

*php*

# Μια φορά κι έναν καιρό...

| Έτος | Έκδοση  |      |             |
|------|---|------|-------------|
| 1994 | Ο Rasmus Lerdorf γράφει σε C μερικά CGI tracking scripts με το όνομα Personal Home Page Tools |      |             |
| 1995 | Δημοσιεύεται ο κώδικας για τα PHP Tools   |      |             |
| 1996 | PHP/FI 2.0  |      |             |
| 1998 | PHP 3.0 (PHP: Hypertext Preprocessor)   |      |             |
| 1999 | Zend Engine (Andi Gutmans and Zeev Suraski)   |      |             |
| 2000 | PHP 4.0   |      |             |
| 2004 | PHP 5.0 – Zend Engine 2.0   |      |             |
| 2005 | PHP 5.1   | Έτος | Έκδοση      |
| 2006 | PHP 5.2   | 2014 | PHP 5.6     |
| 2009 | PHP 5.3   | 2015 | PHP 7.0     |
| 2012 | PHP 5.4   | 2016 | PHP 7.1     |
| 2013 | PHP 5.5   | 2017 | PHP 7.2 RC4 |

# PHP. Τι είναι;

Η PHP είναι μια scripting γλώσσα προγραμματισμού που ενσωματώνεται σε HTML έγγραφα και εκτελείται στην πλευρά του web server. Ως γλώσσα προγραμματισμού υποστηρίζει τόσο διαδικαστικό προγραμματισμό όσο και αντικειμενοστρεφή.

Η σύνταξη της PHP έχει δανειστεί στοιχεία από τις C, Java και Perl.

Σκοπός της PHP είναι να δώσει τη δυνατότητα στους web developers να γράψουν δυναμικά παραγώμενες ιστοσελίδες εύκολα και γρήγορα.

# Πού γράφουμε κώδικα;

Ο κώδικας της PHP γράφεται σε αρχεία με κατάληξη .php τα οποία μπορεί να περιέχουν είτε μόνο πηγαίο κώδικα σε PHP ή κώδικα σε PHP αναμεμειγμένο με HTML.

# Τι κώδικα γράφουμε;

```
<!DOCTYPE html>
<html>
<head>
<title>My first PHP page</title>
</head>
<body>
<?php echo 'Hello world!'; ?>
</body>
</html>
```

PSR-2: Coding Style Guide - <http://www.php-fig.org/psr/psr-2/>

# Πώς γράφουμε σωστά;

```
<?php
namespace Vendor\Package;

use FooInterface;
use BarClass as Bar;
use OtherVendor\OtherPackage\BazClass;

class Foo extends Bar implements FooInterface
{
    public function sampleMethod($a, $b = null)
    {
        if ($a === $b) {
            bar();
        } elseif ($a > $b) {
            $foo->bar($arg1);
        } else {
            BazClass::bar($arg2, $arg3);
        }
    }
}
```

# Μεταβλητές

- Οι μεταβλητές στην PHP ξεκινάνε πάντα με \$ και ακολουθεί το όνομα της μεταβλητής
  - Το όνομα της μεταβλητής πρέπει να ξεκινάει με γράμμα ή με «\_» (underscore)
  - Το όνομα της μεταβλητής δεν πρέπει να ξεκινάει με ψηφίο
  - Το όνομα της μεταβλητής μπορεί να περιέχει μόνο τους χαρακτήρες: A-Z a-z 0-9 \_
  - Τα πεζά γράμματα διαφέρουν από τα κεφαλαία στα ονόματα των μεταβλητών
  - Το όνομα της μεταβλητής δεν πρέπει να είναι το \$this
- <?php

```
$a = 5;  
$b = 10;  
$c = $a + $b;
```

?>

# Εμβέλεια μεταβλητών

Οι τύποι εμβέλειας μεταβλητών που διαθέτει η PHP είναι οι εξής:

- καθολική (global)
- τοπική (local)
- στατική (static)



# Καθολικές μεταβλητές

Καθολικές λέγονται οι μεταβλητές που χρησιμοποιούνται έξω από συναρτήσεις (functions).

```
<?php
```

```
$a = 5;
```

```
function myFunc()  
{  
    global $a;  
    $b = 10;  
    return $a + $b;  
}
```

```
?>
```

