

Министерство образования Республики Беларусь

**Учреждение образования
«Гомельский государственный технический
университет имени П. О. Сухого»**

**Институт повышения квалификации
и переподготовки кадров**

Кафедра «Информатика»

Н. В. Самовендюк

СЕРВЕРНЫЕ ТЕХНОЛОГИИ РАЗРАБОТКИ WEB-САЙТОВ

ПРАКТИКУМ

**по одноименной дисциплине
для слушателей специальности 1-40 01 74
«Web-дизайн и компьютерная графика»
заочной формы обучения**

Гомель 2014

УДК 004.43(075.8)
ББК 32.973-018я73
С17

*Рекомендовано научно-методическим советом
факультета автоматизированных и информационных систем
ГГТУ им. П. О. Сухого
(протокол № 2 от 26.09.2014 г.)*

Рецензент: проф. каф. «Информационные технологии» ГГТУ им. П. О. Сухого
д-р техн. наук, проф. *И. А. Мурашко*

Самовендюк, Н. В.

С17 Серверные технологии разработки Web-сайтов : практикум по одной дисциплине для слушателей специальности 1-40 01 74 «Web-дизайн и компьютерная графика» заоч. формы обучения / Н. В. Самовендюк. – Гомель : ГГТУ им. П. О. Сухого, 2014. – 82 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц ; 32 Mb RAM ; свободное место на HDD 16 Mb ; Windows 98 и выше ; Adobe Acrobat Reader. – Режим доступа: [http:// library.gstu.by/StartEK/](http://library.gstu.by/StartEK/). – Загл. с титул. экрана.

Представлены задания по выполнению лабораторных работ по дисциплине «Серверные технологии разработки Web-сайтов», направленных на разработку Web-сайтов с использованием серверного языка сценария PHP.

Рассмотрены основные возможности языка PHP. Особое внимание уделено разработке серверных сценариев для решения типичных задач разработки Web-сайтов, таких как прием и обработка на сервере данных форм, отправленных клиентом пользователя, взаимодействие с базами данных, создание гостевой книги, сохранение данных пользователя на стороне сервера с использованием сеансов, загрузка и обработка файлов.

Для слушателей специальности 1-40 01 74 «Web-дизайн и компьютерная графика» заочной формы обучения.

УДК 004.43(075.8)
ББК 32.973-018я73

© Учреждение образования «Гомельский
государственный технический университет
имени П. О. Сухого», 2014

СОДЕРЖАНИЕ

ЛАБОРАТОРНАЯ РАБОТА № 1	3
ЛАБОРАТОРНАЯ РАБОТА № 2	9
ЛАБОРАТОРНАЯ РАБОТА № 3	17
ЛАБОРАТОРНАЯ РАБОТА № 4	26
ЛАБОРАТОРНАЯ РАБОТА № 5	33
ЛАБОРАТОРНАЯ РАБОТА № 6	43
ЛАБОРАТОРНАЯ РАБОТА № 7	51
ЛАБОРАТОРНАЯ РАБОТА № 8	61
ЛАБОРАТОРНАЯ РАБОТА № 9	69
ЛАБОРАТОРНАЯ РАБОТА № 10	79
ЛАБОРАТОРНАЯ РАБОТА № 11	87

ЛАБОРАТОРНАЯ РАБОТА № 1

Тема: «Знакомство с серверными технологиями разработки Web-сайтов»

Цель работы: изучить программное обеспечение web-сервера, сервера баз данных, модуля PHP, общие сведения о синтаксисе PHP, способах внедрения PHP-скрипта в HTML-страницу и включения файлов PHP в web-страницы.

Теоретическая часть: теоретические сведения приведены в электронном конспекте лекций.

Практическая часть:

Задание 1.

1. Установите локальный web-сервер, используя дистрибутивы Denwer или XAMP (эти дистрибутивы включают Apache, PHP, MySQL, Perl и т.д.). Главная особенность Denwer — удобство при удаленной работе сразу над несколькими независимыми проектами и возможность размещения на Flash-накопителе, поэтому дальнейшее изложение материала будет основано на этом дистрибутиве. Дистрибутив можно взять у преподавателя или скачать с сайта www.denwer.ru.

2. Запустите web-сервер, используя ярлык Start Denwer на Рабочем столе или программу Run, расположенную в каталоге denwer.

3. В адресной строке браузера наберите **http://localhost/** и убедитесь, что web-сервер работает.

4. В каталоге **home** создайте папку с именем виртуального хоста **mysite.local** и в ней папку **www**. Папка **www** является корневой для вашего локального сайта. Проверьте работоспособность локального сайта, набрав в адресной строке **http://mysite.local/**.

Задание 2.

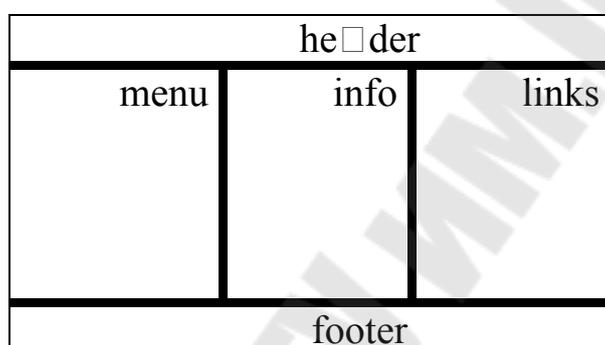
1. В корневой папке сайта создайте файл **index.php** следующего содержания:

```
<?php  
echo "Начало работы с PHP!";  
phpinfo();  
?>
```

2. Протестируйте работу скрипта.

Задание 3.

1. Создайте макет сайта следующей структуры



Макет страницы содержит следующие основные блоки:

1. header – заголовок сайта, содержит логотип (изображение);
2. menu – меню навигации;
3. info – основной раздел в котором располагается вся информативная часть сайта;
4. links – ссылки на сайт университета и института повышения квалификации.
5. footer – содержит ФИО студента, группу, дату создания, email и др.

Пример макета, выполненный средствами HTML и CSS

Файл **maket.html**

```
<html>  
<head>  
<title>Разработка структуры сайта</title>  
<link href="default.css" rel="stylesheet" type="text/css" >
```

```

</head>
<body>
<table id="maket" cellspacing="2" cellpadding="2" border="1">
<tr height=20%>
<td colspan="3">
  <h1>Пример сайта</h1></td>
<tr height=70%>
<td width="20%">
  <b>-=Меню=-</b><br>
  <a href="page1.php">Страница 1</a><br>
  <a href="page2.php">Страница 2</a><br>
  <a href="page3.php">Страница 3</a><br></td>
<td width="60%">
  Контент
</td>
<td > <h3>Ссылки на сайты </h3>
  <a href="http://www.gstu.by/">Сайт университета</a>
  <br>
  <a href="http://www.ipk.gstu.by/">Сайт ИПК</a>
</tr>
<tr>
<td colspan="3">
  <B>&copy; Иванов И.И., 2013</b> </td>
</tr>
<table>
</body>
</html>

```

Файл **default.css**

```

* {
  margin: 0;
  padding: 0;
}
body {
  font: normal small Arial, Helvetica, sans-serif;
  color: #999999;
  font-size: 1.0em;
}

```

```
h1, h2 {
    margin: 0;
    color: #11A6D4;
}
h2 {
    margin-bottom: .5em;
    font-size: 1.8em;
}
a {
    color: #EA672E;
}
a:hover {
    text-decoration: none;
    color: #11A6D4;
}
#maket {
    height: 100%;
    width: 100%;
    margin: 0 auto;
    text-align:center;
}
```

2. Разбейте созданный макет сайта на блоки **header**, **menu**, **info**, **links**, **footer** и сохраните их в файлах с аналогичным названием с расширениями **php** в корневой папке сайта.

Файл **header.php**

```
<h1>Пример сайта</h1>
```

Файл **menu.php**

```
<b>--Меню--</b><br>
<a href="page1.php">Страница 1</a><br>
<a href="page2.php">Страница 2</a><br>
<a href="page3.php">Страница 3</a><br>
```

Файл **info.php**

`<h1>Содержание контента</h1>`

Файл **links.php**

`<h3>Ссылки на сайты </h3>`

`Сайт университета`

`
`

`Сайт ИПК`

Файл **footer.php**

`© Иванов И.И., 2013`

Замените содержимое файла **index.php**, в котором, с помощью оператора включения **include** добавляются ранее созданные блоки.

Пример включения файла

`<td colspan="3">`

`<?php include ("header.php");?>`

`<tr height=70%>`

3. Создайте папку **images** в корневом каталоге сайта. Скопируйте изображения **img1.png** и **img2.jpeg** в созданную папку. Включите в html-код вставку изображений **img1.png** и **img2.jpeg** в файлы **header.php** и **info.php** соответственно. Проверьте работу сайта.

4. В корневом каталоге сайта создайте папку **Lab1** и скопируйте в нее все файлы созданного сайта. Переименуйте файл **index.php** в файл **main.php**.

В этом файле замените строку кода `<?php include ("info.php");?>` на `<?php echo $content;?>`.

Создайте файл **index.php** следующего содержания:

`<?php`

`$content="<h1>Лабораторная работа №1</h1>`

```
<img src='../images/img2.jpeg'>";  
include ("main.php");  
?>
```

Проверьте работоспособность скрипта.

Контрольные вопросы

1. Какое программное обеспечение необходимо для серверных технологий разработки Web-сайтов?
2. Как запустить, остановить перезапустить web-сервер?
3. Как создать локальный хост?
4. Как внедрить PHP-программу в HTML-документ?
5. С использованием каких операторов включается и выполняется PHP-программа, расположенная в отдельном файле?

ЛАБОРАТОРНАЯ РАБОТА № 2

Тема: «Типы данных в PHP и операторы»

Цель работы: изучить основные типы данных и операторы языка серверного скриптования PHP.

Теоретическая часть: теоретические сведения приведены в электронном конспекте лекций.

Практическая часть:

Задание 1. Использование оператора вывода и комментариев в PHP.

1. В корневом каталоге локального сайта создайте папку **Lab2**. В этой папке создайте файл **z1.php** и поместите в него следующий код:

```
<?php
echo "Меня зовут Иван<br>";
// Это однострочный комментарий
// в стиле C++
echo "Моя фамилия Иванов <br>";
/* Это многострочный комментарий.
Здесь можно написать несколько строк.
При исполнении программы все, что
находится здесь (закомментировано),
будет игнорировано. */
echo "Я изучаю PHP";
# Это комментарий в стиле
# оболочки Unix
```

?>

2. Просмотрите в браузере исходный код динамически сформированной страницы.

Задание 2. Определение констант в PHP.

Создайте файл **z2.php** и поместите в него следующий код:

```
<?php
define("PASSWORD","qwerty"); // определяем константу
PASSWORD
define("PI","3.14", True); // определяем регистронезависимую
//константу PI со значением 3.14
echo (PASSWORD);
// будет выведено значение константы PASSWORD
echo constant("PASSWORD"); // тоже будет выведено значение
echo (password);
/* выведет слово password, поскольку мы ввели
регистрозависимую константу PASSWORD */
echo pi; // выведет значение 3.14, поскольку константа PI
// регистронезависимая по определению
?>
```

Задание 3. Задание переменных по значению.

Создайте файл **z3.php** и поместите в него следующий код:

```
<?php
$first = ' Text '; // Присваиваем $first значение ' Text '
```

```

$second = $first; // Присваиваем $second значение переменной
$first
$first = ' New text '; // Изменяем значение $first на ' New text '
echo "Переменная с именем first равна $first <br>";
// выводим значение $first
echo "Переменная с именем second равна $second";
// выводим значение $second
?>

```

Задание 4. Задание переменных по ссылке и динамических переменных.

Создайте файл **z4.php** и поместите в него следующий код:

```

<?php
$first = ' Text '; // Присваиваем $first значение ' Text '
$second = &$first; /* Делаем ссылку на $first через $second.
/* Теперь значения этих переменных будут всегда совпадать */
echo "Переменная с именем second равна $second <br>";
// Изменим значение $first на ' New text '
$first = ' New text ';
echo "Переменная с именем first равна $first <br>";
// выведем значения второй переменной
echo "Переменная с именем second равна $second<br>";
//Создание динамических переменных
$client = "user"; //Присваиваем $client значение "user"
$$client = "Nick"; // Создаем динамическую переменную $user =
"Nick"

```

```
echo "1) $client<br>"; //выводится user
echo '2) $client<br>'; //выводится $client (одинарные кавычки!)
echo "3) $user<br>"; //выводится Nick
echo "4) $$client<br>"; //выводится $user
echo "5) ",$client; //выводится Nick
?>
```

Задание 5. Ознакомиться с основными типами данных.

Создайте файл **z5.php** и поместите в него следующий код:

```
<?php
$i = 6; // целое (integer)
$I = -526; // целое, переменные i и I разные
$x = 4.89; // число с плавающей точкой (float)
$y = 13.2E-2; // число с плавающей точкой в экспоненциальной
форме
$str1 = "PHP для всех!"; // строка
$str2 = 'Строка в одинарных кавычках!'; // строка
echo "Вывод переменных: i=$i, I=$I";
echo "<br>Вывод переменных x и y: ",$x, " ", "$y;
echo "<br>$str2";
echo '<br>Вывод строки : i=$i, I=$I';
echo "<br>Привет, мир! ".$str1;
?>
```

Обратите внимание на вывод данных при использовании двойных и одинарных кавычках.

Задание 6. Ознакомиться с арифметическими операциями в PHP.

