



PHP TUTORIAL#1

Καπετανάκη Λένα

MSc Πανεπιστήμιο Πειραιά

site: <https://lenakapetanaki.gr>

YouTube: youtube.com/c/lenakapetanaki

ΙΣΤΟΡΙΚΑ

- Η PHP, τα αρχικά της οποίας προέρχονται από τις λέξεις PHP **Hypertext Preprocessor**, είναι μια server-side scripting γλώσσα η οποία ενσωματώνεται σε HTML έγγραφα και εκτελείται κάθε φορά που οι χρήστες επισκέπτονται τα έγγραφα αυτά.
- Η PHP αναπτύχθηκε το 1994 από τον Rasmus Lerdorf προκειμένου να εντοπίζει τους επισκέπτες του δικτυακού του τόπου και την επόμενη χρονιά κυκλοφόρησε ως Personal Home Page (πρώιμη ονομασία της PHP) Tools.
- Ο αναλυτής (parser) επανασχεδιάστηκε από τους Zeev Suraski και Andi Gutmans όπως επίσης και η σύνταξη της γλώσσας ώστε να προσεγγίζει την σύνταξη αντικειμενοστραφών γλωσσών προγραμματισμού .



ΕΙΣΑΓΩΓΙΚΑ

- Η PHP είναι μια γλώσσα server scripting και ένα ισχυρό εργαλείο για τη δημιουργία δυναμικών και διαδραστικών ιστοσελίδων.
- Η PHP είναι μια ευρέως χρησιμοποιούμενη και ανοιχτού κώδικα γλώσσα προγραμματισμού
- Η PHP είναι δωρεάν για λήψη και χρήση
- Η PHP 7 είναι η τελευταία έκδοση.

