



ОСНОВЫ ЯЗЫКА PHP

Переменные, типы данных, операторы языка



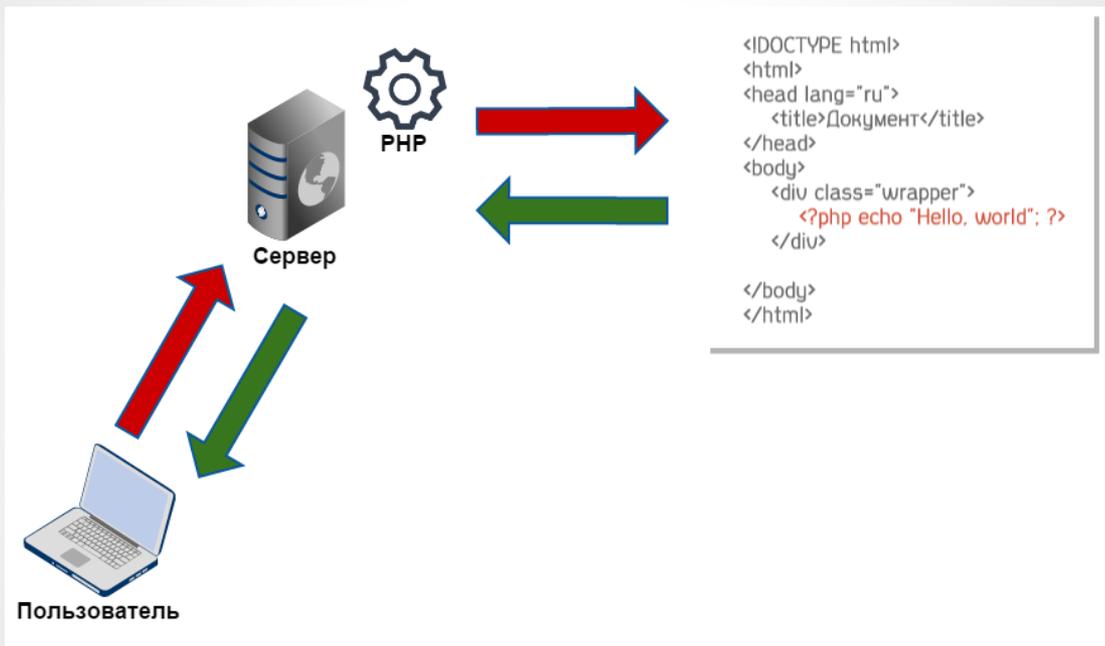
LoftSchool
ОТ МЫСЛИТЕЛЯ К СОЗДАТЕЛЮ

Принцип работы РНР



LoftSchool
ОТ МЫСЛИТЕЛЯ К СОЗДАТЕЛЮ

Как работает PHP



Синтаксис языка



LoftSchool
от мыслителя к создателю

PHP теги

| Так PHP обрабатывает файл

<p>Это будет проигнорировано.</p>

<?php echo 'А это будет обработано.'; ?>

<p>Это тоже будет проигнорировано.</p>

PHP теги

I Два других набора тэгов

- Короткие теги (доступны с директивой `short_open_tag` в `php.ini`)
 - Всегда доступен с PHP 5.4
- Теги в стиле ASP (доступны с директивой `asp_tags` в `php.ini`) - в PHP 7 будут удалены

`<? echo 'это простейшая инструкция обработки'; ?>` // пример коротких тегов

`<% echo 'Вы можете по выбору использовать теги в стиле ASP'; %>`

Комментарии

- I PHP поддерживает комментарии в стиле 'C', 'C++' и оболочки Unix

```
echo "Это тест"; // Это однострочный комментарий в стиле c++  
/* Это многострочный комментарий  
еще одна строка комментария */  
echo "Это еще один тест";  
echo "Последний тест"; # Это тоже однострочный комментарий
```

Особенности PHP

- | `<?php` какая-то инструкция; `?>`
- | `<?php` инструкция 1; инструкция 2; `?>`
- | `<?php`

какая-то

инструкция

в

несколько

строк;

`?>`

Вывод на экран

```
<?php
```

- | `echo "Привет мир!";`
- | `print("Привет мир!");`
- | `print "print можно использовать и без скобок.";`
- | `echo 'Эта ', 'строка ', 'была ', 'создана ', 'с несколькими параметрами.';`
- | `echo strftime('%d-%B-%Y, %A');`

```
?>
```