Создайте файл **z6.php** и поместите в него следующий код:

```
<?php
$a= 5;
$b =3;
```

```

$x = 1.2;
$y = 3.6;
$str1 = "5";
$str2 = "3рубля";
$str3="Рублей 10";
echo "<br>a+b=", $a+$b;
echo "<br>a*x=", $a*$x;
echo "<br>y/b=", $y/$b;
echo "<br>Деление по модулю a на b (остаток от целочисленного
деления) =", $a%$b;
echo "<p><center>Использование в арифметических операциях
переменных строкового типа</center>";
echo "<br>str1+b=", $str1+$b;
echo "<br>str1*x=", $str1*$x;
echo "<br>str2/b=", $str2/$b;
echo "<br>str3+x=", $str3+$x;
echo "<br>str2+str3=", $str2+$str3;
echo "<br>str3+str3=", $str3+$str3;
echo "<p><center>Использование операций инкремента и
декремента</center>";
echo "<br>a", $a;
echo "<br>a++=", $a++;
echo "<br>a=", $a;
echo "<br>++a=", ++$a;
echo "<br>a=", $a;
echo "<p><center>Использование в арифметических операциях
переменных логического типа</center>";
$log1=true;
$log2=false;
echo "<br>log1=$log1";
echo "<br>log2=$log2";
echo "<br>>true+true=", true+true;
echo "<br>true+false=", true+false;
echo "<br>false+false=", false+false;
echo "<p><center>Приведение типов</center>";
echo "<br>Приведение к целочисленному типу x=", (int)$x;
echo "<br>Приведение к вещественному типу", (float)$a;
echo "<br>Приведение к логическому типу", (bool) $y;
?>

```

Индивидуальные задания

Во всех скриптах, в заголовке окна браузера должны быть ваши фамилия и имя!!!

Задание 1.

1. Создайте php-скрипт **lab2-1.php**, в котором должны быть переменная `$breakfast` со значением «gamburger» и динамическая переменная `$$breakfast` со значением «and tea».

Объясните, что и почему будет выведено на экран в случаях (а), (б), (в):

(а) `echo "$breakfast $gamburger";`

(б) `echo "$breakfast"; echo "$$breakfast";`

(в) `echo $breakfast; echo $$breakfast;`

2. В этом же скрипте **lab2-1.php** создайте переменную `$breakfast2` являющуюся ссылкой на переменную `$breakfast` и переменные `$color1`, `$color2` для двух значений цвета (на ваш выбор). Отобразите на экране значение переменной `$breakfast2` цветом `$color1` до, а затем цветом `$color2` после присвоения переменной `$breakfast` значения «pizza». Объясните результат.

Задание 2.

1. В скрипте **lab2-2.php** создайте константу `NUM_E` со значением 2.71828 (число e).

2. Выведите на экран значение этой константы в виде строки «Число e равно ...».

3. Присвойте переменной `$num_e1` значение константы и отобразите тип переменной `$num_e1`.

4. Последовательно измените тип переменной `$num_e1` на строковый, целый и булевский; отобразите тип и получившееся значение переменной `$num_e1`.

Задание 3.

В соответствии с вариантом необходимо написать PHP-скрипт (**lab2-3.php**) для вычисления значения переменной `b`. Исходные данные задаются как константы. При выводе информации

предусмотреть форматирование документа, вывод текста задания на лабораторную работу, включая рисунок исходного выражения, и вывод информации о разработчике скрипта. (Список математических функций, необходимых для вычисления арифметического выражения приведен в лабораторной работе 4).

Вариант	Выражение	Вариант	Выражение
1	$b = \frac{1 + \cos^2(x+z)}{ x^3 - 2y^2 }$	16	$b = x + \frac{\sqrt[3]{zy}}{y + \cos x}$
2	$b = \frac{\ln^2 z }{\sqrt[3]{ x + y }}$	17	$b = \lg\left(\sqrt{e^{x-y} + x^{ y } + z}\right)$
3	$b = \frac{y^3}{x + y^3 \cos^2 z}$	18	$b = 1 + \frac{x^2 + 1}{3 + y^2} + \sin 2z$
4	$b = \sqrt{x + \sqrt[4]{ y }} + \cos^2 z$	19	$b = \cos x + \cos y + 2 \sin^2 z$
5	$b = \frac{\sqrt[3]{e^{\sin x}} \cdot \cos y}{z^2 + 1}$	20	$b = \frac{\ln(y^3)(z - x/2)}{2 \cos^2 x}$
6	$b = z(\operatorname{tg} y - e^{-(x+3)})$	21	$b = \sqrt{10(\sqrt[3]{z} + x^{(y+2)})}$
7	$b = x - y (\sin^2 z + \operatorname{tg} z)$	22	$b = (\sin z)^2 + x + y $
8	$b = \sqrt{y + \sqrt[3]{x}} - 1 + 2z$	23	$b = e^{2z} - \sqrt[3]{y x }$
9	$b = x(\operatorname{tg} z + \cos^2 y)$	24	$b = e^{(x-1)} + \sin y$
10	$b = e^{ x-y }(\operatorname{tg}^2 z + 1)^x$	25	$b = \sqrt{ z }e^{-(y+x/2)}$
11	$b = \cos^2 z + \operatorname{tg} 2x + y $	26	$b = \frac{4y^2 e^{2x} \sin^2 z}{3z^3 + \ln x}$
12	$b = 5 \operatorname{tg} z - 4y^2 + xy $	27	$b = \frac{\sqrt{y \ln x - z x^2}}{1 + \operatorname{tg}^2 x^2} x$
13	$b = (z-x) \frac{y - \ln z}{1 + (y-x)^2}$	28	$b = \frac{\lg(y + \sqrt{z + x^2})}{y + x^2}$
14	$b = y^z + \sqrt{ x + y }$	29	$b = \frac{x^2 + 4}{\sin^2 z^2 + x/2} y$
15	$b = \frac{\lg(\sqrt{x} + \sqrt{y} + 2)}{ 2z }$	30	$b = \frac{\sin x + \sqrt{ z-y }}{y(x-2) + x^2}$

Контрольные вопросы

6. Как в PHP выводятся данные?
7. В чем разница вывода данных, заключенных в одинарные и двойные кавычки?
8. Как задается константа?
9. Как задается переменная?
10. Как вывести значение переменной?
11. Как создается динамическая переменная?
12. Какие типы данных определены в PHP?
13. Как преобразовать данные к заданному типу?
14. Какие арифметические операторы используются в PHP?
15. Как обозначается операция конкатенация (склейки) строк?

ЛАБОРАТОРНАЯ РАБОТА № 3

Тема: «Основные конструкции языка PHP»

Цель работы: изучить основные конструкции языка PHP.

Теоретическая часть: теоретические сведения приведены в электронном конспекте лекций.

Практическая часть:

Операторы условного перехода

Задание 1.

В корневом каталоге сайта создайте папку **Lab3** и в ней файл **z1.php**, содержащий следующий скрипт, демонстрирующий возможности использования условного оператора **IF**:

```
<?php
echo "<center>Использование оператора <b>IF</b></center>";
$age=24;
if ($age<18) echo "Вы слишком молоды для просмотра этого
сайта";// ничего не выводится
$age=10;
if ($age<18) echo "Вы слишком молоды для просмотра этого
сайта";// выводится сообщение

echo "<center>Использование оператора <b>IF</b> с блоком
<b>ELSE</b></center>";
$var = "плохо";
if ($var == "хорошо")
echo "Я в хорошем настроении!";
else echo "Мне $var";

echo "<center>Использование оператора <b>IF</b> с блоками
<b>ELSE</b> и <b>ELSEIF</b> </center>";
if ($var == "хорошо") {echo "Я в хорошем настроении!";} //после
закрывающейся скобки ; можно не ставить
```

```
elseif ($var == "плохо") {echo "Не отчаиваться!";}
else {echo "Непонятно, просто $var";}
```

```
//Пример нахождения большего среди трех чисел
echo "<center>Пример использования оператора IF</center>";
$a=1;$b=22;$c=3;
if ($a>$b&&$a>$c) echo "Наибольшее среди трех чисел - a=", $a;
elseif ($b>$c) echo "Наибольшее среди трех чисел - b=", $b;
else echo "Наибольшее среди трех чисел - c=", $c;
```

```
echo "<br>Наибольшее среди трех чисел a,b,c - ", max($a,$b,$c);
```

```
?>
```

Задание 2.

Создайте файл **z2.php**, содержащий скрипт, демонстрирующий возможности использования оператора **SWITCH**:

```
<?php      echo      "<center>Использование      оператора
<b>SWITCH</b></center>";
    $age =23;
    $tmp=$age;
    if ($age<10 || $age>20) $f=$age%10;
    else $f=5;
    switch ($f) {
    case 1: echo "Вам ",$age, " год"; break;
    case 2: echo "Вам ",$age, " года"; break;
    case 3: echo "Вам ",$age, " года"; break;
    case 4: echo "Вам ",$age, " года"; break;
    default:
    echo "Вам ",$age, " лет";}
?>
```

Задание 3.

Создайте файл **z3.php**, содержащий скрипт, демонстрирующий возможности использования оператора условия **?: :**

```

<?php echo "<center>Использование условного оператора
<b>?:<b></center>";
$a =23;
$b=25;
$max=max($a,$b);
echo "Наибольшее среди двух чисел - ",$max;
//Или
if ($a>$b) echo "<br>Наибольшее среди двух чисел - ",$a;
else echo "<br>Наибольшее среди двух чисел - ",$b;
//Или
$max2=($a>$b) ? $a:$b;
echo "<br>Наибольшее среди двух чисел - ",$max2;
?>

```

Операторы цикла

Задание 4.

Создайте файл **z4.php**, содержащий скрипт, демонстрирующий использование операторов цикла **for**, **while**, **do while**:

```

<?php
echo "<center>Использование оператора цикла
<b>FOR</b></center>";
//Возведение числа x в степень
$x=2;//заданное число
$y=10;//степень, в которую надо возвести число
$z=$x;//результат
for ($i=2;$i<=abs($y);$i++)
    $z=$z*$x;
$z=($y>0)?$z:1/$z;
echo "Число ",$x," в степени ",$y," равно ",$z;

echo "<center>Использование оператора цикла
<b>WHILE</b></center>";
$z1=$x;
$i=2;
while ($i<=abs($y)) {
    $z1=$z1*$x;
    $i++;}

```

```

$z1=($y>0)?$z1:1/$z1;
echo "Число ",$x," в степени ",$y," равно ",$z1;

echo "<center>Использование оператора цикла <b>DO
WHILE</b></center>";
$z2=$x;
$i=2;
do {
$z2=$z2*$x;
$i++;}
while ($i<=abs($y));
$z2=($y>0)?$z2:1/$z2;
echo "Число ",$x," в степени ",$y," равно ",$z2;
?>

```

В языке PHP есть ещё оператор цикла **foreach** (для каждого), специально предназначенный для работы с массивами, который будет рассмотрен в лабораторной работе № 5.

Индивидуальные задания

Во всех скриптах, в заголовке окна браузера должны быть ваши фамилия и имя!!!

Задание 1.

Пусть в скрипте **lab3-1.php** задана переменная \$lang, которая может принимать значения «ru», «en», «fr» или «de». Используя операторы **if-else-elseif** обеспечьте вывод на экран полного названия языка (русский, английский и т.д.) в зависимости от того, что присвоено переменной \$lang. Обязательно предусмотрите случай неверного задания значения переменной \$lang - тогда должна выводиться надпись «Язык неизвестен».

Задание 2.

Сделайте тоже самое, что в задании 1, но используя оператор **switch** (скрипт **lab3-2.php**).

Задание 3.

В соответствии со своим вариантом необходимо написать PHP-скрипт (**lab3-3.php**) для вычисления значения функции y . В выводимом файле предусмотреть форматирование документа, вывод текста задания на лабораторную работу, включая рисунок исходной функции, и вывод информации о разработчике скрипта.

Вариант	Вид функции
1	$y = \begin{cases} 1/x, & \text{если } x \geq -5, x \neq 0 & (1) \\ x^2, & \text{если } x \leq -10 & (2) \\ \sqrt{ x+1 } & \text{в ост. случаях} & (3) \end{cases}$
2	$y = \begin{cases} x^2, & \text{если } x \leq 0, x \neq -10 & (1) \\ \sqrt{x+1}, & \text{если } x > 1 & (2) \\ 1/x & \text{в ост. случаях} & (3) \end{cases}$
3	$y = \begin{cases} x + e^{2x}, & \text{если } x \leq 0, x \neq -1 & (1) \\ \cos^2 x, & \text{если } 0 < x \leq 3,14 & (2) \\ x & \text{в ост. случаях} & (3) \end{cases}$
4	$y = \begin{cases} x^3, & \text{если } x > 5, x \neq 20 & (1) \\ x^2, & \text{если } -5 \leq x \leq 5 & (2) \\ \lg x & \text{в ост. случаях} & (3) \end{cases}$
5	$y = \begin{cases} \sqrt{x}, & \text{если } x \geq 100, x \neq 105 & (1) \\ \sqrt[3]{x}, & \text{если } x = 20 \text{ или } x = 40 & (2) \\ x^2 + 1 & \text{в ост. случаях} & (3) \end{cases}$
6	$y = \begin{cases} \sqrt{x-1}, & \text{если } x \geq 10, x \neq 20 & (1) \\ 1/x + e^{2x}, & \text{если } x < 0 & (2) \\ \ln(x+1) & \text{в ост. случаях} & (3) \end{cases}$
7	$y = \begin{cases} 8x+1, & \text{если } x \geq 5, x \neq 9 & (1) \\ x^2 + x , & \text{если } x \leq 1 & (2) \\ x^3 + \sqrt{x} & \text{в ост. случаях} & (3) \end{cases}$
8	$y = \begin{cases} 1-3x, & \text{если } x > 0, x \neq 8 & (1) \\ x^2 - \sin x, & \text{если } x \leq -1 & (2) \\ \cos x & \text{в ост. случаях} & (3) \end{cases}$

Вариант	Вид функции
9	$y = \begin{cases} x^3 + 1, & \text{если } x \geq 8, x \neq 10 & (1) \\ 2x^2 + \sqrt[3]{ x }, & \text{если } x \leq 1 & (2) \\ \sqrt{x} & \text{в ост. случаях} & (3) \end{cases}$
10	$y = \begin{cases} 2x^2, & \text{если } x > 0, x \neq 3 & (1) \\ \sqrt{x^2 + 1} & \text{если } x \leq -2 & (2) \\ x + 5 & \text{в ост. случаях} & (3) \end{cases}$
11	$y = \begin{cases} \sqrt{ 2x - x^2 - 1 }, & \text{если } x \leq -1, x \neq -4 & (1) \\ \ln(x + 3), & \text{если } x > 0 & (2) \\ x/2 & \text{в ост. случаях} & (3) \end{cases}$
12	$y = \begin{cases} \sqrt{x}, & \text{если } x \geq 4, \text{ или } x = 1 & (1) \\ \ln x + 1 , & \text{если } x \leq -2 & (2) \\ e^{-x} & \text{в ост. случаях} & (3) \end{cases}$
13	$y = \begin{cases} x/3, & \text{если } -3 \leq x \leq 3 & (1) \\ \lg(x^2 + 1), & \text{если } x < -3 \text{ или } x = 4 & (2) \\ \sqrt{x^3 - 2} & \text{в ост. случаях} & (3) \end{cases}$
14	$y = \begin{cases} x^3 + 4 , & \text{если } x \leq -1 \text{ или } x = 0 & (1) \\ \sqrt{x/2}, & \text{если } x \geq 8 & (2) \\ x^3 & \text{в ост. случаях} & (3) \end{cases}$
15	$y = \begin{cases} \sqrt{x + 1}, & \text{если } x \geq 8, x \neq 10 & (1) \\ 0,6x, & \text{если } 0 < x < 8 & (2) \\ \lg x + 3 & \text{в ост. случаях} & (3) \end{cases}$

Задание 4.