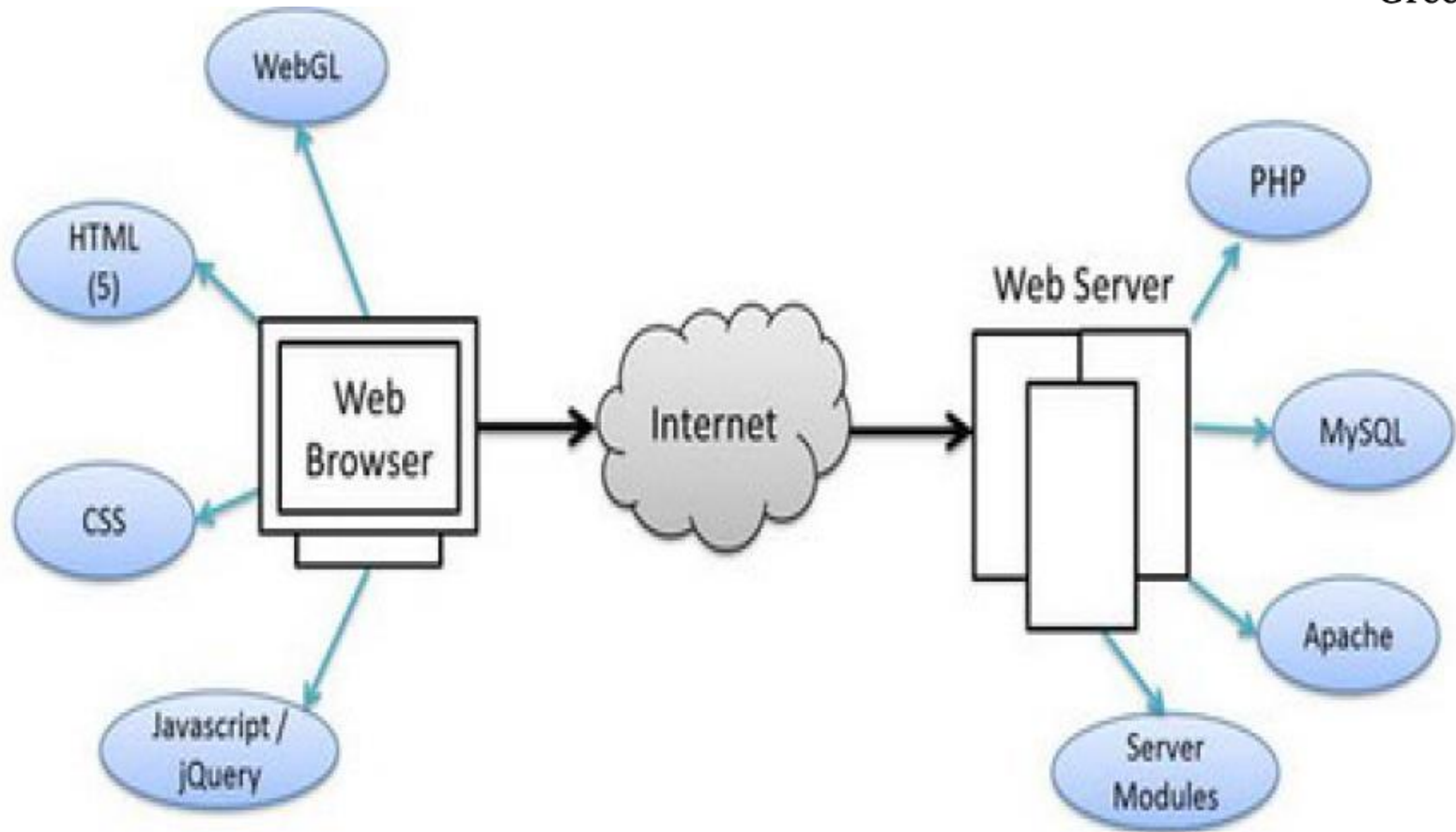
ΑΡΧΕΙΑ PHP

- Τα αρχεία PHP μπορούν να περιέχουν κείμενο, HTML, CSS, JavaScript και κώδικα PHP
- Ο κωδικας PHP εκτελείται στον διακομιστή και το αποτέλεσμα επιστρέφεται στο πρόγραμμα περιήγησης ως απλή HTML
- Τα αρχεία PHP έχουν επέκταση ".php"

ΤΙ ΚΑΝΕΙ Η PHP

- Η PHP μπορεί να δημιουργήσει δυναμικό περιεχόμενο σελίδας
- Η PHP μπορεί να δημιουργήσει, να ανοίξει, να διαβάσει, να γράψει, να διαγράψει και να κλείσει αρχεία στο διακομιστή
- Η PHP μπορεί να συλλέξει δεδομένα φόρμας
- Η PHP μπορεί να στείλει και να λάβει cookie
- Η PHP μπορεί να προσθέσει, να διαγράψει, να τροποποιήσει δεδομένα στη βάση δεδομένων σας
- Η PHP μπορεί να χρησιμοποιηθεί για τον έλεγχο της πρόσβασης των χρηστών Η PHP μπορεί να κρυπτογραφήσει δεδομένα

- Η PHP εκτελείται σε διάφορες πλατφόρμες (Windows, Linux, Unix, Mac OS X κ.λπ.)
- Η PHP είναι συμβατή με σχεδόν όλους τους διακομιστές που χρησιμοποιούνται σήμερα (Apache, IIS κ.λπ.)
- Η PHP υποστηρίζει ένα ευρύ φάσμα βάσεων δεδομένων
- Η PHP είναι εύκολη στην εκμάθηση και λειτουργεί αποτελεσματικά από την πλευρά του διακομιστή



ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ (1)

◉ Επεξεργαστής Κειμένου.

