

Тема: «Комментарии. Типы данных. Переменные. Массивы. Операторы»

Цель: Изучить синтаксис PHP. Ознакомиться с основными инструментами языка, научиться создавать массивы.

Теоретическая часть

Введение

Соединение XHTML и PHP

Код *PHP* обычно объединяется с тегами *XHTML*. *PHP* является встраиваемым языком — это означает, что можно перемещаться между чистым кодом *HTML* и *PHP*, не жертвуя возможностью чтения текста.

Чтобы встроить код *PHP* в *XHTML*, *PHP* должен задаваться обособленно, с помощью начального и конечного тегов *PHP*. Теги *PHP* говорят серверу *Web*, где начинается и заканчивается код *PHP*. Анализатор *PHP* распознает три варианта начального и конечного тегов.

Варианты встраивания сценариев *PHP* в документ *XHTML*

Стандартный стиль	Сокращенный стиль	Стиль сценария (script)
<code><?php</code> Блок кода PHP <code>?></code>	<code><?</code> Блок кода PHP <code>?></code>	<code><script</code> <code>language="php"></code> Блок кода PHP <code></script></code>

Предпочтительным является первый вариант.

Блоки сценария могут размещаться в любом месте документа *XHTML*, в том месте, где *сценарий* создает и показывает свой *вывод*.

Пример:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD/XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml11-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en">
<head>
  <title>Страница Web </title>
</head>
<body>
<p>
<?php echo "Это базовый документ PHP";?>
</p>
<p>
<? print "PHP - это здорово!";?>
</p>
<p>
<script language="php">
$myvar = "Hello World! ";
echo $myvar;
</script>
</p>
</body>
</html>
```

Вывод контента

PHP содержит два основных оператора для вывода текста в браузере *Web*: `echo` и `print`.

Операторы echo и print используют следующий формат:

echo "Выводимый текст"; (одна или несколько строк)

print "Выводимый текст"; (одна строка)

Пример:

```
<p>
```

```
<?php
```

```
echo "Это базовый документ PHP";
```

```
?>
```

```
</p>
```

По умолчанию операторы echo и print не создают автоматические переносы строк, необходимо использовать тег <p> или
 для создания параграфов или переносов строк:

```
echo "<p>Параграф 1</p>";
```

```
echo "<p>Параграф 2</p>";
```

Каждая строка кода PHP должна завершаться терминатором инструкции (признаком конца), в качестве которого используется точка с запятой (;).

Комментарии в коде

Комментарии применяются в PHP для записи собственных замечаний во время процесса разработки кода.

Комментарии в PHP можно определить одним из следующих способов:

// — простой комментарий PHP;

— альтернативный простой комментарий PHP;

/*...*/ — многострочные блоки комментариев.

Пример:

```
<?php
```

```
// Простой комментарий PHP
```

```
# Другой тип простого комментария PHP
```

```
/* Многострочный блок комментария PHP
```

```
Он может занимать любое необходимое
```

```
количество строк */
```

```
?>
```

Переменные (типы данных)

Переменные являются временным местом хранения данных, используемым для представления значений в сценарии PHP. В PHP имеется два основных типа переменных: скалярные и массивы. Скалярные переменные содержат только одно значение в данный момент времени, а переменные массивы — список значений.

Типы переменных:

1. **Целые** – целые числа или числа без десятичной точки (1, 999, 325812841).
2. Числа **с плавающей точкой** – числа, содержащие десятичную точку (1.11, 2.5, .44).
3. **Строки** – текстовая или числовая информация. Строковые данные всегда определяются с помощью кавычек ("Hello World", "478-477-5555").
4. **Булевы значения** – используются для значений true (истина) или false (ложь).

Имена переменных PHP всех типов начинаются со знака "\$". Имена переменных могут содержать буквы, числа, и символ подчеркивания (_); они не могут, однако, начинаться с цифры. В PHP имена переменных различают регистр символов.

Пример:

```
$myvar
```

```
$F_Name
$address1
$my_string_variable
```

Вывод переменных

Переменные выводятся без кавычек.

