $key => $type)
{
    echo(
    "<h2>$key</h2>\n". "<ul>\n");
    foreach($type as $ship)
    {
        echo("\t<li>$ship</li>\n");
    }
}
```

```

}
echo("</ul>\n");
?>

```

Пример 5: Передача нескольких значений из формы:

```

<?php
echo ("<P>");
echo ("name= "); echo $_GET["name"];
echo ("<P>");
echo ("e-mail= "); echo $_GET["e-mail"];
?>

<P>

<form action=03.php method="get">
Name: <input type="text" name="name"><br>
E-mail: <input type="text" name="e-mail"><br>
<input type="submit">
</form>

```

Пример 5: Обработка переключателей и радиокнопок

```

<?
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
echo ('Мой любимый предмет: <i>' . $_POST['kurs'] . '</i><br>');

$favorite_times = count($times);

if( $favorite_times <= 1) {
$times_message = 'не ботан';
} elseif ($favorite_times > 1 && $favorite_times < 4) {
$times_message = 'ботаю иногда';
} else {
$times_message = 'ботан';
}

echo ('Я <i>' . $times_message . '</i><br>');
?>

<FORM TARGET="04.php" METHOD="POST">
любимый предмет:<br>
<INPUT TYPE=RADIO NAME="kurs" VALUE="КТБД">Конструкторско-технологические базы данных<br>
<INPUT TYPE=RADIO NAME="kurs" VALUE="СФМ">Системы функционального моделирования<br>
<INPUT TYPE=RADIO NAME="kurs" VALUE="СИИ">Системы искусственного интеллекта<br>

Когда вы предпочитаете его изучать:<br>
<INPUT TYPE=CHECKBOX NAME="times[]" VALUE="m">за завтраком
<INPUT TYPE=CHECKBOX NAME="times[]" VALUE="n">в обед
<INPUT TYPE=CHECKBOX NAME="times[]" VALUE="d">за ужином
<INPUT TYPE=CHECKBOX NAME="times[]" VALUE="l">поздно ночью

