Τεχνολογίες Διαδικτύου

Εισαγωγή στη php

Δρ. Ρόζας Μαυροπόδη

Η πλευρά του εξυπηρετητή

Ο προγραμματισμός στη πλευρά του εξυπηρετητή αποτελεί μια τεχνική η οποία χρησιμοποιείται στην ανάπτυξη Διαδικτυακών εφαρμογών. Με την τεχνική αυτή χρησιμοποιούνται σενάρια κάποιας γλώσσας προγραμματισμού (π.χ. php, python, asp, ή cgi σε perl, c ή κάτι άλλο) τα οποία παράγουν μια απόκριση προσαρμοσμένη σε κάθε χρήστη διαφορετική αίτηση (επίσκεψη) στην ιστοσελίδα.

Η πλευρά του εξυπηρετητή - Πλεονεκτήματα

Τα σενάρια κώδικα στη πλευρά του εξυπηρετητή:

- μπορούν να συγκεντρώσουν τα χαρακτηριστικά του πελάτη για να προσαρμόσουν την απόκριση βάσει αυτών των χαρακτηριστικών, των απαιτήσεων του χρήστη, των δικαιωμάτων πρόσβασης κλπ.
- επιτρέπουν επίσης στον κάτοχο της ιστοσελίδας να κρύψει τον πηγαίο κώδικα που δημιουργεί τη διασύνδεση
- επιτρέπουν την αποθήκευση ευαίσθητών ή μη δεδομένων για μεγάλο χρονικό διάστημα

Η πλευρά του εξυπηρετητή - μειονεκτήματα

Μειονέκτημα αποτελεί ότι:

- ο πελάτης πρέπει να κάνει περαιτέρω αιτήματα μέσω του δικτύου στο διακομιστή προκειμένου να εμφανίσει νέες πληροφορίες στον χρήστη μέσω του φυλλομετρητή.
- τα αιτήματα, αυτά, ενδέχεται να επιβραδύνουν τη λειτουργία της διαδικτυακής εφαρμογής,
- τα αιτήματα, αυτά, ενδέχεται να τοποθετήσουν περισσότερο φόρτο στον εξυπηρετητή, αλλά και το δίκτυο
- η εφαρμογή σταματά λειτουργεί εφόσον ο χρήστης αποσυνδεθεί από το δίκτυο και τον εξυπηρετητή.

Τί είναι η php

Η **PHP** (Hypertext Preprocessor) (php.net/) είναι μία **ελεύθερα διαθέσιμη** γλώσσα και **διερμηνέας σεναρίων**. Αρχικά χρησιμοποιήθηκε σε κεντρικούς υπολογιστές με περιβάλλον UNIX, αλλά μπορεί να χρησιμοποιηθεί και σε **άλλα λειτουργικά συστήματα**. Μία σελίδα HTML που περιλαμβάνει σενάρια PHP έχει τη χαρακτηριστική επέκταση "**.php**", ".php3", ".php4" ή ".phtml".

Στα τέλη του 1994, ως μία εύκολη Perl. Γράφτηκε από τον Rasmus Lerdorf. Σήμερα κυκλοφορεί η έκδοση 7+.

Τί μπορεί να κάνει

Μπορεί να:

- λάβει πληροφορία από φόρμες και να τη χρησιμοποιήσει με διάφορους τρόπους: να την αποθηκεύσει σε μία βάση δεδομένων, να δημιουργήσει εξαρτημένες/υποθετικές σελίδες που εξαρτώνται από τα περιεχόμενα της φόρμας, να τοποθετήσει cookies στο φυλλομετρητή του χρήστη και να στείλει ηλεκτρονικό ταχυδρομείο (e-mail).
- πιστοποιήσει και να εντοπίσει χρήστες.
- φιλοξενήσει συζητήσεις στην ιστοσελίδα.
- προσαρμόσει την εμφάνιση μιας ιστοσελίδας στους διαφορετικούς φυλλομετρητές ή συσκευές, τους οποίους χρησιμοποιούν οι χρήστες.
- εξυπηρετήσει σελίδες XML και JSON.

Πώς το κάνει

Όταν κάποιος πελάτης (χρήστης) επισκεφτεί την ιστοσελίδα που περιέχει κώδικα PHP, ο **εξυπηρετητής εκτελεί** τον κώδικα και αυτό που επιστρέφεται στον χρήστη είναι **μία ιστοσελίδα HTML.** Όλη τη δουλειά την κάνει ο εξυπηρετητής και όχι κάποιος φυλλομετρητής. Ο πελάτης (χρήστης) από την πλευρά του δε χρειάζεται κάποιο πρόσθετο εργαλείο ή πρόγραμμα για να δει το αποτέλεσμα της PHP, επειδή όπως προαναφέρθηκε, το αποτέλεσμα φτάνει στον χρήστη ως κώδικας HTML.

Πώς ...