- Η PHP μπορεί να γραφτεί σε οποιονδήποτε απλό text editor αρκεί να αποθηκευτεί σαν αρχείο με κατάληξη .php.
- Εμείς θα χρησιμοποιήσουμε το Notepad++ (free download <https://notepad-plus-plus.org/>)
- Άλλοι text editors
 - ◉ Visual Studio Code
 - ◉ Sublime (<https://www.sublimetext.com/>)
 - ◉ Brackets (<http://brackets.io>)



ΠΡΟΑΠΑΙΤΟΥΜΕΝΑ (2)

○ **Server – Local Server**

- Τα αρχεία .php δεν μπορούν να τρέξουν κατευθείαν σε κάποιον browser αλλά μόνο μέσα από κάποιον server.
- Εμείς θα χρησιμοποιήσουμε το XAMPP
(<https://www.apachefriends.org/index.html>)
- Άλλοι servers
 - WAMP server
 - MAMP server



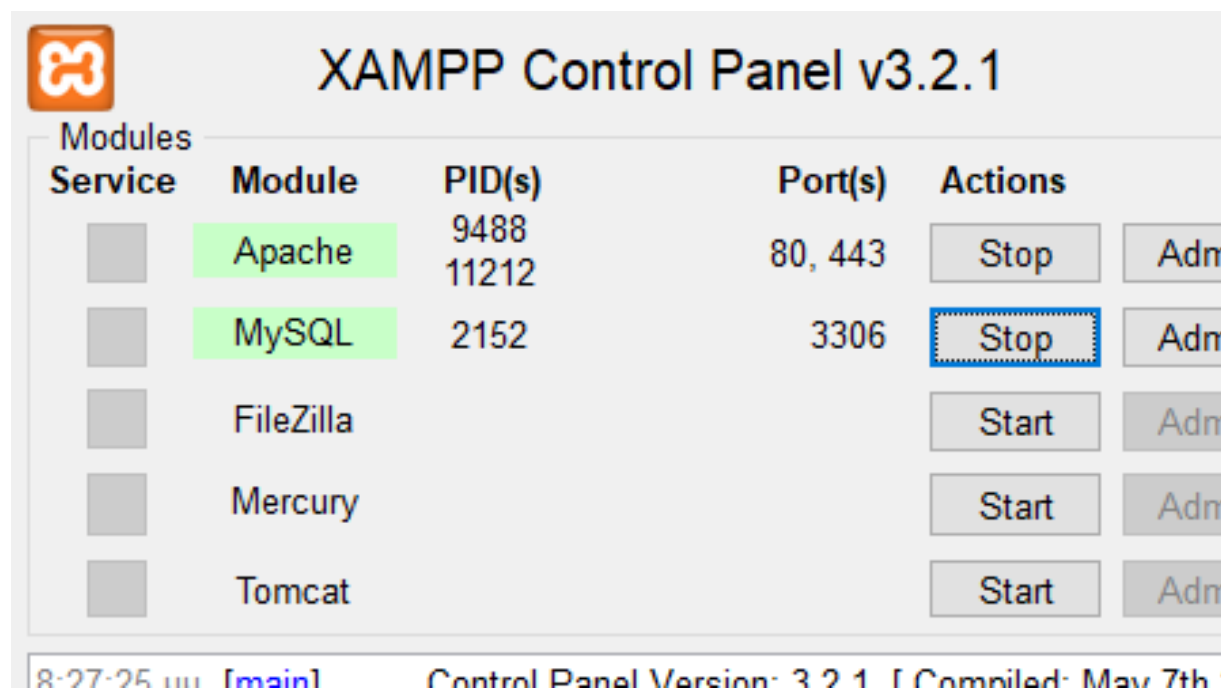
XAMPP (1)

- Το xampp είναι ένα δωρεάν πρόγραμμα με το οποίο μπορούμε να κάνουμε εγκατάσταση τον Apache, τη MySQL και την PHP.
- Το πρόγραμμα εγκαθίσταται αυτόματα στην διαδρομή C:\XAMPP.