<P>
<INPUT TYPE=HIDDEN NAME="stage" VALUE="results">
<INPUT TYPE=SUBMIT VALUE="всегда!">
</FORM>

```

Формы представляют собой простейший способ организации внутри HTML-документа обратной связи между пользователем и сервером. Упрощенно формы можно понимать как набор кнопок, флажков, полей ввода, передаваемых сценарию, в качестве входной информации для обработки. Обработка, принятой сервером информации, ничего общего с HTML не имеет и может выполняться самыми разными средствами.

Спецификация дескриптора задания форм					
Дескриптор	Назначение				
<form> </form>	Формуляр				
	<table border="1"> <thead> <tr> <th>Атрибут</th> <th>Значение</th> </tr> </thead> <tbody> <tr> <td>Action</td> <td>URL-адрес для отправки заполненного формуляра</td> </tr> </tbody> </table>	Атрибут	Значение	Action	URL-адрес для отправки заполненного формуляра
Атрибут	Значение				
Action	URL-адрес для отправки заполненного формуляра				

Enctype	кодирование передаваемых данных
Method	способ передачи формуляра

Типы внутренних конструкций в формах:

- Поля ввода объектов (типы объектов определяются значением атрибута type).
- Поля ввода многострочных текстов.
- Выпадающие меню.
- Поля списков.

Поля ввода.

Спецификация дескриптора задания форм											
Дескриптор	Назначение										
<input> </input>	Поле для ввода строки										
Атрибут	Значение										
Action=URL	URL адрес для отправки заполненной формы										
Align=	<table border="0"> <tr> <td>Bottom</td> <td>выравнивает нижний край кнопки по базовой линии строки</td> </tr> <tr> <td>Left</td> <td>выравнивает кнопку-иллюстрацию по левому краю текста</td> </tr> <tr> <td>Middle</td> <td>центрирует кнопку-иллюстрацию в текстовой строке</td> </tr> <tr> <td>Right</td> <td>выравнивает кнопку-иллюстрацию по правому краю текста</td> </tr> <tr> <td>top</td> <td>выравнивает кнопку-иллюстрацию по верхней границе строки</td> </tr> </table>	Bottom	выравнивает нижний край кнопки по базовой линии строки	Left	выравнивает кнопку-иллюстрацию по левому краю текста	Middle	центрирует кнопку-иллюстрацию в текстовой строке	Right	выравнивает кнопку-иллюстрацию по правому краю текста	top	выравнивает кнопку-иллюстрацию по верхней границе строки
Bottom	выравнивает нижний край кнопки по базовой линии строки										
Left	выравнивает кнопку-иллюстрацию по левому краю текста										
Middle	центрирует кнопку-иллюстрацию в текстовой строке										
Right	выравнивает кнопку-иллюстрацию по правому краю текста										
top	выравнивает кнопку-иллюстрацию по верхней границе строки										
Checked	Установленный флажок или выбранное положение переключателя										
Maxlength	Максимальная длина вводимых текстов										
Name	Название элемента ввода/управления										
Src	Источник графического файла картинки кнопки										

Type =	checkbox	Тип элемента управления /ввода флажки, независимы друг от друга, их можно установить/сбросить в любой комбинации
	file	поле ввода для имени файла, рядом отображается кнопка, "Пролистать/Browse" открывающая
	image	стандартное диалоговое окно выбора файла пользовательская командная кнопка вместо стандартных, получаемых с помощью
	hidden	type=submit или type=reset
	password	параметры передаваемые на сервер, которые не могут быть изменены пользователем
	radio	текстовое поле, вводимые данные отображаются "звездочками"
	reset	селекторные кнопки (переключатели), из группы можно выбрать только одну
	submit	командная кнопка, возвращает формуляр к исходному состоянию; данные не
	text	пересылаются командная кнопка, отправляет
Value		Установленное по умолчанию значение

Поля ввода многострочных текстов

Спецификация дескриптора ввода многострочного текстового поля		
Дескриптор	Назначение	
<textarea> </textarea>	Многострочное поле для ввода текста в форме	
	Атрибут	Значение
	Cols=N	Количество символов /столбцов в поле ввода
	Name=char	Имя поля ввода
	Rows=N	Количество строк поля ввода
	Wrap=	
	off	верстка не выполняется (сервер получает текст одним куском)
	physical	автоматическая верстка, с переносом строк по мере достижения правого края (сервер получает текст с разрывами строк)
	virtual	автоматическая верстка, с переносом строк по

Формирование выпадающих меню и полей списков

Спецификация дескриптора меню или поля списка в форме		
Дескриптор	Назначение	
<select> </select>	Меню или поле списка в форме	
	Атрибут	Значение

Multiple	Возможность выбора нескольких опций
Name=char	Имя элемента
Size=N	Количество одновременно отображаемых

Спецификация дескриптора задания элемента списка/меню в форме	
Дескриптор	Назначение
<option> </option>	Элемент списка/меню в форме.
Атрибут	Значение
Selection	Выбран по умолчанию
Value	Параметры элемента

Лабораторная работа 3

Пример 3: Сессии в PHP:

page1.php:

```
<?
session_start();
$_SESSION['username'] = "maksim";
echo 'Привет, ' . $_SESSION['username'] . "<br>";
?>
<a href="page2.php">На следующую страницу </a>
```

page2.php:

```
?
session_start();
echo $_SESSION['username'] . ' , ты пришел на другую страницу этого сайта!';
echo("<br>");
?>
<a href="page3.php">На следующую страницу </a>
```

page3.php:

```
?
session_start();
unset($_SESSION['username']); // разрегистровали переменную
echo 'Привет, ' . $_SESSION['username'];
/* теперь имя пользователя уже не выводится */
session_destroy(); // разрушаем сессию
?>
```

Пример4: Регулярные выражение PHP

bool `ereg(string pattern, string string [, array regs])`

Данная функция ищет в строке `string` соответствие регулярному выражению, заданному в шаблоне `pattern`. Если соответствия подвыражений с шаблоном будут найдены, то они сохраняются в массиве соответствий `regs`. При этом `$regs[0]` содержит копию строки `string`, `$regs[1]` содержит подстроку, начинающуюся с первой левой скобки, `$regs[2]` хранит подстроку, начинающуюся со второй левой скобки и т.д.

