



ΜΟΝΑΔΕΣ ΑΡΙΣΤΕΙΑΣ
ΑΝΟΙΧΤΟΥ ΛΟΓΙΣΜΙΚΟΥ

1^ο ΣΧΟΛΕΙΟ ΚΩΔΙΚΑ

*«Βασικά Θέματα Προγραμματισμού
στην Ανάπτυξη Δυναμικών Διαδικτυακών Εφαρμογών»
(Part 4 - PHP)*

Γιάννης Σαμωνάκης



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Υπουργείο Παιδείας και Θρησκευμάτων,
Πολιτισμού και Αθλητισμού



Περιεχόμενα

- Τι είναι η PHP
- Που μπορεί να χρησιμοποιηθεί η PHP
- Τι χρειάζεται για την ανάπτυξη μιας εφαρμογής
- Οι τύποι δεδομένων, οι τελεστές, οι μεταβλητές και η εμφάνισή τους στην PHP
- Οι συναρτήσεις στην PHP
- Οι εντολές διακλάδωσης και επανάληψης της PHP

Τι είναι η PHP

- Server-side scripting γλώσσα προγραμματισμού
- Μπορεί επίσης να χρησιμοποιηθεί και ως γλώσσα προγραμματισμού γενικού σκοπού

Που μπορεί να χρησιμοποιηθεί

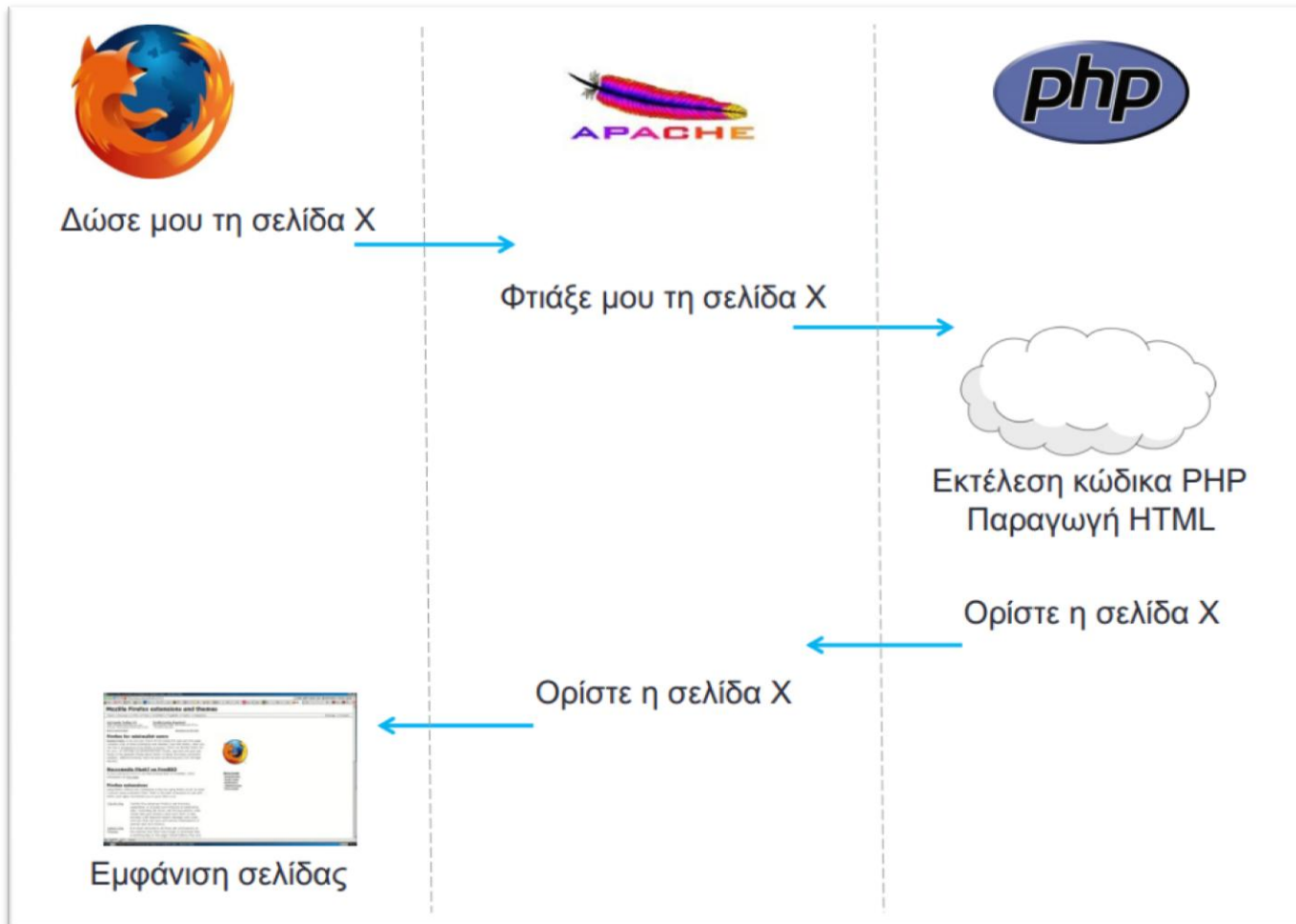
- Στην ανάπτυξη διαδικτυακών εφαρμογών (server-side scripting)
- Στην ανάπτυξη εφαρμογών που τρέχουν από τη γραμμή εντολών. Κατάλληλο για την εκτέλεση επαναλαμβανόμενων εργασιών περιβάλλον Linux και Windows
- Στην ανάπτυξη desktop εφαρμογών, με τη βοήθεια σχετικών επεκτάσεων (PHP-GTK)

