

Πώς να Ξεφύγουμε από την HTML

Υπάρχουν τέσσερις τρόποι για να μπορέσουμε να ξεφύγουμε από την HTML και να μπούμε στην μέθοδο συγγραφής κώδικα της PHP (PHP code mode) :

1ος τρόπος

```
<? echo ("Είναι η απλούστερη, μια εντολή επεξεργασίας SGML \n"); ?>
```

2ος τρόπος

```
<?php echo ("Αν θέλουμε να εξυπηρετήσουμε XML έγγραφα \n"); ?>
```

3ος τρόπος

```
<script language="php">
    echo ("Σε μερικούς editors, όπως ο FrontPage, δεν αρέσουν οι
εντολές επεξεργασίας");
</script>
```

4ος τρόπος

```
<% echo ("Μπορούμε να χρησιμοποιήσουμε και tags με στυλ ASP"); %>
```

Ο πρώτος τρόπος είναι διαθέσιμος μόνο αν έχουμε ενεργοποιήσει τα σύντομα (short) tags. Αυτό μπορεί να γίνει με τη συνάρτηση *short_tags()*, ενεργοποιώντας το *short_open_tag configuration setting* στο αρχείο *config* της PHP ή μεταγλωττίζοντας την PHP με την επιλογή *--enable-short-tags option*.

Ο τέταρτος τρόπος είναι διαθέσιμος μόνο αν έχουν ενεργοποιηθεί τα tags με στυλ ASP με το *asp_tags configuration setting*. Η υποστήριξη για τα ASP-style tags προστέθηκε στην έκδοση 3.0.4.

Οι εντολές διαχωρίζονται με τον ίδιο τρόπο όπως και στην C ή την Perl - τερματίζουμε κάθε εντολή με ένα ερωτηματικό (;)

Το tag κλεισίματος (?>) επίσης υποδηλώνει το τέλος μιας έκφρασης-δηλώσης, συνεπώς τα ακόλουθα είναι ισοδύναμα:

```
<?php
    echo "This is a test";
?>
<?php echo "This is a test" ?>
```

Σχόλια

Η PHP χρησιμοποιεί το στυλ της C στα σχόλια, καθώς και τα σχόλια που χρησιμοποιούμε στο Shell

```
<html>
<title>PHP Comments</title>
<?
echo "Three Kinds of PHP comments";
//This is a PHP single line comment;
/*This is a
multiple line
comment*/
echo "This is a test";
echo "One Final Test"; # This is shell-style style comment
?>
```

```
<!--This of course is an HTML style comment-->
</html>
```

1^{ος} τρόπος: // (για κάθε γραμμή)

2^{ος} τρόπος: /* */ (για block γραμμών)

3^{ος} τρόπος: # comments

Είναι καλή πρακτική, να σχολιάζεται πάντα ο κώδικας για μελλοντική επεξεργασία / αλλαγή.

Μεταβλητές (Variables)

Μια περιοχή της μνήμης που ορίζεται από τον προγραμματιστή για να αποθηκεύει προσωρινά δεδομένα.

- **Γενικά**

Οι μεταβλητές ξεκινούν με \$.

Αντιστοιχίζεται η τιμή με το =

Όλες οι μεταβλητές πρέπει να ξεκινούν με letter ή _(underscore).

Δεν επιτρέπεται μια μεταβλητή να έχει τα +, -, &, *

Το όνομα της μεταβλητής απαγορεύεται να ξεκινάει με ψηφίο

Παραδείγματα μεταβλητών:

```
$today="Monday"; //(κείμενο, string)
```

```
$total=1000; //(number)
```

Οι μεταβλητές είναι case-sensitive

π.χ. \$total και \$Total είναι διαφορετικές μεταβλητές.

Ωστόσο μπορεί να αλλάξει η τιμή της μεταβλητής συνέχεια:

```
$total=10;
```

```
$total=100;
```

Είναι καλή προγραμματιστική τεχνική, πάντα να αρχικοποιούνται οι μεταβλητές πριν την χρήση τους. Ωστόσο στην php δεν είναι απαραίτητο να οριστεί πρώτα η μεταβλητή και μετά να της αποδοθεί τιμή. Μπορούμε κατευθείαν αν της αποδώσουμε τιμή και κατ' αυτόν τον τρόπο έχει οριστεί ως μεταβλητή.

Παράδειγμα 1 ε τη χρήση μεταβλητών

```
<html>
<head> <title>Variable Example </title> </head>
<body>
<?php
$first_num = 12;
$second_num = 356;
$temp = $first_num;
$first_num = $second_num;
$second_num = $temp;
print ("first_num= $first_num <br>
second_num=$second_num" );
?>
</body>
</html>
```

Τύποι Δεδομένων της PHP

Ο τύπος δεδομένων μιας μεταβλητής δεν ορίζεται συνήθως από τον προγραμματιστή αλλά αποφασίζεται την ώρα της εκτέλεσης (runtime) από την PHP. Αν θέλουμε να κάνουμε μια μεταβλητή να μετατραπεί σε ένα συγκεκριμένο τύπο, μπορούμε να **μετατρέψουμε (cast)** τη μεταβλητή ή να χρησιμοποιήσουμε τη συνάρτηση `settype()` σε αυτή.

Πρέπει να έχουμε υπόψη μας ότι μια μεταβλητή μπορεί να συμπεριφερθεί διαφορετικά σε συγκεκριμένες καταστάσεις, ανάλογα με το τι τύπο δεδομένων έχει εκείνη τη στιγμή.

Τύποι δεδομένων που υποστηρίζει η PHP

Τύπος	Παράδειγμα	Περιγραφή
Integer	8	Ακέραιοι αριθμοί
Double	4.254	Αριθμοί κινητής υποδιαστολής
String	"Goodmorning"	Μια ακολουθία χαρακτήρων
Boolean	true	Μια από τις δυο λογικές τιμές (true ή false)

Έλεγχος Τύπου Δεδομένων

Μπορείτε να χρησιμοποιείτε τη συνάρτηση `gettype()` της PHP για να ελέγχετε τον τύπο δεδομένων οποιασδήποτε μεταβλητής. Εάν τοποθετήσετε το όνομα μιας μεταβλητής μέσα στις παρενθέσεις σε μια κλήση αυτής της συνάρτησης, η `gettype()` θα επιστρέψει ένα αλφαριθμητικό, το οποίο αντιπροσωπεύει τον τύπο δεδομένων της μεταβλητής. Ο παρακάτω κώδικας αναθέτει πέντε διαφορετικούς τύπους δεδομένων σε μια μεταβλητή και την ελέγχει με τη συνάρτηση `gettype()` κάθε φορά.

```
<HTML>
<HEAD>
<TITLE>Έλεγχος του τύπου δεδομένων μιας μεταβλητής</TITLE>
</HEAD>
<BODY>
<?php
$testing; //Δήλωση χωρίς ανάθεση
echo gettype($testing); //null
echo "<BR>";
$testing=5;
echo gettype($testing); //integer
echo "<BR>";
$testing="five";
echo gettype($testing); // string
echo "<BR>";
$testing=5.0;
echo gettype($testing); // double
echo "<BR>";
$testing=true;
echo gettype($testing); //boolean
echo "<BR>";
?>
</BODY>
</HTML>
```