- Όταν ο επισκέπτης κάνει 'προβολή κώδικα σελίδας' βλέπει μόνον κώδικα HTML.
- Ο προγραμματιστής μπορεί να χρησιμοποιήσει οποιονδήποτε κειμενογράφο ώστε να γράψει κώδικα PHP.
- Τα αποτελέσματα της εκτέλεσης ενός αρχείου με κώδικα PHP μπορεί να εμφανιστούν μόνο εφόσον το αρχείο έχει 'δημοσιευτεί' σε κάποιον εξυπηρετητή Ιστού (web server), π.χ. Apache, IIS, gws κλπ. Στη συνέχεια ο επισκέπτης προβάλει το αρχείο αυτό στον φυλλομετρητή του.

Βασικά στοιχεία

- Αποτελεί μία γλώσσα σεναρίων (χαλαρή με απλή δομή και σύνταξη).
- Τα αρχεία με τον κώδικα της δεν χρειάζονται μεταγλώττιση (παραγωγή .exe) απλά ερμηνεύονται μέσα από τον διερμηνέα (php.exe) της PHP.

Βασικά στοιχεία

- Όλα τα σενάρια PHP περικλείονται στις εντολές: `<?php` και `?>`. Π.χ.

```
<?php
      print ("Αυτό είναι ένα παράδειγμα!");
?>
```

- Οι εντολές τερματίζουν με το σύμβολο του `;`
- Όπως και στην HTML, η μορφοποίηση του κώδικα PHP,
- δηλαδή τα κενά, οι αλλαγές γραμμής, κτλ., δεν επηρεάζει
- το αποτέλεσμα.
- Διάκριση πεζών/κεφαλαίων

Εισαγωγή σχολίων #, //, /**/

```php
<?php
// Αυτές οι γραμμές θα αγνοηθούν.
# Αποτελούν σχόλια
print ("Αυτό είναι ένα παράδειγμα!");
/* και αυτές οι γραμμές θα αγνοηθούν.
Αποτελούν ένα τμήμα σχολίων. */
?>
```

Συναρτήσεις εξόδου

Για την εκτύπωση κάποιου κειμένου στην οθόνη του φυλλομετρητή μπορεί να χρησιμοποιηθούν: `print()`, `echo()`.

```
<?php
      echo 'Γειά σου Κόσμε!<br>’;
      echo ('Γειά σου Κόσμε!<br>');
      print 'Γειά σου Κόσμε!<br>';
      print('Γειά σου Κόσμε!<br>') ;
?>
```

Συναρτήσεις εξόδου

Ενώ και οι δύο είναι συναρτήσεις, εντούτοις αποτελούν εξαίρεση καθώς λειτουργούν με ή χωρίς τις παρενθέσεις. Οπότε, μπορεί να βρίσκονται με τη μορφή `print ()` και `echo ()` ή `print` και `echo`.

Συναρτήσεις εξόδου

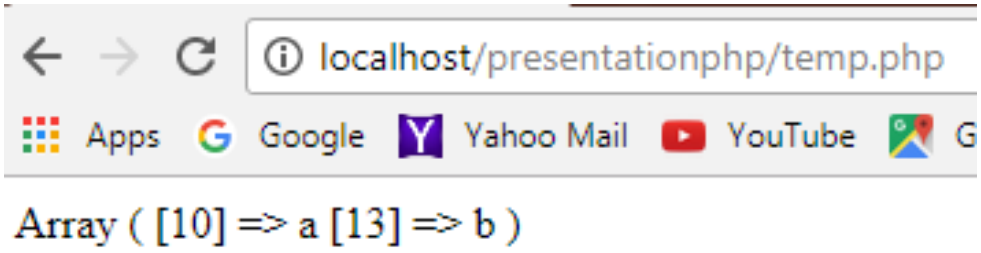
Η `print` δέχεται μια παράμετρο ως δεδομένο εισόδου και επιστρέφει κάποια τιμή. Ενώ η `echo` μπορεί να έχει περισσότερες της μιας παραμέτρους εισόδου και δεν επιστρέφει κάποια τιμή, το οποίο την καθιστά σχετικά γρηγορότερη. Αντίθετα, εξαιτίας της μη επιστροφής τιμής, η `echo` δε μπορεί να χρησιμοποιηθεί ως τμήμα πολύπλοκων διαδικασιών.

```
echo "κάποιο κείμενο 1", " κάποιο κείμενο 2";
```

Συναρτήσεις εξόδου

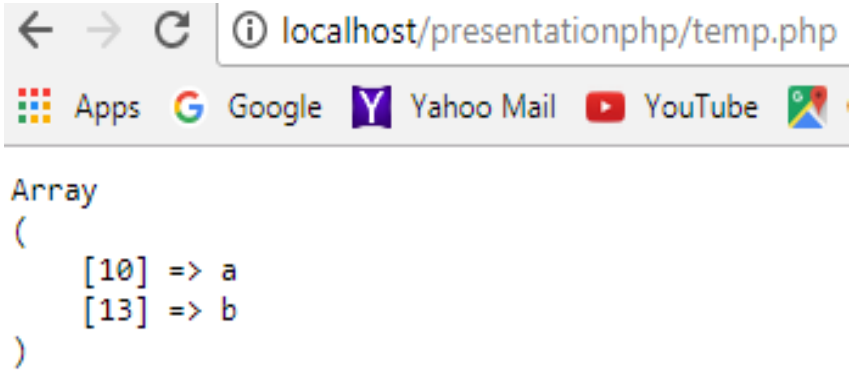
Υπάρχει και η print_r() χρήσιμη για την εκτύπωση πινάκων.

```php
<?php
      $letters = array (10=>'a' , 13=>'b' );
      print_r($letters);
?>
```