Τι θα χρειαστούμε

Για την ανάπτυξη μιας διαδικτυακής εφαρμογής θα πρέπει:

- Να είναι εγκατεστημένη η PHP στο μηχάνημα που θα τρέχει η εφαρμογή
- Να είναι εγκατεστημένος ένας web server (λχ τον Apache HTTP server), στο μηχάνημα που θα τρέχει η εφαρμογή
- Να έχουμε πρόσβαση σε μια βάση δεδομένων, στην περίπτωση που θέλουμε να αναπτύξουμε δυναμικές διαδικτυακές εφαρμογές
- Να έχουμε στη διάθεσή μας ένα φυλλομετρητή ιστού

Ένα τυπικό σενάριο



Ένα απλό πρόγραμμα σε PHP

```
<!DOCTYPE html>  
<html>  
<body>  
  
<h1>My first PHP page</h1>  
  
<?php  
    echo "Hello World!";  
?>  
  
</body>  
</html>
```

Μεταβλητές στην PHP

```
<!DOCTYPE html>  
<html>  
<body>
```

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

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

```
</body>  
</html>
```

Τα ονόματα των μεταβλητών:

- Πρέπει να ξεκινούν με το \$
- Μετά το \$ θα πρέπει να ακολουθεί είτε γράμμα του λατινικού αλφαβήτου είτε ο χαρακτήρας «_»
- Μπορούν να περιέχουν μόνο λατινικά γράμματα (A-z), αριθμούς (0-9) και το χαρακτήρα «_»
- Είναι case-sensitive, δηλ. το \$age και το \$AGE είναι δύο διαφορετικές μεταβλητές

Τύποι δεδομένων στην PHP

- Οι μεταβλητές μπορούν να αποθηκεύσουν δεδομένα διαφορετικών τύπων.
- Ανάλογα με τον τύπο των μεταβλητών, καθορίζονται και οι ενέργειες που μπορούμε να εκτελέσουμε στις μεταβλητές αυτές
- String
- Integer
- Float (λέγεται και double)
- Boolean
- Array
- Object
- NULL
- Resource

Ο τύπος δεδομένων String

- Είναι μια ακολουθία από χαρακτήρες.
- Η ακολουθία αυτή θα πρέπει να περικλείεται από μονά (' ') ή διπλά (" ") εισαγωγικά

```
<!DOCTYPE html>  
<html>  
<body>
```

