



PHP

Данные PHP

Урок 2



1. Что такое PHP?
2. Какие операторы в PHP позволяют вывести текст на странице?

php



Типы данных PHP

Тип	Пример	Описание
Integer	3	Целое число
Double	3.234	Число с плавающей точкой
String	"hello"	Строка символов
Boolean	true	Логическое значение (true / false)
Object		Объект
Array		массив



Скалярные переменные

Целые – целые числа или числа без десятичной точки (8, 777, 432512512).

Числа с плавающей точкой – числа, содержащие десятичную точку (2.22, 2.8, .54).

Строки – текстовая или числовая информация. Строковые данные всегда определяются с помощью кавычек («Привет Мир», «8-800-3535»).

Булевы значения – используются для значений true (истина) или false (ложь).



Объявление переменных

Формат скалярных переменных:

```
$hostname = "lui"
```

```
$computer_name = "PC"
```

```
$file_name = "Doc"
```



Вывод переменных

```
1 ▼ <html>
2 ▼ <head>
3   <title>Страница Web </title>
4 </head>
5 ▼ <body>
6 ▼ <p>
7   <?php
8     $string_var = "Это моя программа на PHP";
9     $integer_var = 1000;
10    $float_var = 13.17;
11
12    echo $string_var;
13    echo $integer_var;
14    echo $float_var;
15    ?>
16  </p>
17
18  </body>
19 </html>
```



Вывод переменных

← → ↻ ⓘ localhost/dashboard/test/nov.php

Это моя программа на PHP100013.17



Вывод переменных

```
1 ▼ <html>
2 ▼ <head>
3   <title>Страница Web </title>
4 </head>
5 ▼ <body>
6 ▼ <p>
7   <?php
8     $string_var = " PHP программ YEAH". "<br/>";
9     $integer_var = 1111 . "<br/>";
10    $float_var = 12.18;
11    echo $string_var;
12 echo $integer_var;
13 echo $float_var;
14 ?>
15   </p>
16
17   </body>
18 </html>
```




Вывод переменных

10

← → ↻ ⓘ localhost/dashboard/test/nov.php

PHP programm YEAH
1111
12.18



Соединение переменных

```
1 ▼ <html>
2 ▼ <head>
3   <title>Страница Web </title>
4 </head>
5 ▼ <body>
6 ▼ <p>
7   <?php
8
9   $fname = "Anakin";
10  $lname = "Skywalker";
11
12  echo "The user\'s name is " . $fname ." ". $lname;
13
14  ?>
15
16    </p>
17
18    </body>
19  </html>
```



Соединение переменных

12

← → ↻ ⓘ localhost/dashboard/test/nov.php

The user's name is Anakin Skywalker



```
1 ▼ <html>
2 ▼ <head>
3   <title>Страница Web </title>
4 </head>
5 ▼ <body>
6 ▼ <p>
7   <?php
8
9   $fname = "Anakin";
10  $lname = "Skywalker";
11
12  echo "The user\'s name is $fname $lname";
13  ?>
14
15   </p>
16
17   </body>
18 </html>
```



Соединение переменных

14

← → ↻ ⓘ localhost/dashboard/test/index.php

The user's name is Anakin Skywalker



ПРОСТОЙ МАССИВ

	0	1	2	3	4
\$name =	A	B	C	D	E

АССОЦИАТИВНЫЙ МАССИВ

	white	black	red	green	blue
\$name =	белый	черный	красный	зеленый	синий



Массивы с числовыми индексами

```
1 ▼ <html>
2 ▼ <head>
3   <title>Страница Web </title>
4 </head>
5 ▼ <body>
6 ▼ <p>
7   <?php
8
9   $my_array = array('blue', 'purple', 'orange');
10
11  echo "Значение массива - " . $my_array[0]."<br/>";
12  echo "Значение массива - " . $my_array[1]."<br/>";
13  echo "Значение массива - " . $my_array[2]."<br/>";
14  ?>
15
16   </p>
17
18   </body>
19 </html>
```




Массивы с числовыми индексами

17

← → ↻ ⓘ localhost/dashboard/test/index.php

Значение массива - blue

Значение массива - purple

Значение массива - orange



count() – функция `count` используется для подсчета числа элементов в массиве.

sort() – функция `sort` используется для сортировки элементов существующего массива.

shuffle() – функция `shuffle` используется для случайного перемешивания элементов в заданном массиве.

sizeof() – функция `sizeof` является синонимом (алиасом) функции `count()`.



`array_slice($array_name, offset, length)` – функция `array_slice` нужна для изъятия части существующего массива. `$array_name` является именем разрезаемого массива, `offset` показывает позицию, где будет начинаться разрез, `length` - число элементов, которое будет вырезано из массива.

`array_merge($array_name, $array_name)` – функция `array_merge` - объединяет или сливает два или большее количество существующих массивов. Имена массивов разделяются запятыми.



Суперглобальные переменные:

\$_SET[]

\$_COOKIE[]

\$_REQUEST[]

\$_COOKIE[]

\$_FILES[]

\$_SERVER[]

\$_ENV[]

\$_BACK[]



Целые – целые числа или числа без десятичной точки (8, 777, 432512512).

