

## Тема работы: ИЗУЧЕНИЕ ЯЗЫКА PHP

**Цель работы:** Изучить основы языка серверного скриптования PHP.

### Теоретическая часть:

**PHP** (официальное название "PHP: Hypertext Preprocessor") — язык скриптования на стороне сервера, встраиваемый в HTML-документ.

Для того, чтобы понять, что это означает, разберем пример простого скрипта:

```
<html>
  <head>
    <title>Пример</title>
  </head>
  <body>
    <?php
      echo "Привет, это PHP-скрипт!";
    ?>
  </body>
</html>
```

Обратите внимание, не программный код выводит на экран теги html-документа, сам гипертекстовый документ содержит небольшой встроенный программный код. Иными словами, вебмастер по-прежнему может создавать гипертекстовые страницы с помощью привычного ему редактора, а затем, при необходимости, встраивать элементы динамической обработки — небольшие скрипты. В рассмотренном примере вся программа состоит из одной строки:

```
echo "Привет, это PHP-скрипт!";
```

Сервер "узнает" о том, что это исполняемая программа по специальным тегам:

```
<? ... ?>
```

Отличие языка PHP от языков клиентского программирования (таких, как JavaScript), состоит в том, что программный код выполняется на стороне сервера, т.е. еще до того, как страницу получит браузер пользователя.

Все страницы, содержащие программный код, должны иметь расширение .php (или .phtml).

## ПРИМЕРЫ с комментряями:

### 1) Вывод на экран и переменные в PHP.

С помощью текстового редактора "Блокнот" (Notepad) создайте файл testphp1.phtml и поместите в начало страницы следующий код:

```
<html>
<head>
  <title>Вывод на экран и переменные в PHP</title>
  <?php
    echo "Привет, мир!";
  ?>
</head>
<body>
</body>
</html>
```

Попробуем ввести в скрипт переменные каждого из перечисленных типов:

```
<html>
<head>
  <title>Вывод на экран и переменные в PHP</title>
</head>
<body>
  <?php
    $i = 6; // целое
    $d = 4.89; // дробное
    $str = "PHP для всех!"; // строка
    echo ($i + $d);
    echo "<br>Привет, мир! ".$str;
  ?>
</body>
</html>
```

Обратите внимание, что все переменные в PHP должны начинаться с символа dollar (\$), тип переменной не задается в явном виде, он вычисляется препроцессором PHP в зависимости от контекста. В результате работы скрипта на экран первой строкой будет выведена сумма значений двух переменных \$i и \$d и результат конкатенации двух строк "<br>Привет, мир!" и значения переменной \$str.

Поддерживаются все арифметические операции и функции, многоуровневые скобки, логические операции, операции увеличения или уменьшения на единицу и многое другое. Кроме того, очень просто и естественно организуется сравнение если — то — иначе. Для этого в PHP применяется конструкция:

```

if (условие)
{
    код для выполнения N1;
}
else
{
    код для выполнения N2;
}

```

Рассмотрим несложный пример:

```

<html>
<head>
<title>Вывод на экран и переменные в PHP</title>
</head>
<body>
<?php
    $i = 6; // целое
    $v = 7;
    $d = 4.89; // дробное
    $str = "PHP для всех!"; // строка
    echo ($i + $d);
    echo "<br>Привет, мир!".$str;
    if($i == $v)
    {
        echo $i + $v;
    }
    else
    {
        echo $i.$v;
    }
?>
</body>
</html>

```

При сравнении на истину применяется два знака равенства для того, чтобы интерпретатор мог без труда отличить сравнение от присваивания. Результат работы скрипта — 67, т.к \$i не равно \$v, а команда echo \$i.\$v; объединяет две строки: "6" и "7". Неравенство (ложь) обозначается символами !=, допустимы все остальные арифметические и логические символы и операторы (например, or, and, >, <= и т.д.).

В PHP есть средства быстрого изменения переменной на единицу в сторону увеличения или уменьшения. Для этого нужно указать имя переменной и за ней, без знака равенства, — подряд два плюса или минуса

соответственно. Например, `$a++`; — переменная `$a` будет увеличена на единицу.

Поддерживается одновременное присваивание одного значения нескольким переменным — `$a = $b = 4`; Введите на странице `testphp1.php` следующий код:

```
<html>
<head>
  <title>Вывод на экран и переменные в PHP</title>
</head>
<body>
  <?php
    $b = $a = 5;
    echo "<br>переменная a=$a, b=$b";
    $c = $a++;
    echo "<br>переменная a=$a, c=$c";
    $e = $d = ++$b;
    echo "<br>переменная e=$e, d=$d, b=$b";
    $g = 10;
    $h = $g += 10;
    echo "<br>переменная e=$e, d=$d, b=$b";
  ?>
</body>
</html>
```

Обратите внимание, что операция присваивания тоже дает свой результат, так операция `$a = 5` дала результат 5, именно поэтому переменная `b` была проинициализирована в 5.

В операции `$c = $a++`; сначала было произведено присваивание, а затем переменная `$a` была увеличена на 1.

В операции `$d = ++$b`; сначала была увеличена на 1 переменная `$b`, а уже затем присвоено получившееся значение.

