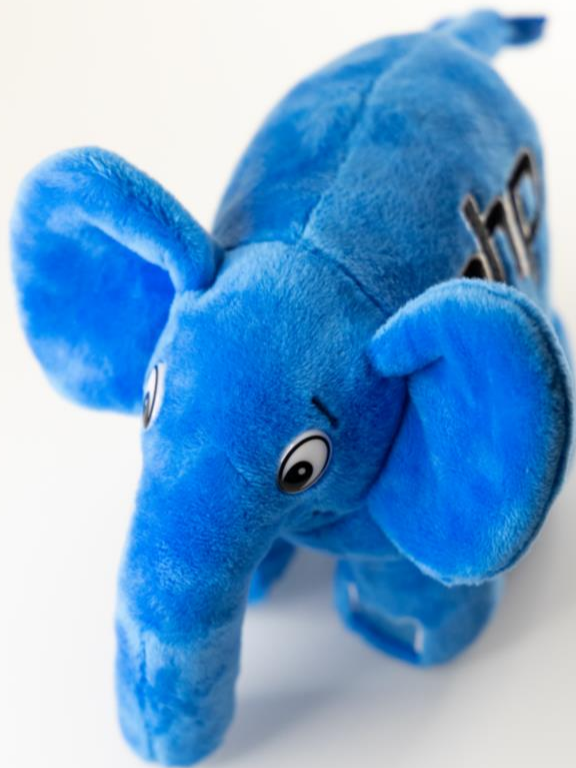


Useful Functions and Expressions

Disclaimer: вы смотрите просто запись лекции, это HE специально подготовленный видеокурс!



Disclaimer

Most of these functions and expressions will be reviewed in more details later.

We also assume that you are familiar with **if ... else** expression 😊.

Output almost anything: echo

Echo is NOT a function, do NOT use parentheses!

```
<?php

$someVariable = 5;
$someAnotherVariable = 10;
$someYetAnotherVariable = 'The SUM result is ';

echo $someVariable . "\n";
echo $someAnotherVariable . "\n";
echo $someYetAnotherVariable . ($someVariable + $someAnotherVariable) . "\n";

/*
5
10
The SUM result is 15
*/
```

Output literally anything (for debug!): `var_dump()` and `print_r()`

These functions are for debug ONLY! Never make a “real code” with them!

```
<?php

$someArray = ['name' => 'John', 'surname' => 'Smith'];

var_dump($someArray); /* array(2) {
    'name' =>
    string(4) "John"
    'surname' =>
    string(5) "Smith"
} */

print_r($someArray); /* Array
(
    [name] => John
    [surname] => Smith
)
*/
```

Check if a variable or an array element exists: isset

Most likely you'll use this function with arrays:

```
<?php

$someVar = 10;
$someAnotherVar = 20;
$someArray[12] = 5;

if (isset($someVar)) {
    echo 'OK'; // OK
}

if (isset($someVar, $someAnotherVar)) {
    echo 'OK'; // OK
}

if (isset($someArray[12])) {
    echo 'OK'; // OK
}

if (isset($someNonExistingVar)) {
    echo 'OK'; // nothing
}

if (isset($someArray[27])) {
    echo 'OK'; // nothing
}
```

Delete a variable or an array element: unset

And again – most likely you'll use this function with arrays:

```
<?php

$someVar = 10;
$someAnotherVar = 20;
$someArray[12] = 5;

// Simple: we are deleting a variable.
unset($someVar);

// We may easily delete even non-existing variable.
unset($someAnotherVar, $someNonExistingVar);

// This is how we delete an array element.
unset($someArray[12]);

// We may easily delete even non-existing array element.
unset($someArray[27]);
```

Terminate a script execution: exit

This is extremely useful in CLI scripts!

```
<?php

if ($argc < 3) {
    exit('USAGE: php scan.php SOURCE_DIR DESTINATION_DIR');
}
```

Access CLI parameters: \$argc and \$argv

This is extremely useful in CLI scripts!

```
<?php

if ( $argc > 0 ) {
    foreach ( $argv as $argumentNumber => $argumentValue ) {
        echo $argumentNumber . ' = ' . $argumentValue . "\n";
    }
}

// php 06_argc_and_argv.php One Two "Some more"

/*
0 = 06_argc_and_argv.php
1 = One
2 = Two
3 = Some more
*/
```


Iterate through an array (and change values): foreach

This expression works with one dimension only! For multidimensional cases use recursion.

```
<?php

$someArray = ['name' => 'John', 'surname' => 'Smith', 'age' => 50];

foreach ($someArray as $key => $value) {
    if (is_string($value)) {
        echo '"' . $value . "\"\n";
    } elseif (is_int($value)) {
        echo '[' . $value . "]\n";
        $someArray[$key] = $value + 1;
    }
}

print_r($someArray);
```

```
/*
"John"
"Smith"
[50]

Array
(
    [name] => John
    [surname] => Smith
    [age] => 51
)
*/
```

Combine a script from several files: include (and variations)

These are NOT functions! Do NOT use parentheses!

```
<?php
function someCoolFunction(): void
{
}

```

library.php

```
<?php
include 'library.php';
setcookie("Test", 999, time() + 3600);
echo "Hi!";

```

index.php

WARNING on failure

include

include_once

ERROR on failure

require

require_once

...**_once** ignore
multiple attempts to
use the same file

Concatenate strings: .

If this was not yet obvious... 😊

```
<?php
```

```
$resultLabel = 'The result of SUM is ';
```

```
$operandOne = 2;
```

```
$operandTwo = 5;
```

```
echo $resultLabel . ($operandOne + $operandTwo);
```

```
// The result of SUM is 7
```

See also...

<https://front-line-php.com/cheat-sheet>

<https://overapi.com/php>

<https://www.codewithharry.com/blogpost/php-cheatsheet>

Useful Functions and Expressions

Disclaimer: вы смотрите просто запись лекции, это HE специально подготовленный видеокурс!

