

УВАЖАЕМЫЕ СТУДЕНТЫ! Изучите теоретические сведения к лабораторной работе, выполните пример и задание согласно вашему варианту.

Результаты работы, фотоотчет, предоставить преподавателю на e-mail: igor-gricenko-95@mail.ru **в течении ТРЕХ дней**

Требования к отчету:

Отчет предоставляется преподавателю в электронном варианте и должен содержать:

- название работы, постановку цели, вывод;
- ответы на контрольные вопросы, указанные преподавателем.

При возникновении вопросов по приведенному материалу обращаться по следующему номеру телефона: (072)132-63-42,

ВНИМАНИЕ!!! При отправке работы, не забывайте указывать ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).

Лабораторная работа № 4

Тема: Создание простых программ на языке PHP. Построение логических выражений на языке PHP. Построение скриптов оператора условного перехода и циклических конструкций в языке PHP.

Цель работы: Изучить основы языка PHP.

Базовые сведения:

Синтаксис PHP

Синтаксис PHP во многом заимствован из таких языков как C, Java и Perl. Файл, обрабатываемый сервером как правило имеет расширение php.

Поэтому, если вы знакомы хотя бы с одним из них - вам не составит особого труда просто сесть и начать писать программы на PHP.

PHP-код включаются в html-код в следующем виде:

```
<?PHP текст_кода ?>
```

или

```
<? текст_кода;?>
```

Комментарии

PHP поддерживает комментарии 'C', 'C++' и оболочки Unix. Например:

```
<?php echo "This is a test"; // Это однострочный комментарий в стиле c++
```

```
/* Это многострочный комментарий,  
это ещё одна его строка */  
echo "This is yet another test"; echo "One Final Test";  
# Это комментарий в shell-стиле ?>
```

echo

```
<?php echo "Эта информация будет выведена в HTML";?>
```

Присвоение значений переменным

Переменные в программах на PHP, отделяются символами \$.

```
$city = "Tula";
```

city - переменная

Tula - значение

Некоторые операции

инкремента/декремента;

++\$a Pre-increment Увеличивает \$a на 1, затем возвращает \$a.

\$a++ Post-increment Возвращает \$a, затем увеличивает \$a на 1.

--\$a Pre-decrement Уменьшает \$a на 1, затем возвращает \$a.

\$a-- Post-decrement Возвращает \$a, затем уменьшает \$a на 1.

арифметические:

\$a + \$b Сложение Сумма \$a и \$b.

\$a - \$b Вычитание Разность \$a и \$b.

\$a * \$b Умножение Произведение \$a и \$b.

\$a / \$b Деление Частное от деления \$a на \$b.

\$a % \$b Modulus Целочисленный остаток от деления \$a на \$b.

строковые:

Имеются две строковые операции. Первая - операция ('.'), которая возвращает объединение из правого и левого аргументов. Вторая - операция присвоения ('.='), которая присоединяет правый аргумент в левому аргументу.

```
$a = "Hello "; $b = $a . "World!"; // теперь $b содержит "Hello World!"
```

```
$a = "Hello "; $a .= "World!"; // теперь $a содержит "Hello World!"
```

Выражения сравнения

Выражения сравнения вычисляются в 0 или 1, означая FALSE или TRUE (соответственно).

PHP поддерживает

> (больше),

`>=` (больше или равно),
`==` (равно),
`!=` (не равно),
`<` (меньше) и `<=` (меньше или равно).

Эти выражения чаще всего используются внутри условных операторов, таких как `if`.

сравнения:

`$a == $b` равно `TRUE`, если `$a` равно `$b`.

`$a != $b` не равно `TRUE`, если `$a` не равно `$b`.

`$a <> $b` не равно `TRUE`, если `$a` не равно `$b`.

`$a < $b` меньше `TRUE`, если `$a` строго меньше `$b`.

`$a > $b` больше `TRUE`, если `$a` строго больше `$b`.

`$a <= $b` меньше или равно `TRUE`, если `$a` меньше или равно `$b`.

`$a >= $b` больше или равно `TRUE`, если `$a` больше или равно `$b`.

Некоторые операторы

`include` "имя файла"

- команда для включения содержимого одного файла в другой. Содержимое файла, имя которого указывается в команде, целиком и полностью вставляется на то место, где располагается эта команда, при этом все коды PHP, содержащиеся во вставляемом файле, исполняются так же, как если бы они были на месте этой команды. (Помните, что файл именно вставляется - т. е., например, пути к картинкам, которые должны присутствовать во вставляемом файле, следует указывать от местонахождения того файла, в котором находилась команда `include`.) Если файл, включаемый в страницу при помощи команды `include`, отсутствует, то вместо него размещается уведомление об этом, а программа на PHP выполняется дальше. (При необходимости завершения обработки и выдачи web-страницы в случае отсутствия включаемого файла, вместо команды `include` следует использовать команду `require`.)