Συναρτήσεις εξόδου

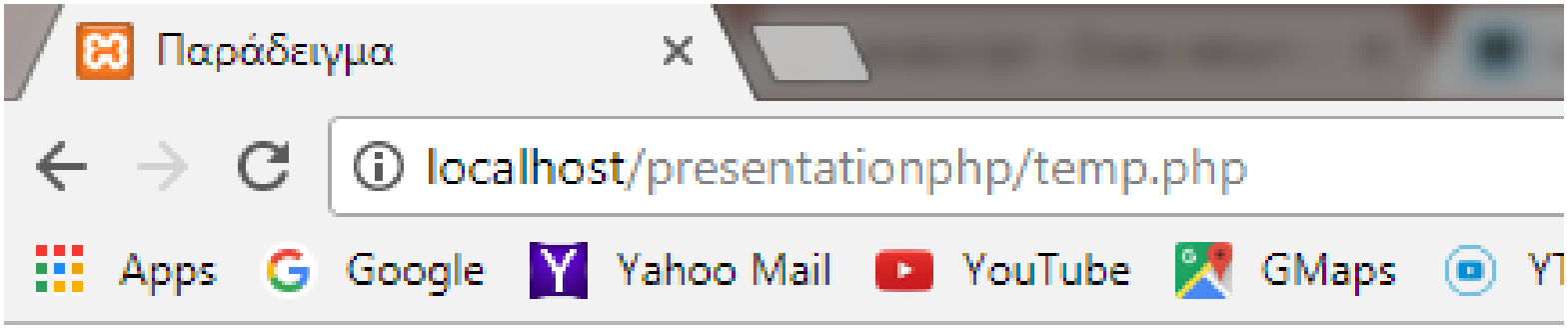
```php
<?php
	$letters = array (10=>'a' , 13=>'b' );
	echo "<pre>";
	print_r($letters);
	echo "</pre>";
?>
```



Ενσωμάτωση σε HTML

```
<html>
<head>
<title> Παράδειγμα </title>
</head>
<body>
<p style="color: red;">
Ο κώδικας PHP δημιουργεί μια σελίδα που λέει:
</p>
<p><?php print ("Αυτό είναι ένα παράδειγμα!"); ?>
</p>
</body>
</html>
```

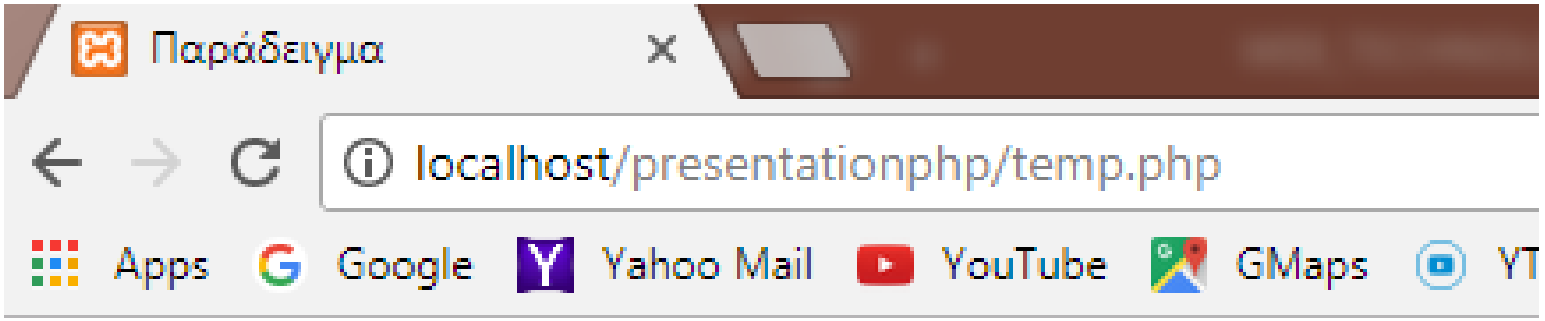
Ενσωμάτωση σε HTML..



Ενσωμάτωση σε HTML..

```php
<?php
echo "<html>";
echo "<head> ";
echo "<title> Παράδειγμα </title>";
echo "</head> ";
echo "<body> ";
echo "<p style='color: red;'>Ο κώδικας PHP δημιουργεί μια σελίδα που λέει:</p> ";
echo "<p>";
echo "Αυτό είναι ένα παράδειγμα!";
echo "</p>";
echo date('l, d-m-Y');
echo "</body> ";
echo "</html>"; ?>
```

Ενσωμάτωση σε HTML..



Ενσωμάτωση σε HTML..