Αλλαγή Τύπου Δεδομένων

Η PHP διαθέτει τη συνάρτηση `settype()` για την αλλαγή του τύπου δεδομένων μιας μεταβλητής. Για να χρησιμοποιήσετε τη συνάρτηση `settype()`, θα πρέπει να τοποθετήσετε μέσα στις παρενθέσεις την μεταβλητή της οποίας τον τύπο δεδομένων θέλετε να αλλάξετε, καθώς και τον τύπο δεδομένων στον οποίο θέλετε να αλλάξετε αυτή τη μεταβλητή, διαχωρίζοντάς τα με ένα κόμμα. Το παρακάτω πρόγραμμα μετατρέπει την τιμή 3.14 (μία τιμή τύπου `double`) σε κάθε έναν από τους τέσσερις τύπους δεδομένων.

```
<HTML>
<HEAD>
<TITLE>Αλλαγή του Τύπου Δεδομένων μιας Μεταβλητής</TITLE>
</HEAD>
<BODY>
<?php
$undecided=3.14;
echo gettype($undecided);
echo " is $undecided<BR>";
settype($undecided,'string');
echo gettype($undecided);
echo " is $undecided<BR>";
settype($undecided,'integer');
echo gettype($undecided);
echo " is $undecided<BR>";
settype($undecided,'double');
echo gettype($undecided);
echo " is $undecided<BR>";
settype($undecided,'boolean');
echo gettype($undecided);
echo " is $undecided<BR>";
?>
</BODY>
</HTML>
```

Ρητή Δήλωση Τύπου

Τοποθετώντας το όνομα ενός τύπου δεδομένων σε παρενθέσεις μπροστά από το όνομα μιας μεταβλητής, δημιουργείτε ένα αντίγραφο της τιμής αυτής της μεταβλητής, το οποίο μετατρέπεται στον προσδιοριζόμενο τύπο δεδομένων.

Η βασική διαφορά μεταξύ της συνάρτησης `settype()` και της ρητής δήλωσης τύπου (`cast`) είναι το γεγονός ότι η ρητή δήλωση τύπου παράγει ένα αντίγραφο, αφήνοντας άθικτη την αρχική μεταβλητή.

```
<HTML>
<HEAD>
<TITLE>Ρητή Δήλωση του Τύπου μιας Μεταβλητής</TITLE>
</HEAD>
<BODY>
<?php
$undecided=3.14;
$holder=(double) $undecided;
echo gettype($holder);
echo " is $holder<BR>";
//string
```

```

$holder=(string) $undecided;
echo gettype($holder);
echo " is $holder<BR>";
//integer
$holder=(integer) $undecided;
echo gettype($holder);
echo " is $holder<BR>";
//double
$holder=(double) $undecided;
echo gettype($holder);
echo " is $holder<BR>";
//boolean
$holder=(boolean) $undecided;
echo gettype($holder);
echo " is $holder<BR>";
echo "original variable type: ";
echo gettype($undecided);
?>
</BODY>
</HTML>

```

Τελεστές

Τελεστής Εκχώρησης

Ο τελεστής εκχώρησης τιμής αντιπροσωπεύεται από τον χαρακτήρα `=`. Ο τελεστής εκχώρησης τιμής παίρνει την τιμή του δεξιού τελεστέου του και την εκχωρεί στον αριστερό τελεστέο του:

```
$name="George"
```

Αριθμητικοί Τελεστές

Οι αριθμητικοί τελεστές λειτουργούν με τον γνωστό τρόπο και εκτελούν αριθμητικές πράξεις.

Τελεστής	Όνομα	Παράδειγμα	Αποτέλεσμα
+	Πρόσθεση	7+3	10
-	Αφαίρεση	7-3	4
*	Πολλαπλασιασμός	7*3	21
/	Διαίρεση	7/3	2.333333333
%	Ακέραιο υπόλοιπο	7%3	1

Ο Τελεστής Συνένωσης

Ο τελεστής συνένωσης (concatenation) αντιπροσωπεύεται από τον χαρακτήρα της τελείας. Αντιμετωπίζοντας και τους δυο τελεστέους του σαν αλφαριθμητικά, προσαρτά τον δεξιό τελεστέο στο τέλος του αριστερού. Συνεπώς, ο κώδικας

```
"hello"." world"
```

επιστρέφει το αποτέλεσμα

```
"hello world"
```

Σημειώστε ότι το κενό μεταξύ των λέξεων εμφανίζεται επειδή υπάρχει ένα κενό διάστημα στην αρχή του δεύτερου τελεστέου.

Σύνθετοι τελεστές εκχώρησης τιμής

Αν και στην πραγματικότητα υπάρχει μόνο ένας τελεστής εκχώρησης τιμής, η PHP υποστηρίζει ορισμένους σύνθετους τελεστές, οι οποίοι μεταβάλλουν τον αριστερό τελεστέο τους και επιστρέφουν ένα αποτέλεσμα. Οι σύνθετοι τελεστές εκχώρησης είναι μια συντόμευση και σας εξοικονομούν κόπο και χρόνο, επειδή δεν χρειάζεται να χρησιμοποιείτε ξεχωριστά δυο τελεστές. Για παράδειγμα:

```
$x=6;
$x=$x+3; // Η x ισούται με 9
το παραπάνω μπορεί να γραφεί ως εξής:
$x+=3; // Η x ισούται με 9
```

Παρακάτω παρουσιάζονται οι Σύνθετοι Τελεστές Εκχώρησης

Τελεστής	Παράδειγμα	Αποτέλεσμα
+=	<code>\$x+=3</code>	<code>\$x=\$x+3</code>
-=	<code>\$x-=3</code>	<code>\$x=\$x-3</code>
=	<code>\$x=3</code>	<code>\$x=\$x*3</code>
/=	<code>\$x/=3</code>	<code>\$x=\$x/3</code>
%=	<code>\$x%=3</code>	<code>\$x=\$x%3</code>
.=	<code>\$x.="test"</code>	<code>\$x=\$x."test"</code>

Ερωτήσεις

1. Ποια από τα ακόλουθα ονόματα μεταβλητών δεν είναι έγκυρο;

```
$this_is_a_value
$__test__
$metritis
$the counter
$my-name
$count
$2test
```

2. Ποιο είναι το αποτέλεσμα του ακόλουθου αποσπάσματος;

```
$test_val=6.5466;
settype($test_val, "integer");
echo $test_val;
```

3. Ποιο θα είναι το αποτέλεσμα της ακόλουθης εντολής;

```
echo gettype("8");
```

Δομές Ελέγχου

Η ροή εκτέλεσης του κώδικα των **scripts** που δημιουργήσαμε μέχρι τώρα ακολουθούσε μόνο μια κατεύθυνση. Κάθε φορά που εκτελείται κάποιο από αυτά τα **script** εκτελούνται οι ίδιες ακριβώς εντολές, με την ίδια ακριβώς σειρά. Αυτό δεν παρέχει όμως μεγάλη ευελιξία. Σ' αυτό το μάθημα θα εξετάσουμε ορισμένες δομές οι οποίες δίνουν στα **scripts** τη δυνατότητα να προσαρμόζονται ανάλογα με τις συνθήκες που ισχύουν ανά πάσα στιγμή.

if

Η δομή **if** είναι ένα από τα πιο σημαντικά χαρακτηριστικά σε πολλές γλώσσες, συμπεριλαμβανομένης και της PHP. Επιτρέπει την υπο συνθήκη εκτέλεση κομματιών κώδικα.

```
if (expr)
    statement
```

Όπως περιγράφηκε στο τμήμα σχετικά με τις εκφράσεις, η **expr** υπολογίζεται στην Boolean τιμή της. Αν η **expr** είναι **TRUE**, η PHP θα εκτελέσει τη *δήλωση*, και αν είναι **FALSE** - θα την αγνοήσει.