XAMPP (2)

- Η εκτέλεση του προγράμματος εμφανίζει το πιο κάτω παράθυρο.
- Πατάμε start στον Apache και MySQL για να ξεκινήσει ο server.



XAMPP (3)

- Για να δούμε αν έχει εγκατασταθεί θα πρέπει να ανοίξουμε ένα πρόγραμμα πλοήγησης (browser) και να δώσουμε στην γραμμή διευθύνσεων <http://localhost> (ή

ένα



XAMPP for Windows

XAMPP
1.8.3
[PHP: 5.5.15]

Welcome
Status
Security
Documentation
Components
Applications

Php
phpinfo()
CD Collection
Biorhythm
Instant Art
Phone Book

Perl
perlinfo()
Guest Book

Welcome to XAMPP for Windows!

Congratulations:
You have successfully installed XAMPP on this system!

Now you can start using Apache and Co. You should first try »Status« on the

For OpenSSL support please use the test certificate with <https://127.0.0.1> or

Good luck, Kay Vogelgesang + Kai 'Oswald' Seidler

Install applications on XAMPP using BitNami

Apache Friends and BitNami are cooperating to make dozens of open source deployed with one-click installers. Visit the [BitNami XAMPP page](#) for details o



XAMPP (4)

- Το xampp αποθηκεύει τις ιστοσελίδες στον φάκελο C:\XAMPP\htdocs.
- Προκειμένου να ανοίξουμε μια ιστοσελίδα .php πρέπει λοιπόν να την αποθηκεύσουμε σε αυτόν τον φάκελο.
 - Πχ. Έστω ότι θέλουμε να τρέξουμε την σελίδα site1.php
 - Την αποθηκεύουμε στον φάκελο htdocs.
 - Στην γραμμή διεύθυνσης κάποιου browser πληκτρολογούμε <http://localhost/site1.php>



ΒΑΣΙΚΗ ΣΥΝΤΑΞΗ PHP

- Ένα PHP script μπορεί να τοποθετηθεί οπουδήποτε στο έγγραφο.
- Ένα PHP script ξεκινά με **<? Php** και τελειώνει με **?>**

```
<?php
// PHP code goes here
?>
```

- Η προεπιλεγμένη επέκταση αρχείου για αρχεία PHP είναι ".php".
- Ένα αρχείο PHP περιέχει συνήθως ετικέτες HTML και κάποιο κώδικα δέσμης ενεργειών PHP.

ΣΧΟΛΙΑ

- Τα σχόλια μπορούν να χρησιμοποιηθούν για:
 - Για να επιτρέψεις σε άλλους να καταλάβουν τον κωδικό σου
 - Για να υπενθυμίσεις στον εαυτό σου τι εκανες

```
<?php
// This is a single-line comment
```

```
# This is also a single-line comment
?>
```

```
<?php
/*
This is a multiple-lines comment block
that spans over multiple
lines
*/
```

ECHO ΚΑΙ PRINT

- Με την PHP, υπάρχουν δύο βασικοί τρόποι εξόδου (output): **echo** and **print**.
- Είναι λίγο πολύ τα ίδια. Και τα δύο χρησιμοποιούνται για την έξοδο δεδομένων στην οθόνη.
- Με το echo μπορείς να ενώσεις πολλές συμβολοσειρές ενώ με το Print μόνο μία.

```
<?php
echo "<h2>PHP is Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters.";
?>
```


PHP ΜΕΤΑΒΛΗΤΕΣ

- Στην PHP, μια μεταβλητή ξεκινά με το σύμβολο \$, ακολουθούμενο από το όνομα της μεταβλητής

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

- Μετά την εκτέλεση των παραπάνω δηλώσεων, η μεταβλητή **\$txt** θα διατηρήσει την τιμή Hello world !, η μεταβλητή **\$x** θα διατηρήσει την τιμή 5 και η μεταβλητή **\$y** θα διατηρήσει την τιμή 10.5