```
<?php
```

```
$string_var = "Моя программа PHP";
$integer_var = 500;
$float_var = 2.25;
```

```
echo $string_var;
echo $integer_var;
echo $float_var;
```

```
?>
```

Значения переменных будут выведены в одну строку:

```
Моя программа PHP5002.25
```

Для разбивки на строки нужно в оператор **echo** добавить тэг "**
**"

Массивы

Массив - это пронумерованный набор переменных, представляющих собой единое целое.

Переменные, входящие в массив, называются *элементами*. Доступ к каждому элементу выполняется по его порядковому номеру, называемому *индексом*. А общее число элементов этого массива называется его *размером*.

Нумерованные массивы

Первый способ создания массива (создание ячеек).

Создается массивы с именем *\$name* и *размером* равным 5.

```
<? php                               <? php
$name [ 0 ] = "A";                    $name [ ] = "A";
$name [ 1 ] = "B";                    $name [ ] = "B";
$name [ 2 ] = "C";                    $name [ ] = "C";
$name [ 3 ] = "D";                    $name [ ] = "D";
$name [ 3 ] = "E";                    $name [ ] = "E";
?>                                     ?>
```

Оба массива одинаковы. Во втором индексы устанавливаются автоматически, начиная со значения равного "0"

Для вывода значения нужной ячейки массива используем оператор **echo**:

```
/*Вывод значения ячейки массива с индексом 2 на экран*/
```

```
echo $name[2]; /*Результат "C"*/
```

Второй способ создания массива (использование оператора *array*).

Создается такой же массив с именем *\$name* и *размером* равным 5, при условии, что индексы устанавливаются по умолчанию, т.е. начиная с нуля.

```
$name = array ("A", "B", "C", "D", "E");
echo $name[2]; /*Результат "C"*/
```

Если нумерация индексов произвольная, то нужно задавать значения массива совместно с индексами:

```
$name = array (2 => "A", 3 => "B", 4 => "C", 5 => "D", 6 => "E");  
echo $name[2]; /*Результат "A"*/
```

Ассоциативные массивы

В ассоциативных массивах индексы обозначаются текстовыми символами (словами).

```
$color["white"] = "белый";  
$color["black"] = "черный";  
$color["red"] = "красный";  
$color["green"] = "зеленый";  
$color["blue"] = "синий";
```

Второй способ создания ассоциативного массива.

```
$color = array("white" => "белый", "black" => "черный", "red" =>  
"красный", "green" => "зеленый", "blue" => "синий");
```

Вывод на экран

```
echo "Выбран ".$color["red"]." цвет";  
/*Результат: «Выбран красный цвет»*/
```

Многомерные массивы PHP

В качестве элементов массива могут выступать другие массивы, в этом случае говорят о многомерных массивах:

```
$arr = array(  
    "Овощи" => array("Огурец", "Помидор", "Тыква"),  
    "Фрукты" => array("Абрикос", "Апельсин", "Банан")  
);
```

Выводим на экран с помощью специальной функцией для работы с массивами `print_r`:

```
echo "<pre>";  
print_r ($arr);  
echo "</pre>";
```

Получим на экране двумерный массив:

```
Array  
(  
    [Овощи] => Array  
        (  
            [0] => Огурец  
            [1] => Помидор  
            [2] => Тыква  
        )  
    [Фрукты] => Array  
        (  
            [0] => Абрикос  
            [1] => Апельсин  
            [2] => Банан  
        )  
)
```

)

Операторы

Оператор – это команда языка PHP, выполняющая какое-либо действие над данными (аргументом) или сценарием. Один из операторов – это оператор вывода данных **echo**.

Арифметические операторы служат для выполнения арифметических действий над аргументами (таблица 1).

Оператор	Описание	Действие
-	Смена знака числа на противоположный	$\$r = -\r
+	Сложение	$\$a = \$b + \$c$
-	Вычитание	$\$c = \$a - \$b$
*	Умножение	$\$c = \$a * \$b$
/	Деление	$\$c = \$a / \$b$
++	Инкремент (увеличение на единицу)	$\$t = \$r++$
--	Декремент (уменьшение на единицу)	$\$t = \$r--$

Таблица 1. Арифметические операторы

Оператор объединения строк (.) – точка позволяет соединить переменные и строки.