Έστω ότι το αρχείο test.php περιέχει το παρακάτω:

```
<?php print ("Αυτό είναι ένα παράδειγμα!"); ?>
```

Και στο αρχείο example1.php

```
<html>
<head>
<title> Παράδειγμα </title>
</head>
<body>
<p style="color: red;">Ο κώδικας PHP δημιουργεί μια σελίδα που λέει:</p>
<p> <?php include 'test.php'; ?> </p>
</body>
</html>
```

Εισαγωγή αρχείου κώδικα

- include, require

Η βασικότερη διαφορά ανάμεσα στην include και στην require αποτελεί το γεγονός ότι εάν το αρχείο (test.php) δε βρεθεί κατά την κλήση της ιστοσελίδας, τότε η μεν πρώτη (include) θα εκτυπώσει προειδοποιητικό μήνυμα (warning), αλλά θα συνεχίσει να εκτελεί το πρόγραμμα, η δε δεύτερη (require) θα εκτυπώσει μήνυμα λάθους (error) και θα σταματήσει την εκτέλεση.

- include_once, require_once

Παρόμοια λειουργία με τα ανωτέρω, αλλά εφόσον έχει ήδη γίνει include ή require προηγούμενα δεν το επαναεισάγει

Μεταβλητές

- Όλες οι μεταβλητές ξεκινούν με το σύμβολο του δολαρίου $.
- Το όνομά της αρχίζει με γράμμα
- Να αποτελείται από γράμματα, αριθμούς και τον χαρακτήρα υπογράμμισης (_);
- Να μην πρόκειται για κάποιο δεσμευμένο όνομα όπως για παράδειγμα το "print".
- Διάκριση πεζών / κεφαλαίων

Μεταβλητές

Δεν χρειάζονται ορισμό, μπορούν να χρησιμοποιηθούν άμεσα.

```
<?php
        $name ='Phil';
        $age = 23;
        echo $name;
        echo ' is ';
        echo $age ;
        // Phil is 23
?>
```

Μονά ή διπλά εισαγωγικά

Στα μονά εισαγωγικά οι μεταβλητές δεν μεταφράζονται στις τιμές τους.

```
<?php
	$name = 'Phil' ;
	$age = 23;
	echo "$name is $age";
	// Phil is 23
	echo '$name is $age';
	// $name is $age
?>
```

Χαρακτήρας διαφυγής "\"

Αν θελήσουμε να εκτυπώσουμε το κείμενο με τα εισαγωγικά πρέπει να χρησιμοποιήσουμε το χαρακτήρα διαφυγής "\", ο οποίος ορίζει στην PHP να μη χρησιμοποιήσει τον επόμενο χαρακτήρα ως μέρος του κώδικα.

```
print (" \" Αυτό είναι ένα παράδειγμα!\"" );
// "Αυτό είναι ένα παράδειγμα!"
```

Βαθμωτές μεταβλητές

Μία βαθμωτή μεταβλητή μπορεί να περιέχει αριθμούς, γράμματα, φράσεις, κτλ. Τι μπορώ να κάνω:

- να τους θέσουμε τιμή με το σύμβολο = , π.χ. $stuff = "Τι κάνεις;";
- να χρησιμοποιούμε εισαγωγικά εάν πρόκειται για χαρακτήρες,
  π.χ. "Τι κάνεις;". Οι αριθμοί δεν απαιτούν εισαγωγικά.
- να τελειώνουμε κάθε εντολή με το σύμβολο του ερωτηματικού (;).

Καλούμε τη μεταβλητή απλά αναφέροντας το όνομά της.

Πίνακες

Απλή μεταβλητή:

```
$student1 = "Ανδριαννή";
$student2 = "Ηλιάνα";
$student3 = "Λευτέρης";
```

Μεταβλητή πίνακας. Αποθηκεύονται πολλές τιμές ταυτόχρονα.

```
$class = array("Ανδριαννή", "Ηλιάνα", "Λευτέρης");
```

Κλειδιά

Ο συγκεκριμένος τύπος πίνακα εκχωρεί αυτόματα ένα αριθμημένο **κλειδί** σε κάθε στοιχείο του πίνακα, δίνοντας στο πρώτο στοιχείο τον αριθμό 0, στο δεύτερο τον αριθμό 1 κ.ο.κ. Έτσι, η Ανδριαννή είναι το στοιχείο [0] κλπ. Αναφερόμαστε σε στοιχείο

```
echo $class[2];
```

το οποίο θα αποδώσει την τιμή “Λευτέρης”.