ΚΑΝΟΝΕΣ ΜΕΤΑΒΛΗΤΩΝ

- Μια μεταβλητή μπορεί να έχει ένα σύντομο όνομα (όπως x και y) ή ένα πιο περιγραφικό όνομα (ηλικία, carname, total_volume).
- Μια μεταβλητή ξεκινά με το σύμβολο \$, ακολουθούμενο από το όνομα της μεταβλητής
- Ένα όνομα μεταβλητής πρέπει να ξεκινά με ένα γράμμα ή τον χαρακτήρα υπογράμμισης
- Ένα όνομα μεταβλητής δεν μπορεί να ξεκινά με έναν αριθμό
- Ένα όνομα μεταβλητής μπορεί να περιέχει μόνο αλφαριθμητικούς χαρακτήρες και κάτω παύλες (A-z, 0-9 και _)
- Τα ονόματα των μεταβλητών είναι πεζών-κεφαλαίων (το \$ age και το \$ AGE είναι δύο διαφορετικές μεταβλητές)

- Οι σταθερές είναι σαν τις μεταβλητές αλλά μόλις οριστούν, δεν μπορούν να αλλάξουν ξανά τιμή
- **`define(name, value, case-insensitive)`**
 - `name`: Καθορίζει το όνομα της σταθεράς
 - `value`: Καθορίζει την τιμή της σταθεράς
 - `case-insensitive`: Καθορίζει εάν το όνομα της σταθεράς θα πρέπει να είναι χωρίς διάκριση πεζών-κεφαλαίων. Η προεπιλογή είναι ψευδής

```
<?php
define("GREETING", "Welcome to W3Schools.com!");
echo GREETING;
?>
```

PHP DATA TYPES

- Οι μεταβλητές μπορούν να αποθηκεύουν δεδομένα διαφορετικών τύπων
 - String
 - Integer
 - Float (floating point numbers - also called double)
 - Boolean
 - Array
 - Object
 - NULL
 - Resource

PHP STRING

- Ένα string είναι μια ακολουθία χαρακτήρων, όπως το "Hello world!"

```
<?php
$x = "Hello world!";
$y = 'Hello world!';

echo $x;
echo "<br>";
echo $y;
?>
```

PHP INTEGER

- Ένας ακέραιος τύπος δεδομένων είναι ένας μη δεκαδικός αριθμός μεταξύ -2.147.483.648 και 2.147.483.647
 - Ένας ακέραιος δεν πρέπει να έχει δεκαδικό σημείο
 - Ένας ακέραιος μπορεί να είναι είτε θετικός είτε αρνητικός

```
<?php
$x = 5985;
var_dump($x);
?>
```

PHP FLOAT

- Είναι ένας αριθμός με δεκαδικό σημείο ή αριθμός σε εκθετική μορφή
- Στο παρακάτω παράδειγμα, το \$ x είναι float. Η συνάρτηση PHP var_dump () επιστρέφει τον τύπο και την τιμή δεδομένων

```
<?php
$x = 10.365;
var_dump($x);
?>
```

PHP BOOLEAN

- Αντιπροσωπεύει δύο πιθανές καταστάσεις: TRUE ή FALSE

```
$x = true;  
$y = false;
```


PHP ARRAY

- Ένας πίνακας αποθηκεύει πολλές τιμές σε μία μόνο μεταβλητή.
- Στο παρακάτω παράδειγμα το \$ cars είναι ένας πίνακας

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
var_dump($cars);
?>
```

PHP STRING FUNCTIONS

- **strlen ()** - Επιστρέφει το μήκος μιας συμβολοσειράς

```
<?php
echo strlen("Hello world!"); // outputs 12
?>
```

- **str_word_count ()** - Καταμέτρηση λέξεων σε συμβολοσειρά
- **strrev ()** – Αντιστρέφει μια συμβολοσειρά
- **strpos ()** - Αναζήτηση κειμένου σε μια συμβολοσειρά