# Τοπικές μεταβλητές

Οι τοπικές μεταβλητές χρησιμοποιούνται εντός των συναρτήσεων.

```
<?php
```

```
function addAllThree($a, $b = 5)
{
    $c = 10;
    return $a + $b + $c;
}
```

```
addAllThree(-5);
```

```
?>
```

# ΣΤΑΤΙΚΕΣ-ΤΟΠΙΚΕΣ μεταβλητές

Οι στατικές-τοπικές μεταβλητές χρησιμοποιούνται εντός των συναρτήσεων και διατηρούν την τιμή τους μεταξύ των κλήσεων μιας συνάρτησης.

```
<?php
```

```
function incByTwo($a)
{
    static $executions = 0;
    $executions += 1;
    return $a + 2;
}
```

```
incByTwo(10);
incByTwo(100);
```

```
?>
```

# Εμφάνιση δεδομένων

Η εμφάνιση δεδομένων γίνεται με τις εντολές echo και print.

```
<?php
```

```
echo "1 + 1 = ";  
echo 2;  
echo PHP_EOL;
```

```
?>
```

# Τύποι δεδομένων

Τα καλά νέα είναι ότι στην PHP δε χρειάζεται να ορίσουμε τον τύπο των δεδομένων που περιέχονται σε μια μεταβλητή. Η PHP επιλέγει τον τύπο αυτομάτως.

Οι τύποι δεδομένων που υποστηρίζονται είναι οι εξής:

- String - `<?php $a = "abc"; ?>`
- Integer - `<?php $a = 15; ?>`
- Float - `<?php $a = 15.5;?>`
- Boolean - `<?php $a = true; ?>`
- Array - `<?php $a = array(); ?>`
- Object - `<?php $a = new DbQuery(); ?>`
- NULL - `<?php $a = null; ?>`
- Resource - `<?php $a = mysqli_connect("localhost", "user", "pass", "db"); ?>`

# Σταθερές

Οι σταθερές είναι πρακτικά η αναφορά ενός ονόματος σε μια σταθερή τιμή.

Τα ονόματα των σταθερών **δεν** πρέπει να ξεκινάνε με \$ και μπορούν να περιέχουν χαρακτήρες και το γράμμα \_

Οι σταθερές είναι αυτομάτως καθολικής εμβέλειας.

```
<?php
```

```
define("PI", 3.14159265359);
```

```
?>
```

# Αριθμητικοί Τελεστές

| Τελεστής | Πράξη              | Παράδειγμα   |
|----------|--------------------|--------------|
| -        | Αφαίρεση           | $\$x - \$y$  |
| +        | Πρόσθεση           | $\$x + \$y$  |
| *        | Γινόμενο           | $\$x * \$y$  |
| /        | Πηλίκιο διαίρεσης  | $\$x / \$y$  |
| %        | Υπόλοιπο διαίρεσης | $\$x \% \$y$ |
| **       | Ύψωση σε δύναμη    | $\$x ** \$y$ |

# Τελεστές Εκχώρησης

| Τελεστής | Πράξη              | Παράδειγμα |
|----------|--------------------|------------|
| =        | Εκχώρηση τιμής     | \$x = 15   |
| +=       | Πρόσθεση           | \$x += 2   |
| -=       | Αφαίρεση           | \$x -= 5   |
| *=       | Γινόμενο           | \$x *= 2   |
| /=       | Πηλίκιο διαίρεσης  | \$x /= 2   |
| %=       | Υπόλοιπο διαίρεσης | \$x %= 2   |



# Τελεστές Σύγκρισης

| Τελεστής | Πράξη                       | Παράδειγμα    |
|----------|-----------------------------|---------------|
| ==       | Ίσο ως προς τιμή            | $\$x == \$y$  |
| ===      | Ίσο ως προς τιμή και τύπο   | $\$x === \$y$ |
| !=       | Διάφορο ως προς τιμή        | $\$x != \$y$  |
| <>       | Διάφορο ως προς τιμή        | $\$x <> \$y$  |
| !==      | Διάφορο ως προς τιμή ή τύπο | $\$x !== \$y$ |
| <        | Μικρότερο                   | $\$x < \$y$   |
| >        | Μεγαλύτερο                  | $\$x > \$y$   |
| <=       | Μικρότερο ή ίσο             | $\$x <= \$y$  |
| >=       | Μεγαλύτερο ή ίσο            | $\$x >= \$y$  |