В скрипте **lab3-4.php**, используя вложенные циклы **for**, отобразите на экране таблицу 10×10, в ячейках которой идут числа от 1 до 100.

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

При этом красным цветом выделены «треугольные» числа, т.е. числа вида $n(n+1)/2$ (где $n=1,2,\dots, 14$). Все другие числа черного цвета. Массивы **не использовать!** Ширина рамки таблицы равна 1, отступ содержимого ячеек от границы равен 5.

Задание 5.

В скрипте **lab3-5.php** вычислить и вывести на экран значения функции, используя стандартные функции PHP и с помощью разложения функции в ряд Тейлора. При написании PHP-скрипта воспользоваться оператором цикла **do ... while**.

Вариант	Разложение функции в ряд Тейлора	
1.	$\ln \frac{x+1}{x-1} = 2 \sum_{n=0}^{\infty} \frac{1}{(2n+1)x^{2n+1}} = 2 \left(\frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + \dots \right)$	$ x > 1$
2.	$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} - \dots$	$ x < \infty$
3.	$e^{-x} = \sum_{n=0}^{\infty} \frac{(-1)^n x^n}{n!} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \dots$	$ x < \infty$
4.	$\ln(x+1) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{n+1}}{n+1} = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} - \dots$	$-1 < x \leq 1$
5.	$\ln \frac{1+x}{1-x} = 2 \sum_{n=0}^{\infty} \frac{x^{2n+1}}{2n+1} = 2 \left(x + \frac{x^3}{3} + \frac{x^5}{5} + \dots \right)$	$ x < 1$
6.	$\ln(1-x) = - \sum_{n=1}^{\infty} \frac{x^n}{n} = - \left(x + \frac{x^2}{2} + \frac{x^3}{3} + \dots \right)$	$-1 \leq x \leq 1$
7.	$\operatorname{arctg} x = \frac{\pi}{2} + \sum_{n=0}^{\infty} \frac{(-1)^{n+1} x^{2n+1}}{2n+1} = \frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5} - \dots$	$ x \leq 1$
8.	$\operatorname{arctg} x = \frac{\pi}{2} + \sum_{n=0}^{\infty} \frac{(-1)^{n+1}}{(2n+1)x^{2n+1}} = \frac{\pi}{2} - \frac{1}{x} + \frac{1}{3x^3} - \frac{1}{5x^5} + \dots$	$x > 1$
9.	$\operatorname{arctg} x = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)} = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$	$ x \leq 1$
10.	$\operatorname{Arth} x = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{2n+1} = x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots$	$ x < 1$
11.	$\operatorname{Arth} x = \sum_{n=0}^{\infty} \frac{1}{(2n+1)x^{2n+1}} = \frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + \dots$	$ x > 1$
12.	$\operatorname{arctg} x = - \frac{\pi}{2} + \sum_{n=0}^{\infty} \frac{(-1)^{n+1}}{(2n+1)x^{2n+1}} = - \frac{\pi}{2} - \frac{1}{x} + \frac{1}{3x^3} - \frac{1}{5x^5} + \dots$	$x < -1$
13.	$e^{-x^2} = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{n!} = 1 - x^2 + \frac{x^4}{2!} - \frac{x^6}{3!} + \frac{x^8}{4!} - \dots$	$ x < \infty$
14.	$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$	$ x < \infty$
15.	$\frac{\sin x}{x} = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n+1)!} = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \frac{x^6}{7!} - \dots$	$ x < \infty$

Контрольные вопросы

1. Как операторы управления вы знаете?
2. Для чего используется оператор **break** в операторе **switch**?
3. Как записать сложное условие?

4. В чем разница между ключевыми словами **else** и **elseif**?
5. Какие операторы цикла вы знаете?
6. В чем отличие операторов цикла **while** и **do while**?
7. Что записывается в заголовке оператор цикла **for**?
8. Как инициализируется счетчик цикла?
9. Как задается шаг изменения счетчика цикла?
10. Как принудительно выйти из цикла?

ЛАБОРАТОРНАЯ РАБОТА № 4

Тема: «Использование функций в PHP»

Цель работы: научиться использовать стандартные и создавать пользовательские функции в PHP.

Теоретическая часть: основные теоретические сведения приведены в электронном конспекте лекций.

Дополнительно:

Список математических функций, используемых в PHP

abs — Модуль числа
acos — Арккосинус
acosh — Гиперболический арккосинус
asin — Арксинус
asinh — Гиперболический арксинус
atan — Арктангенс
atanh — Гиперболический арктангенс
ceil — Округляет дробь в большую сторону
cos — Косинус
cosh — Гиперболический косинус
deg2rad — Преобразует значение из градусов в радианы
exp — Вычисляет число e в степени
floor — Округляет дробь в меньшую сторону
fmod — Возвращает дробный остаток от деления по модулю
log10 — Десятичный логарифм
log — Натуральный логарифм
max — Возвращает наибольшее значение
min — Находит наименьшее значение
pi — Возвращает число π
pow — Возведение в степень
rad2deg — Преобразует значение из радианов в градусы
rand — Генерирует случайное число
round — Округляет число типа float
sin — Синус

`sinh` — Гиперболический синус
`sqrt` — Квадратный корень
`srand` — Изменяет начальное число генератора псевдослучайных чисел
`tan` — Тангенс
`tanh` — Гиперболический тангенс

Для вызова функции необходимо указать её имя и в круглых скобках передать фактические параметры. Например:

```
echo pow(2, 8); //будет выведено число 256
```

Все математические функции возвращают значение, которое может использоваться в качестве аргумента другой функции, либо в качестве значения переменной.

Практическая часть:

Создание пользовательских функций

Задание 1.

В корневом каталоге локального сайта создайте папку **Lab4** и в ней файл **z1.php**, демонстрирующий создание пользовательской функции с аргументами и обращение к ней из основной части скрипта:

```
<?php
//Описание функции вывода строки с переходом на новую
function PrintBR($txt) {
print ("$txt<br>\n");
}
//Обращение к функции с передачей фактических параметров
PrintBR("Это первая строка");
PrintBR("Это следующая строка");
PrintBR("Это еще одна строка");
?>
```

Протестируйте работу скрипта.

Задание 2.

Создайте файл **z2.php**, демонстрирующий создание пользовательской функции, возвращающей значение:

```
<?php
//Описание функции, возвращающей значение
function AddNums($firstnum, $secondnum) {
    $result = $firstnum + $secondnum;
    return $result;
}
echo "3+5=".AddNums(3,5); //будет выведено 8
$a=5;
$b=25;
echo "<br>$a+$b=".AddNums($a,$b); //будет выведено 30
?>
```

Задание 3.

Создайте файл **z3.php**, демонстрирующий использование функций-переменных в PHP:

```
<?php
// Описание функции с приветствием на русском языке
function Russian($name) {
    echo "<p><b>Здравствуйте, $name !";
}
// Описание функции с приветствием на английском языке
function English($name) {
    echo "<p><b>Hello, $name!";
}
$language = "Russian"; // Выбрали русский язык
$name="Вася";
$language($name); // Выполнение функции-переменной
$language = "English"; // Выбрали английский язык
$name="Mr.Smit";
$language($name); // Выполнение функции-переменной
?>
```

Задание 4.

Создайте файл **z4.php**, демонстрирующий использование функции с необязательным параметром:

```
<?php
function FontSize($txt, $size=12) {
echo "<div style=\"font-size:\".$size.\"pt\">$txt</div>";
}
FontSize("<p>Крупный шрифт",16);
FontSize("<p>Нормальный шрифт, первая строка");
FontSize("<p>Нормальный шрифт, вторая строка");
FontSize("<p>Мелкий шрифт",8);
?>
```

Задание 5.

Создайте файл **z5.php**, демонстрирующий использование функции с передачей параметра по значению и по ссылке:

```
<?php
function AddFive($num) {
$num +=5;
}
$var = 10;
AddFive($var); //передача параметра по значению
echo "<br>".$var; // выводится 10
AddFive(&$var); //передача параметра по значению
echo "<br>".$var; // выводится 15
?>
```

Индивидуальные задания

Во всех скриптах, в заголовке окна браузера должны быть ваши фамилия и имя!!!

Задание 1.

В скрипте **lab4-1.php** создайте 4 функции с именами Ru(), En(), Fr(), De(). Каждая функция выводит на экран приветствие на соответствующем языке:

Ru() - «Здравствуйте!»,

En() - «Hello!»,

Fr() - «Bonjour!»,

De() - «Guten Tag!».

Эти функции имеют аргумент \$color, который определяет цвет выводимого текста. Используя функцию-переменную \$lang(), отобразите на экране одно из приветствий, причем какое приветствие будет выведено и каким цветом - задать как параметры в строке вызова скрипта:

Lab4-1.php?lang=Ru&color=[назв-е цвета](En/Fr/De)

Если настройки PHP запрещают обращаться к переменным, переданным в заголовке запроса по имени, измените значение параметра **register_globals** на **On** в конфигурационном файле **php.ini**.

Задание 2.

В скрипте **lab4-2.php**, используя аргументы \$size, \$day и \$color функции WeekDay(), отобразите названия дней недели уменьшающимся размером (начиная с 7) и разными цветами:

понедельник

вторник

среда

четверг

пятница

суббота

воскресенье

Задание 3.

В соответствии со своим вариантом напишите скрипт **lab4-3.php** для вычисления значения функции **b=f(x,y,z)**. Значения x, y и z

должны передаваться в адресной строке. При выводе информации предусмотреть форматирование документа, вывод текста задания, включая рисунок исходной функции, и вывод информации о разработчике скрипта.

Вариант	Вид функции	Вариант	Вид функции
1	$b = \frac{1 + \cos^2(x+z)}{ x^3 - 2y^2 }$	16	$b = x + \frac{\sqrt[3]{zy}}{y + \cos x}$
2	$b = \frac{\ln^2 z }{\sqrt[3]{ x + y }}$	17	$b = \lg\left(\sqrt{e^{x-y} + x^{ y } + z}\right)$
3	$b = \frac{y^3}{x + y^3 \cos^2 z}$	18	$b = 1 + \frac{x^2 + 1}{3 + y^2} + \sin 2z$
4	$b = \sqrt{x + \sqrt[4]{ y }} + \cos^2 z$	19	$b = \cos x + \cos y + 2 \sin^2 z$
5	$b = \frac{\sqrt[3]{e^{\sin x}} \cdot \cos y}{z^2 + 1}$	20	$b = \frac{\ln(y^3)(z - x/2)}{2 \cos^2 x}$
6	$b = z(\operatorname{tg} y - e^{-(x+3)})$	21	$b = \sqrt{10(\sqrt[3]{z} + x^{(y+2)})}$
7	$b = x - y (\sin^2 z + \operatorname{tg} z)$	22	$b = (\sin z)^2 + x + y $
8	$b = \sqrt{y + \sqrt[3]{x}} - 1 + 2z$	23	$b = e^{2z} - \sqrt[3]{y x }$
9	$b = x(\operatorname{tg} z + \cos^2 y)$	24	$b = e^{(x-1)} + \sin y$
10	$b = e^{ x-y }(\operatorname{tg}^2 z + 1)^x$	25	$b = \sqrt{ z }e^{-(y+x/2)}$
11	$b = \cos^2 z + \operatorname{tg} 2x + y $	26	$b = \frac{4y^2 e^{2x} \sin^2 z}{3z^3 + \ln x}$
12	$b = 5 \operatorname{tg} z - 4y^2 + xy $	27	$b = \frac{\sqrt{y \ln x - z x^2}}{1 + \operatorname{tg}^2 x^2} x$
13	$b = (z - x) \frac{y - \ln z}{1 + (y - x)^2}$	28	$b = \frac{\lg(y + \sqrt{z + x^2})}{y + x^2}$
14	$b = y^z + \sqrt{ x + y }$	29	$b = \frac{x^2 + 4}{\sin^2 z^2 + x/2} y$
15	$b = \frac{\lg(\sqrt{x} + \sqrt{y} + 2)}{ 2z }$	30	$b = \frac{\sin x + \sqrt{ z - y }}{y(x - 2) + x^2}$

Контрольные вопросы

1. Что представляет из себя функция?
2. Как создать пользовательскую функцию?
3. Что такое формальные и фактические параметры?
4. Чем отличается описание функций, не возвращающих и возвращающих значения?
5. В чем отличие передачи параметров по значению и по ссылке?
6. Как создается функция-переменная?
7. Чем отличаются глобальные и локальные переменные?
8. Как значение параметра по умолчанию?

ЛАБОРАТОРНАЯ РАБОТА № 5

Тема работы: «Работа с массивами в PHP»

Цель работы: изучить возможности PHP для создания и обработки одномерных и двумерных массивов.

Теоретическая часть: теоретические сведения приведены в электронном конспекте лекций.

Практическая часть:

Создание массивов

Задание 1.