```
<?php
echo strpos("Hello world!", "world"); // outputs 6
?>
```

- Το NaN σημαίνει Not a Number.
- Το NaN χρησιμοποιείται για αδύνατες μαθηματικές πράξεις.
- Η PHP έχει τις ακόλουθες λειτουργίες για να ελέγξει εάν μια τιμή δεν είναι αριθμός: `is_nan()`
- Ωστόσο, η συνάρτηση PHP `var_dump()` επιστρέφει τον τύπο και την τιμή δεδομένων

PHP CASTING STRINGS AND FLOATS TO INTEGERS

- Μερικές φορές χρειάζεται να κάνουμε cast μια αριθμητική τιμή σε έναν άλλο τύπο δεδομένων
- Η συνάρτηση (int), (integer) ή intval() χρησιμοποιείται συχνά για τη μετατροπή μιας τιμής σε ακέραιο.

```
<?php
// Cast float to int
$x = 23465.768;
$int_cast = (int)$x;
echo $int_cast;

echo "<br>";

// Cast string to int
$x = "23465.768";
$int_cast = (int)$x;
echo $int_cast;
?>
```

PHP ARITHMETIC OPERATORS

Operator	Name	Example	Result	Show it
+	Addition	$\$x + \y	Sum of $\$x$ and $\$y$	Try it »
-	Subtraction	$\$x - \y	Difference of $\$x$ and $\$y$	Try it »
*	Multiplication	$\$x * \y	Product of $\$x$ and $\$y$	Try it »
/	Division	$\$x / \y	Quotient of $\$x$ and $\$y$	Try it »
%	Modulus	$\$x \% \y	Remainder of $\$x$ divided by $\$y$	Try it »
**	Exponentiation	$\$x ** \y	Result of raising $\$x$ to the $\$y$ 'th power	Try it »

PHP ASSIGNMENT OPERATORS

Assignment	Same as...	Description	Show it
<code>x = y</code>	<code>x = y</code>	The left operand gets set to the value of the expression on the right	Try it »
<code>x += y</code>	<code>x = x + y</code>	Addition	Try it »
<code>x -= y</code>	<code>x = x - y</code>	Subtraction	Try it »
<code>x *= y</code>	<code>x = x * y</code>	Multiplication	Try it »
<code>x /= y</code>	<code>x = x / y</code>	Division	Try it »

PHP COMPARISON OPERATORS

Operator	Name	Example	Result
==	Equal	<code>\$x == \$y</code>	Returns true if <code>\$x</code> is equal to <code>\$y</code>
===	Identical	<code>\$x === \$y</code>	Returns true if <code>\$x</code> is equal to <code>\$y</code> , and they are of the same type
!=	Not equal	<code>\$x != \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code>
<>	Not equal	<code>\$x <> \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code>
!==	Not identical	<code>\$x !== \$y</code>	Returns true if <code>\$x</code> is not equal to <code>\$y</code> , or they are not of the same type
>	Greater than	<code>\$x > \$y</code>	Returns true if <code>\$x</code> is greater than <code>\$y</code>
<	Less than	<code>\$x < \$y</code>	Returns true if <code>\$x</code> is less than <code>\$y</code>
>=	Greater than or equal to	<code>\$x >= \$y</code>	Returns true if <code>\$x</code> is greater than or equal to <code>\$y</code>
<=	Less than or equal to	<code>\$x <= \$y</code>	Returns true if <code>\$x</code> is less than or equal to <code>\$y</code>
<=>	Spaceship	<code>\$x <=> \$y</code>	Returns an integer less than, equal to, or greater than zero, depending on if <code>\$x</code> is less than, equal to, or greater than <code>\$y</code> . Introduced in PHP 7.

PHP IF...ELSE...ELSEIF STATEMENTS

- Η if εκτελεί κάποιο κωδικό εάν μια συνθήκη είναι αληθής