`mail` ("Кому", "Тема", "Текст сообщения", "Дополнительные заголовки")

- отправка почтового сообщения. При выполнении данной команды на сервере в соответствии с указанными параметрами формируется электронное письмо и отправляется с помощью установленной на сервере почтовой программы. В качестве параметра "Кому" может выступать набор адресов, разделенных запятыми. "Дополнительные заголовки" могут быть любые (естественно, допустимые

почтовыми протоколами!), разделяться они должны комбинацией символов /n, которая в РНР означает перевод строки. (Если среди "Дополнительных заголовков" не указано поле From, то оно заполняется по умолчанию почтовой программой web-сервера, например, именем "Unprivileged User".)

echo ("текст")

- вывод на web-страницу какого-либо текста. Чтобы вывести на web-страницу значение какой-либо переменной, достаточно просто написать ее имя внутри выводимой строки: команда echo "это цифра \$a" выведет в web-страницу текст "это цифра 1", если ранее переменной \$a было присвоено значение, равное единице. В случае необходимости использовать в выводимой строке кавычки или иные специальные символы перед этими символами следует ставить символ " .

if (условие) {...команды, которые должны выполняться, если условие верно...;}

else {...команды, которые должны выполняться, если условие неверно...}

-команда, позволяющая выполнить то или иное действие в зависимости от истинности верности или ложности того или иного условия. В фигурных скобках может располагаться несколько команд, разделенных точкой с запятой.

for (начальное значение счетчика, условие продолжения цикла, изменение счетчика на каждом цикле) { ...команды... ;}

- цикл, т. е. повторение указанных в нем команд столько раз, сколько позволит условие изменения счетчика цикла (т. с. переменной, специально выделенной для подсчета числа выполнений команд цикла).

while (условие) { ...команды... }

- цикл с условием. Команды в фигурных скобках выполняются до тех пор, пока выполняется условие в заголовке цикла. Для того чтобы цикл прервался, нужно, чтобы условие выполняться перестало - поэтому внутри цикла необходимо предусмотреть возможность влиять на это условие.

Цикл do { . . .команды. . . } while (условие)

работает так же, однако команды, указанные в фигурных скобках, будут выполнены по меньшей мере один раз - даже если условие выполняться не будет. Прервать выполнение любого цикла можно оператором break -дальнейшее выполнение программы пойдет с команды, следующей после закрывающей фигурной скобки.

Оператор же continue прерывает текущую стадию выполнения цикла, т. е. после этого оператора дальнейшее выполнение программы начнется с очередной проверки условия заголовка цикла.

switch (выражение) {case значение: ... команды...; break; case другое значение: ... команды...; break;}

- оператор выбора. При его работе содержимое, заключённое в фигурные скобки, просматривается сверху вниз. Как только будет найден оператор case со значением, совпадающим со значением выражения, PHP начнёт выполнять весь код, следующий за этим оператором case до последней фигурной скобки оператора switch или до первого оператора break, в зависимости от того, что появится раньше. (Обратите внимание, что если команду break не указать в конце кода, относящегося к одному варианту значения выражения в заголовке оператора switch, PHP будет выполнять код дальше - т. е. тот, который принадлежит уже следующему оператору case! Это - одно из отличий данного оператора от аналогичных в других языках программирования.) В конце оператора switch можно указать оператор default. Код, стоящий после него, выполнится в том случае, если значение выражения в заголовке оператора не совпадет ни с одним из значений после операторов case.

foreach (переменная as массив) { . . .команды. .. ;}

- поочередное считывание всех элементов массива. Foreach считывает в указанную в его параметрах переменную поочередно все элементы указанного в них же массива, выполняя каждый раз указанный в фигурных скобках код, в котором может использоваться указанная переменная. (Значения элементов массива этим оператором только считываются, их модификация при помощи команды foreach невозможна.)

Оператор foreach может быть использован только в PHP версии 4.0 и выше.

Программа на PHP может прерываться кодом web-страницы - для этого достаточно вставить закрывающий тэг до этого кода и открывающий - после. Все, что находится между ними, будет выдаваться в браузер без какой-либо обработки, рассматриваясь как выводимое с помощью команды echo. Иными словами, код

```
<?php if ($a==1) { ?<p>Переменная a равна 1</p><?php } ?>
```

эквивалентен коду

```
<?php if ($a==1) {echo "<p> Переменная a равна 1</p>";} ?>
```

Однако, первый вариант меньше нагружает процессор компьютера, на котором расположен интерпретатор PHP. Из сказанного также следует, что все программы на PHP, расположенные на одной web-странице, представляют собой одну большую программу, несмотря на то, что они разделяются блоками обычного текста страницы. Именно поэтому переменная, объявленная в расположенном в начале страницы коде, сохраняет свое значение не только до ее конца, но и во всех присоединяемых с помощью команды include файлах.

Пример

```
<?
$rows=5;
$columns=3;
echo '<html><body>';
echo '<table border="1">';
for ($i=1;$i<=$rows;$i++){
echo '<tr>';
  for ($j=1;$j<=$columns;$j++)
  {
  if (((($i+$j) % 2)==0)
($color="#000000");)
else
($color="#ffffff");)
echo "<td bgcolor=$color>$i,$j</td>";
} /* end of for $j*/
echo '</tr>';
} /* end of for $i*/
echo '</table>';
echo '</body></html>';
?>
```

Figure: Пример php.

Задания на лабораторную работу

1. Создайте отдельную страницу, на которой будут размещаться ссылки на php скрипты.
2. Реализуйте указанный пример, выясните его принцип работы.
3. Получите информацию о настройках php с помощью команды `phpinfo()`;
4. Создайте php-скрипт, который создает страницу "Hello World!!!";
5. Создайте php-скрипт, который загружает в web-страницу файл с информацией о разработчике.
6. Создайте php-скрипт, который создает страницу с таблицей возможных цветов HTML.
7. Создайте php-скрипт, который формирует web-страницу с таблицей умножения.