Переменные и типы данных



Переменные

- | Переменные в PHP начинаются со знака доллара (\$)
- | Имя переменной **должно начинаться** с буквы или символа подчеркивания
- | Последующие символы в имени переменной могут быть буквами, цифрами или символом подчеркивания в любом количестве
- | Имя переменной **чувствительно к регистру**

Типы переменных

- | boolean
- | integer
- | float (число с плавающей точкой)
- | string
- | array
- | object
- | resource
- | NULL

Операторы языка



LoftSchool
ОТ МЫСЛИТЕЛЯ К СОЗДАТЕЛЮ

Операторы - арифметические

Пример	Название	Результат
$-\$a$	отрицание	Смена знака $\$a$
$\$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	деление по модулю	Целочисленное деление
$\$a ** \b	возведение в степень	$\$a$ в степени $\$b$

Операторы - строковый

Оператор конкатенации ('.') возвращает объединение левого и правого аргумента

```
$a = "Hello ";
```

```
$b = $a . "World!";
```

```
// $b теперь содержит строку "Hello World!"
```

```
$a = "Hello"; $b = "World!";
```

```
$c = $a . " " . $b;
```

```
// $c теперь содержит строку "Hello World!"
```

```
$c = "$a $b"; // $c тоже "Hello World!"
```

Полный список операторов представлен - <http://bit.ly/1prxMLZ>

Полезные функции - getType

```
echo getType($value);
```

- Возвращает тип переменной
- Возможными значениями возвращаемой строки являются:
 - | "boolean""integer"
 - | "double" (по историческим причинам "double" возвращается в случае типа float, а не просто "float")
 - | "string"
 - | "array"
 - | "object"
 - | "resource"
 - | "NULL"

Полезные функции - setType

```
$foo = "5bar"; // строка
```

```
setType($foo, "integer"); // $foo теперь 5 (целое)
```

Допустимыми значениями типа являются:

- | "boolean" (или "bool")
- | "integer" (или "int")
- | "float" (или "double")
- | "string"
- | "array"
- | "object"
- | "null"

Приведение типов переменных

```
$foo = 10; // $foo это целое число
```

```
$bar = (boolean) $foo; // $bar это булев тип
```

Допускаются следующие приведения типов:

(int), (integer) - приведение к integer

(bool), (boolean) - приведение к boolean

(float), (double), (real) - приведение к float

(string) - приведение к string

(array) - приведение к array

(object) - приведение к object

(unset) - приведение к NULL (PHP 5)

Константы

- Константы можно определить с помощью функции `define()`, а не присваиванием значения `define("CONSTANT", "Здравствуй, мир.");`

`const CONSTANT = "Здравствуй, мир."; // PHP 5.3`

- У констант нет приставки в виде знака доллара (`$`)
- `echo CONSTANT; // выводит "Здравствуй, мир."`
- Константы могут быть определены и доступны в любом месте без учета области видимости
 - Константы не могут быть определены или аннулированы после первоначального объявления
 - Константы могут иметь только скалярные значения
 - Объявляем константу регистро-независимой

`define("CONSTANT", "Здравствуй, мир.", true);`

Управляющие конструкции

if, if-else, else-if, switch-case



Управляющие конструкции: if

Структура if реализована в PHP по аналогии с языком C:

if (условие)
инструкция

Пример

```
1) if ($shop)
    echo "Иду в магазин";
```

```
2) if ($shop) {
    echo "Иду в магазин";
    echo "Покупаю хлеб";
}
```

```
3) if (defined("CONSTANT"))
    echo CONSTANT;
```

Полезные функции

isset(\$var)