```
<?
$date = "2003-03-21";
if (ereg ("([0-9]{4})-([0-9]{1,2})-([0-9]{1,2})", $date, $regs))
{
    echo "$regs[3].$regs[2].$regs[1]";
}
else
{
    echo "Неверный формат даты: $date";
}
?>
```

Вид отображения на экране

split()

array **split** (*string pattern*, *string string* [, *int limit*])

Эта функция возвращает массив строк, которые представляют собой подстроки строки *string*, образованные в результате деления строки *string* на подстроки в соответствии с регулярным выражением *pattern*. Если указан необязательный параметр *limit*, то в возвращаемом массиве будет не больше *limit* элементов, последний из которых содержит неразделенную часть строки.

Эта функция полезна при разделении дат, доменных имен и т.д.

К примеру:

```
<?
$url = "www.softtime.ru";
$array = split ("\.", $url);
foreach($array as $index => $val)
{
    echo("$index -> $val <br />");
}
?>
```

Результат:

```
0 -> www
1 -> softtime
2 -> ru
```

То же самое можно проделать с датой:

```
<?
$date = "10-12-2003";
$array = split("-", $date);
foreach($array as $index => $val)
{
    echo("$index -> $val <br />");
}
?>
```

Вид отображения на экране

Выражение

- "x.[0-9]" - соответствует строке, содержащей символ x, за которым идет любой другой символ и цифры от 0 до 9.
- "ru|com|net" - содержатся ли в строке подстроки "ru", "com" или "net":
- "[^а-я]" – любой символ кроме символом от «а» до «я»
- "а+" - хотя бы один а (строки "абв" и "абва" соответствуют этому выражению, а строка "укр" - нет);
- "а?" - ноль или один а (строки "абв" и "укр" соответствуют этому выражению, а строка "абва" - нет);
- "а*" - ноль или более а (строки "абв" и "абва" и "укр" соответствуют этому выражению).
- "ху{2}" соответствует строке, в которой за х следует два у;

- "xу{2,}" соответствует строке, в которой за x следует не менее двух у (может быть и больше);
- "xу{2,6}" соответствует строке, в которой за x следует от двух до шести у;
- "x(yz){2,6}" соответствует строке, в которой за x следует от двух до шести последовательностей yz;
- "x(yz)*" соответствует строке, в которой за x следует ноль и более последовательностей yz;
- "^ху" - соответствует любой строке, начинающейся с ху.
- "ху\$" – соответствует любой строке, заканчивающейся на ху.
- "\\$" – соответствует символу \$;

Классами символов называются сокращенные обозначения для предопределенных символов.

- Класс [[:alnum:]] - буквенно-цифровые символы
- Класс [[:digit:]] - десятичные цифровые символы
- Класс [[:xdigit:]] - шестнадцатеричные цифровые символы
- Класс [[:alpha:]] - буквенные символы
- Класс [[:upper:]] - прописные буквенные символы
- Класс [[:lower:]] - строчные буквенные символы
- Класс [[:punct:]] - знаки пунктуации
- Класс [[:space:]] - символы пробела
- Класс [[:blanc:]] - символы табуляции и пробела
- Класс [[:print:]] - печатные символы
- Класс [[:cntrl:]] - управляющие символы
- Класс [[:graph:]] - печатные символы, за исключением пробельные
- Эквивалентом выражения "[a-zA-Z_0-9]" является выражение "[[:alnum:]]";
- Выражению "[0-9]" эквивалентно выражение "[[:digit:]]"
- Выражению "[a-Z]" эквивалентно регулярное выражение "[[:alpha:]]".

Лабораторная работа 4

Подключение к базе данных MySQL