Числа с плавающей точкой – числа, содержащие десятичную точку (2.22, 2.8, .54).

Строки – текстовая или числовая информация. Строковые данные всегда определяются с помощью кавычек («Привет Мир», «8-800-3535»).

Булевы значения – используются для значений true (истина) или false (ложь).



Вывод констант

```
1 ▼ <html>
2 ▼ <head>
3   <title>Страница Web </title>
4 </head>
5 ▼ <body>
6 ▼ <p>
7   <?php
8   define("STRING_CONST","PHP program YEAH");
9   define("INTEGER_CONST",1000);
10  define("FLOAT_CONST",12.18);
11
12
13  echo STRING_CONST;
14  echo INTEGER_CONST;
15  echo FLOAT_CONST;
16  ?>
17    </p>
18
19    </body>
20  </html>|
```



← → ↻ ⓘ localhost/dashboard/test/index.php

PHP program YEAN100012.18



Виды строковых литералов:

- строки в одиночных кавычках
- строки в двойных кавычках
- строки в синтаксисе heredoc



Функции для работы со строками

strlen(string) – определяет длину строки `string`.

ltrim(string) – удаляет символы-разделители в начале строки `string`.

rtrim(string) – удаляет символы-разделители в конце строки `string`.

strpbrk(string, char) – ищет в строке `string` символ `char`. Возвращает `false` или строку, начинающуюся с найденного символа.



Функции для работы со строками

strtoupper(string) – преобразует строку в верхний регистр.

strtolower(string) – преобразует строку в нижний регистр.

strrev(string) – возвращает строку `string` в обратном порядке.

eregi(pattern, subject) – выполняет независимое от регистра символов сравнение с выражением. В строке `subject` происходит поиск регулярного выражения, заданного строкой `pattern`.



Функции для работы со строками

```
1 ▼ <html>
2 ▼ <head>
3   <title>Страница Web </title>
4 </head>
5 ▼ <body>
6 ▼ <p>
7   <?php
8   $string = "Hello World";
9   $another_string = "Welcome to the club PHP";
10
11   echo strlen($string);
12   echo strtoupper($another_string);
13   echo strrev($another_string);
14   echo strpbrk($string, "W");
15
16   ?>
17   </p>
18
19   </body>
20 </html>|
```



Функции для работы со строками

28

← → ↻ ⓘ localhost/dashboard/test/index.php

11WELCOME TO THE CLUB PHPPHP bulc eht ot emocleWWorld



Виды операторов:

- Арифметические операторы
- Операторы присваивания
- Операторы сравнения
- Логические операторы



Операторы PHP

Оператор	Описание
+	сложение
-	вычитание
*	умножение
/	деление
%	сравнение по модулю (возвращает остаток от деления)
++	увеличивает значение
--	уменьшает значение



```
1 ▼ <html>
2 ▼ <head>
3   <title>Страница Web </title>
4 </head>
5 ▼ <body>
6 ▼ <p>
7   <?php
8   $sum = 8 + 3; //Это оператор суммы
9   $sum = 11;
10  $newsum = $sum + 5; //Это оператор суммы
11  echo "Из сумма равна " . $newsum;
12  $difference = $newsum - 6;
13  echo "Их вычитание равно " . $difference;
14  $product = $difference * 3; //Это оператор произведения
15  echo "Их произведение равно " . $product;
16  $quotient = $product / $difference; //Это оператор разности
17  echo "Их разность равна " . $quotient;
18  $quotient++; //Оператор инкремента
19  echo $quotient;
20  $quotient --; //Оператор декремента
21  echo $quotient;
22  |?>
23    </p>
24
25    </body>
26 </html>
```



localhost/dashboard/test/index.php

Из сумма равна 16 Их вычитание равно 10 Их произведение равно 30 Их разность равна 343



Порядок выполнения операций

Оператор	Описание
=	Переменной слева присваивается вычисленное справа значение
+=	Складывает значение слева со значением справа и присваивает результат переменной слева
-=	Вычитает значение справа из значения слева и присваивает результат переменной слева
*=	Умножает значение справа на значения слева и присваивает результат переменной слева
/=	Делит значение слева на значение справа и присваивает результат переменной слева
%=	Делит значение слева на значение справа и присваивает остаток (по модулю) переменной слева
.=	Значение слева соединяется (конкатенация) со значением справа, и результат присваивается переменной слева

Задания для самостоятельного решения

34

Напишите программу, которая считывает целое число и выводит текст, аналогичный в примере.

Помните – пробелы, знаки препинания, заглавные и строчные буквы важны!

Входные данные:

Вводится целое число

Выходные данные:

Ответ на задачу

Пример:

Входные данные:

99

Выходные данные:

Следующее число после 99 это 100

Предыдущие число перед 99 это 98



```
1 ▾ <html>
2 ▾ <head>
3   <title>Страница Web </title>
4 </head>
5 ▾ <body>
6 ▾ <p>
7   <?php
8   $n = 99;
9
10   $previous =$n - 1;
11   $following = $n+1;
12   echo " Следующее число после 99 " . $n. "это " . $following ."/br";
13   echo " Предыдущее число перед 99 " . $n. "это " . $previous ."/br";
14   ?>
15   </p>
16
17   </body>
18 </html>
```



Молодцы!