В корневом каталоге локального сайта создайте папку **Lab5** и в ней файл **z1.php**, содержащий следующий скрипт:

```
<?php

echo "<center><b>Создание массива с помощью функции
array()</b></center>";
$colors = array("red", "green", "blue", "gray");
print_r($colors);
echo "<br>";
var_dump($colors);

echo "<p><center><b>Создание элементов массива с помощью
идентификатора</b></center>";
$colors1[]="red";
$colors1[]="green";
$colors1[]="blue";
$colors1[]="gray";
print_r($colors1);

echo "<p><center><b>Создание ассоциативного массива с
помощью функции array()</b></center>";
```

```
$adres = array ("name" => "Иванов", "street" => "Ильича", "home"
=> 12);
print_r($adres);
```

```
echo "<p><center><b>Непосредственное создание
ассоциативного массива</b></center>";
$adres1["name"]="Иванов";
$adres1["street"]="Ильича";
$adres1["home"]=12;
print_r($adres1);
```

```
echo "<p><center><b>Создание многомерного
массива</b></center>";
$adres3=array(
    array ("name" => "Иванов",
           "street" => "Ильича",
           "home" => 12),
    array ("name" => "Петров",
           "street" => "Ленина",
           "home" => 22),
    array ("name" => "Сидоров",
           "street" => "Ульянова",
           "home" => 32)
);
print_r($adres3);
```

```
echo "<p><center><b>Обращение к элементам
массива</b></center>";
echo "<br>Второй элемент массива color - ".$colors[1];
//индексация начинается с 0
echo "<br>Первый элемент ассоциативного массива adres -
".$adres["name"];
echo "<br>Третий элемент многомерного ассоциативного
массива adres3 - ".$adres3[2]["street"];
?>
```



```

echo "<center><b>Операции сложения массивов</b></center>";
$a = array("и"=>"Информатика", "м"=>"Математика");
$b = array("и"=>"История", "м"=>"Биология", "ф"=>"Физика");
$c = $a + $b;
$d = $b + $a;
print_r($c);
echo "<br>";
print_r($d);
echo "<br>";

```

```

echo "<center><b>Сравнение массивов</b></center>";
$a = array("и"=>"Информатика", "м"=>"Математика");
echo "Исходный массив А<br>";
foreach($a as $key=>$elem) echo
"[$key]=>$elem    ";
echo "<br>";
$b = array("м"=>"Математика", "и"=>"Информатика");
echo "Исходный массив В<br>";
foreach($b as $key=>$elem) echo
"[$key]=>$elem    ";
echo "<br>";
if ($a == $b) echo "Массивы равны и";
else echo "Массивы НЕ равны и ";
if ($a === $b) echo " эквивалентны";
else echo " НЕ эквивалентны";

```

```

echo "<center><b>Функция поиска значения в массиве
array_search</b></center>";
$langs = array("Lisp", "Python", "Java", "PHP", "Perl");
echo "Исходный массив <br>";
foreach($langs as $key=>$elem) echo
"[$key]=>$elem    ";
echo "<br>";

if (!array_search("C#", $langs))
echo "Может быть надо изучить C#<br>";
$k = array_search("PHP", $langs);
echo "PHP стоит на $k-м";

```

```
echo "<center><b>Объединение массивов функцией  
array_merge</b></center>";
```

```
$a = array(1,2,3);  
echo "Исходный массив A<br>";  
foreach($a as $key=>$elem) echo  
"[$key]=>$elem    ";  
echo "<br>";  
$b = array(4,5,6,7);  
echo "Исходный массив B<br>";  
foreach($b as $key=>$elem) echo  
"[$key]=>$elem    ";  
echo "<br>";
```

```
echo "Объединение массивов A и B<br>";  
$c = array_merge($a, $b);  
echo "Результирующий массив C<br>";  
foreach($c as $key=>$elem) echo  
"[$key]=>$elem    ";  
echo "<br>";
```

```
echo "<center><b>Выделение части массива с использованием  
функции array_slice</b></center>";
```

```
$d=array_slice($c,2,3);  
echo "Массив D, выделенный из массива C<br>";  
foreach($d as $key=>$elem) echo  
"[$key]=>$elem    ";  
echo "<br>";
```

```
echo "<center><b>Сортировка массива</b></center>";  
echo "<center>Сортировка по возрастанию элементов</center>";
```

```
$a=$langs;  
sort($a);  
foreach($a as $key=>$elem) echo  
"[$key]=>$elem    ";  
echo "<br>";  
$adres = array ("name" => "Иванов", "street" => "Ильича", "home"  
=> 12);
```

```
echo "Массив до сортировки<br>";
foreach($adres as $key=>$selem) echo
"[$key]=>$selem    ";
echo "<br>";
$a=$adres;
sort($a);
echo "Массив после сортировки<br>";
foreach($a as $key=>$selem) echo
"[$key]=>$selem    ";
echo "<br>";
echo "Обратите внимание что индексы были заменены
цифрами!!!<br>";
```

```
echo "<center>Сортировка по индексу</center>";
$a=$adres;
ksort($a);
echo "Массив после сортировки по индексу<br>";
foreach($a as $key=>$selem) echo
"[$key]=>$selem    ";
echo "<br>";
```

?>

Индивидуальные задания

Во всех скриптах, в заголовке окна браузера должны быть ваши фамилия и имя!!!

Задание 1.

В скрипте **lab5-1.php**:

1. Создайте массив `$treug[]` «треугольных» чисел, т.е. чисел вида $n(n+1)/2$ (где $n=1,2,\dots, 10$) и выведите значения этого массива на экран в строку (через 2 пробела).
2. Создайте массив `$kvad[]` квадратов натуральных чисел от 1 до 10, выведите значения этого массива на экран в строку.
3. Объедините эти 2 массива в массив `$rez[]`, выведите результат на экран.

4. Отсортируйте массив \$rez[] в обратном порядке, выведите результат на экран.

Задание 2.

В скрипте **lab5-2.php**:

1. Сформируйте одномерный массив (не менее 15 элементов, для генерации значений массива можно воспользоваться функцией rand(n1,n2), где n1 и n2 границы диапазона), выведите значения этого массива на экран в строку.

2. Выполните задание в соответствии с вариантом, не используя встроенные функции

Вариант	Задание
1	Найти максимальный элемент и поменять его местами с последним элементом массива
2	Найти минимальный элемент и поменять его местами с предыдущим элементом массива
3	Найти минимальный элемент и поменять его местами с последующим элементом массива
4	Найти максимальный элемент и поменять его местами с шестым элементом массива
5	Найти максимальный элемент, присвоить его значение последнему элементу массива, а вместо максимального числа записать - 1
6	Найти минимальный элемент, присвоить его значение первому элементу массива, а вместо минимального элемента записать число 9999
7	Найти минимальный элемент и поменять его местами с третьим элементом массива
8	Найти минимальный элемент и заменить его на сумму первого и последнего элементов
9	Найти максимальный элемент и поменять его местами с предпоследним элементом массива
10	Найти минимальный элемент, присвоить его значение последнему элементу массива, а вместо минимального элемента записать значение 3N
11	Найти минимальный элемент и поменять его местами с

Вариант	Задание
	элементом массива, номер которого задан
12	Найти максимальный элемент и поменять его местами со вторым элементом массива
13	Найти минимальный элемент и поменять его местами с последним элементом массива
14	Найти минимальный элемент и поменять его местами с предпоследним элементом массива
15	Найти минимальный элемент и поменять его местами с третьим элементом массива

Задание 3.

В скрипте **lab5-3.php**:

1. Создайте двумерный ассоциативный массив `$depart[]` в котором на первом уровне содержатся ключи: ФАИС, ЭФ, МСФ, которые являются сокращенными названиями факультетов ГГТУ им. П.О. Сухого. Эти ключи на втором уровне связаны с ассоциативными массивами, в которых ключами являются кафедра 1, кафедра 2, кафедра 3 и т.д., а значениями ключей — названия кафедр (без слова "кафедра"). Названия заносятся в массивы точно как в таблице.

Факультеты

ФАИС

кафедра 1	Информационные технологии
кафедра 2	Промышленная электроника
кафедра 3	Автоматизированные электроприводы
кафедра 4	Высшая математика
кафедра 5	Информатика

ЭФ

кафедра 1	Электроснабжение
кафедра 2	Промышленная теплоэнергетика
кафедра 3	Иностранные языки
кафедра 4	Теоретические основы электротехники
кафедра 5	Физика

МСФ

кафедра 1	Технология машиностроения
кафедра 2	Металлорежущие станки
кафедра 3	Разработка и эксплуатация нефтяных

	месторождений
кафедра 4	Гидропневмоавтоматика
кафедра 5	Техническая механика

2. С помощью функций сортировки массивов выведите на экран в алфавитном порядке названия факультетов, а под названием каждого факультета — список его кафедр тоже в алфавитном порядке (как маркированный список), причем перед названием кафедры должно быть слово «кафедра».

Контрольные вопросы

1. Что такое массив?
2. Какие массивы используются в PHP?
3. Как индексируются элементы массив?
4. Что такое ассоциативный массив?
5. Как создается многомерный массив?
6. Как просмотреть структуру и значения элементов массива?
7. Как вывести значения элементов массива в окно браузера?
8. Как отсортировать массив по значениям (по возрастанию и убыванию) ?
9. Как отсортировать элементы массива по индексам?
10. Какие стандартные функции обработки массивов вы знаете?

ЛАБОРАТОРНАЯ РАБОТА № 6

Тема: «Работа со строками в PHP»

Цель работы: получить навыки обработки символьной информации в PHP.

Теоретическая часть: теоретические сведения приведены в электронном конспекте лекций.

Практическая часть:

Задание 1. Вывода текстовой информации в PHP

В корневом каталоге локального сайта создайте папку **Lab6**. В этой папке создайте файл **z1.php** и поместите в него следующий код:

```
<?
$x=12;
echo '\Вывод имени переменной $x и кавычек \<br>';
echo "'Вывод значения переменной x=$x и кавычек '<br>";
echo <<<M1
'использование heredoc-синтаксиса
для вывода значения переменной x=$x и кавычек'<br>
M1;
// использовать функцию print
$x=25;
print '\Вывод имени переменной $x и кавычек \<br>';
print "'Вывод значения переменной x=$x и кавычек '<br>";
?>
```

Задание 2. Конкатенация строк (объединение)

Создайте файл **z2.php**, демонстрирующий объединение строк:

```
<?
$s1="мир";
$s2="труд";
```

```
$s3="май";

echo "<h2>".$s1.", ".$s2.", ".$s3."!!!";
?>
```

Задание 3. Поиск элемента в строке

Создайте файл **z3.php**, демонстрирующий поиск элемента в строке:

```
<?
$s1="мир";
$s2="труд";
$s3="май";
$s_rez=$s1.", ".$s2.", ".$s3."!!!";
echo $s_rez."<br>";
$poz=strpos($s_rez,$s2);
echo "Искомая строка начинается с $poz символа<br>";
if ($poz=strpos($s_rez,$s1) !== false) echo "Искомая строка $s1
присутствует в результирующей<br>";
?>
```

Обратите внимание, что для сравнения используется знак **!==** (**не эквивалентность**). Дело в том, что в PHP операнды сравниваются только в том случае, если оба они являются строками. В противном случае они сравниваются как числа. Попробуйте изменить оператор сравнения **!==** на **!=** и посмотреть результаты выполнения скрипта.

Задание 4. Выделение подстроки

Создайте файл **z4.php**, демонстрирующий выделение подстроки:

```
<?
$s1="мир";
$s2="труд";
$s3="май";
$s_rez=$s1.", ".$s2.", ".$s3."!!!";
echo $s_rez."<br>";
$s11=strstr($s_rez,$s2);
```

```

echo "$s11<br>";
$s22=substr($s_rez,5);
$s33=substr($s_rez,5,-8);
echo "s22=$s22<br>";
echo "s33=$s33<br>";
?>

```

Задание 5. Вывод отдельных символов

Если необходимо вывести какой-то символ строки, можно обратиться к нему по индексу:

```
$s_rez[5]; // выведет т.
```

Для вывода одиночных символов используется функция **chr**, аргументом которой является код символа в ASCII-таблице. Функция полезна при вставке в строку непечатаемых символов (например, нулевой код, символ прогона страницы, при работе бинарных файлов).

Создайте файл **z5.php** и поместите в него следующий PHP-скрипт, который выводит ASCII-код и соответствующий ему символ:

```

<?
for ($i=0,$x=0;$x<16;$x++) {
    for ($y=0;$y<16;$y++) {
        $matr[$x][$y]=array($i,chr($i));
        $i++;
    }
}
echo "<table border=1 cellpadding=1 cellspacing=1>";
foreach ($matr as $row) {
    echo "<tr>";
    foreach ($row as $cell) {
        echo "<td>$cell[0]";
        echo "<b><tt>$cell[1]</tt></b></td>";
    }
}
?>

```

Задание 6. Определение длины строки

Создайте файл **z6.php**, демонстрирующий определение строки:

```
<?
$s1="мир";
$s2="труд";
$s3="май";
$s_rez=$s1.", ".$s2.", ".$s3."!!!";
echo $s_rez."<br>";
echo "Длина строки s1 - ",strlen($s1),"<br>";
echo "Длина строки s_rez - ",strlen($s_rez),"<br>";
?>
```

Задание 7. Функции замены

Функция **str_replace(\$s1,\$s2,\$s_ist)** заменяет в строке **\$s_ist** все вхождения подстроки **\$s1** на подстроку **\$s2** и возвращает результат в виде новой строки.

Функция **substr_replace(\$s_ist,\$s2,\$start[\$,\$len])** заменяет в строке **\$s_ist** символы, начиная с **\$start** и длиной **\$len**, на подстроку **\$s2** и возвращает результат в виде новой строки.