Προσθήκη νέου στοιχείου

```
$class[ ] = "Ανδριαννή";
$class[ ] = "Ηλιάνα";
$class[ ] = "Λευτέρης";
```

Για να προσθέσουμε ένα νέο μαθητή γράφουμε (ανεξάρτητα από τον τρόπο που χρησιμοποιήσαμε για τη δημιουργία του πίνακα):

```
$class[ ] = "Βασίλης";
```

Συσχετιζόμενοι πίνακες

Οι συσχετιζόμενοι πίνακες διαχωρίζουν τα περιεχόμενα στοιχεία όχι με αριθμούς αλλά με **λεκτικά** που καθορίζει ο προγραμματιστής. Το ζευγάρι κλειδί και τιμή (key - value), ορίζει πλήρως κάθε στοιχεί ενός πίνακα key=>"value".

```
$students = array (
              "name"=>"Τόνυ",
              "haircolor"=>"μαύρα",
              "eyecolor"=>"καστανά",
              "age"=>5
              );
```

Αναφορά σε συγκεκριμένο στοιχείο.

```
print $students[eyecolor];
```

Σταθερές

Παρόμοιο μηχανισμό με τις μεταβλητές για την αποθήκευση / κατοχή πληροφοριών, αποτελούν και οι σταθερές (**constants**). Υπάρχει, όμως, μια ουσιαστική διαφορά, σε σχέση με τις μεταβλητές. Στις σταθερές **δεν είναι δυνατή η αλλαγή της τιμής** τους από τη στιγμή που τους έχει οριστεί, δηλαδή τους έχει εκχωρηθεί, κάποια αρχική τιμή.

```
define("ROOT_LOCATION", "/usr/local/www/");
```

Σταθερές

- Το όνομα τους είναι, συνήθως, με κεφαλαία.
- Δεν χρησιμοποιούν το $
- Χρησιμοποιούνται όπως όλες οι άλλες μεταβλητές.

```php
<?php
      define ( 'NAME' 'Phil' ) ;
      define ( 'AGE' , 23) ;
      echo NAME;
      echo ' is ';
      echo AGE;
      // Phil is 23
?>
```

Αριθμητικοί τελεστές

Τελεστής	Περιγραφή
+	Αθροίζει δύο τιμές.
-	Αφαιρεί δύο τιμές.
*	Πολλαπλασιάζει δύο τιμές.
/	Διαιρεί δύο τιμές.
%	Παρουσιάζει το υπόλοιπο της διαίρεσης δύο αριθμών.
++	Αυξάνει την τιμή κατά ένα.
--	Ελαττώνει την τιμή κατά ένα.

Incrementing/Decrementing Operators

Example	Name	Effect
++$a	Pre-increment	Increments *$a* by one, then returns *$a*.
$a++	Post-increment	Returns *$a*, then increments *$a* by one.
--$a	Pre-decrement	Decrements *$a* by one, then returns *$a*.
$a--	Post-decrement	Returns *$a*, then decrements *$a* by one.

Σημείωση

- Ανάλογα με τη θέση της μεταβλητής σε σχέση με τον αριθμητικό τελεστή ++, επιστρέφεται και διαφορετικό αποτέλεσμα. Για παράδειγμα αν γραφεί το ++$a, τότε αυξάνεται η τιμή κατά μία μονάδα και στη συνέχεια επιστρέφεται/αποδίδεται η μεταβλητή προς χρήση. Ενώ αν γραφεί το $a++, τότε, πρώτα επιστρέφεται και στη συνέχεια αυξάνεται η τιμή της μεταβλητής κατά μια μονάδα Το παραπάνω χρησιμοποιείται συχνά στους βρόχους επανάληψης.

Σημείωση παράδειγμα

Έστω ότι $a=1

Το ++$a είναι: 2 το $a είναι:2

$a=1

Το $a++ είναι: 1 το $a είναι:2

Τελεστές εκχώρησης

Τελεστής	Περιγραφή
=	Εκχωρεί την τιμή δεξιά του συμβόλου στη μεταβλητή που υπάρχει αριστερά.
+=	Προσθέτει τη ποσότητα που βρίσκεται στην δεξιά πλευρά του τελεστή, στη μεταβλητή που υπάρχει αριστερά.
-=	Αφαιρεί την ποσότητα που βρίσκεται στην δεξιά πλευρά του τελεστή, στη μεταβλητή που υπάρχει αριστερά.
*=	Πολλαπλασιάζει την ποσότητα που βρίσκεται στην δεξιά πλευρά του τελεστή με τη μεταβλητή που υπάρχει αριστερά. Το αποτέλεσμα εκχωρείται στη μεταβλητή που υπάρχει αριστερά.
/=	Διαιρεί τη μεταβλητή που βρίσκεται στην αριστερά πλευρά του τελεστή με τη ποσότητα που υπάρχει δεξιά. Το αποτέλεσμα εκχωρείται στη μεταβλητή που υπάρχει αριστερά.
.=	Προσθέτει / συνδυάζει την αλφαριθμητική τιμή (string) που βρίσκεται στην δεξιά πλευρά του τελεστή με τη μεταβλητή που υπάρχει αριστερά. Το αποτέλεσμα εκχωρείται στη μεταβλητή που υπάρχει αριστερά. Χρησιμοποιείται για αλφαριθμητικές (string) τιμές και μεταβλητές.
%=	Αποδίδει το υπόλοιπο (modulo) μετά τη διαίρεση της μεταβλητής που βρίσκεται στην αριστερά πλευρά του τελεστή με τη ποσότητα που υπάρχει δεξιά. Το αποτέλεσμα εκχωρείται στη μεταβλητή που υπάρχει αριστερά.