Το ακόλουθο παράδειγμα θα εμφάνιζε ότι το **a** είναι μεγαλύτερο από το **b** αν όντως το **\$a** είναι μεγαλύτερο από το **\$b**:

```
<?php
if ($a > $b)
    print "a is bigger than b";
?>
```

Συχνά θα θέλετε να εκτελείτε περισσότερες από μία δηλώσεις σε μια υποθετική συνθήκη. Φυσικά, δεν χρειάζεται να εμπερικλείετε κάθε δήλωση μέσα σε μια **if** δομή. Αντιθέτως, μπορείτε να συμπεριλάβετε αρκετές δηλώσεις σε ένα statement group. Για παράδειγμα, αυτός ο κώδικας θα εμφάνιζε **a is bigger than b** αν το **\$a** είναι μεγαλύτερο από το **\$b**, και τότε θα ανέθετε την τιμή του **\$a** στη μεταβλητή **\$b**:

```
<?php
if ($a > $b) {
    print "a is bigger than b";
    $b = $a;
}
?>
```

Οι δηλώσεις με **If** μπορούν να εμφωλευτούν απεριόριστα μέσα σε άλλες δηλώσεις **if**, κάτι το οποίο σας δίνει μεγάλη ευελιξία για την υπό συνθήκη εκτέλεση πολλών μερών του προγράμματός σας.

else

Συχνά θα θέλετε να εκτελέσετε μια δήλωση αν πληρείται μια συγκεκριμένη συνθήκη, και μια διαφορετική δήλωση αν αυτό δε συμβαίνει. Γι' αυτό το λόγο χρησιμοποιείται το **else**. Το **else** είναι επέκταση μιας δήλωσης **if** και εκτελεί μια δήλωση στην περίπτωση που η έκφραση στη δήλωση **if** είναι **FALSE**. Για παράδειγμα, ο ακόλουθος κώδικας θα εμφάνιζε **a is bigger than b** αν το **\$a** είναι μεγαλύτερο από το **\$b**, και **a is NOT bigger than b** στην αντίθετη περίπτωση:

```
<?php
if ($a > $b) {
    print "a is bigger than b";
} else {
    print "a is NOT bigger than b";
}
?>
```

elseif

Η εντολή *elseif*, όπως λέει και το όνομα της, είναι ένας συνδυασμός των *if* και *else*. Όπως το και *else*, έχει ως επέκταση μία *if* έκφραση με σκοπό να εκτελέσει μια διαφορετική έκφραση σε περίπτωση που η αρχική *if* συνθήκη πάρει την τιμή **FALSE**. Παρόλαυτα, σε αντίθεση με το *else*, θα εκτελέσει αυτή την εναλλακτική έκφραση μόνο αν η *elseif* υποθετική συνθήκη πάρει την τιμή **TRUE**. Για παράδειγμα, το ακόλουθο κομμάτι κώδικα θα εμφανίσει **a is bigger than b, a equal to b or a is smaller than b**:

```
<?php
if ($a > $b) {
    print "a is bigger than b";
} elseif ($a == $b) {
    print "a is equal to b";
} else {
    print "a is smaller than b";
}
?>
```

Μπορούν να υπάρχουν πολλά *elseif* μέσα στην ίδια έκφραση *if*. Η πρώτη *elseif* έκφραση (αν υπάρχει) που θα πάρει την τιμή **TRUE** θα είναι και αυτή που θα εκτελεστεί. Στην PHP, μπορείτε επίσης να γράψετε 'else if' (σε δυο λέξεις) και η συμπεριφορά να είναι όμοια με αυτή του 'elseif' (μία λέξη). Η συντακτική έννοια είναι ελαφρώς διαφορετική αλλά το τελικό αποτέλεσμα είναι ότι και οι δυο εκφράσεις θα καταλήξουν στην ίδια ακριβώς συμπεριφορά.

Η έκφραση *elseif* εκτελείται μόνο αν η προηγούμενη έκφραση *if* και οποιοσδήποτε προηγούμενες εκφράσεις *elseif* έχουν πάρει την τιμή **FALSE**, και η τρέχουσα έκφραση *elseif* πάρει την τιμή **TRUE**.

Τελεστές Σύγκρισης

Τελεστής	Όνομα	Επιστρέφει αληθές εάν...
==	Ισοδυναμία	Ο αριστερός τελεστής είναι ισοδύναμος με τον δεξιό
!=	Μη-ισοδυναμία	Ο αριστερός τελεστής δεν είναι ισοδύναμος με τον δεξιό
===	Ταυτοποίηση	Ο αριστερός τελεστής είναι ισοδύναμος με Τον δεξιό και έχουν τον ίδιο τύπο δεδομένων
>	Μεγαλύτερο	Ο αριστερός τελεστής είναι μεγαλύτερος από τον δεξιό
>=	Μεγαλύτερο ή ίσο	Ο αριστερός τελεστής είναι μεγαλύτερος ή ίσος από τον δεξιό
<	Μικρότερο	Ο αριστερός τελεστής είναι μικρότερος από Τον δεξιό
<=	Μικρότερο ή ίσο	Ο αριστερός τελεστής είναι μικρότερος ή ίσος από τον δεξιό

Λογικοί Τελεστές

Τελεστής	Όνομα	Επιστρέφει αληθές εάν...
	Or (διάζευξη)	Ο αριστερός τελεστής ή ο δεξιός είναι true
or	Or (διάζευξη)	Ο αριστερός τελεστής ή ο δεξιός είναι true
xor	Xor (Αποκλειστική διάζευξη)	Ο αριστερός ή ο δεξιός τελεστής είναι true, αλλά όχι ταυτόχρονα και οι δύο
&&	And (σύζευξη)	Ο αριστερός και ο δεξιός τελεστής είναι true
and	And (σύζευξη)	Ο αριστερός και ο δεξιός τελεστής είναι true
!	Not (άρνηση)	Ο ένας και μοναδικός τελεστής δεν είναι true

ΑΣΚΗΣΕΙΣ

1. Δίνονται δύο αριθμοί. Ο $\alpha = 2$ και $\beta = 4$. Με μια δομή επιλογής να ελέγξετε προγραμματιστικά ποιος είναι ο μεγαλύτερος και να τυπώσετε ένα αντίστοιχο μήνυμα.

