

# Особености на езика



Максим Крижановски

## ще говорим за

- ✓ особености на скриптовите езици
- ✓ особености на server-side програмиране
- ✓ особености на езика php
- ✗ променливи и цикли
- ✗ синтаксис за деклариране на функции
- ✗ прости оператори
- ✗ неща, които са същите в езици като C/C++
- ✓ неща, които се отличават от C/C++
- ✓ особености на циклите и функциите на php

# php

- ✓ Динамичен
- ✓ Слабо типизиран
- ✓ Сървърен
- ✓ web и не само

# променливи

```
$a = 'b';
```

```
$b = 'a';
```

```
echo $$b; // b
```

```
echo $$$$$$$$$$$$b; // ?!?
```

// това не е КОНТАКТЕНАЦИЯ

```
echo $a, $b, $bb;
```

```
print $a, $b, $bb;
```

Parse error: parse error

# променливи променливи, функции и методи

```
function foo(){  
    echo 'foo';  
}
```

```
function bar(){  
    echo 'bar';  
}
```

```
$f = 'foo';  
$f();
```

# функции и езикови конструкции

```
$f = 'echo';
```

```
$f('Hello world!');
```

Fatal error: Call to undefined function echo()

```
$f = 'printf';
```

```
$f('Hello world!');
```

Hello world

# интерполяция

```
$name = 'Maxim';
```

```
echo "Hello, my name is $name";
```

```
echo <<<NAME
```

```
Hello, my name is $name,
```

```
and this syntax is called HEREDOC
```

```
NAME;
```

# конкатенация и работа с низове

```
$first = 'Maxim';  
$last = 'Krizhanovsky';  
$full = $first . ' ' . $last;  
  
echo strlen($full);  
echo mb_strlen($full);
```



# Референции

```
$a = 'abc';
```

```
$b =& $a;
```

```
$c =& $b;
```

```
$b = 'bcd'; // $a == 'bcd';
```

```
unset($b); // $b == null, $a == 'bcd',
```

```
$c = bcd
```

```
$d = 'abc';
```

```
$c =& $d; // abc = 'bcd', $c = 'abc'
```

# Предаване по референция

```
function hello(&$count = 0) {  
    ++$count;  
}
```

```
$a = 1;  
hello($a);  
hello();
```

# функции

```
function double(){  
    global $x;  
    $x = $x * 2;  
}  
$x = 2;  
double();
```

# функции

```
function sum(){  
    $sum = 0;  
    foreach (func_get_args() as $value)  
    {  
        $sum += $value;  
    }  
    return $sum;  
}
```

```
$sum = sum(1,2,3,4); // 10
```

# функции

✗ В php няма предефиниране на функции

✓ функциите се дефинират runtime

```
if ($x) {  
    function foo(){}  
}
```

✓ всички функции имат глобален scope

✓ за да върнете стойност по референция:

```
function &get_num(){}  
$a =& get_num();
```

# функции

✓ можете да използвате `function_exists()` или `is_callable()`

✓ php поддържа анонимни функции

```
$message = function(){  
    echo 'Hello world!';  
};  
if (is_callable($message)){  
    $message();  
}
```

# Сравнение

- ✓ `0 == ""; // true`
- ✓ `0 == false; // true`
- ✓ `1 == "1"; // true`
- ✓ `1 == "1abc"; // true`
- ✗ `"1" == "1abc"; // false`
- ✗ `1 === "1"; // false`
- ✓ `"00000" == false; // true`
- ✓ `empty($var); // true if $var is 0, "", false or null`

## is\_\*

- ✓ is\_bool(\$a)
- ✓ is\_int(\$a)
- ✓ is\_numeric(\$a)
- ✓ is\_null(\$a)
- ✓ is\_array(\$a)
- ✓ и други...