Τελεστές σύγκρισης

Τελεστής	Περιγραφή
==	Επιστρέφει την τιμή ‘αληθής’ (true), εάν η τιμή της μεταβλητής στα αριστερά είναι ίση με αυτή στα δεξιά. Αλλιώς επιστρέφει την τιμή ‘ψευδής’ (false).
===	Επιστρέφει την τιμή ‘αληθής’ (true), εάν η τιμή της μεταβλητής αριστερά είναι ίση με αυτή στα δεξιά, και είναι και του ίδιου τύπου. Αλλιώς επιστρέφει την τιμή ‘ψευδής’ (false).
!=	Επιστρέφει την τιμή ‘αληθής’ (true), εάν η τιμή της μεταβλητής αριστερά δεν είναι ίση με αυτή στα δεξιά. Αλλιώς επιστρέφει την τιμή ‘ψευδής’ (false).
<>	Επιστρέφει την τιμή ‘αληθής’ (true), εάν η τιμή της μεταβλητής αριστερά δεν είναι ίση με αυτή στα δεξιά. Αλλιώς επιστρέφει την τιμή ‘ψευδής’ (false).
!==	Επιστρέφει την τιμή ‘αληθής’ (true), εάν η τιμή της μεταβλητής αριστερά δεν είναι ίση με αυτή στα δεξιά και δεν είναι και του ίδιου τύπου. Αλλιώς επιστρέφει την τιμή ‘ψευδής’ (false).
<	Επιστρέφει την τιμή ‘αληθής’ (true), εάν η τιμή της μεταβλητής αριστερά είναι μικρότερη από αυτή στα δεξιά. Αλλιώς επιστρέφει την τιμή ‘ψευδής’ (false).
>	Επιστρέφει την τιμή ‘αληθής’ (true), εάν η τιμή της μεταβλητής αριστερά είναι μεγαλύτερη από αυτή στα δεξιά. Αλλιώς επιστρέφει την τιμή ‘ψευδής’ (false).
<=	Επιστρέφει την τιμή ‘αληθής’ (true), εάν η τιμή της μεταβλητής αριστερά είναι μικρότερη ή ίση με αυτή στα δεξιά. Αλλιώς επιστρέφει την τιμή ‘ψευδής’ (false).
>=	Επιστρέφει την τιμή ‘αληθής’ (true), εάν η τιμή της μεταβλητής αριστερά είναι μεγαλύτερη ή ίση με αυτή στα δεξιά. Αλλιώς επιστρέφει την τιμή ‘ψευδής’ (false).

Λογικοί τελεστές

Τελεστής	Περιγραφή
and	Επιστρέφει την τιμή ‘αληθής’ (true), εάν αν η τιμή της μεταβλητής αριστερά και στα δεξιά είναι ‘αληθής’. Αλλιώς αν κάποια από τις δύο είναι ‘ψευδής’ επιστρέφει την τιμή ‘ψευδής’ (false).
or	Επιστρέφει την τιμή ‘αληθής’ (true), εάν η τιμή μιας μεταβλητής είναι ‘αληθής’ είτε και οι δύο είναι ‘αληθής’. Αλλιώς αν και οι δύο είναι ‘ψευδής’ επιστρέφει την τιμή ‘ψευδής’ (false).
xor	Επιστρέφει την τιμή ‘αληθής’ (true), εάν η τιμή μιας μεταβλητής (αριστερά ή στα δεξιά) είναι ‘αληθής’. Αλλιώς αν και οι δύο είναι ‘ψευδής’ ή ‘αληθής’, επιστρέφει την τιμή ‘ψευδής’ (false).
!	Επιστρέφει την τιμή ‘αληθής’ (true), εάν η τιμή της μεταβλητής δεν είναι ‘ψευδής’. Αλλιώς αν η τιμή της μεταβλητής είναι ‘αληθής’ επιστρέφει την τιμή ‘ψευδής’ (false).
&&	Επιστρέφει την τιμή ‘αληθής’ (true), εάν η τιμή της μεταβλητής αριστερά και στα δεξιά είναι ‘αληθής’. Αλλιώς αν κάποια από τις δύο είναι ‘ψευδής’ επιστρέφει την τιμή ‘ψευδής’ (false). Ίδιος με τον τελεστή ‘and’.
||	Επιστρέφει την τιμή ‘αληθής’ (true), εάν η τιμή της μεταβλητής αριστερά ή στα δεξιά είναι ‘αληθής’, είτε είναι ‘αληθής’ και οι δύο. Αλλιώς αν και οι δύο είναι ‘ψευδής’ επιστρέφει την τιμή ‘ψευδής’ (false). Ίδιος με τον τελεστή ‘or’.