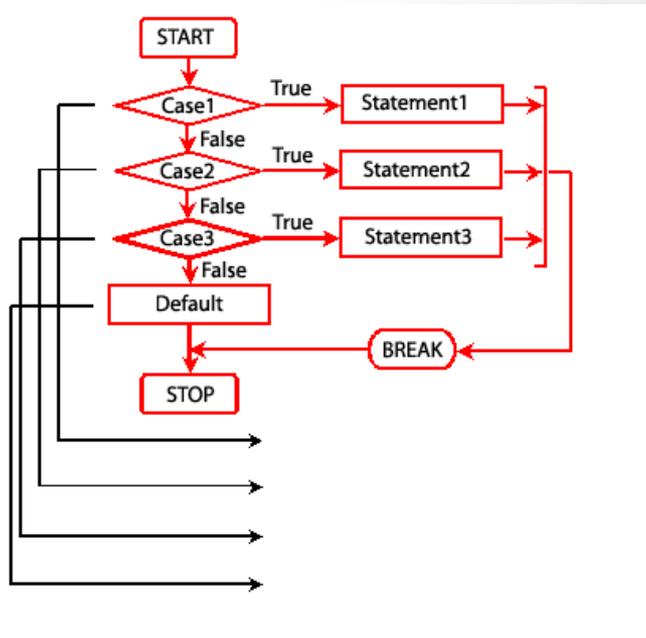
- Устанавливает, определена ли переменная
- Возвращает TRUE, если переменная определена; FALSE в противном случае

empty(\$var)

- Определяет, считается ли переменная пустой
- Возвращает FALSE, если переменна

Управляющие конструкции: switch

```
$i = 1;  
switch ($i) {  
case 0:  
    echo "Результат: 0";  
    break;  
case 1:  
    echo "Результат: 1";  
    break;  
case 2:  
    echo "Результат: 2";  
    break;  
}
```



Работа с массивами

Создание пустого массива

- `$arr = array();`

Создание массива с элементами

- `$arr = array("John","root","1234");`

Обращение к элементу массива

- `echo $arr[1]; // root`

Добавление элементов массива

- `$arr[] = 25;`
- `$arr[] = true;`

Ассоциативные массивы

Ключ может быть либо `integer`, либо `string`

```
$arr = array(  
    "name" => "John",  
    "login" => "root",  
    "pass" => "1234",  
    "age" => 25);
```

Обращение к элементам массива

```
echo $arr["name"]; // John  
echo $arr[0]; // 1
```

Циклы

for, foreach, while, do-while

Операторы инкремента / декремента

“++” - Увеличивает значение переменной на единицу

“--” - Уменьшает значение переменной на единицу

PRE инкремент/декремент

++\$a и --\$a

- Увеличивает или уменьшает \$a на единицу и возвращает значение \$a.

POST инкремент/декремент

\$a++ и \$a--

- Возвращает значение \$a, а затем увеличивает или уменьшает \$a на единицу.

Цикл for

```
for (Часть А; Часть В; Часть С) {  
    // Тело цикла  
}
```

```
for ($i = 1; $i <= 10; $i++) {  
    echo $i;  
}
```

Цикл while

```
while (Условие) {  
    // Тело цикла  
}
```

```
$i = 1;  
while ($i <= 10) {  
    echo $i++;  
}
```

Цикл do-while

```
$i = 100;
```

```
do {
```

```
    echo $i++;
```

```
} while ($i <= 10);
```

Управление циклами: break

```
$i = 1;  
while ($i <= 10) {  
    echo $i++;  
    if($i == 5)  
        break;  
    /* Тело цикла;. */  
}
```

Выведет: 1 2 3 4

Управление циклами: continue

```
$i = 1;  
while ($i <= 10) {  
    echo $i++;  
    if($i == 5)  
        continue;  
    /* Тело цикла;. */  
}
```

Выведет: 1 2 3 4 6 7 8 9 10

Цикл foreach

```
foreach (array as $value) {  
    // Тело цикла  
}
```

```
foreach (array as $key => $value) {  
    // Тело цикла  
}
```

Цикл foreach: примеры

```
$arr = array('a'=>'one','b'=>'two', 'c'=>'three');
```

- `foreach ($arr as $val) { echo "$val
"; }`
- `foreach ($arr as $key => $val) { echo "$key => $val
"; }`
- `foreach ($arr as &$val) { $val = "_$val_"; }`

Что нужно сделать после вебинара?

1. Пересмотреть запись вебинара
2. Пересмотреть видео по [основам PHP](#)
3. Прочитать методичку
4. Продолжить чтение книг из списка [“рекомендованной литературы”](#)
5. Задавать свои вопросы в общем чате
6. Ожидать следующий вебинар