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

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

```
</body>  
</html>
```

Ο τύπος δεδομένων Integer

- Δεν επιτρέπεται να έχει δεκαδικό μέρος
- Μπορεί να είναι θετικός ή αρνητικός

```
<!DOCTYPE html>  
<html>  
<body>
```

```
<?php  
$x = 5;  
echo "The number is:" . $x;  
?>  
</body>  
</html>
```

Ο τύπος δεδομένων Float

- Έχει δεκαδικό μέρος
- Μπορεί να είναι θετικός ή αρνητικός

```
<!DOCTYPE html>  
<html>  
<body>
```

```
<?php  
$x = 5.25;  
echo "The number is:" . $x;  
>
```

```
</body>  
</html>
```

Ο τύπος δεδομένων Array

- Αποθηκεύει πολλές τιμές κάτω από το ίδιο όνομα
- Η προσπέλαση των τιμών πραγματοποιείται με τη βοήθεια ενός αριθμητικού δείκτη
- Ο δείκτης ξεκινάει πάντα από το μηδέν (0)

```
<!DOCTYPE html>  
<html>  
<body>
```

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

```
</body>  
</html>
```

Τελεστές

- Χρησιμοποιούνται για την εκτέλεση πράξεων στις μεταβλητές
- Η PHP υποστηρίζει τα ακόλουθα είδη τελεστών
 - Αριθμητικοί τελεστές
 - Τελεστές ανάθεσης
 - Τελεστές σύγκρισης
 - Τελεστές αυξομείωσης
 - Λογικοί τελεστές
 - Τελεστές συμβολοσειρών (string operators)
 - Τελεστές πάνω σε arrays

Αριθμητικοί Τελεστές

Operator	Name	Example	Result
+	Addition	$\$x + \y	Sum of $\$x$ and $\$y$
-	Subtraction	$\$x - \y	Difference of $\$x$ and $\$y$
*	Multiplication	$\$x * \y	Product of $\$x$ and $\$y$
/	Division	$\$x / \y	Quotient of $\$x$ and $\$y$
%	Modulus	$\$x \% \y	Remainder of $\$x$ divided by $\$y$
**	Exponentiation	$\$x ** \y	Result of raising $\$x$ to the $\$y$ 'th power (Introduced in PHP 5.6)

Τελεστές Ανάθεσης

Assignment	Same as...	Description
$x = y$	$x = y$	The left operand gets set to the value of the expression on the right
$x += y$	$x = x + y$	Addition
$x -= y$	$x = x - y$	Subtraction
$x *= y$	$x = x * y$	Multiplication
$x /= y$	$x = x / y$	Division
$x \% = y$	$x = x \% y$	Modulus

Τελεστές Σύγκρισης

Operator	Name	Example	Result
==	Equal	$x == y$	Returns true if x is equal to y
===	Identical	$x === y$	Returns true if x is equal to y , and they are of the same type
!=	Not equal	$x != y$	Returns true if x is not equal to y
<>	Not equal	$x <> y$	Returns true if x is not equal to y
!==	Not identical	$x !== y$	Returns true if x is not equal to y , or they are not of the same type
>	Greater than	$x > y$	Returns true if x is greater than y
<	Less than	$x < y$	Returns true if x is less than y
>=	Greater than or equal to	$x >= y$	Returns true if x is greater than or equal to y
<=	Less than or equal to	$x <= y$	Returns true if x is less than or equal to y

Τελεστές Αυξομείωσης

Operator	Name	Description
++\$x	Pre-increment	Increments \$x by one, then returns \$x
\$x++	Post-increment	Returns \$x, then increments \$x by one
--\$x	Pre-decrement	Decrements \$x by one, then returns \$x
\$x--	Post-decrement	Returns \$x, then decrements \$x by one

Συναρτήσεις (1)