# МАСИВИ

```
$numeric = array(1,2,3,4,5);  
$colors = array('red' => '#ff0000',  
                'green' => '#00ff00', 'blue' =>  
                '#0000ff');  
$numeric[] = 6;  
$numeric['thousand'] = 1000;  
$colors['black'] = '#000000';
```



```
$new = [1,2,3,4];
```

# функции за работа с масиви

- ✓ над 75 вградени функции за работа с масиви!
- ✓ стойностите могат да са от всякакъв тип
- ✓ могат да се сместват типове стойности
- ✓ добавяне и премахване на елементи без ограничение
- ✓ добавяне в началото на масива с `array_unshift`

`array_pop`, `array_push`, `array_shift`, `array_unshift`

# най-често използвани функции за масиви

- ✓ `sort, asort, usort, uasort, ksort, uksort`
- ✓ `in_array, array_search, array_key_exists`
- ✓ `array_unique, array_keys, array_values`
- ✓ `array_map, array_filter`
- ✓ `array_merge`
- ✓ `count`

## \$array1 + array2

```
$data = array(1 => 1,2,3);  
$more = array(4 => 4,5,6);  
$all = $data + $more; // [1 =>  
1,2,3,4,5,6]
```

```
$data = array(1,2,3);  
$more = array(4,5,6);  
$all = $data + $more; // [1,2,3]  
$all = array_merge($data + $more); //  
[1,2,3,4,5,6]
```

# foreach

```
foreach ($arr as $value) {  
    echo $value;  
}
```

```
foreach ($arr as $key => $value) {  
    $arr[$key] = $value * 2;  
}
```

```
foreach ($arr as &$amp;value) {  
    $value *= 2;  
}
```

# list

```
list(,$month,$day) = explode('-',  
'2013-03-20');
```

```
list($a,$b,$c) = array(1,2,3);
```

```
extract(array('a' => 1, 'b' => 2, 'c'  
=> 3))
```

```
function foo($args) {  
    extract($args);  
}
```

# example

```
function foo($size = 'm', $color = 'red',  
amount = 1) {  
  
}
```

PHP:

```
foo('m', 'red', 2);
```

Python:

```
foo(amount = 2)
```

# example

```
function foo($properties = array()) {  
    $size = 'm';  
    $color = 'red';  
    $amount = 1;  
    extract($properties);  
}
```

```
function foo($properties = array()) {  
    $defaults = array('size' => 'm',  
                      'color' =>  
'red',  
                      'amount' => 1);  
    $data = array_merge($defaults, $properties)  
}
```



# break / continue

```
foreach ($array as $subarray) {  
    foreach ($subarray as $value) {  
        if (is_null($value)) {  
            break 2;  
        }  
    }  
}
```

# HTML

- ✓ php се вгражда в HTML (или обратното)
- ✓ Всичко, което не е в `<?php ?>` тагове се output-ва необработено
- ✓ php предлага алтернативен синтаксис на операторите за по-удобна работа в HTML темплейт

if:            switch:            for:            while:

endif;    endswitch;    endfor;    endwhile;

# Includes

- ✓ include, include\_once
- ✓ require, require\_once
- ✓ eval
- ✓ \_\_DIR\_\_
- ✓ include връща стойност
- ✗ не я използвайте

# Еднакви и не съвсем

- ✓ `die()` и `exit()` са еднакви
- ✗ `print` и `echo` не са
- ✓ `sizeof()` и `count()` са еднакви
- ✗ `and` и `&&`, `or` и `||` не са

# Debugging

- ✓ `print_r`, `var_dump`
- ✗ Не используйте `@`
- ✓ `error_reporting`
- ✓ `display_errors`

# Някои забавни неща

```
$a = 'string';  
echo ++$a; // strinh
```

```
$array = array(true, true);  
var_dump(in_array('test', $array)); //  
true
```

```
${'a' . 'b'} = 'c';  
echo $ab; // c
```

## Някои забавни неща

```
$country = 'Bulgaria';  
$city = 'Sofia';  
$data = compact('country', 'city');
```

е същото като

```
$data = array(  
    'country' => $country,  
    'city' => $city  
);
```

# Contacts

Maxim Krizhanovsky

darhazer@gmail.com

<http://www.linkedin.com/in/darhazer>

PHP Bulgaria @ Facebook

<https://www.facebook.com/groups/459347284112503/>

php.dev.bg