```
$s1 = "Иванов";
```

```
$s2 = "Петр";
```

```
$s3 = "Александрович";
```

```
$output = $s1 . " " . $s2 . " " . $s3; (кавычками обозначены пробелы)
```

На экране будет результат: **Иванов Петр Александрович**

Операторы присваивания (=). С помощью оператора присваивания переменной присваивается значение, находящееся справа от знака присваивания:

```
$a = 2; $c = $a - $b; $a = $a + $b; или $a += $b;
```

Операторы сравнения. Они сравнивают два аргумента согласно какому-то условию и возвращают логическое значение «истина» (true) или «ложь» (false). В числовом выражении true равно 1, а false равно 0. Все операторы сравнения приведены в таблице 2.

Оператор	Описание
<	Меньше
>	Больше
==	Равно
<=	Меньше и равно
>=	Больше и равно
!= или <>	Не равно
===	Строго равно
!==	Строго не равно

Таблица 2. Операторы сравнения PHP

Логические операторы. Выполняют действия над логическими значениями (Таблица 3.).

Оператор	Описание
!	НЕ (логическая инверсия)
?? или and	И (логическое умножение)
или or	ИЛИ (логическое сложение)

Практическая часть

Для устранения проблемы с кодировкой страницы установите в начале странице следующий код:

```
<?php // header("Content-type: text/html; charset=utf-8"); ?>
```

Задание 1. Встраивание PHP код в HTML код

1. Создайте переменную **name** и присвойте ей значение содержащее Ваше имя, например "Игорь" (обязательно в кавычках!)
2. Создайте переменную **age** и присвойте ей значение содержащее Ваш возраст, например 40
3. Выведите с помощью echo (или print) фразу "**Меня зовут: ваше_имя**", например: "**Меня зовут: Игорь**" подставляя значение переменной name
4. Выведите фразу: "**Мне ваш_возраст лет**", например: "**Мне 40 лет**" подставляя значение переменной age
5. Измените код так, чтобы каждая фраза начиналась с новой строки

Задание 2. Работа с переменными и массивами

1. Создать две переменные:

```
$linkAddr = "http://www.volkpb.ru";  
$linkText = "Это Мой колледж!";
```
2. Создать ссылку, в которой адрес и текст выводится средствами PHP с использованием этих переменных.

```
<a href="$linkAddr" >$linkText</a>
```
3. Создайте массив с перечислением дней недели. Просмотреть массив с помощью функции **print_r**
4. Создайте массив с выводом номеров рабочих мест и ваших фамилий. Просмотреть массив с помощью функции **print_r**.
5. Создайте массив с занесением оценок по предметам: Иванов (алгебра – 4, химия – 5, литература – 3); Петров (алгебра – 3, химия – 3, литература – 5); Сидоров (алгебра – 5, химия – 3, литература – 4). Вывести на экран весь массив. Вывести на экран оценки Петрова.
 Просмотреть массив с помощью функции **print_r**.
6. Создайте массив **\$arr** с элементами 'a', 'b', 'c', 78 двумя различными способами.
7. Создайте массив **\$arr** с элементами 'Я', 'учу', 'PHP', '!'. С его помощью выведите на экран фразу 'Я учу PHP!'.

Задание 5. Операции с целыми числами

1. Выполните следующие операции с целыми числами

```
$a = 3;
```

```
$b = 2;
```

Найдите сумму и произведение этих чисел с выводом результатов на экран: «Сумма чисел a и b равна 5»; «Произведение чисел a и b равно 6». Создайте переменные **\$a=10** и **\$b=2**.

Выведите на экран их сумму, разность, произведение и частное (результат деления).

3. Создайте переменные **\$c=15** и **\$d=2**. Просуммируйте их, а результат присвойте переменной **\$result**. Выведите на экран значение переменной **\$result**.

4. Создайте переменные **\$a=10**, **\$b=2** и **\$c=5**. Выведите на экран их сумму.
5. Создайте переменные **\$a=17** и **\$b=10**. Отнимите от **\$a** переменную **\$b** и результат присвойте переменной **\$c**. Затем создайте переменную **\$d**, присвойте ей значение 7. Сложите переменные **\$c** и **\$d**, а результат запишите в переменную **\$result**. Выведите на экран значение переменной **\$result**.