- Μια συνάρτηση είναι ένα σύνολο από εντολές οι οποίες εκτελούν μια συγκεκριμένη εργασία και μπορούν να επαναχρησιμοποιηθούν.
- Οι συναρτήσεις δεν εκτελούνται από μόνες τους. Θα πρέπει εμείς να τις καλέσουμε, χρησιμοποιώντας το όνομα που της έχουμε δώσει.
- Το όνομα μιας συνάρτησης μπορεί να ξεκινάει με λατινικό γράμμα ή το χαρακτήρα “_” ενώ δεν υπάρχει διάκριση ανάμεσα στα πεζά και τα κεφαλαία

Συναρτήσεις (2)

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<?php
```

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

```
    writeMsg();
```

```
?>
```

```
</body>
```

```
</html>
```

Συναρτήσεις (3)

- Στα μαθηματικά: $y=f(x)$
- Σε μια συνάρτηση είναι δυνατό να διοχετεύσουμε την πληροφορία που επιθυμούμε, με τη βοήθεια μεταβλητών που ονομάζονται παράμετροι.
- Η πληροφορία αυτή, αξιοποιείται από τη συνάρτηση που συνήθως παράγει ένα αποτέλεσμα
- Μπορούμε να περάσουμε όσες παραμέτρους θέλουμε, χωρισμένους με κόμμα.

Συναρτήσεις (4)

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
function sum($x, $y) {  
    $z = $x + $y;  
    return $z;  
}
```

```
echo "5 + 10 = " . sum(5,10) . "<br>";  
echo "7 + 13 = " . sum(7,13) . "<br>";  
echo "2 + 4 = " . sum(2,4);
```

```
?>
```

```
</body>
```

```
</html>
```

Εμβέλεια μεταβλητών (1)

- Η εμβέλεια έχει να κάνει με το εύρος ισχύος μιας μεταβλητής. Δηλαδή σε ποια έκταση του κώδικά μας μπορούμε να την προσπελάσουμε και επομένως να τη χρησιμοποιήσουμε.
- Η PHP έχει τρία διαφορετικά variable scopes:
 - **Local**: μια μεταβλητή που έχει δηλωθεί μέσα σε μια συνάρτηση, μπορεί να χρησιμοποιηθεί μόνο μέσα στη συνάρτηση αυτή
 - **Global**: μια μεταβλητή που έχει δηλωθεί έξω από μια συνάρτηση, δεν μπορεί να χρησιμοποιηθεί μέσα σε συνάρτηση.
 - **Static**: όταν ολοκληρώνει την εκτέλεσή της μια συνάρτηση οι μεταβλητές της διαγράφονται. Αν δεν επιθυμούμε κάτι τέτοιο μπορούμε να τις δηλώσουμε ως “static”

Εμβέλεια μεταβλητών (2)

```
<!DOCTYPE html>  
<html>  
<body>
```

```
<?php
```

```
    $x = 5; // global scope
```

```
    function myTest() {  
        // using x inside this function will generate an error  
        echo "<p>Variable x inside function is: $x</p>";  
    }  
    myTest();
```

```
    echo "<p>Variable x outside function is: $x</p>";
```

```
?>
```

```
</body>  
</html>
```


Εμβέλεια μεταβλητών (3)

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
function myTest() {  
    $x = 5; // local scope  
    echo "<p>Variable x inside function is: $x</p>";  
}  
myTest();
```

```
// using x outside the function will generate an error  
echo "<p>Variable x outside function is: $x</p>";
```

```
?>
```

```
</body>
```

```
</html>
```

Εμβέλεια μεταβλητών (4)

Αν θέλουμε μέσα σε μια συνάρτηση να χρησιμοποιήσουμε μεταβλητές με global scope, χρησιμοποιούμε τη λέξη κλειδί “global”