2. Να γραφτεί script το οποίο ανάλογα με το βαθμό που θα δίνετε σε μια μεταβλητή θα εμφανίζει και το ανάλογο μήνυμα.

A → «Άριστα»

B → «Αρκετά Καλά»

Γ → «Όχι Άσχημα»

Δ → «Μόλις που τα κατάφερες»

E → «Καλοκαιρινό Φροντιστήριο»

Αν δεν δοθεί κάποιος από τους παραπάνω βαθμούς να εμφανίζει το μήνυμα «Αυτός δεν είναι βαθμός».

3. Να γραφτεί script που να βρίσκει αν ένας ακέραιος αριθμός είναι μονός ή ζυγός.

4. Να γραφτεί script που να βρίσκει τον μεγαλύτερο από τρεις αριθμούς a, b, c. Χρειάζεται να ορίσουμε μια βοηθητική μεταβλητή max, όπου εκεί θα καταχωρούμε προσωρινά τον μεγαλύτερο αριθμό.

5. Σε έναν αυτόματο πωλητή αναψυκτικών υπάρχουν τα προϊόντα 1. Πορτοκαλάδα με 0.7€, 2. Λεμονάδα 0.5€, 3. Ανθρακούχο Νερό 0.4€, 4. Χωνευτικό ποτό 0.8€. Όταν επιλέξεις ένα από τυπώνεται ένα μήνυμα με την αντίστοιχη τιμή του προϊόντος. Να φτιάξετε έναν script που να τυπώνει το ποσό και το προϊόν που επιλέξαμε αν κάναμε την επιλογή 3.

6. Μία εταιρεία ταχυδρομικών υπηρεσιών εφαρμόζει για τα έξοδα αποστολής ταχυδρομικών επιστολών εσωτερικού και εξωτερικού, χρέωση σύμφωνα με τον παρακάτω πίνακα:

Βάρος επιστολής σε γραμμάρια	Χρέωση εσωτερικού σε Ευρώ	Χρέωση εξωτερικού σε Ευρώ
από 0 έως και 500	2,0	4,8
από 500 έως και 1000	3,5	7,2
από 1000 έως και 2000	4,6	11,5

Για παράδειγμα τα έξοδα αποστολής μιας επιστολής βάρους 800 γραμμαρίων και προορισμού εσωτερικού είναι 3,5 Ευρώ. Να γράψετε αλγόριθμο ο οποίος:

A. Να διαβάζει το βάρος της επιστολής.

B. Να διαβάζει τον προορισμό της επιστολής. Η τιμή "ΕΣ" δηλώνει προορισμό εσωτερικού και η τιμή "ΕΞ" δηλώνει προορισμό εξωτερικού.

Γ. Να υπολογίζει τα έξοδα αποστολής ανάλογα με τον προορισμό και το βάρος της επιστολής.

Δ. Να εκτυπώνει τα έξοδα αποστολής.

Παρατήρηση. Θεωρείστε ότι ο αλγόριθμος δέχεται τιμές για το βάρος μεταξύ του 0 και του 2000 και για τον προορισμό μόνο τις τιμές "ΕΣ" και "ΕΞ".

Η εντολή switch

Η εντολή **switch** εξετάζει μια μόνο συνθήκη και εκτελεί διαφορετικά τμήματα κώδικα ανάλογα με το αποτέλεσμα αυτού του ελέγχου.

```
switch (expression) {  
    case value1:  
        statements;  
        break;  
  
    case value2:  
        statements;  
        break;  
  
    default:  
        statements;
```

Εάν βρεθεί μια ισότητα, εκτελείται ο κώδικας που ακολουθεί μετά από την συγκεκριμένη εντολή **case**. Η εντολή **break** τερματίζει ολοκληρωτικά την εκτέλεση της **switch**. Εάν παραληφθεί, εξετάζεται η έκφραση ελέγχου της επόμενης **case**. Εάν η ροή εκτέλεσης φτάσει την προαιρετική εντολή **default**, εκτελείται ο κώδικας της.

ΠΡΟΣΟΧΗ! Χωρίς την **break**, η ροή εκτέλεσης του προγράμματος θα συνεχίσει στην επόμενη **case** και τελικά θα φτάσει στην **default**. Στις περισσότερες περιπτώσεις, αυτή δεν είναι η επιθυμητή συμπεριφορά.

ΒΡΟΧΟΙ

Η εντολή while

Η δομή **while** είναι η απλούστερη μορφή επανάληψης στην PHP.

```
while (έκφραση) {  
    εντολές  
}
```

Για όσο χρόνο η έκφραση της **while** είναι **true**, εκτελείται το τμήμα κώδικα κατ' επανάληψη. Η τιμή της έκφρασης ελέγχεται κάθε φορά στην αρχή του **loop**, έτσι ώστε ακόμη και αν αυτή η τιμή αλλάξει κατά τη διάρκεια της εκτέλεσης των εμφωλευμένων συνθηκών, η εκτέλεση δε θα σταματήσει μέχρι το τέλος της επανάληψης (κάθε φορά που η PHP εκτελεί τις εκφράσεις στο **loop** αποτελεί μια επανάληψη). Μερικές φορές, αν η έκφραση **while** πάρει

την τιμή **FALSE** από την αρχή, η εμφωλευμένη εντολή-έκφραση δε θα εκτελεστεί ούτε μια φορά.

Εναλλακτική σύνταξη της **while**

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

```
<?php  
/* example 1 */  
  
$i = 1;  
while ($i <= 10) {  
    print $i++;  
}  
  
/* example 2 */  
  
$i = 1;  
while ($i <= 10):  
    print $i;  
    $i++;  
endwhile;  
?>
```

Η εντολή **do...while**

Η εντολή **do...while** είναι παρόμοια με την **while** εκτός από το ότι η έκφραση αλήθειας ελέγχεται στο τέλος κάθε επανάληψης και όχι στην αρχή. Η κύρια διαφορά από τα κανονικά **while** loops είναι ότι η πρώτη επανάληψη ενός **do..while** loop εγγυάται την εκτέλεση του (η αλήθεια της έκφρασης ελέγχεται μόνο στο τέλος της επανάληψης).

```
do {  
    εντολές  
while (έκφραση);  
  
<?php  
$i = 0;  
do {  
    print $i;  
} while ($i > 0);  
?>
```

ΑΣΚΗΣΕΙΣ

1. Να γίνει script στην PHP όπου θα δίνουμε σε μια μεταβλητή με το όνομα **lang** μια από τις τιμές **fr, es, en, de, gr**. Ανάλογα με την τιμή που θα έχει η μεταβλητή θα εκτυπώνεται και ανάλογο μήνυμα για την κάθε γλώσσα (Γαλλικά, Ισπανικά, Αγγλικά, Γερμανικά, Ελληνικά). Σε περίπτωση που η μεταβλητή δεν έχει καμιά από τις τιμές που αναφέρθηκαν θα εμφανίζεται το μήνυμα «Άγνωστη Γλώσσα!»
2. Με τη χρήση της **while** να γίνει script που να υπολογίζει την προπαίδεια του 7.
3. Να γράψετε ένα πρόγραμμα που να τυπώνει τον πίνακα της προπαίδειας όπως στο παρακάτω παράδειγμα:

1.	1	2	3	4	5	6	7	8	9	10
2.	2	4	6	8	10	12	14	16	18	20
3.	3	6	9	12	15	18	21	24	27	30
4.	4	8	12	16	20	24	28	32	36	40
5.	5	10	15	20	25	30	35	40	45	50
6.	6	12	18	24	30	36	42	48	54	60
7.	7	14	21	28	35	42	49	56	63	70
8.	8	16	24	32	40	48	56	64	72	80
9.	9	18	27	36	45	54	63	72	81	90
10.	10	20	30	40	50	60	70	80	90	100

Για να μπορέσουν οι αριθμοί να εμφανιστούν σε στήλες μπορείτε να χρησιμοποιήσετε στην printf τον κωδικό εξόδου %4d (εμφάνιση του αριθμού σε χώρο 4 χαρακτήρων) αντί του κωδικού %d.

4. Να γραφτεί script όπου με σε ένα πίνακα να εμφανίζει την τιμή ενός προϊόντων πολλαπλασιαζόμενη επί την ποσότητα.

Ποσότητα	Τιμή
10	50
20	100
30	150
40	200
50	250
60	300
70	350
80	400
90	450
100	500

5. Να γράψετε ένα πρόγραμμα που να εισάγετε κείμενο σε μια φόρμα. Οι χαρακτήρες που διαβάζει να εκτυπώνονται σύμφωνα με το διεθνές φωνητικό αλφάβητο: Alpha Bravo Charlie Delta Echo Foxtrot Golf Hotel India Juliet Kilo Lima Mike November Oscar Papa Quebec Romeo Sierra Tango Uniform Victor Whiskey Xray Yankee Zulu. Τα ψηφία 1-9 να εκτυπώνονται με επαναλαμβανόμενα αστερίκια (*) αντίστοιχα με το ψηφίο. Χαρακτήρες για τους οποίους δεν υπάρχει αντιστοιχία να εκτυπώνονται ως έχουν.

Παράδειγμα:

Είσοδος:

Hello 2 you!.

ΑΠΟΤΕΛΕΣΜΑ:

Hotel Echo Lima Lima Oscar ** Yankee Oscar Uniform!

for

Τα *for* loops είναι τα πιο περίπλοκα loops στην PHP. Η σύνταξη ενός *for* loop είναι:

for (expr1; expr2; expr3) statement

Η πρώτη έκφραση (*expr1*) εκτελείται χωρίς να λάβουμε υπόψη κάποια συνθήκη στην αρχή του loop.

Στην αρχή κάθε επανάληψης, η *expr2* υπολογίζεται. Αν πάρει την τιμή **TRUE**, το loop συνεχίζει και οι εμφωλευμένες εντολές εκτελούνται. Αν πάρει την τιμή **FALSE**, η εκτέλεση του loop σταματά.

Στο τέλος κάθε επανάληψης, υπολογίζεται η τιμή της *expr3*.

```
<?php
/* Παράδειγμα 1 */

for ($i = 1; $i <= 10; $i++) {
    print $i;
}

/* Παράδειγμα 2 */

for ($i = 1; ; $i++) {
    if ($i > 10) {
        break;
    }
    print $i;
}

/* Παράδειγμα 3 */

$i = 1;
for (;;) {
    if ($i > 10) {
        break;
    }
    print $i;
    $i++;
}

/* Παράδειγμα 4 */

for ($i = 1; $i <= 10; print $i, $i++);
?>
```

for (expr1; expr2; expr3) statement

Ασκήσεις

1. Να γραφεί **script** που να κάνει τη διαίρεση του 2000 με το 1-10 και να εμφανίζει το πηλίκο με το εξής μήνυμα: Του πηλίκο του 2000 διά του είναι το
2. Γράψτε την προπαίδεια με τη χρήση της εντολής **for**.

Χρήση Φορμών για εισαγωγή στοιχείων

Μέχρι τώρα στα **scripts** που έχουμε δει δεν δίνεται η δυνατότητα στο χρήστη-επισκέπτη της ιστοσελίδας να εισάγει τα δικά του δεδομένα και πληροφορίες. Στην **HTML** η εισαγωγή στοιχείων επιτυγχάνεται με τη χρήση φορμών και είναι το βασικό μέσο με το οποίο στέλνονται οι πληροφορίες από τον χρήστη στον **server**.

Για την ώρα θα κρατήσουμε τον κώδικα **HTML** ξεχωριστά από τον κώδικα της **PHP**. Ο παρακάτω κώδικας δημιουργεί μια απλή φόρμα **HTML**.

Form.php

```
<HTML>
<HEAD>
<TITLE>Φόρμα για την εισαγωγή στοιχείων</TITLE>
</HEAD>
<BODY>
<FORM ACTION="test.php" METHOD="POST">
<P><STRONG>Name:</STRONG><BR>
<INPUT TYPE="TEXT" NAME="USER"></P>
<P><STRONG>Address:</STRONG><BR>
<TEXTAREA NAME="ADDRESS" ROWS="5" COLS="40"></TEXTAREA></P>
<P><INPUT TYPE="SUBMIT" VALUE="SEND"></P>
</FORM>
</BODY>
</HTML>
```

Test.php

```
<HTML>
<HEAD>
<TITLE>Εμφάνιση στοιχείων από φόρμα</TITLE>
</HEAD>
<BODY>
<?php
$user=$_POST[USER];
$address=$_POST[ADDRESS];
echo "<p>Welcome <b>$user</b></p>";
echo "<p>Your address is:<br><b>$address</b></p>";
?>
</BODY>
</HTML>
```

Self.php

```
<HTML>
<HEAD>
<TITLE>Φόρμα </TITLE>
</HEAD>
<BODY>
<FORM ACTION="<?php echo $_SERVER[PHP_SELF] ?>" METHOD="POST">
<P><STRONG>Name:</STRONG><BR>
<INPUT TYPE="TEXT" NAME="USER"></P>
<P><STRONG>Address:</STRONG><BR>
<TEXTAREA NAME="ADDRESS" ROWS="5" COLS="40"></TEXTAREA></P>
<P><INPUT TYPE="SUBMIT" VALUE="SEND"></P>
</FORM>
<?php
```

```
$user=$_POST[USER];  
$address=$_POST[ADDRESS];  
if ($user){  
echo "<p>Welcome <b>$user</b></p>";  
echo "<p>Your address is:<br><b>$address</b></p>";  
}  
?>  
</BODY>  
</HTML>
```

Πίνακες (Arrays)

Οι πίνακες χρησιμοποιούνται για την αποθήκευση και οργάνωση των δεδομένων. Οι απλές μεταβλητές μπορούν να αποθηκεύουν μόνο μια τιμή κάθε φορά. Για παράδειγμα η μεταβλητή \$name μπορεί να περιέχει μόνο μια τιμή (Γιώργος, Μαρία, Νίκος κλπ). Δεν μπορεί να χρησιμοποιηθεί για να αποθηκευτεί μια λίστα τιμών. Σε αντίθεση λοιπόν, οι πίνακες είναι ειδικοί τύποι μεταβλητών οι οποίοι μας δίνουν τη δυνατότητα να αποθηκεύουμε όσες τιμές θέλουμε. Οι πίνακες οργανώνονται βάσει δείκτη, πράγμα το οποίο σημαίνει ότι το κάθε στοιχείο τους αποτελείται από ένα κλειδί (key) και μια τιμή (value). Το κλειδί είναι η θέση του στοιχείου στον πίνακα, ξεκινώντας από το 0. Η τιμή είναι η οποιαδήποτε τιμή έχετε συσχετίσει με κάποια θέση του πίνακα και μπορεί να είναι ένα αλφαριθμητικό, ένας ακέραιος, ένας πραγματικός κλπ.