Создайте файл **z7.php**, демонстрирующий функции замены:

```
<?
$s1="учиться";
$s2="учит";
$s3="учитель";
$s_rez=$s3.", ".$s2.", ".$s1."!";
echo $s_rez."<br>";
$sr1=str_replace($s2,"мирит",$s_rez);
echo "$sr<br>";
$sr1=substr_replace($s_rez,"мирит",$poz=strpos($s_rez,$s1));
echo "$sr1<br>";
?>
```

Задание 8. Использование регулярных выражений

Создайте файл **z8.php**, демонстрирующий возможности использования регулярных выражений:

```
<?
//Проверяем, является ли строка правильным доменным именем
$str1="http://www.gstu.by";
$pattern1="!^(http|https)://([A-z][A-z0-9_-]*(?:.[A-z0-9][A-z0-9_-
]*)+)!";
if (preg_match($pattern1,$str1))
{echo "OK";}
else
{echo "Bad";}

<?
//строка, в которой нужно что-то найти
$str = "Мой телефонный номер: ".
"33-22-44. Номер моего редактора: ".
"222-44-55 и 323-22-33";
//шаблон, по которому искать, задает поиск семизначных
номеров.
$pattern = "[0-9]{3}-\d{2}-\d{2}/m";
//функция, осуществляющая поиск
$num_match = preg_match_all ($pattern, $str, $result);
//вывод результатов поиска
for ($i=0;$i<$num_match;$i++)
    echo "Совпадение $i: ".
        $result[0][$i]."<br>";

//Проверка E-mail
$str1="sam-nick@gs_u.b.ru";
$pattern="!^[A-Za-z0-9][A-z0-9-]*@[([A-Za-z0-9][A-z0-9-]*\.)+[A-
Za-z]{2,}$!";
if (preg_match($pattern,$str1))
{echo "OK";}
else
{echo "Bad";}
?>
```

Индивидуальные задания

Во всех скриптах, в заголовке окна браузера должны быть ваши фамилия и имя!!!

Задание 1. В соответствии со своим вариантом написать PHP-скрипт **lab6-1.php**, выполняющий следующие действия

Вариант	Задание
01	Подсчитать общее количество символов '+' и '-' и заменить каждый символ ';' на ',' и '.'
02	После каждого символа ';' вставить пробел и подсчитать количество букв 'A' и 'B' отдельно
03	Заменить символ '*' на '++' и подсчитать общее количество букв 'F' и 'D'
04	Подсчитать количество букв 'C' и 'D' отдельно и заменить каждую пару символов '**' на символ '.'
05	После каждого символа '!' вставить символ 'I' и подсчитать общее количество цифр в строке
06	Удалить каждую пару символов 'PQ' и подсчитать общее количество символов '.' и ',' в строке
07	Подсчитать количество пар символов '+ -' и заменить каждый символ '*' на '/-'
08	После каждой цифры вставить такую же цифру и подсчитать количество пар 'AC' в строке
09	Удалить каждый символ 'A', стоящий после ',' и подсчитать количество пар 'BC'
10	Подсчитать количество символов '.' , стоящих перед пробелом, и заменить каждую пару символов 'ST' на символ 'P'
11	После каждого символа 'A' вставить пробел и подсчитать количество символов 'B', стоящих между знаками '+' и '-'
12	Удалить каждый символ '?', стоящий после ';', и подсчитать общее количество символов 'o' и 'O'
13	Подсчитать количество символов '+', стоящих между 'A' и 'B', заменить каждый символ '0' на 'OO'
14	В каждую пару символов 'AB' вставить символ '*', подсчитать, сколько раз в строке символ 'I' стоит перед '2'

15	Вставить символ ';' после каждого символа 'А' и после каждого 'В', подсчитать, сколько раз символ 'С' встречается между символами '*' и '/'
----	---

Задание 2. В соответствии со своим вариантом написать PHP-скрипт **lab6-2.php**, выполняющий следующие действия

Вариант	Задание
01	После каждого слова поставить запятую. Подсчитать количество слов, в которых есть буква 'п'
02	Подсчитать количество букв в третьем слове. Поменять местами первое и последнее слова
03	Во втором слове после каждой буквы вставить пробел. Определить количество слов, которые заканчиваются на 'e'
04	Перед первой буквой каждого слова вставить символ '*'. Определить количество слов, в которых нет ни одной буквы 'г'
05	Для первого слова указать, сколько букв 'и' в нем содержится. Переставить первое слово в конец строки
06	Определить количество слов, начинающихся с буквы 'А'. После каждой буквы предпоследнего слова вставить символ '*'
07	Подсчитать количество букв во втором слове. Каждое слово заключить в кавычки.
08	Подсчитать количество слов, длина которых больше 5. Удалить пробелы, стоящие между первым и вторым словом.
09	Определить количество слов, в которых буква 'и' встречается хотя бы один раз. Поменять местами первое и второе слово
10	Третье слово строки поставить после первого. Определить количество слов, в которых первая и последняя буквы совпадают
11	Определить количество слов, вторая буква которых 'р'. Удалить последнюю букву в каждом слове
12	Подсчитать количество букв в предпоследнем слове. В каждом слове поменять местами первую и последнюю буквы

13	Перед каждой буквой третьего слова поставить '-'. Определить количество слов, после которых один пробел
14	После последней буквы каждого слова вставить точку. Для пятого слова указать, сколько букв 'И' в нем содержится.
15	Удалить все пробелы из строки, кроме тех, которые стоят между первым и вторым словом. Определить количество слов, которые по длине меньше 3

Контрольные вопросы

1. Как выводится текстовая информация в PHP?
2. Для чего используется **heredoc-синтаксис**?
3. Что осуществляет операция конкатенация строк?
4. Какая функция осуществляет поиск заданного элемента в строке?
5. Как определить длину строки?
6. Как из исходной строки выделить подстроку?
7. Как выполнить замену подстроки в исходной строке?
8. Для чего используются регулярные выражения?
9. Какие функции используются в PHP для работы с регулярными выражениями?
10. Как вывести подстроки в соответствии с шаблоном?

ЛАБОРАТОРНАЯ РАБОТА № 7

Тема: «Работа с формами»

Цель работы: изучить возможности PHP по обработке данных HTML-форм.

Теоретическая часть: теоретические сведения приведены в электронном конспекте лекций.

Практическая часть:

1. В корневом каталоге локального сайта создайте папку **Lab7**. В этой папке создайте файл с HTML-формой **lab7.html**, следующего содержания:

```
<html>
<head>
<title>Пример работы с формой данных</title>
</head>
<body>
<!-- localhost должен быть заменен на имя локального сайта!!! -->
<form action="http://localhost/Lab7/form.php" method="POST">

<center>Поля для ввода</center><br>
Поле для ввода текста<input type="text" name="txt"><br>
Поле для ввода пароля<input type="password"
name="pswd"><br>
Скрытое поле<input type="hidden" name="hid" value="Это поле
не отображается"><br>

<center>Текстовая область</center><br>
<textarea name="t_area" cols="20" rows="5"
wrap="virtual">Область текста шириной в 20 символов и с 5
строками</textarea><br>
<hr>
```

<center>Возможность выбора нескольких вариантов</center>

<input type="checkbox" name="var1" value="1" checked>Вариант1
(флаг включен по умолчанию)

<input type="checkbox" name="var2" value="2">Вариант2

<input type="checkbox" name="var3" value="3">Вариант3

<hr>

<center>Возможность выбора одного из нескольких вариантов</center>

<center>Выберите пол</center>

<input type="radio" name="sex" value="Мужской" checked>Мужской (по умолчанию)

<input type="radio" name="sex" value="Женский">Женский

<hr>

<center>Список без множественного выбора</center>

<select name="month" size=1>

<option value=1>Январь</option>

<option value=2>Февраль</option>

<option value=3>Март</option>

<option value=4>Апрель</option>

<option value=5>Май</option>

<option value=6>Июнь</option>

<option value=7>Июль</option>

<option value=8>Август</option>

<option value=9>Сентябрь</option>

<option value=10>Октябрь</option>

<option value=11>Ноябрь</option>

<option value=12>Декабрь</option>

</select>

<hr>

<center>Список с множественным выбором</center>

<select name="months[]" size=12 multiple>

<option value=01>Январь</option>

<option value=02>Февраль</option>

<option value=03>Март</option>

<option value=04>Апрель</option>

```

<option value=05>Май</option>
<option value=06>Июнь</option>
<option value=07>Июль</option>
<option value=08>Август</option>
<option value=09>Сентябрь</option>
<option value=10>Октябрь</option>
<option value=11>Ноябрь</option>
<option value=12>Декабрь</option>
</select>
<hr>

```

```

<center>Кнопки для выбора действий</center><br>
<input type="submit" name="go" value="Отправить данные">
<input type="reset" value="Очистить поля">
</form>
</body>
</html>

```

2. Создайте файл сценария **form.php**, который получает и отображает данные, переданные с помощью формы.

```

<?
/* Проверка нажатия кнопки */
if (!isset ($_POST["go"]))
{
echo "Не указаны параметры<br>";
exit(1);
}
else
{
/* Обработка введенных данных */
echo "<html><head><title>Обработка
данных</title></head><body>";
echo "<center><b>Текстовые поля</b></center><br>";
echo "Поле для ввода текста - ".$_POST["txt"]."<br>";
echo "Поле для ввода пароля - ".$_POST["pswd"]."<br>";
echo "Скрытое поле - ".$_POST["hid"]."<br>";
echo "Текстовая область -
<br><pre>".$_POST["t_area"]."</pre><br>";

```

```

        echo          "<center><b>Выбора
вариантов</b></center><br>";
        if    (isset($_POST["var1"]))    echo    "Вариант    1    -
".$_POST["var1"]."<br>";
        if    (isset($_POST["var2"]))    echo    "Вариант    2    -
".$_POST["var2"]."<br>";
        if    (isset($_POST["var3"]))    echo    "Вариант    3    -
".$_POST["var3"]."<br>";

```

```

        echo          "<center><b>Выбор    одного    из    нескольких
вариантов</b></center><br>";
        echo "Выбранный пол - ".$_POST["sex"]."<br>";

```

```

        echo          "<center><b>Список    без    множественного
выбора</b></center><br>";
        switch ($_POST["month"]) {
        case 1: echo "Месяц - Январь<br>";break;
        case 2: echo "Месяц - Февраль<br>";break;
        case 3: echo "Месяц - Март<br>";break;
        case 4: echo "Месяц - Апрель<br>";break;
        case 5: echo "Месяц - Май<br>";break;
        case 6: echo "Месяц - Июнь<br>";break;
        case 7: echo "Месяц - Июль<br>";break;
        case 8: echo "Месяц - Август<br>";break;
        case 9: echo "Месяц - Сентябрь<br>";break;
        case 10: echo "Месяц - Октябрь<br>";break;
        case 11: echo "Месяц - Ноябрь<br>";break;
        case 12: echo "Месяц - Декабрь<br>";break;
        }

```

```

        echo          "<center><b>Список    со    множественным
выбором</b></center><br>";
        foreach ($_POST["months"] as $key=>$value)
        echo "$key = $value<br>";
        }
        ?>

```

3. Создайте файл сценария **form1.php**, измените в файле **lab7.html** имя файла обработчика данных, и просмотрите данные, переданные с помощью формы, используя суперглобальные массивы **\$_POST**, **\$_REQUEST**, **\$_SERVER**.

```
<?
/* Проверка нажатия кнопки */
if (!isset ($_POST["go"]))
{
echo "Не указаны параметры<br>";
exit(1);
}
else
{
/* Просмотр переданных данных */
foreach ($_POST as $key=>$value)
{
if (gettype($value) == "array")
{
print "$key = <br>\n";
foreach ($value as $v ) {print "$v<br>";}
}
else {
print "$key = $value<br>\n";}
}
print "<hr>";

foreach ($_REQUEST as $key=>$value)
{
if (gettype($value) == "array")
{
print "$key = <br>\n";
foreach ($value as $v ) {print "$v<br>";}
}
else {
print "$key = $value<br>\n";}
}
print "<hr>";
```

```

foreach ($_SERVER as $key=>$value)
{
if (gettype($value) == "array")
{
print "$key = <br>\n";
foreach ($value as $v ) {print "$v<br>";}
}
else {
print "$key = $value<br>\n";}
}
print "<hr>";
}
?>

```

4. Создайте файл сценария **form2.php**, с помощью которого на экране браузера изображается небольшой прямоугольник — таблица, содержащая всего одну ячейку, и предлагается задать цвет для этого прямоугольника. После выбора нужного цвета и нажатия кнопки "закрасить", можно видеть, что прямоугольник тут же меняет свой цвет.

```

<?php
if ($_REQUEST['bg']==null) $bgc = "silver";
else $bgc = $_REQUEST['bg'];
echo "<p>Выберите цвет фона прямоугольника";
echo "<p><table border=1 width=100>";
echo "<tr> <td bgcolor=$bgc>&nbsp;&nbsp;&nbsp;<br>&nbsp;&nbsp;&nbsp;<br>&nbsp;&nbsp;&nbsp;</td></tr></table>";
echo "<form action='$_SERVER[\"PHP_SELF\"]' "
method='GET'>";
echo "<p><input type='radio' name='bg' value='red'>красный";
echo "<p><input type='radio' name='bg' value='green'>зеленый";
echo "<p><input type='radio' name='bg' value='blue'>синий";
echo "<p><input type='radio' name='bg' value='silver'>исходный";
echo "<p><input type='submit' value='закрасить'>";
echo "</form>";
?>

```

Индивидуальное задание:

Задание 1. Создайте файл **lab7-1.php** с HTML-формой, которая вызывает сама себя. В форме можно выбрать:

- а) горизонтальное (align, варианты значений: left, center, right);
 - б) вертикальное (valign, варианты значений: top, middle, bottom)
- расположение текста в таблице.

Для (а) необходимо использовать переключатели, для (б) - флажки. По умолчанию необходимо установить такие значения: для align - «left», для valign - «top».

Обработка формы выполняется при нажатии на кнопку «Выполнить». Над формой должна быть таблица, состоящая из одной ячейки шириной и высотой 100 пикселей. В ячейку таблицы следует поместить слово «Текст»

Текст

Выберите горизонтальное расположение:

- слева
- по центру
- справа

Выберите вертикальное расположение:

- сверху
- посередине
- внизу

Выполнить

Задание 2.

1. Создайте файл **lab7-2.htm** с HTML-формой анкеты, определяющей характер человека.