```
<!DOCTYPE html>
<html>
<body>
<?php
    $x = 5;
    $y = 10;

    function myTest() {
        global $x, $y;
        $y = $x + $y;
    }

    myTest(); // run function
    echo $y; // output the new value for variable $y
?>
</body>
</html>
```

Εντολές διακλάδωσης (1)

- Συχνά, όταν προγραμματίζουμε, χρειάζεται να εκτελέσουμε διαφορετικές ενέργειες, με βάση διαφορετικές συνθήκες.
- Για το σκοπό αυτό η PHP προσφέρει τις ακόλουθες εντολές:
 - **if statement:** αν η συνθήκη είναι αληθής εκτελούνται κάποιες εντολές
 - **if...else statement:** όταν η συνθήκη είναι αληθής εκτελούνται κάποιες εντολές, αλλιώς εκτελούνται κάποιες άλλες.
 - **if...elseif...else statement:** ορίζει μια επιπλέον συνθήκη για να εξεταστεί, αν η πρώτη είναι ψευδής.
 - **switch statement:** Επιλογή ενός από πολλά μπλοκ εντολών για εκτέλεση

Εντολές διακλάδωσης (2)

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
    $t = date("H");
```

```
    if ($t < 10) {
```

```
        echo "Have a good morning!";
```

```
    } elseif ($t < 20) {
```

```
        echo "Have a good day!";
```

```
    } else {
```

```
        echo "Have a good night!";
```

```
    }
```

```
?>
```

```
</body>
```

```
</html>
```

Εντολές επανάληψης (1)

- Συχνά, όταν προγραμματίζουμε, θέλουμε το ίδιο τμήμα κώδικα να εκτελεστεί περισσότερες από φορές.
- Για το σκοπό αυτό η PHP προσφέρει τις ακόλουθες εντολές επανάληψης:
 - **while**: εκτελείται ένα μπλοκ εντολών όσο μια συνθήκη είναι αληθής
 - **do...while**: ίδια με την προηγούμενη μόνο που πρώτα ελέγχεται η συνθήκη και αν εξακολουθεί να είναι αληθής εκτελείται το μπλοκ των εντολών
 - **for**: εκτελείται ένα μπλοκ εντολών τόσες φορές όσες καθορίσουμε εμείς
 - **foreach**: εκτελείται ένα μπλοκ εντολών για κάθε ένα στοιχείο ενός array.

Εντολές επανάληψης (2)

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<?php
```

```
    $x = 1;
```

```
    while($x <= 5) {
```

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

```
        $x++;
```

```
    }
```

```
?>
```

```
</body>
```

```
</html>
```

Εντολές επανάληψης (3)

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<?php
```

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

```
?>
```

```
</body>
```

```
</html>
```

PHP & Βάσεις Δεδομένων (1)

- Η PHP μας δίνει τη δυνατότητα σύνδεσης με βάσεις δεδομένων, για τη δημιουργία δυναμικών ιστοσελίδων
- Αντί να γράφουμε το περιεχόμενο μιας ιστοσελίδας μέσα στο `<body> </body>`, το αντλούμε από τις εγγραφές ενός πίνακα της βάσης δεδομένων
- Επιπλέον, έχουμε τη δυνατότητα επεξεργασίας της πληροφορίας, πριν την προβάλλουμε.

PHP & Βάσεις Δεδομένων (2)

- Ακολουθούμε πέντε βήματα:
 - Σύνδεση με τη βάση δεδομένων
 - Ορισμός του character set σε utf8
 - Άντληση των δεδομένων
 - Επεξεργασία και προβολή των δεδομένων
 - Κλείσιμο της σύνδεσης με τη βάση δεδομένων

PHP & Βάσεις Δεδομένων (3)

- Για τη σύνδεση με MySQL βάση δεδομένων, θα χρησιμοποιήσουμε μια επέκταση της MySQL που ονομάζεται MySQLi
- Η επέκταση αυτή συνήθως εγκαθίσταται με την εγκατάσταση του package “php5 mysql”

PHP & Βάσεις Δεδομένων (4)

**Παράδειγμα σύνδεσης PHP με MySQL
βάση δεδομένων**

Παραπομπές

- W3schools.com - PHP Tutorial:
 - <http://www.w3schools.com/php/default.asp>
- Εισαγωγικό tutorial για PHP:
 - <http://devzone.zend.com/6/php-101-php-for-the-absolute-beginner/>
- Επίσημος ιστότοπος της PHP:
 - <http://php.net>
- Πληροφορίες για server-side scripting:
 - http://en.wikipedia.org/wiki/Server-side_scripting