Δημιουργία Πίνακα

Μπορείτε να δημιουργήσετε ένα πίνακα είτε χρησιμοποιώντας τη συνάρτηση array(), είτε τον τελεστή πινάκων, [].

- Η συνάρτηση array() χρησιμοποιείται συνήθως όταν θέλετε να δημιουργήσετε μια νέα διάταξη με περισσότερα από ένα στοιχεία ταυτόχρονα.
- Ο τελεστής [] χρησιμοποιείται όταν θέλετε να δημιουργήσετε ένα πίνακα με ένα μόνο στοιχείο (προς το παρόν), ή όταν θέλετε να προσθέσετε ένα στοιχείο σε έναν υπάρχον πίνακα.

```
$cars = array("Toyota", "Mercedes", "BMW", "Ferrari", "Renault");
```

```
$cars[] = "Toyota";
$cars[] = "Mercedes";
$cars[] = "BMW";
$cars[] = "Ferrari";
$cars[] = "Renault";
```

```
$cars[0] = "Toyota";
$cars[1] = "Mercedes";
$cars[2] = "BMW";
$cars[3] = "Ferrari";
$cars[4] = "Renault";
```

Άσκησης

1. Να δημιουργήσετε ένα πίνακα που να περιέχει τα παρακάτω στοιχεία ('Perl', 'PHP', 'C', 'Java', 'Pascal', 'Cobol', 'Visual Basic') και στη συνέχεια να τα εμφανίσετε στην οθόνη.
2. Να δημιουργήσετε ένα πίνακα που να περιέχει τα παρακάτω στοιχεία (8, 12, 21, 3, 15, 6, 7, 32, 9) και να εμφανίσετε τα στοιχεία του στην οθόνη.
3. Βρείτε το μεγαλύτερο και το μικρότερο αριθμό από τα στοιχεία του πίνακα της άσκησης 2.
4. Ταξινομήσετε τον πίνακα της άσκησης 2 και εμφανίσετε τα στοιχεία του.
5. Να υπολογίσετε την προπαίδεια του 8 και να αποθηκεύσετε τα αποτελέσματα σε έναν πίνακα. Ακολουθώντας αφού τελειώσετε, τυπώστε τα αποτελέσματα αυτά με την αντίθετη σειρά.

6. Για το πρόβλημα, να αποθηκεύεται τις ενδιάμεσες τιμές σε έναν μονοδιάστατο πίνακα αφού υπολογίσετε το $N!$ να εμφανίσετε τα ενδιάμεσα και το τελικό αποτέλεσμα, δηλαδή τα περιεχόμενα του πίνακα.
7. Με την χρήση της αναδρομής να φτιάξετε ένα php script το οποίο θα υπολογίζει το $N!$ ($1*2*3*4*5*.....*N$). Το αποτέλεσμα να επιστρέφεται και να εκτυπώνεται στον χρήστη με κατάλληλο μήνυμα.
8. Για το παραπάνω πρόβλημα να εισάγεται την τιμή του $N!$ από τον browser με την χρήση των φορμών HTML.

Η συνάρτηση date()

string date (string format [, int timestamp])

Επιστρέφει ένα string σχηματισμένο ανάλογα με το δοθέν format string, χρησιμοποιώντας τη δοθείσα integer παράμετρο *timestamp* ή την τοπική ώρα ένα δεν έχει δοθεί timestamp.

Το έγκυρο πεδίο τιμών για ένα timestamp είναι τυπικά από Fri, 13 Dec 1901 20:45:54 GMT έως Tue, 19 Jan 2038 03:14:07 GMT. (Αυτές είναι οι ημερομηνίες που αντιστοιχούν στην ελάχιστη και μέγιστη τιμή, αντίστοιχα, ενός προσημασμένου integer των 32-bit). Στα windows το πεδίο περιορίζεται από την 01-01-1970 έως τις 19-01-2038.

Για να δημιουργήσετε ένα timestamp από μία ημερομηνία που αναπαρίσταται από string, μπορείτε να χρησιμοποιήσετε τη συνάρτηση [strtotime\(\)](#). Επιπλέον, μερικές βάσεις δεδομένων έχουν συναρτήσεις για να πετατρέπουν τις ημερομηνίες σε timestamps (όπως η συνάρτηση UNIX_TIMESTAMP της MySQL).

Οι ακόλουθοι χαρακτήρες αναγνωρίζονται από το format string:

- a - "am" ή "pm"
- A - "AM" ή "PM"
- B - Swatch Internet time
- d - ημέρα του μήνα, 2 ψηφία με ηγετικά μηδενικά ("01" έως "31").
- D - ημέρα της εβδομάδας, παρατίθεται με 3 γράμματα. π.χ. "Fri"
- F - μήνας, παρατίθεται το πλήρες όνομα. π.χ. "January"
- g - ώρα, σε 12ωρη μορφή χωρίς ηγετικά μηδενικά ("1" έως "12").
- G - ώρα, 24ωρη μορφή χωρίς ηγετικά μηδενικά ("0" έως "23").
- h - ώρα, 12ωρη μορφή ("01" έως "12").
- H - ώρα, 24ωρη μορφή ("00" έως "23").
- i - λεπτά ("00" έως "59").
- I (κεφαλαίο i) - "1" εάν είναι Θερινή Ώρα, αλλιώς "0".
- j - ημέρα του μήνα χωρίς ηγετικά μηδενικά ("1" έως "31").
- l (μικρό 'l') - ημέρα της εβδομάδας, παρατίθεται το πλήρες όνομα. π.χ. "Friday"
- L - boolean για το αν είναι δίσεκτο έτος ("0" ή "1").
- m - μήνας ("01" έως "12").
- M - μήνας, παρατίθεται με 3 γράμματα. π.χ. "Jan"
- n - μήνας χωρίς ηγετικά μηδενικά ("1" έως "12").
- O - διαφορά με την Greenwich time σε ώρες. π.χ. "+0200"
- r - ημερομηνία σε μορφή RFC 822; π.χ. "Thu, 21 Dec 2000 16:01:07 +0200" (προστέθηκε στην PHP 4.0.4)
- S - δευτερόλεπτα ("00" έως "59")
- S - το αγγλικό τακτικό επίθεμα για την ημέρα του μήνα, 2 χαρακτήρων ("st", "nd", "rd" ή "th").
- t - πλήθος ημερών στο δοσμένο μήνα ("28" έως "31").
- T - Η ρύθμιση ζώνης ώρας της μηχανής; π.χ. "EST" ή "MDT"
- U - δευτερόλεπτα από την Unix Epoch (January 1 1970 00:00:00 GMT)
- W - ημέρα της εβδομάδας, αριθμητικά ("0" (Κυριακή) to "6" (Σάββατο)).
- W - πλήθος εβδομάδων του έτος κατά το ISO-8601, οι εβδομάδες αρχίζουν Δευτέρα (προστέθηκε στην PHP 4.1.0)
- Y - έτος, 4 ψηφία; π.χ. "1999"