**2) Передача параметров по ссылке, передача параметров из формы (GET и POST -- запросы).**

С помощью текстового редактора "Блокнот" (Notepad) создайте файл `testphp2.php` и поместите в начало страницы следующий код:

```
<html>
<head>
  <title>Вывод на экран входящих параметров</title>
</head>
<body>
  <?php
    echo $message.", ".$name;
```

```
?>
</body>
</html>
```

Обратите внимание, что переменные `message` и `name` передаются в адресной строке, отделяются от адреса страницы знаком `?`, между собой разделены амперсантом (`&`). Вы можете изменить значения переменных прямо в адресной строке, нажать `Enter` и получить другой результат работы скрипта.

Создайте страницу `testphp2_form.html`. Задайте на ней форму:

```
<form name="myform" method="get" action="testphp2.phtml">
  <br>Сообщение:<br>
  <input type="text" name="message">
  <br>Имя:<br>
  <input type="text" name="name">
  <br><input type="submit" value="Обработать данные">
</form>
```

Проверьте работоспособность скрипта.

Обратите внимание, что на странице с формой элементам назначены имена, соответствующие именам переменных в принимающем скрипте.

Попробуйте изменить метод `GET` на метод `POST`, теперь значения переменных не должны передаваться в открытом виде.

Вместо элемента `<input type="text" name="name">` определите на странице выпадающее меню, такое, как приведено ниже:

Измените работу скрипта таким образом, чтобы пользователь мог выбрать имя из списка, ввести приветствие, нажать кнопку "Обработать данные" и получить приветствие для выбранного имени.

Измените страницу с формой так, чтобы пользователь мог выбрать приветствие, нажав кнопку зависимой фиксации (`radiobutton`):

```
Привет
Здравствуйте
Рад Вас снова видеть
Добрый день
```

Измените страницу с формой так, чтобы пользователь мог выбрать фон страницы (средство выбора — на Ваше усмотрение).

### 3) Динамическое формирование страницы.

Создайте 3 файла с именами: `testphp3_inc1.html`, `testphp3_inc2.html` и `testphp3_inc3.html`. Поместите небольшой текст в каждый.

Создайте файл `testphp3.phtml`, поместите в него следующий код:

```

<html>
<head>
  <title>Динамическое формирование страниц</title>
</head>
<body bgcolor=silver>
  <?php
    $file = "";
    if ($link == 1) { $file = "testphp3_inc1.html"; }
    if ($link == 2) { $file = "testphp3_inc2.html"; }
    if ($link == 3) { $file = "testphp3_inc3.html"; }
    if ($file == "") { ?>
      <h3>Пожалуйста, выберите стихотворение:</h3>
      <a href="testphp3.phtml?link=1"> текст N1</a><br>
      <a href="testphp3.phtml?link=2"> текст N2</a><br>
      <a href="testphp3.phtml?link=3"> текст N3</a>
      <? } else {
        include($file);
      }
    ?>
  </body>
</html>

```

Сохраните созданные файлы, затем запустите файл testphp3.phtml.

При выборе ссылки скрипту передается в качестве параметра идентификатор страницы, содержание которой должно быть включено в результирующий файл.

Измените скрипт таким образом, чтобы тексты включались в таблицу с желтым фоном; ссылка на страницу, которая импортирована в файл в настоящий момент, была неактивна.

Создайте массив \$titles, который будет содержать названия текстов. Из этого массива должно формироваться содержимое тега <title>...</title> в зависимости от импортируемой страницы.

Внизу страницы поместите ссылки Предыдущее, К оглавлению и Следующее. Ссылка Предыдущее должна вести к предыдущему (по порядковому номеру) стихотворению, ссылка Следующее должна вести к следующему (по порядковому номеру) стихотворению. Ссылки должны исчезать со страницы, если предыдущие/следующие стихотворения исчерпаны. Ссылка К оглавлению должна приводить страницу в первоначальный вид и исчезать на странице оглавления.

Перенесите файлы testphp3\_inc1.html, testphp3\_inc2.html и testphp3\_inc3.html в папку includes и внесите соответствующие изменения в скрипт.

#### 4) Работа с файлами.

Создайте файл `testphp4_form.html`, поместите в него форму, которая будет содержать текстовое поле `<textarea name=message cols=10 rows=4></textarea>`, поле редактирования `<input type=text name=person>` и кнопку для отправки данных.

Настройте форму таким образом, чтобы данные направлялись скрипту `testphp4.phtml`.

Создайте файл `testphp4.phtml`, поместите в него следующий код:

```
<html>
<head>
  <title>Гостевая книга</title>
</head>
<body bgcolor=silver>
  <?php
    $filename = "messages.txt";
    $fp = fopen($filename,"a");
    if($fp)
    {
      fputs($fp, $message." ".$person);
      fclose($fp);
    }
    include($filename);
  ?>
</body>
</html>
```

Создайте пустой файл `messages.txt` — в него будут сохраняться все сообщения пользователей.

Запустите файл `testphp4_form.html`, убедитесь в работоспособности скрипта.

Должно происходить следующее: переменные `$message` и `$person` принимаются скриптом `testphp4.phtml`, открывается файл `messages.txt` (делается это методом `fopen`, подробнее о нем читайте), помещает в его конец строку, состоящую из значений переданных переменных, разделенных пробелом (делается это методом `fputs`, подробнее о нем читайте).

Измените работу скрипта таким образом, чтобы сообщения и имена пользователей выводились в отформатированном виде, например, поместите их в ячейки таблицы.

Измените работу скрипта таким образом, чтобы форма находилась в файле `testphp4.phtml`, иными словами, удалите файл `testphp4_form.html`.

В начало страницы поместите `guestbook.jpg`.