# Σύγκριση δεδομένων με διαφορετικό τύπο

<http://php.net/manual/en/types.comparisons.php>

# Τελεστές αύξησης/μείωσης

| Τελεστής | Εντολές                           | Παράδειγμα |
|----------|-----------------------------------|------------|
| ++       | Αύξηση κατά 1 και επιστροφή τιμής | ++\$x      |
|          | Επιστροφή τιμής και αύξηση κατά 1 | \$x++      |
| --       | Μείωση κατά 1 και επιστροφή τιμής | --\$x      |
|          | Επιστροφή τιμής και μείωση κατά 1 | \$x--      |

# Τελεστές κατά bit

| Τελεστής | Πράξη             | Παράδειγμα       |
|----------|-------------------|------------------|
| &        | AND               | $\$x \& \$y$     |
|          | OR                | $\$x   \$y$      |
| ^        | XOR               | $\$x \wedge \$y$ |
| ~        | NOT               | $\sim \$x$       |
| >>       | Δεξιά ολίσθηση    | $\$x >> \$y$     |
| <<       | Αριστερή ολίσθηση | $\$x << \$y$     |

# Λογικοί τελεστές

| Τελεστής | Πράξη | Παράδειγμα  |
|----------|-------|-------------|
| and      | AND   | \$x and \$y |
| or       | OR    | \$x or \$y  |
| xor      | XOR   | \$x xor \$y |
| &&       | AND   | \$x && \$y  |
|          | OR    | \$x    \$y  |
| !        | NOT   | !\$x        |

# Τελεστές συμβολοσειρών

| Τελεστής | Πράξη                | Παράδειγμα   |
|----------|----------------------|--------------|
| .        | Συνένωση             | $\$x . \$y$  |
| .=       | Συνένωση με εκχώρηση | $\$x .= \$y$ |

# Τελεστές Πινάκων

| Τελεστής | Πράξη  | Παράδειγμα    |
|----------|--|---------------|
| +        | Ένωση  | $\$x + \$y$   |
| ==       | Ισότητα κλειδιών/τιμών                           | $\$x == \$y$  |
| ===      | Ισότητα κλειδιών/τιμών<br>με ίδια σειρά και τύπο | $\$x === \$y$ |
| !=       | Διάφορο  | $\$x != \$y$  |
| <>       | Διάφορο  | $\$x <> \$y$  |
| !==      | Διάφορο κατά τιμή ή<br>τύπο                      | $\$x !== \$y$ |

# Προτεραιότητα τελεστών

<http://php.net/manual/en/language.operators.precedence.php>



# Δομή ελέγχου ροής if...elseif...else

```
<?php
```

```
if (συνθήκη_1) {  
} elseif (συνθήκη_2) {  
} else {  
}  
}
```

```
?>
```

```
<?php
```

```
if ($x > 0) {  
    $a = "positive";  
} elseif ($x < 0) {  
    $a = "negative";  
} else {  
    $a = "zero";  
}  
}
```

```
?>
```

# Δομή ελέγχου ροής switch

```
<?php
```

```
switch (μεταβλητή) {  
    case τιμή1:
```

```
        break;
```

```
    case τιμή2:
```

```
        break;
```

```
    default:
```

```
}
```

```
?>
```

```
<?php
```

```
switch ($x) {
```

```
    case 1:
```

```
        $a = "one";
```

```
        break;
```

```
    case 2:
```

```
        $a = "two";
```

```
        break;
```

```
    default:
```

```
        $a = "not one or two";
```

```
}
```

```
?>
```

# Δομή επανάληψης while

```
<?php
```

```
while (συνθήκη) {
```

```
}
```

```
?>
```

```
<?php
```

```
while ($x > 0) {
```

```
    $a *= 2;
```

```
    $x -= 1;
```

```
}
```

```
?>
```

# Δομή επανάληψης do...while

```
<?php
```

```
do {
```

```
} while (συνθήκη);
```

```
?>
```

```
<?php
```

```
do {
```

```
    $a *= 2;
```

```
    $x += 1;
```

```
} while ($x < 0);
```

```
?>
```

# Δομή επανάληψης for

```
<?php
```

```
for (αρχικοποίηση; συνθήκη; τελευταία_εντολή) {  
}
```

```
?>
```

```
<?php
```

```
for ($i = 0; $i < 10; $i++) {  
    $a *= 2;  
}
```

```
?>
```

# Δομή προσπέλασης foreach

```
<?php
```

```
foreach (πίνακας as κλειδί => τιμή) {
```

```
}
```

```
?>
```

```
<?php
```

```
foreach ($my_array as $k => $v) {
```

```
    $my_array[$k] = $v + 2;
```

```
}
```

```
?>
```

# Συναρτήσεις (functions)

Τα ονόματα των συναρτήσεων στην PHP πρέπει να ξεκινάνε από γράμμα ή το χαρακτήρα «\_» (underscore).