- **Y** - έτος, 2 ψηφία; π.χ. "99"
- **Z** - ημέρα του έτους ("0" έως "365").
- **Z** - διαφορά της ζώνης ώρας σε δευτερόλεπτα ("-43200" έως "43200"). Η διαφορά για ζώνες ώρας δυτικά του UTC είναι πάντα αρνητική, και για αυτές ανατολικά του UTC πάντα θετική.

Παραδείγματα της date()

```
echo date("l dS of F Y h:i:s A");
echo "July 1, 2000 is on a " . date("l", mktime(0,0,0,7,1,2000));
```

Μπορείτε να αποτρέψετε τη λειτουργία ενός αναγνωρίσιμου χαρακτήρα στο format string με το να θέσετε ένα backslash πριν από αυτόν. Εάν ο χαρακτήρας με το backslash είναι ήδη μία ειδική ακολουθία, μπορεί να χρειαστεί να τοποθετήσετε ένα επιπλέον backslash.

Χαρακτήρες escape στην date()

```
echo date("l \\t\\h\\e jS"); // prints something like 'Saturday the 8th'
```

Μερικά επιπλέον παραδείγματα της **date()**. Πρατηρείστε ότι θα πρέπει να θέσετε backslash σε όλους τους χαρακτήρες, καθώς οποιοσδήποτε έχει μία ιδιαίτερη σημασία θα παράξει ανεπιθύμητα αποτελέσματα, και επειδή μπορεί να δοθεί νόημα και σε άλλους χαρακτήρες στις μελλοντικές εκδόσεις της PHP. Όταν κάνετε escape ένα χαρακτήρα, βεβαιωθείτε ότι χρησιμοποιείτε απλά εισαγωγικά για χαρακτήρες όπως ο \n, που θα γίνει νέα γραμμή.

date() Formatting

```
/* Today is March 10th, 2001, 5:16:18 pm */
$today = date("F j, Y, g:i a"); // March 10, 2001, 5:16 pm
$today = date("m.d.y"); // 03.10.01
$today = date("j, n, Y"); // 10, 3, 2001
$today = date("Ymd"); // 20010310
$today = date('h-i-s, j-m-y, it is w Day z '); // 05-16-17, 10-03-01, 1631 1618 6
Fripm01
$today = date("\i\t \i\s \t\h\e jS \d\ay."); // It is the 10th day.
$today = date("D M j G:i:s T Y"); // Sat Mar 10 15:16:08 MST 2001
$today = date('H:m:s \m \i\s\ \m\o\n\t\h'); // 17:03:17 m is month
$today = date("H:i:s"); // 17:16:17
```

Η συνάρτηση **strtotime()**

strtotime -- Μετατρέπει σχεδόν οποιαδήποτε ημερομηνία ή ώρα που είναι σε μορφή Αγγλικού κειμένου σε ένα UNIX timestamp

int strtotime (string time [, int now])

Η συνάρτηση περιμένει να της δοθεί ένα string που περιέχει μία ημερομηνία στα Αγγλικά και θα προσπαθήσει να τη μετατρέψει σε ένα UNIX timestamp σχετικό με αυτό που δίνεται με την παράμετρο *now*, ή την τρέχουσα ώρα εάν δε δοθεί κανένα. Σε περίπτωση αποτυχίας, ένα -1 επιστρέφεται.

Παραδείγματα της **strtotime()**

```
<?
echo strtotime ("now"), "\n";
echo strtotime ("10 September 2000"), "\n";
echo strtotime (" +1 day"), "\n";
echo strtotime (" +1 week"), "\n";
echo strtotime (" +1 week 2 days 4 hours 2 seconds"), "\n";
echo strtotime ("next Thursday"), "\n";
echo strtotime ("last Monday"), "\n";
$test = strtotime ("2000/5/15");
$today = date("d.M.y", $test);
echo $today;
?>
```

Μεταφορά Αρχείων

Η PHP είναι ικανή για να λαμβάνει upload αρχείων από οποιοδήποτε συμβατό browser. Αυτό το χαρακτηριστικό επιτρέπει στους χρήστες να κάνουν upload τόσο κειμένου, όσο και binary αρχεία. Με τις συναρτήσεις αναγνώρισης και χειρισμού αρχείων της PHP, έχετε πλήρη έλεγχο στο ποιος επιτρέπεται να κάνει upload και τι θα γίνει με το αρχείο από τη στιγμή που έχει γίνει upload και μετά.

Μια οθόνη για upload αρχείου μπορεί να φτιαχτεί δημιουργώντας μια ειδική φόρμα η οποία μοιάζει κάπως έτσι:

Φόρμα για Upload Αρχείου

```
<HTML>
<HEAD>
<TITLE> Upload Files </TITLE>
</HEAD>
<BODY>
<form action="upload.php" method="POST" enctype="multipart/form-
data">
<input type="hidden" value="512000" name="MAX_FILE_SIZE" />
<p><strong>File to Upload:</strong><input type="file"
name="userfile" /></p>
<p><input type="submit" value="upload!" /></p>
</form>
</BODY>
</HTML>
```

Το MAX_FILE_SIZE κρυφό πεδίο (hidden) πρέπει να προηγείται του input πεδίου για το αρχείο και η τιμή του είναι το μέγιστο μέγεθος αρχείου που επιτρέπεται. Η τιμή είναι σε byte.

Σημείωση

Το MAX_FILE_SIZE είναι ενημερωτικό για τον browser. Είναι εύκολο να παρακαμφθεί αυτή η μέγιστη τιμή. Έτσι μην στηρίζετε ότι ο browser υπακούει την επιθυμία σας! Οι PHP-ρυθμίσεις όμως, για το μέγιστο μέγεθος (maximum-size), δεν μπορούν να ξεγελαστούν. Καλύτερα να προσθέτετε το MAX_FILE_SIZE ούτως ή άλλως γιατί προστατεύει τους χρήστες από τον κόπο να περιμένουν για ένα μεγάλο αρχείο να μεταφερθεί μόνο και μόνο για να μάθουν πως ήταν πολύ μεγάλο μετά.