Συνδυασμός αλφαριθμητικού

String concatination:

```
<?php
$firstname = 'Rob ';
$surname ='Tuley';
// displays 'Rob Tuley'
echo $firstname . $surname ;
?>
```

Εντολές διακλάδωσης στην PHP

```
if (συνθήκη) {
/* Ο κώδικας αυτός θα εκτελεστεί εάν η συνθήκη είναι αληθής */
}
if (συνθήκη) {
/* Ο κώδικας αυτός θα εκτελεστεί εάν η συνθήκη είναι αληθής */
} else {
/* Ο κώδικας αυτός θα εκτελεστεί εάν η συνθήκη είναι ψευδής */
}
```

Εντολές διακλάδωσης στην PHP

```
if (συνθήκη1) {
/* Ο κώδικας αυτός θα εκτελεστεί εάν η συνθήκη1 είναι αληθής */
} elseif (συνθήκη2) {
/* Ο κώδικας αυτός θα εκτελεστεί εάν η συνθήκη2 είναι αληθής */
} else {
/* Ο κώδικας αυτός θα εκτελεστεί εάν η συνθήκη1 και συνθήκη2 είναι ψευδής */
}
```

Εντολές διακλάδωσης στην PHP

```
<?php
$b = 13;
if ($a<$b) {
      echo 'a is smaller than b' ;
} elseif ($a==$b) {
      echo 'la is equal to b' ;
} else {
      echo 'a is bigger than b' ;
}
?>
```

Μια γρήγορη διακλάδωση

Η παρακάτω γραμμή κώδικα αποτελεί παράδειγμα πολύπλοκων διαδικασιών, στις οποίες δε μπορεί να χρησιμοποιηθεί η συνάρτηση echo.

```
$b ? print "Είναι TRUE" : print "Είναι FALSE";
```

Το παραπάνω αποτελεί μια ερώτηση για την τιμή της μεταβλητής $b. Εάν είναι αληθής (true) τότε θα εκτυπωθεί το πρώτο λεκτικό, δηλαδή η φράση "Είναι TRUE". Στην αντίθετη περίπτωση, δηλαδή η μεταβλητή $b είναι ψευδής (false), τότε θα εκτυπωθεί το δεύτερο λεκτικό.

Μια γρήγορη διακλάδωση

Είναι ισοδύναμα:

```
$b ? print "Είναι TRUE" : print "Είναι FALSE";
```

ή

```
if ($b) {
    print "Είναι TRUE";
} else {
    print "Είναι FALSE" ;
}
```

Switch

```
switch ($FavoriteDay) {
case "Δευτέρα":
      echo "Μου αρέσει η Δευτέρα πολύ!!";
      break;
case "Τρίτη":
      echo "Μου αρέσει η Τρίτη πολύ!!";
      break;
case "Τετάρτη":
      echo "Μου αρέσει η Τετάρτη πολύ!!";
      break;
case "Πέμπτη":
      echo "Μου αρέσει η Πέμπτη πολύ!!"; break;
default:
echo "Δεν έχω κάποια προτίμηση.";
}
```

Βρόχοι επανάληψης

```
while (συνθήκη που είναι αληθής) {
    εντολές προς επανάληψη;
}
```

ή εναλλακτικά

```
while (συνθήκη):
    εντολές
endwhile;
```

Βρόχοι επανάληψης

```
do {
     εντολές προς επανάληψη ;
} while (συνθήκη είναι αληθής)


for (αρχικοποίηση; έλεγχος; μεταβολή) {
     εντολές προς επανάληψη;
}
ή
for (αρχικοποίηση; έλεγχος; μεταβολή ):
     εντολές
endfor;
```

Βρόχοι επανάληψης σε πίνακα

```
foreach (όνομα_πίνακα as τιμή_μεταβλητή) {
εντολές προς επανάληψη;
}
```

```
<?php
$students = array ("Ανδριαννή","Ηλιάνα","Λευτέρης","Γιάννης");
foreach ($students as $each_value) {
        echo $each_value;
}
?>
```

Δηλαδή πραγματοποιείται το: $each_value= $students[0]

Συναρτήσεις

Γιατί να τις προτιμήσω;

- Μειώνουν τις επαναλήψεις κώδικα
- Διευκολύνουν τη συντήρηση του προγράμματος
- Διευκολύνουν την επίλυση λαθών
- Μπορεί να χρησιμοποιηθούν από διαφορετικές εφαρμογές

Συναρτήσεις

```
function MyFunction () {
    //κώδικας συνάρτησης
}

<?php
function MyMessage(){
print "Αυτό είναι ένα απλό παράδειγμα συνάρτησης";
}
?>
```

Συναρτήσεις

- Η δήλωση function MyMessage () ορίζει τη συνάρτηση με όνομα MyMessage. Το περιεχόμενο της συνάρτησης, το οποίο περικλείεται σε άγκιστρα { }, εκτυπώνει τη φράση "Αυτό είναι ένα απλό παράδειγμα συνάρτησης".