Пользователю предлагается в текстовом поле ввести свое имя, а затем ответить «да» или «нет» на следующие вопросы:

1. Считаете ли Вы, что у многих ваших знакомых хороший характер?
2. Раздражают ли Вас мелкие повседневные обязанности?
3. Верите ли Вы, что ваши друзья преданы Вам?
4. Неприятно ли Вам, когда незнакомый человек делает Вам замечание?
5. Способны ли Вы ударить собаку или кошку?
6. Часто ли Вы принимаете лекарства?
7. Часто ли Вы меняете магазин, в который ходите за продуктами?
8. Продолжаете ли Вы отстаивать свою точку зрения, поняв, что ошиблись?
9. Тяготят ли Вас общественные обязанности?
10. Способны ли Вы ждать более 5 минут, не проявляя беспокойства?
11. Часто ли Вам приходят в голову мысли о Вашей невезучести?
12. Сохранилась ли у Вас фигура по сравнению с прошлым?
13. Можете ли Вы с улыбкой воспринимать подтрунивание друзей?
14. Нравится ли Вам семейная жизнь?
15. Злопамятны ли Вы?
16. Находите ли Вы, что стоит погода, типичная для данного времени года?
17. Случается ли Вам с утра быть в плохом настроении?
18. Раздражает ли Вас современная живопись?
19. Надоедает ли Вам присутствие чужих детей в доме более одного часа?

Ответ на каждый из вопросов необходимо представить с помощью переключателя («да» или «нет»):

Анкета "Ваш характер"

Введите Ваше имя

Ответьте да или нет на следующие вопросы:

1. Считаете ли Вы, что у многих ваших знакомых хороший характер?

да нет

2. Раздражают ли Вас мелкие повседневные обязанности?

да нет

3. Верите ли Вы, что ваши друзья преданы Вам?

да нет

4. Неприятно ли Вам, когда незнакомый человек делает Вам замечание?

да нет

5. Способны ли Вы ударить собаку или кошку?

да нет

По умолчанию задайте везде значение «да».

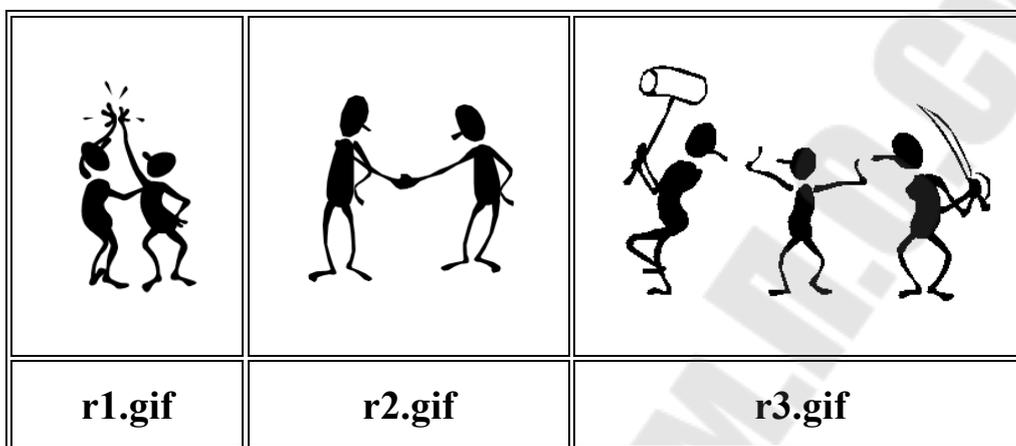
2. Напишите сценарий обработки анкеты (скрипт **lab7-2.php**), в котором:

Ставится 1 балл, если был ответ «да» в вопросах с номерами 3, 9, 10, 13, 14, 19 и «нет» в вопросах с номерами 1, 2, 4, 5, 6, 7, 8, 11, 12, 15, 16, 17, 18. Посчитайте сумму набранных баллов.

Если сумма оказалась более 15, то формируется фраза-результат: «У Вас покладистый характер»; если сумма в интервале от 8 до 15, то результат: «Вы не лишены недостатков, но с вами можно

ладить»; если сумма менее 8 баллов, то результат: «Вашим друзьям можно посочувствовать».

Отобразите на экране имя анкетированного, фразу-результат анкетирования и соответствующее этому результату изображение.



Контрольные вопросы

1. Как в PHP обрабатываются данные пользовательской формы?
2. Чем отличаются методы передачи данных GET и POST?
3. Какие суперглобальные массивы используются для хранения данных, переданных пользователем?
4. Какие данные сохраняются в массиве `$_SERVER`?
5. В чем отличие массивов `$_POST` и `$_REQUEST`?
6. Как отобразить значения элементов массива `$_POST`?
7. Как расположить PHP-программу, обрабатывающую данные, на той же странице, что и форму, передающую такие данные?
8. Для чего используется атрибут `action` в теге `form`?

ЛАБОРАТОРНАЯ РАБОТА № 8

Тема: «Переходы и передача данных между Web-страницами»

Цель работы: Изучить основные возможности PHP для осуществления переходов и передачи данных между Web-страницами.

Теоретическая часть: теоретические сведения приведены в электронном конспекте лекций.

Практическая часть:

Переходы между страницами в PHP:

Задание 1. Использование ссылок.

5. В корневом каталоге локального сайта создайте папку **Lab8**. В этой папке создайте два файла **page1.php** и **page2.php** следующего содержания:

```
<html>
<head><title>Страница 1</title></html>
<body bgcolor=blue>
<?php
echo "<a href='per2.php'><font color=white size=5>Переход на
другую страницу</a><br>";
?>
</body>
</html>
```

```
<html>
<head><title>Страница 2</title></html>
<body bgcolor=red>
<a href="per1.php"><font color=white size=5>Переход на первую
страницу</a>
</body>
</html>
```

6. Протестируйте переходы между страницами.

Задание 2. Применение форм.

Создайте файла формы **form.php** без каких-либо элементов для ввода, а только с кнопкой **submit**, щелчок на которой приводит к переходу на страницу, **URL**-адрес которой указан в атрибуте **action** тега **<form>**:

```
<html>
<head><title>Переход на другую страницу</title></html>
<body>
<form action="page1.php" method="POST">
<input type="submit" name="go" value="Переход на другую
страницу">
</body>
</html>
```

Задание 3. Использование встроенной PHP-функции **header()**

Создайте файл формы **form2.php** для ввода данных о пользователе (имя, пароль и т.д.). Данные передаются на сервер, где обрабатываются сценарием **redirect.php**, который перенаправляет пользователя в зависимости от введенных данных на страницы **per1.php** и **per2.php**.

form2.php

```
<html>
<head><title>Страница 1</title></html>
<body bgcolor=blue>
<?
echo "Привет - ".$_REQUEST['login'].!"<br>";
?>
<a href=per2.php?name=Guest><font color=white size=5>Вы
можете перейти на другую страницу с именем Guest</a><br>
</body>
</html>
```

redirect.php

```
<?php
if      (strtolower($_REQUEST["login"])=="admin"      &&
strtolower($_REQUEST["pswd"])=="12345")
{header("Location:per1.php");}
else {header("Location:per2.php");}
?>
```

per1.php

```
<html>
<head><title>Страница 1</title></html>
<body bgcolor=blue>
<?
echo "Привет - ".$_REQUEST['login']."!<br>";
?>
<a href=per2.php?name=Guest><font color=white size=5>Вы
можете перейти на другую страницу с именем Guest</a><br>
</body>
</html>
```

per2.php

```
<html>
<head><title>Страница 2</title></html>
<body bgcolor=red text=white size=6>
<?php
echo "Привет - ".$_GET["name"]."<br>";
echo "<a href='per1.php'><font color=white size=5>Переход на
другую страницу</a>";
?>
</body>
</html>
```

Функцию **header()** нельзя использовать после текста, HTML-кода и операторов **echo** или **print**.

Передача данных между страницами

Задание 4. Проанализируйте содержимое файлов **per1.php** и **per2.php** предыдущего задания. Обратите внимание, что информация, которую можно добавить к URL-адресу, состоит из элементов вида *имя_переменной=значение*. Если таких элементов несколько, то между ними ставится знак **&** в качестве разделителя. Вся последовательность элементов дописывается к URL-адресу через знак вопроса (?).

Задание 5. Применение **cookie**.

1. Создайте файл **cookie.php**, следующего содержания:

```
<?php
setcookie("message", "Привет");
?>
<HTML>
<HEAD>
<TITLE>
Установка cookie
</TITLE>
</HEAD>
<BODY>
<CENTER>
<h1>Установка cookie</h1>
<p><b>Cookie</b> установлен! Для просмотра следует перейти
по
ссылке <a href='phpgetcookie.php'>phpgetcookie.php</a>
</CENTER>
</BODY>
</HTML>
```

2. Для чтения значения **cookie** создайте файл **phpgetcookie.php**, следующего содержания.

```
<HTML>
<HEAD>
<TITLE>
```

Чтение значения cookie

```
</TITLE>
</HEAD>
<BODY>
<CENTER>
<h1>Чтение значения cookie</h1>
Cookie содержит значение:
<?php
if (isset ($_COOKIE ['message']))
{
echo 'Cookie содержит значение: '.$_COOKIE ['message'];
}
else
{
echo 'Cookie не установлен';
}
?>
</CENTER>
<BODY>
</HTML>
```

3. Используя **cookie**, создайте простой счетчик посещений страницы сайта

```
<?
if (isset($_COOKIE))
{$counter=$_COOKIE["counter"];
$counter++;
setcookie("counter",$counter);
echo("Вы посетили эту страницу $counter раз");
}
else
{echo "Cookie не установлены";
setcookie("counter",0);
}
?>
```

Обновите страницу несколько раз и убедитесь в работоспособности счетчика.

Задание 6. Применение сеансов PHP

1. Создайте PHP-скрипты **sessions.php** и **phpsession.php**, демонстрирующие сохранение и чтение данных в сеансе. Элементу массива `$_SESSION['temperature']` присваивается значение температуры. Формируемая страница содержит ссылку для перехода на следующую страницу, в которой данные будут прочитаны.

sessions.php

```
<?php
session_start();
?>
<HTML>
<HEAD>
<TITLE>
Сохранение данных сеанса
</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>
Сохранение данных сеанса
</H1>
<?php
$_SESSION['temperature']=36.6;
echo "Температура: ".$_SESSION['temperature'];
?>
<BR>
Значение температуры сохранится и после перехода по
<a href="phpsession.php">ссылке</a>.
<BR>
Для удаления значения температуры перейдите по
<a href="phpsession2.php">ссылке</a>.
</CENTER>
<BODY>
</HTML>
```

phpsession.php

```

<?php
session_start();
?>
<HTML>
<HEAD>
<TITLE>
Чтение данных сеанса
</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>
Чтение данных сеанса
</H1>
<?php
if (isset ($_SESSION ["temperature"]))
{
echo          "Значение          температуры          составляет
".$_SESSION['temperature'];
}
?>
</CENTER>
<BODY>
</HTML>

```

2. Для удаления данных сеанса создайте PHP-скрипт **phpsession.php**:

```

<?php
session_start ();
unset ($_SESSION ["temperature"]);
?>
<HTML>
<HEAD>
<TITLE>
Удаление данных сеанса
</TITLE>
</HEAD>

```

```
<BODY>
<CENTER>
<H1>
Удаление данных сеанса
</H1>
Проверьте удаление данных. Перейдите по
<a href="phpsession.php">ссылке</a>.
</CENTER>
<BODY>
</HTML>
```

Убедитесь в работоспособности скрипта.

Контрольные вопросы

1. Как в PHP-скрипте осуществить переход на другую страницу?
2. Как осуществить переход на другую страницу, используя пользовательскую форму?
3. Для чего используется PHP-функции **header()**?
4. Какова особенность использования функции **header()**?
5. Как можно передать данные между страницами?
6. Как передаются данные методом **GET**?
7. Для чего используется механизм **cookie**?
8. Как установить значение **cookie**?
9. Как удалить значение **cookie**?
10. Для чего используется механизм сеансов?
11. Как происходит чтение данных сеанса?
12. Как удалить данные сеанса?

ЛАБОРАТОРНАЯ РАБОТА № 9

Тема: «Работа с СУБД»

Цель работы: получить навыки работы с СУБД MySQL.

Теоретическая часть:

Запуск MySQL

Управление сервером обычно осуществляется из командной строки. Запуск сервера MySQL осуществляется вместе с запуском Web-сервера.

После запуска можно попытаться войти в сервер. В случае, если предполагается управление сервером через консоль, то необходимо использовать команду **mysql**. Изначально существует единственный пользователь, которому предоставляется право входа - **root**, который не имеет пароля. Первое, что необходимо сделать войти под именем **root** и зарегистрировать нового пользователя и установить для него пароль. Команда **mysql** может использовать следующие опции:

Таблица 1 - Опции команды MySQL

-?, --help	Справка
-h, --hostname=[hostname]	Имя сервера mysql.
-u, --user=[user]	Имя пользователя для доступа к mysql.
-p, --password=[password]	Пароль пользователя для доступа к mysql.
-P, --port=[port]	Порт для соединения с сервером.
-V, --version	Информация о версии

Для запуска команд используется режим командной строки. В файловом менеджере лучше всего перейти в каталог **\usr\local\mysql-5.1\bin** и запустить команду **cmd.exe**, после этого необходимо ввести команду:

mysql -u root

Рисунок 1 - Успешный запуск консоли

Если вы это получили, значит вы успешно вошли в консоль `mysql`, которая используется для администрирования сервера.

Просмотр списка баз данных (БД), доступных на сервере, осуществляется командой **SHOW DATABASES**. Ввод любой команды заканчивается `;` или `\g`.

Для выбора соответствующей БД используется команда **USE Name**, где **Name** – имя БД.

БАЗА ДАННЫХ MYSQL И ТАБЛИЦЫ ПРИВИЛЕГИЙ

БД `mysql` содержит 5 таблиц, которые используются для предоставления доступа к базам данных и таблицам в них пользователям. Для просмотра таблиц используется команда **SHOW TABLES**.