```
if (condition) {
    code to be executed if condition is true;
}
```

```
if (condition) {
    code to be executed if this condition is true;
} elseif (condition) {
    code to be executed if first condition is false and this condition is true;
} else {
    code to be executed if all conditions are false;
}
```


SWITCH

- Επιλογή ενός από τα πολλά μπλοκ κώδικα που θα εκτελεστεί

```
<?php
$favcolor = "red";

switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    case "green":
        echo "Your favorite color is green!";
        break;
    default:
        echo "Your favorite color is neither red, blue, nor green!";
}
?>
```

PHP LOOPS

- Συχνά, όταν γράφουμε κώδικα, θέλουμε το ίδιο μπλοκ κώδικα να εκτελείται ξανά και ξανά για έναν ορισμένο αριθμό επαναλήψεων. Έτσι, αντί να προσθέσουμε αρκετές σχεδόν ίσες γραμμές κώδικα σε έναν κώδικα, μπορούμε να χρησιμοποιήσουμε βρόχους
- Οι βρόχοι χρησιμοποιούνται για την εκτέλεση του ίδιου μπλοκ κώδικα ξανά και ξανά, αρκεί μια συγκεκριμένη συνθήκη να ισχύει.
- Στην PHP, έχουμε τους ακόλουθους τύπους βρόχων:
 - while – επανάληψη ενός μπλοκ κώδικα για όσο ισχύει μια καθορισμένη συνθήκη
 - do... while- εκτελεί ένα μπλοκ κώδικα μία φορά και μετά επαναλαμβάνει τον βρόχο όσο ισχύει μια καθορισμένη συνθήκη
 - for- εκτέλεση ενός μπλοκ κώδικα για καθορισμένο αριθμό επαναλήψεων
 - foreach – εκτέλεση ενός μπλοκ κώδικα για κάθε στοιχείο σε έναν πίνακα

THE PHP WHILE LOOP

```
while (condition is true) {
    code to be executed;
}
```

```
<?php
$x = 1;

while($x <= 5) {
    echo "The number is: $x <br>";
    $x++;
}
?>
```

PHP DO WHILE LOOP

```
do {
    code to be executed;
} while (condition is true);
```

```
<?php
```

```
$x = 1;
```

```
do {
```

```
    echo "The number is: $x <br>";
```

```
    $x++;
```

```
} while ($x <= 5);
```

```
?>
```

THE PHP FOR LOOP

```
for (init counter; test counter; increment counter) {  
    code to be executed for each iteration;  
}
```

- init counter: αρχικοποίηση της τιμής του μετρητή
- test counter: συνθήκη που αν ισχύει συνεχίζει ο βρόχος αλλιώς σταματάει
- increment counter: Αυξάνει την τιμή μετρητή του βρόχου

```
<?php  
for ($x = 0; $x <= 10; $x++) {  
    echo "The number is: $x <br>";  
}  
?>
```

THE PHP FOREACH LOOP

- Δουλεύει μόνο σε πίνακες

```
foreach ($array as $value) {
    code to be executed;
}
```

```
<?php
$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {
    echo "$value <br>";
}
?>
```

BREAK - CONTINUE

- Ο βρόχος σταματά όταν το x είναι ίσο με 4

```
<?php
for ($x = 0; $x < 10; $x++) {
    if ($x == 4) {
        break;
    }
    echo "The number is: $x <br>";
}
?>
```

PHP FUNCTIONS

- Οι συναρτήσεις της php μπορούν να καλούνται μόνο μέσω php.
- Οι συναρτήσεις στην php ορίζονται με την λέξη κλειδί function.
- Μπορούν να δέχονται παραμέτρους (μπορούν να είναι και κενές)
- Μπορούν να επιστρέφουν τιμές ή απλά να εκτελούνται



PHP FUNCTIONS

