



PHP Functions

kurz Webové technológie
Eduard Kuric

Funkcie

```
function myFunc()  
{  
    echo 'Ahoj Zem';  
}  
  
myFunc(); // Ahoj Zem
```

Funkcie /2

- funkcia môže byť zavolaná ešte predtým, ako je definovaná

```
foo(); // ok
```

```
function foo() {}
```

- toto ale nie je ok:

```
myFunc();
```

```
// je definovaná za podmienky
```

```
if (true) { function bar() {} }
```

```
myFunc(); // ok
```

Parametre

```
function myFunc($x, $y)
{
    echo $x . $y;
}
myFunc('Ahoj', 'Svet'); // Ahoj Svet
```

Predvolené hodnoty parametrov

```
function myFunc($x, $y = 'Zem')  
{  
    echo $x . $y;  
}  
myFunc('Ahoj'); // Ahoj Zem
```

Premenlivý zoznam parametrov

- funkciu nemožno zavolať s menším počtom argumentov, ako je počet parametrov, ale s väčším počtom áno

```
function myArgs()  
{  
    $x = func_get_arg(0);  
    $y = func_get_arg(1);  
    $z = func_get_arg(2);  
    echo $x . $y . $z;  
}  
myArgs('Fiit', 'ká', 'ri');  
// Fiitkári
```

Premenlivý zoznam parametrov /2

```
function myArgs2 ()
{
    $num = func_num_args ();
    $args = func_get_args ();
    for ($i = 0; $i < $num; $i++)
        echo $args[$i];
}

myArgs2 ('Fiit', 'ká', 'ri');
// Fiitkári
```

Premenlivý zoznam parametrov /3

- dd PHP 5.6

```
function myArgs3 (...$args)
{
    foreach($args as $v) {
        echo $v;
    }
}
```

```
myArgs3(1, 2, 3); // 123
```

```
$a = [1, 2, 3];
```

```
myArgs3(...$a); // 123
```


Príkaz return

```
function myFunc()  
{  
    return; // exit  
    echo 'Ahoj'; // nikdy sa nevykoná  
}
```

```
function myFunc()  
{  
    return 'Ahoj';  
}  
echo myFunc(); // Ahoj
```

Príkaz return /2

- funkcia bez `return` automaticky vracia `null`

```
function myFuncNull() {}
```

```
if (myFuncNull() === null)  
    echo 'true'; // true
```

Platnosť premenných

```
$x = 'Svet'; // globálna premenná
```

```
function myFunc()  
{  
    // tu $x neexistuje  
    $y = 'Svet'; // lokálna premenná  
}
```

Platnosť premenných /2

```
$x = 'Ahoj'; // globálna $x
function myFunc()
{
    global $x; // použij globálnu $x
    $x .= 'Svet'; // zmen globálnu $x
}

myFunc();
echo $x; // Ahoj Svet
```

Platnosť premenných /3

```
function myFunc()  
{  
    // zmen globálnu $x  
    $GLOBALS['x'] .= 'Svet';  
}
```

- **v PHP v riadiacich štruktúrach platí globálna platnosť**

```
if(true)  
{  
    $x = 10; // toto je globálna  
}  
echo $x; // 10
```

Anonymné funkcie

- od PHP 5.3

```
$povedzNieco = function($name)
{
    echo "Ahoj" . $name;
};
$povedzNieco("Svet"); // Ahoj Svet
```

Anonymné funkcie /2

- zvyčajne sa používajú ako callback funkcie

```
function myCaller($myCallback)  
{  
    echo $myCallback ();  
}  
  
// "Ahoj"  
myCaller( function() { echo "Ahoj"; } );
```

Closure

- anonymná funkcia je schopná zachytiť kontext - hodnoty premenných
- `use` „zachytí“ hodnoty počas definovania funkcie

```
$x = 1;  
$y = 2;  
$myClosure = function($z) use ($x, $y)  
{  
    return $x + $y + $z;  
};  
echo $myClosure(3); // 6
```


Closure /2

```
$global_variable = 1;
$closure = function() use ($global_variable)
{
    return $global_variable;
};
$closure2 = function() {
    global $global_variable;
    return $global_variable;
};
$global_variable = 99;
echo $closure();      // 1
echo $closure2();     // 99
```

Generátory série hodnôt

```
function getNum()  
{  
    for ($i = 0; $i < 5; $i++) {  
        yield $i;  
    }  
}  
  
foreach (getNum() as $v)  
    echo $v; // 01234
```

Generátory série hodnôt /2

```
function countToFive()  
{  
    yield 1;  
    // za from moze byt iny generator,  
    // iterator, alebo pole  
    yield from [2, 3, 4];  
    yield 5;  
}  
  
foreach (countToFive() as $v)  
    echo $v; // 12345
```