Кратко рассмотрим функции каждой из таблиц:

Таблица User

Определяет, разрешено ли пользователю, пытающемуся подключиться к серверу делать это, содержит имя пользователя, пароль, а также набор привилегий. Если ввести команду **SHOW COLUMNS FROM USER;** то получим следующее:

Таблица 2- Структура таблицы **User**

Field	Type	Null	Key	Default	Extra
Host	char(60)		PRI		
User	char(16)		PRI		
Password	char(41)				
Select_priv	enum('N','Y')			N	

Field	Type	Null	Key	Default	Extra
Insert_priv	enum('N','Y')			N	
Update_priv	enum('N','Y')			N	
Delete_priv	enum('N','Y')			N	
Create_priv	enum('N','Y')			N	
Drop_priv	enum('N','Y')			N	
Reload_priv	enum('N','Y')			N	
Shutdown_priv	enum('N','Y')			N	
Process_priv	enum('N','Y')			N	
File_priv	enum('N','Y')			N	
Grant_priv[1]	enum('N','Y')			N	
References_priv	enum('N','Y')			N	
Index_priv	enum('N','Y')			N	
Alter_priv	enum('N','Y')			N	
Show_db_priv	enum('N','Y')			N	
Super_priv	enum('N','Y')			N	
Create_tmp_table_priv	enum('N','Y')			N	
Lock_tables_priv	enum('N','Y')			N	
Execute_priv	enum('N','Y')			N	
Repl_slave_priv	enum('N','Y')			N	
Repl_client_priv	enum('N','Y')			N	
Create_view_priv	enum('N','Y')			N	
Show_view_priv	enum('N','Y')			N	
Create_routine_priv	enum('N','Y')			N	
Alter_routine_priv	enum('N','Y')			N	
Create_user_priv	enum('N','Y')			N	
Event_priv	enum('N','Y')			N	
Trigger_priv	enum('N','Y')			N	
ssl_type	enum('','ANY','X509','SPECIFIED')				
ssl_cipher	blob			NULL	
x509_issuer	blob			NULL	

Field	Type	Null	Key	Default	Extra
x509_subject	blob				
max_questions	int(11) unsigned				
max_updates	int(11) unsigned				
max_connections	int(11) unsigned				
max_user_connections	int(11) unsigned				

Изначально эта таблица содержит пользователя root без пароля. По умолчанию root может входить с любого хоста, имеет все привилегии и доступ ко всем базам данных.

```

C:\WINDOWS\system32\cmd.exe - mysql -u root
mysql> select host,user,password from user;
+-----+-----+-----+
| host      | user  | password |
+-----+-----+-----+
| localhost | root  |          |
| 127.0.0.1 | root  |          |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>

```

Для добавления нового пользователя **your_name**, можно выполнить следующие операторы языка SQL (**Insert**):

**Insert into user (host, user, password)
values ('localhost', 'username1', '12345');**

```

C:\WINDOWS\system32\cmd.exe - mysql -u root
mysql> insert into user (host,user,password) values ('localhost','username1','12345');
Query OK, 1 row affected, 3 warnings (0.00 sec)

mysql> select host,user,password from user;
+-----+-----+-----+
| host      | user      | password |
+-----+-----+-----+
| localhost | root      |          |
| 127.0.0.1 | root      |          |
| localhost | username  | *BCC5BA3BA0E22487B12142A8B62183201C8B3816 |
| localhost | username1 | 12345    |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>

```

После добавления пользователя в БД дополнительно необходимо назначить права пользователю, после чего можно начинать настраивать таблицы привилегий, вводить новых пользователей, создавать базы данных и таблицы, то есть делать все

то, что называется администрированием. Назначение привилегий на практике обычно осуществляется в таблице **DB**, чтобы привилегии пользователя не распространялись на все базы данных.

Таблица Db

Определяет к каким базам данных каким пользователям и с каких хостов разрешен доступ. В этой таблице можно предоставлять каждому пользователю доступ к базам данных и назначать привилегии.

Таблица Host

Таблица **HOST** используется для расширения диапазона доступа в таблице **DB**. К примеру, если доступ к какой-либо базе данных должен быть предоставлен более чем одному хосту, тогда следует оставить пустой колонку **host** в таблице **db**, и внести в таблицу **host** необходимые имена хостов.

Привилегии, предоставляемые MySQL

Таблица 2 - Привилегии пользователя

Привилегия	Колонка	Где используется
select	Select_priv	таблицы
insert	Insert_priv	таблицы
Update	Update_priv	таблицы
delete	Delete_priv	таблицы
index	Index_priv	таблицы
alter	Alter_priv	таблицы
create	Create_priv	БД, таблицы, индексы
drop	Drop_priv	БД или таблицы
grant	Grant_priv	БД или таблицы
References	References_priv	БД или таблицы
reload	Reload_priv	администрирование сервера
Shutdown	Shutdown_priv	администрирование сервера
Process	Process_priv	администрирование сервера
file	File_priv	доступ к файлам на сервере

Более подробно привилегии будут рассмотрены в отдельном курсе MySQL.

Практическая часть:

Создание базы данных в MySQL

Задание. Необходимо создать базу данных «Оборудование» на основе следующих исходных данных:

- фамилия ответственного за оборудование;
- должность;
- цех;
- оборудование;
- количество единиц оборудования;
- цена за единицу оборудования.

Порядок выполнения

1. Разобьем таблицу на две: Таблицы будут связаны по полю «Фамилия» отношением один (таблица 1) ко многим (таблица 2)

Таблица 1 будет иметь поля :

Фамилия - ключевое поле
Должность
Цех

Таблица 2 будет иметь поля:

Фамилия
Оборудование
Количество
Цена

Последовательность действий в MySQL:

1. Создадим новую базу данных с именем **Oborud**
mysql> CREATE DATABASE OBORUD;
2. Просмотрим базы данных, и выберем ту, которую мы будем использовать
mysql> SHOW DATABASES;

3. Сделаем созданную базу текущей.
mysql> USE OBORUD;

```
C:\WINDOWS\system32\cmd.exe - mysql -u root
mysql> create database oborud;
Query OK, 1 row affected (0.08 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| oborud |
| phpmyadmin |
+-----+
4 rows in set (0.02 sec)

mysql> use oborud;
Database changed
mysql>
```

4. Создадим таблицу «Otvetstv»:
mysql> CREATE TABLE OTVETSTV> (fam char(25),dolgn char
(15), unit char(15));
5. Просмотрим структуру сформированной таблицы:
mysql> DESCRIBE OTVETSTV;

```
C:\WINDOWS\system32\cmd.exe - mysql -u username1 -p mysql
mysql> describe otvetstv;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| fam   | char(25) | NO   | PRI |          |       |
| dolgn | char(15) | YES  |     | NULL    |       |
| unit  | char(15) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.14 sec)

mysql> _
```

6. Аналогично создадим и выведем структуру для таблицы «OBORUD»:

```
C:\WINDOWS\system32\cmd.exe - mysql -u username1 -p mysql
mysql> create table oborud
-> (fam char(25),oborud char(15),unit char(10),cena int(8));
Query OK, 0 rows affected (0.39 sec)

mysql> describe oborud;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| fam   | char(25) | YES  |     | NULL    |       |
| oborud | char(15) | YES  |     | NULL    |       |
| unit  | char(10) | YES  |     | NULL    |       |
| cena  | int(8)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.17 sec)

mysql>
```

7. Посмотрим список созданных таблиц
mysql> SHOW TABLES;

```
C:\WINDOWS\system32\cmd.exe - mysql -u username1 -p mysql
mysql> show tables
-> ;
+-----+
| Tables_in_oborud |
+-----+
| oborud            |
| otvetstv          |
+-----+
2 rows in set (0.06 sec)
mysql> _
```

8. Осуществим ввод данных.

```
C:\WINDOWS\system32\cmd.exe - mysql -u username1 -p mysql
mysql> insert into otvetstv
-> value ('Петров','слесарь','инструментальный');
Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> insert into otvetstv
-> value ('Сидоров','монтер','сборочный');
Query OK, 1 row affected (0.03 sec)

mysql> select * from otvetstv;
+-----+-----+-----+
| fam   | dolgn | unit   |
+-----+-----+-----+
| Иванов | мастер | инструментальный |
| Петров | слесарь | инструментальный |
| Сидоров | монтер | сборочный |
+-----+-----+-----+
3 rows in set (0.00 sec)
mysql>
```

```
C:\WINDOWS\system32\cmd.exe - mysql -u username1 -p mysql
mysql> insert into oborud
-> value ('Сидоров','газовый ключ',9,10);
Query OK, 1 row affected (0.00 sec)

mysql> select * from oborud;
+-----+-----+-----+-----+
| fam   | oborud | unit | cena |
+-----+-----+-----+-----+
| Иванов | станок1 | 7    | 1000 |
| Иванов | станок2 | 4    | 500  |
| Иванов | станок3 | 8    | 700  |
| Петров | паяльник | 7    | 20   |
| Петров | дрель   | 8    | 160  |
| Сидоров | дрель   | 6    | 160  |
| Сидоров | газовый ключ | 9    | 10   |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)
mysql>
```

Выполнение запросов

9. Создадим запрос, позволяющий отобразить все оборудование числящееся за конкретным лицом.

```
C:\WINDOWS\system32\cmd.exe - mysql -u username1 -p mysql
mysql> select * from oborud
-> where fam='Иванов';
+-----+-----+-----+-----+
| fam      | oborud  | unit  | cena  |
+-----+-----+-----+-----+
| Иванов  | станок1 | 7     | 1000  |
| Иванов  | станок2 | 4     | 500   |
| Иванов  | станок3 | 8     | 700   |
+-----+-----+-----+-----+
3 rows in set (0.20 sec)
mysql>
```

10. Создадим запрос, который подсчитывает общую стоимость по каждому виду оборудования, записи сортируются по алфавиту.

```
C:\WINDOWS\system32\cmd.exe - mysql -u username1 -p mysql
mysql> select fam,oborud,unit*cena from oborud;
+-----+-----+-----+
| fam      | oborud  | unit*cena |
+-----+-----+-----+
| Иванов  | станок1 | 7000      |
| Иванов  | станок2 | 2000      |
| Иванов  | станок3 | 5600      |
| Петров  | паяльник | 140       |
| Петров  | дрель   | 1280      |
| Сидоров | дрель   | 960       |
| Сидоров | газовый ключ | 90       |
+-----+-----+-----+
7 rows in set (0.01 sec)
mysql>
```

11. Создадим запрос, который подсчитывает общую стоимость оборудования по каждому материально-ответственному.

```
C:\WINDOWS\system32\cmd.exe - mysql -u username1 -p mysql
mysql> select fam,sum(unit*cena) from oborud
-> group by fam;
+-----+-----+
| fam      | sum(unit*cena) |
+-----+-----+
| Петров  | 1420           |
| Иванов  | 14600          |
| Сидоров | 1050           |
+-----+-----+
3 rows in set (0.01 sec)
```

Контрольные вопросы

1. Как войти в сервер MySQL?
2. Как создать нового пользователя на сервере?
3. Как просмотреть список баз данных, доступных на сервере?
4. Как создать новую базу данных?
5. Как просмотреть структуру базы данных?
6. Как просмотреть структуру таблицы?
7. Как просмотреть содержимое таблицы?

8. Как внести данные в таблицу базы данных?
9. Как просмотреть данные, удовлетворяющие определенным критериям?
10. Как выполнить групповые операции?

ЛАБОРАТОРНАЯ РАБОТА № 10

Тема: «Взаимодействие PHP и MySQL»

Цель работы: получить навыки работы по созданию, обработке и управлению базами данных на стороне сервера.

Теоретическая часть: теоретические сведения приведены в электронном конспекте лекций.

Практическая часть:

Задание 1. Создание базы данных

Создание базы данных рассмотрим на примере создание базы данных «Оборудование», структура которой описана в лабораторной работе № 9.

1. Создайте PHP-скрипт следующего содержания:

```
<HTML>
<HEAD>
<TITLE>
Создание базы данных
</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>Создание базы данных</H1>
<?php

//Подключение к MySQL
$connection = mysql_connect ("localhost", "root")
or die ("Ошибка соединения с сервером");

//Создание базы данных Оборудование
$query = "CREATE DATABASE IF NOT EXISTS OBORUD1";
$result = mysql_query ($query)
```

```

or die("Ошибка при выполнении запроса: ".mysql_error);

//Соединение с созданной базой данных
$db = mysql_select_db ("OBORUD1", $connection)
or die ("Ошибка при выборе базы данных");
echo "База данных 'OBORUD1' успешно создана<p>";

//Создание таблиц
$query = "CREATE TABLE OTVETSTV(fam char(25),dolgn
char(15),podrazd char(25))";
$result = mysql_query ($query)
or die ("Ошибка при выполнении запроса:".mysql_error);
echo "Таблица 'OTVETSTV' успешно создана<p>";

$query = "CREATE TABLE OBORUD(fam_otv char(25),oborud
char(25),kol int,cena int)";
$result = mysql_query ($query)
or die ("Ошибка при выполнении запроса:".mysql_error);
echo "Таблица 'OBORUD' успешно создана<p>";

//Закрытие соединения
mysql_close ($connection);
?>
</CENTER>
</BODY>
</HTML>

```

2. Сохраним файл под названием BD1.php
3. Для просмотра существующих баз данных, а также структуры базы данных «OBORUD1», создайте файл BD1.php:

```

<HTML>
<HEAD>
<TITLE>
Отображение структуры баз данных и таблиц
</TITLE>
</HEAD>
<BODY>

```

```

<CENTER>
<H1>Отображение баз данных</H1>
<p>
<?php

//Подключение к MySQL
$connection = mysql_connect ("localhost", "root")
or die ("Ошибка соединения с сервером");

//Просмотр существующих баз данных
$r=mysql_list_dbs($connection);
echo "<h3>Существующие базы данных</h3>";
while($row=mysql_fetch_row($r)) {
echo "<h4>$row[0]<br>";
}

//Просмотр таблиц в базе данных OBORUD1
$r=mysql_query("SHOW TABLES IN OBORUD1");
echo "<h3>Таблицы в базе данных OBORUD1</h3>";
while($row=mysql_fetch_row($r)) {
echo "<h4>$row[0]<br>";
}

//Просмотр полей в таблице OTVETSTV
echo "<h3>Названия полей в таблице OTVETSTV</h3>";
$fields=mysql_list_fields("OBORUD1", "OTVETSTV",
$connection);
$numcols=mysql_num_fields($fields); // число полей в таблице
// Далее распечатаем имена всех полей таблицы
for($i=0;$i<$numcols;$i++) {
    echo mysql_field_name($fields,$i)."<BR>";
}

//Просмотр структура таблицы OBORUD
$db = mysql_select_db ("OBORUD1", $connection);
$r=mysql_query("DESCRIBE OBORUD");
echo "<h3>Структура таблицы OBORUD</h3>";
while($row=mysql_fetch_row($r)) {
foreach ($row as $b) {