Συναρτήσεις προκαθορισμένες τιμές

```php
<?php
function MySum($num1, $num2=30) {
      $sum = $num1 + $num2;
      echo "Το άθροισμα είναι: $sum";
}
MySum(10, 40); // άθροισμα = 50
MySum(10); //άθροισμα =40
?>
```

- Στη δεύτερη κλήση της συνάρτησης έχει παραληφθεί το δεύτερο όρισμα. Εντούτοις, δεν υπάρχει λάθος, καθώς χρησιμοποιείται η προκαθορισμένη τιμή (30).
- Οι παράμετροι που θα έχουν προκαθορισμένες τιμές θα πρέπει να είναι τελευταίες στη λίστα με τις παραμέτρους του ορισμού της συνάρτησης.

by reference

```php
<?php
function MySum(&$num1, $num2) {
$num1 += $num2;
echo "Το άθροισμα είναι: $num1"; }
$n = 10;
MySum($n, 20); //άθροισμα =30
echo "Εκτός συνάρτησης είναι: $n";
//δίνει 30
?>
```

Επιστροφή τιμής

Έτσι μπορεί να χρησιμοποιηθεί σε κάποια συνάρτηση εξόδου ή να αποθηκευτεί σε κάποια μεταβλητή

```
<?php
function MySum($num1, $num2) {
return $num1 + $num2;
}
echo "Το άθροισμα είναι:". MySum(10, 20);
$temp= MySum(10, 20); //αποθήκευση σε μεταβλητή
?>
```

Εμβέλεια

- **Τοπικές** -- Οι τοπικές μεταβλητές είναι προσβάσιμες μόνο από το τμήμα του κώδικα, στο οποίο έχουν οριστεί.
- **Στατικές** – Οι στατικές μεταβλητές μοιάζουν αρκετά με τις τοπικές. Για τον ορισμό τους χρησιμοποιείται η εντολή static. Έχουν εμβέλεια μόνο μέσα στη συνάρτηση όπου έχουν οριστεί, αλλά δεν χάνουν την τιμή τους όταν ολοκληρωθεί η συνάρτηση.
- **Καθολικές** – Οι καθολικές μεταβλητές μπορεί να προσπελαστούν από οποιοδήποτε τμήμα του κώδικα της εφαρμογής.

Τοπικές μεταβλητές

```php
<?php
function MySum($num1, $num2) {
    $sum = $num1 + $num2;
    echo "Το άθροισμα είναι: $sum";
}
MySum(10, 20);//άθροισμα =30
echo $sum; //λάθος
?>
```

Στατικές μεταβλητές ορισμός

```
<?php
function some_function(){
    static $int = 0; // σωστό
    static $int = 1+3; // σωστό (ισχύει στην PHP 5.6+)
    static $int = max(1,2,3); // λάθος
/*κώδικας συνάρτησης*/
}

?>
```

Στατικές μεταβλητές

```php
<?php
function my_static() {
    static $count = 0;
    echo $count; //θα δώσει 0
    $count++;
}
echo $count; //θα δώσει 1
?>
```

Καθολικές μεταβλητές

```php
<?php
function some_function(){
     global $int;
     $int = 0; //σωστό
     global $int = 0; //λάθος
     $GLOBALS["var"]=0;//σωστός ορισμός και
     αρχικοποίηση
     $GLOBALS["int"]=99; //εκχώρηση τιμής
     echo $int; //εκτυπώνει 99
}
echo $int; //εκτυπώνει 99
?>
```

Παράδειγμα

```
global logged_in;
```

Οπότε, όταν συνδέεται ο χρήσης στη μεταβλητή τίθεται η τιμή 1 και όποτε αποσυνδέεται η τιμή 0. Σε αυτή την περίπτωση κάθε τμήμα της εφαρμογής γνωρίζει την τιμή της, χρησιμοποιώντας έναν απλό έλεγχο, όπως:

```
if ($logged_in==1) {/*κάνε κάτι */}
```

Υπερκαθολικές μεταβλητές

Μεταβλητή	Περιγραφή
$GLOBALS	Περιέχει όλες τις καθολικές μεταβλητές.
$_SERVER	Περιέχει πληροφορίες όπως: επικεφαλίδες πρωτοκόλλου, μονοπάτια και τοποθεσίες σεναρίων. Τα περιλαμβανόμενα στοιχεία, και η τιμή, τους παρέχονται από τον εξυπηρετητή Ιστού (web server) και είναι δυνατόν να παρέχονται όλα, κάποια ή κανένα, ανάλογα την εγκατάσταση.
$_ENV	Περιέχει μεταβλητές που παρέχονται από το περιβάλλον εγκατάστασης της PHP.
$_GET	Περιέχει μεταβλητές που παρέχονται μέσω της μεθόδου HTTP Get.
$_POST	Περιέχει μεταβλητές που παρέχονται μέσω της μεθόδου HTTP Post.
$_COOKIE	Περιέχει τις μεταβλητές που χρησιμοποιούνται ως HTTP cookies.
$_REQUEST	Περιέχει πληροφορίες που παρέχει ο φυλλομετρητής. Αποτελεί γονικό πίνακα των $_GET, $_POST, and $_COOKIE.
$_SESSION	Περιέχει τις μεταβλητές που χρησιμοποιούνται σε σύνοδο (session).
$_FILES	Περιέχει τα αρχεία που έχουν ‘ανεβεί’ (upload) στον εξυπηρετητή Ιστού με τη χρήση της μεθόδου HTTP Post.