Για να δηλώσουμε μια νέα συνάρτηση το όνομά της δεν πρέπει να έχει χρησιμοποιηθεί ήδη είτε από τις ενσωματωμένες συναρτήσεις (built-in) ή από δική μας συνάρτηση. Επομένως κάθε συνάρτηση δηλώνεται μια φορά μόνο.

```
<?php
function myFunc($a, $b, $c = 10, $d = 20)
{
    $a += $b * $c * $d;
    return $a;
}
?>
```

# Πίνακες

```
<?php
```

```
$my_array = array(1, 2, 3);
```

```
$my_array_2 = array(  
    "one",  
    "two",  
    "three",  
);
```

```
$my_array_3 = array(  
    "name" => "George",  
    "surname" => "Filippakis",  
    "married" => true,  
);
```

```
?>
```

```
<?php
```

```
echo $my_array[0];
```

```
echo $my_array_1[0];  
echo $my_array_2[1];  
echo $my_array_3[2];
```

```
echo $my_array_3['name'];  
echo $my_array_3['surname'];
```

```
?>
```



# Χρήσιμες συναρτήσεις

- Μαθηματικές συναρτήσεις:  
<http://php.net/manual/en/ref.math.php>
- Συναρτήσεις χειρισμού συμβολοσειρών:  
<http://php.net/manual/en/ref.strings.php>
- Συναρτήσεις χειρισμού πινάκων:  
<http://php.net/manual/en/ref.array.php>
- Συναρτήσεις χειρισμού βάσης δεδομένων MySQL:  
<http://php.net/manual/en/mysqli.summary.php>
- Άλλες συναρτήσεις:  
<http://php.net/manual/en/funcref.php>

# Superglobals

- `$GLOBALS` – Πίνακας που περιέχει όλες τις μεταβλητές καθολικής εμβέλειας
- `$_POST` – Πίνακας που περιέχει όλα τα δεδομένα που έχουν σταλεί με τη μέθοδο POST
- `$_GET` – Πίνακας που περιέχει όλα τα δεδομένα που έχουν σταλεί με τη μέθοδο GET
- `$_COOKIE` – Πίνακας που περιέχει όλα τα δεδομένα που έχουν σταλεί με χρήση cookies
- `$_FILES` – Πίνακας που περιέχει όλα τα αρχεία που έχουν σταλεί από μια φόρμα

# Έλεγχος εγκυρότητας δεδομένων

<http://php.net/manual/en/book.filter.php>

Παραδείγματα ελέγχου εγκυρότητας (έλεγχος email κ.ά.):

<http://php.net/manual/en/filter.examples.validation.php>



# Σύνδεση σε βάση δεδομένων MySQL

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = mysqli_connect($servername, $username, $password);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
echo "Connected successfully";
?>
```

# CRUD

## Create Read Update Delete

- INSERT – Εισαγωγή εγγραφών (δεδομένων) στη βάση δεδομένων  
<https://dev.mysql.com/doc/refman/5.7/en/insert.html>
- SELECT – Ανάκτηση εγγραφών από τη βάση δεδομένων  
<https://dev.mysql.com/doc/refman/5.7/en/select.html>
- UPDATE – Ενημέρωση εγγραφών στη βάση δεδομένων  
<https://dev.mysql.com/doc/refman/5.7/en/update.html>
- DELETE – Διαγραφή εγγραφών από τη βάση δεδομένων  
<https://dev.mysql.com/doc/refman/5.7/en/delete.html>

# INSERT

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

# SELECT

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // output data of each row
    while($row = mysqli_fetch_assoc($result)) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}

mysqli_close($conn);
?>
```



# UPDATE

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";

if (mysqli_query($conn, $sql)) {
    echo "Record updated successfully";
} else {
    echo "Error updating record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```

# DELETE

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=3";

if (mysqli_query($conn, $sql)) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . mysqli_error($conn);
}

mysqli_close($conn);
?>
```