```

```

echo $b." ";
}
echo "<br>";
}
//Закрытие соединения
mysql_close ($connection);
?>
</CENTER>
</BODY>
</HTML>

```

Задание 2. Добавление данных в существующую базу данных

1. Заполните данными таблицы «OBORUD», «OTVETSTV». Для этого создайте PHP-скрипт BD3.php следующего содержания:

```

<HTML>
<HEAD>
<TITLE>
Добавление данных в базу
</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>Добавление данных в базу</H1>
<p>
<?php

//Подключение к MySQL
$connection = mysql_connect ("localhost", "root")
or die ("Ошибка соединения с сервером");

//Выбор базы OBORUD1
$db = mysql_select_db ("OBORUD1", $connection)
or die ("Ошибка при выборе базы данных");

//Заполнение таблицы OTVETSTV
$query = "INSERT INTO OTVETSTV(fam,dolgn,podrazd)
VALUES ('Иванов','мастер','инструментальный) ";

```

```

$result = mysql_query ($query)
or die ("Ошибка при выполнении запроса: ".mysql_error ());
$query = "INSERT INTO OTVETSTV(fam,dolgn,podrazd )
VALUES ('Петров','слесарь','инструментальный') ";
$result = mysql_query ($query)
or die ("Ошибка при выполнении запроса: ".mysql_error ());
$query = "INSERT INTO OTVETSTV(fam,dolgn,podrazd )
VALUES ('Сидоров','монтер','сборочный') ";
$result = mysql_query ($query)
or die ("Ошибка при выполнении запроса: ".mysql_error ());

//Заполнение таблицы OBORUD
$query = "INSERT INTO OBORUD(fam_otv,oborud,kol,cena)
VALUES ('Иванов','станок1',7,1000) ";
$result = mysql_query ($query)
or die ("Ошибка при выполнении запроса: ".mysql_error ());
$query = "INSERT INTO OBORUD(fam_otv,oborud,kol,cena)
VALUES ('Иванов','станок2',4,500) ";
$result = mysql_query ($query)
or die ("Ошибка при выполнении запроса: ".mysql_error ());
$query = "INSERT INTO OBORUD(fam_otv,oborud,kol,cena)
VALUES ('Иванов','станок3',8,700) ";
$result = mysql_query ($query)
or die ("Ошибка при выполнении запроса: ".mysql_error ());
$query = "INSERT INTO OBORUD(fam_otv,oborud,kol,cena)
VALUES ('Петров','паяльник',7,20) ";
$result = mysql_query ($query)
or die ("Ошибка при выполнении запроса: ".mysql_error ());
$query = "INSERT INTO OBORUD(fam_otv,oborud,kol,cena)
VALUES ('Петров','дрель',8,160) ";
$result = mysql_query ($query)
or die ("Ошибка при выполнении запроса: ".mysql_error ());
$query = "INSERT INTO OBORUD(fam_otv,oborud,kol,cena)
VALUES ('Сидоров','дрель',6,160) ";
$result = mysql_query ($query)
or die ("Ошибка при выполнении запроса: ".mysql_error ());
$query = "INSERT INTO OBORUD(fam_otv,oborud,kol,cena)
VALUES ('Сидоров','гаечный ключ',6,160) ";
$result = mysql_query ($query)

```

```

or die ("Ошибка при выполнении запроса: ".mysql_error ());

echo "<H3>Данные успешно добавлены</h3>";

//Закрытие соединения
mysql_close ($connection);
?>
</CENTER>
</BODY>
</HTML>

```

Задание 3. Просмотр содержимого базы данных

Для просмотра содержимого таблиц базы данных создайте следующий сценарий и сохраните под именем BD4.php:

```

<HTML>
<HEAD>
<TITLE>
Просмотр содержимого базы данных
</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>Содержимое базы данных OBORUD1</H1>
<p>
<?php

//Подключение к MySQL
$connection = mysql_connect ("localhost", "root")
or die ("Ошибка соединения с сервером");

//Выбор базы OBORUD1
$bd="OBORUD1";
$db = mysql_select_db ($bd, $connection)
or die ("Ошибка при выборе базы данных");

//Просмотр таблиц в базе данных OBORUD1
$r=mysql_query("SHOW TABLES IN OBORUD1");

```

```

while($row=mysql_fetch_row($r)) {
echo "<h3>Таблица ".$row[0]."</h3><br>";
$fields=mysql_list_fields($bd,$row[0],$connection);
$col=mysql_num_fields($fields);
$query = "SELECT * FROM ".$row[0];
$result = mysql_query ($query)
or die ("Ошибка при выполнении запроса: ".mysql_error ());
echo "<table border=1>";
for($i=0;$i<$col;$i++) {
echo "<th><b>".mysql_field_name($fields,$i)."</b></th>";
}
while ($row1 = mysql_fetch_row ($result)) {
echo "<TR>";
for ($i=0;$i<$col;$i++) {
echo "<TD>".$row1 [$i]."</TD>";
}
echo "</TR>";
}
echo "</table>";
}

//Закрытие соединения
mysql_close ($connection);
?>
</CENTER>
</BODY>
</HTML>

```

Индивидуальное задание

Необходимо написать PHP-скрипт для вывода данных из таблиц, на основании запросов, рассмотренных в лабораторной работе № 9.

Контрольные вопросы

1. Как в PHP устанавливается соединение с сервером баз данных?
2. Как происходит выполнение запросов?

3. Как создать новую базу данных?
4. Как выбрать необходимую базу данных?
5. Как закрыть соединение с сервером?
6. Для чего используется функция `mysql_list_dbs`?
7. Для чего используется функция `mysql_fetch_row`?
8. Для чего используется функция `mysql_list_fields`?
9. Для чего используется функция `mysql_num_fields`?
10. Какие аргументы использует функция `mysql_field_name`?

ЛАБОРАТОРНАЯ РАБОТА № 11

Тема: «Работа с файлами»

Цель работы: научиться работать с файлами на сервере через функции PHP.

Теоретическая часть: теоретические сведения приведены в электронном конспекте лекций.

Практическая часть:

1. В корневом каталоге локального сайта создайте папку **Lab11**.
2. Создайте PHP-скрипт **Lab11_1.php**, осуществляющих запись в файл **text.txt**

```
<HTML>
<HEAD>
<TITLE>
Запись в файл при помощи fwrite
</TITLE>
</HEAD>
<BODY>
<CENTER><H1>Запись в файл при помощи fwrite</H1>
<?php
$handle = fopen ("text.txt","wb");
$text = "Просто\nстрока\nтекста.\n";
if (fwrite ($handle, $text) == FALSE) {
echo "Ошибка записи в файл 'text.txt'.";
}
else
{
echo "Файл 'text.txt' успешно создан.";
}
fclose ($handle);
?>
</CENTER>
</BODY>
```

```
</HTML>
```

Обратите внимание, после завершения работы с файлом его необходимо закрыть с помощью функции **fclose**.

Содержимое файла можно просмотреть в папке откуда запускался сценарий.

3. Создайте скрипт **Lab11_2.php**, демонстрирующий добавление данных в файл `text.txt`:

```
<HTML>
<HEAD>
<TITLE>
Добавление к файлу при помощи fwrite
</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>Добавление к файлу при помощи fwrite</H1>
<?php
$handle = fopen ("text.txt", "ab");
$text = "Это\ndобавленный\ntекст.\n";
if (fwrite ($handle, $text) == FALSE)
{
echo "Ошибка записи в файл text.txt.";
}
else
{
echo "Текст добавлен к файлу text.txt.";
}
fclose ($handle);
?>
</CENTER>
</BODY>
</HTML>
```

4. Создайте скрипт **Lab11_3.php**, осуществляющий запись в файл `text1.txt` с использованием функции **file_put_contents**:

```
<HTML>
<HEAD>
<TITLE>
Создание файла при помощи file_put_contents
</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>Создание файла при помощи file_put_contents</H1>
<?php
$text = "Просто\nнемного\ntекста.\n";
if (file_put_contents ("text1.txt", $text) == FALSE)
{
echo "Ошибка записи в файл text1.txt.";
}
else
{
echo "Текст записан в файл text1.txt.";
}
?>
</CENTER>
</BODY>
</HTML>
```

5. Создайте PHP-скрипт **Lab11_4.php** для считывания строк из созданного файла text1.txt:

```
<HTML>
<HEAD>
<TITLE>
Чтение файла
</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1> Чтение файла</H1>
<?php
$handle = fopen ("text.txt", "r");
while (!feof ($handle))
```

```

{
$text = fgets ($handle);
echo $text, "<BR>";
}
fclose ($handle);
?>
</CENTER>
</BODY>
</HTML>

```

Обратите внимание. После открытия файла он считывается построчно при помощи функции **fgets**. Для организации цикла используется оператор **while** и функция **feof**, которая возвращает значение TRUE при достижении конца файла. В теле цикла производится чтение строки текста при помощи **fgets** и ее отображение.

6. Создайте скрипт **Lab11_5.php**, демонстрирующий использование посимвольного чтения и отображения содержимого файла `text1.txt`. Следует обратить внимание, что символ конца строки заменяется на тег `
`.

```

<HTML>
<HEAD>
<TITLE>
Посимвольное чтение файла
</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>Посимвольное чтение файла</H1>
<?php
$handle = fopen ("text1.txt", "rb");
while ($char = fgetc ($handle))
{
if ($char == "\n")
{
$char = "<BR>";
}
}

```

```
echo $char;
}
fclose ($handle);
?>
</CENTER>
</BODY>
</HTML>
```

7. Создайте скрипт **Lab11_6.php**, демонстрирующий возможности считывания файла целиком.

```
<HTML>
<HEAD>
<TITLE>
Чтение файла при помощи fread
</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>Чтение файла при помощи fread</H1>
<?php
$handle = ("text.txt", "rb");
$text = fread ($handle, filesize ("text.txt"));
$br_text = str_replace ("\n", "<BR>", $text);
echo $br_text;
fclose ($handle);
?>
</CENTER>
</BODY>
</HTML>
```

8. Создайте скрипт **Lab11_7.php**, демонстрирующий использование функции **file_get_contents**:

```
<HTML>
<HEAD>
<TITLE>
Чтение содержимого файла при помощи file_get_contents
</TITLE>
```

```

</HEAD>
<BODY>
<CENTER>
<H1>Чтение содержимого файла при помощи
file_get_contents</H1>
<?php
$text = file_get_contents ("text.txt");
$br_text = str_replace ("\n", "<BR>", $text);
echo $br_text;
?>
</CENTER>
</BODY>
</HTML>

```

9. Создайте скрипт **Lab11_8.php**, демонстрирующий использование функций для работы с каталогами:

```

<HTML>
<HEAD>
<TITLE>
Просмотр содержимого каталога
</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>Просмотр содержимого каталога</H1>
<?php
$dir=opendir("d:/users/");
chdir("d:/users");
echo "<table width=50% border=1>";
while ($r=readdir($dir)) {
echo "<tr><td>";
if (is_dir($r)) echo "$r</td><td>Каталог</td>";
if (is_file($r)) echo "$r</td><td>.filesize($r).</td>";
echo "</tr>";
}
echo "</table>";
closedir($dir);
?>

```

```
</CENTER>
</BODY>
</HTML>
```

10. Создайте файл формы для отправки файла **Lab11_9.html**:

```
<HTML>
<HEAD>
<TITLE> Форма для загрузки файла на сервер
</TITLE>
</HEAD>
<BODY>
    <form enctype="multipart/form-data" action="obr_files.php"
method="post">
    Загрузить файл: <input type="file" name="myfile" /><br>
    <input type="submit" value="Отправить файл" />
    </form>
</BODY>
</HTML>
```

11. Создайте сценарий **obr_files.php**, который сохраняет выбранный пользователем файл на сервере.

```
<?
$uploaddir = 'd:/users/temp/';
// будем сохранять загружаемые
// файлы в эту директорию
$destination = $uploaddir.$_FILES['myfile']['name'];
// имя файла оставим неизменным

if
(move_uploaded_file($_FILES['myfile']['tmp_name'],$destination))
{
/* перемещаем файл из временной папки
в выбранную директорию для хранения */

print "Файл успешно загружен <br>";
}
```

```
else {echo "Произошла ошибка при загрузке файла. Некоторая  
отладочная информация:<br>";  
print_r($_FILES);  
}
```

?>

Контрольные вопросы

1. На какие группы можно функции PHP для работы с файлами?
2. Какая функция в PHP открывает файл для чтения и записи?
3. Какие модификаторы доступа к файлу вы знаете?
4. Какие функции PHP используются для записи данных в файл?
5. Как обозначается конец строки в разных операционных системах?
6. Какая системная константа PHP хранит представление конца строки для текущей ОС?
7. Почему после завершения работы с файлом его необходимо закрыть с помощью функции **fclose**?
8. Какие функции PHP предназначены для работы с каталогами?
9. Как осуществляется загрузка файлов на сервер?
10. Какие параметры, связанные с загрузкой файлов на сервер, используются в конфигурационном файле **php.ini**?
11. Какая информация сохраняется в суперглобальном массиве **\$_FILES**?

Самовендюк Николай Владимирович

**СЕРВЕРНЫЕ ТЕХНОЛОГИИ
РАЗРАБОТКИ WEB-САЙТОВ**

**Практикум
по одноименной дисциплине
для слушателей специальности 1-40 01 74
«Web-дизайн и компьютерная графика»
заочной формы обучения**

Подписано к размещению в электронную библиотеку
ГГТУ им. П. О. Сухого в качестве электронного
учебно-методического документа 09.12.14.

Пер. № 144Е.
<http://www.gstu.by>