```
<?php
$link = mysql_connect("mysql_host", "mysql_login", "mysql_password")
or die (mysql_error());
echo "Successfully connected to MySQL. ";
mysql_select_db ("my_database") or die (mysql_error());
echo "Successfully connected to my_database.";
mysql_close($link);
?>
```

Пример использования операторов CREATE, INSERT, UPDATE, DROP

```
<?php
$link = mysql_connect("localhost", "ivanovi", "12345")
or die (mysql_error());
echo "Successfully connected to MySQL. ";
mysql_select_db ("ivanovi") or die (mysql_error());
echo "Successfully connected to my_database. ";
$query = "CREATE TABLE test (id INT NOT NULL,name VARCHAR(20),student_group
VARCHAR(20),department VARCHAR(100),PRIMARY KEY (id))";
$result = mysql_query ($query);
if ( $result ) echo "QUERY OK. ";
else {
$error = mysql_error();
echo "$error";
}
$query = "INSERT INTO test VALUES ('1','Ivanov','IU4-21','Department of Designing and
Technology
of Computers and Telecommunications Systems)";
$result = mysql_query ($query)
or die (mysql_error());
$query = "SELECT * FROM test";
$result = mysql_query ($query)
or die ("Query failed");
// printing HTML result
print "<table cellpadding='10px' cellspacing='10px'>";
while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
print "\t<tr>\n";
while(list($col_name, $col_value) = each($line)) {
print "\t\t<td>$col_name</td>\n";
}
print "\t</tr>\n";
}
$query = "SELECT * FROM test";
$result = mysql_query ($query)
or die ("Query failed");
while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
print "\t<tr>\n";
while(list($col_name, $col_value) = each($line)) {
```

```

print "\t<td>$col_value</td>\n";
}
print "\t</tr>\n";
}
print "</table>\n";
$query = "UPDATE test SET name='Petrov' WHERE id='1'";
$result = mysql_query ($query)
or die ("Query failed");
// printing HTML result
$query = "SELECT * FROM test";
$result = mysql_query ($query)
or die ("Query failed");
print "<table cellpadding='10px' cellspacing='10px'>";
while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
print "\t<tr>\n";
while(list($col_name, $col_value) = each($line)) {
print "\t\t<td>$col_name</td>\n";
}
print "\t</tr>\n";
}
$query = "SELECT * FROM test";
$result = mysql_query ($query)
or die ("Query failed");
while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
print "\t<tr>\n";
while(list($col_name, $col_value) = each($line)) {
print "\t\t<td>$col_value</td>\n";
}
print "\t</tr>\n";
}
print "</table>\n";
$query = "DELETE FROM `test` WHERE `id`='1'";
$result = mysql_query ($query)
or die (mysql_error());
// printing HTML result
$query = "SELECT * FROM test";
$result = mysql_query ($query)
or die ("Query failed");
print "<table cellpadding='10px' cellspacing='10px'>";
while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
print "\t<tr>\n";
while(list($col_name, $col_value) = each($line)) {
print "\t\t<td>$col_name</td>\n";
}
print "\t</tr>\n";
}
$query = "SELECT * FROM test";
$result = mysql_query ($query)
or die ("Query failed");
while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
print "\t<tr>\n";
while(list($col_name, $col_value) = each($line)) {

```

```
print "\t<td>$col_value</td>\n";
}
print "\t</tr>\n";
}
print "</table>\n";
$query = "DROP TABLE test";
$result = mysql_query ($query)
or die (mysql_error());
echo "QUERY OK. ";
mysql_close($link);
?>
```