```
function functionName() {  
    code to be executed;  
}
```

```
<?php  
function writeMsg() {  
    echo "Hello world!";  
}
```

```
writeMsg(); // call the function  
?>
```

FUNCTIONS EXAMPLES

- Η `gettype($var)` μας επιστρέφει τον τύπο της μεταβλητής που βρίσκεται στην παρένθεση:

```
function say_hello() {
    echo "HelloWorld!<br/>";
}
say_hello();
```

```
function say_hello2($word) {
    echo "Hello{$word}!<br/>";
}
say_hello2("World");
```



- Ένας πίνακας είναι μια ειδική μεταβλητή, η οποία μπορεί να αποθηκεύει και να διατηρεί περισσότερες από μία τιμές τη φορά.
- Στην PHP, υπάρχουν τρεις τύποι πινάκων
 - Indexed arrays: Πίνακες με αριθμητικό ευρετήριο
 - Associative arrays
 - Multidimensional arrays: Πίνακες που περιέχουν έναν ή περισσότερους πίνακες

PHP INDEXED ARRAYS

○ Δυο τρόποι δημιουργίας

- `$cars = array("Volvo", "BMW", "Toyota");`
ή
- `$cars[0] = "Volvo";`
`$cars[1] = "BMW";`
`$cars[2] = "Toyota";`

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
?>
```

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo count($cars);
?>
```

- Εμφάνιση των στοιχείων του πίνακα

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
$arlength = count($cars);

for($x = 0; $x < $arlength; $x++) {
    echo $cars[$x];
    echo "<br>";
}
?>
```

PHP ASSOCIATIVE ARRAYS

```
<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
echo "Peter is " . $age['Peter'] . " years old.";
?>
```

```
<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```

PHP SORTING ARRAYS

- `sort()` - sort arrays in ascending order
- `rsort()` - sort arrays in descending order
- `asort()` - sort associative arrays in ascending order, according to the value
- `ksort()` - sort associative arrays in ascending order, according to the key
- `arsort()` - sort associative arrays in descending order, according to the value
- `krsort()` - sort associative arrays in descending order, according to the key

PHP GLOBAL VARIABLES - SUPERGLOBALS

- \$GLOBALS
- \$_SERVER
- \$_REQUEST
- \$_POST
- \$_GET
- \$_FILES
- \$_ENV
- \$_COOKIE
- \$_SESSION

ISSET()

- Με την isset() ελέγχουμε αν μια μεταβλητή έχει πάρει τιμή ή όχι
- Επιστρέφει true/false
- Είναι καλός έλεγχος για να δούμε αν έχουμε πάρει δεδομένα από κάποια εξωτερική πηγή ή όχι
- Επιστρέφει true ή false

```
<?php
```

```
$var1 = 13;  
$var2 = "hello";  
$var3 = NULL;
```

```
echo  isset($var1);  
echo  isset($var2);  
echo  isset($var3);
```

```
?>
```

GETTYPE()

- Η `gettype($var)` μας επιστρέφει τον τύπο της μεταβλητής που βρίσκεται στην παρένθεση:

```
$var1 = "3 pink dogs";  
$var2 = 13;
```

```
echo gettype($var1);  
echo "<br/>";  
echo gettype($var2);  
echo "<br/>";
```



INCLUDE - REQUIRE

- Υπάρχουν 2 τρόποι να εισάγουμε ένα php στην html μας:
 - include: Η include παίρνει όλο το κώδικα του αρχείου php και το μεταφέρει (αντιγράφει) στο αρχείο που το καλεί. Αν υπάρξει πρόβλημα τότε θα το αναφέρει αλλά το script θα συνεχίσει να εκτελείται
 - Require: με την require γίνεται η ίδια λειτουργία αλλά αν υπάρχει error τότε δεν αφήνει να εκτελεστεί τίποτε!

```
<?php include header.php' ;?>
```

```
<?php require header.php' ;?>
```