Οι μεταβλητή που ορίζεται για αρχεία που έχουν γίνει upload είναι η autoglobal μεταβλητή [\\$ _FILES](#) και υπάρχει από την PHP 4.1.0. Το [\\$HTTP_POST_FILES](#) array υπάρχει από την PHP 4.0.0. Αυτοί οι πίνακες θα περιέχουν όλες τις πληροφορίες των αρχείων που έχουν γίνει upload. Η χρήση της [\\$_FILES](#) προτιμάται.

Τα περιεχόμενα της [\\$_FILES](#) από το παράδειγμα μας είναι ως ακολούθως. Σημειώστε πως αυτό υποθέτει τη χρήση του ονόματος αρχείου να είναι *userfile*, όπως χρησιμοποιείται στο παράδειγμα παραπάνω.

[\\$_FILES\['userfile'\]\['name'\]](#)

Το αρχικό όνομα του αρχείου στο μηχανήμα του client.

`$_FILES['userfile']['type']`

Το mime type του αρχείου, αν ο browser έχει δώσει αυτή τη πληροφορία. Ένα παράδειγμα θα ήταν *"image/gif"*.

`$_FILES['userfile']['size']`

Το μέγεθος, σε byte, του αρχείου που έχει γίνει upload.

`$_FILES['userfile']['tmp_name']`

Το προσωρινό όνομα του αρχείου στο οποίο έχει αποθηκευτεί το αρχείο που έχει γίνει upload στον server.

`$_FILES['userfile']['error']`

Ο κωδικός σφάλματος που σχετίζεται με αυτό το upload αρχείου. Το *['error']* έχει προστεθεί στην PHP 4.2.0

Προσοχή: Στις εκδόσεις της PHP πριν την 4.1.0 αυτό έχει όνομα *\$HTTP_POST_FILES* και δεν είναι μια autoglobal μεταβλητή όπως είναι η *\$_FILES*. Η PHP 3 δεν έχει υποστήριξη για την *\$HTTP_POST_FILES*.

Τα αρχεία προεπιλεγμένα θα αποθηκεύονται στον προσωρινό κατάλογο του server, εκτός και αν κάποια άλλη τοποθεσία δοθεί με το upload tmp dir directive στο *php.ini*. Ο προεπιλεγμένος κατάλογος μπορεί να αλλάχθει ορίζοντας την μεταβλητή περιβάλλοντος *TMPDIR* στο περιβάλλον στο οποίο εκτελείται η PHP. Ορίζοντας το με την putenv() μέσα από ένα PHP script δεν θα δουλέψει. Αυτή η μεταβλητή περιβάλλοντος μπορεί επιπλέον να χρησιμοποιηθεί για να σιγουρευτείτε πως και άλλες ενέργειες επιτελούνται στα αρχεία που έχουν γίνει upload, επιπλέον.

is_uploaded_file()

Πριν μετακινήσουμε το αρχείο από την προσωρινή του θέση στην θέση που καθορίζεται ελέγχουμε ότι το αρχείο υπάρχει. Η συνάρτηση *is_uploaded_file()* δέχεται σαν όρισμα τη διαδρομή ενός αρχείου και επιστρέφει *true* μόνο αν το εν λόγω αρχείο είναι έγκυρο.

move_uploaded_file()

Η συνάρτηση αυτή αντιγράφει ένα αρχείο από μια θέση σε μια άλλη. Απαιτεί σαν ορίσματα τη διαδρομή προς το πηγαίο αρχείο και τη διαδρομή προορισμού. Επιστρέφει *true* αν η μετακίνηση του αρχείου είναι επιτυχής και *false* αν το αρχείο δεν βρέθηκε ή δεν είναι έγκυρο.

Το παρακάτω παράδειγμα θα επεξεργαστεί ένα upload αρχείου που προέρχεται από μια φόρμα.

Επιτύρωση upload αρχείων

```
<?php

$uploadaddir = "C:/Program Files/Apache Group/Apache2/htdocs/upload";

if (move_uploaded_file($_FILES['userfile']['tmp_name'], $uploadaddir
. $_FILES['userfile']['name'])) {
    echo "File is valid, and was successfully uploaded.";
} else {
    echo "Possible file upload attack!";
}
?>
```

Το PHP script το οποίο λαμβάνει το αρχείο πρέπει να εκτελέσει οποιαδήποτε λογική είναι απαραίτητη για να αποφασίσει τι θα γίνει με τα αρχεία. Μπορείτε για παράδειγμα να χρησιμοποιήσετε την `$_FILES['userfile']['size']` μεταβλητή για να πετάξετε οποιαδήποτε αρχεία είναι είτε πολύ μικρά είτε πολύ μεγάλα. Μπορείτε να χρησιμοποιήσετε την `$_FILES['userfile']['type']` μεταβλητή για να πετάξετε οποιαδήποτε αρχεία δεν ταιριάζουν σε ορισμένα κριτήρια του τύπου του αρχείου. Από την PHP 4.2.0, μπορείτε να χρησιμοποιείτε την `$_FILES['userfile']['error']` και να προγραμματίζετε τη λογική σας σύμφωνα με τους κωδικούς σφαλμάτων. Οποιαδήποτε η λογική, θα πρέπει είτε να διαγράψετε το αρχείο από τον προσωρινό κατάλογο ή να το μετακινήσετε κάπου αλλού.

Το αρχείο θα διαγραφεί από τον προσωρινό κατάλογο στο τέλος του request αν δεν έχει μετακινηθεί ή διαγραφεί.

Σύνδεση PHP με MySQL

Για να δημιουργήσετε μία βάση δεδομένων, πρέπει πρώτα να συνδεθείτε με το διακομιστή της MySQL. Αυτό γίνεται με την ακόλουθη εντολή:

mysql_connect(servername,username,password);

Όπου **servername** αντιστοιχεί στη διεύθυνση του διακομιστή, **username** το όνομα του χρήστη που έχει δικαίωμα πρόσβασης σε αυτόν και **password** ο μυστικός κωδικός του χρήστη. Για παράδειγμα, αν η MySQL έχει εγκατασταθεί τοπικά στον υπολογιστή σας και ο administrator της MySQL σας έχει ανοίξει λογαριασμό με username "username" και password "password" τότε ο ακόλουθος κώδικας PHP σας συνδέει με την MySQL. Στην περίπτωση που η σύνδεση αποτύχει, τότε η PHP μέσω της εντολής εντολής **mysql_error()** επιστρέφει ένα μήνυμα λάθους.

```
<html>
<body>

<?php

$con = mysql_connect("localhost","username","password");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}

?>

</body>
</html>
```

Αφού έχετε συνδεθεί με το διακομιστή της βάσης μπορείτε να δημιουργήσετε μία νέα βάση δεδομένων με την εντολή:

mysql_query("CREATE DATABASE my_db",\$con);

Έτσι ο κώδικας σύνδεσης και δημιουργίας νέας βάσης γίνεται:

```
<html>
<body>

<?php

$con = mysql_connect("localhost","utest","qwerty!1");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
```



```
if (mysql_query("CREATE DATABASE my_db", $con))
{
    echo "Database created";
}
else
{
    echo "Error creating database: ".mysql_error();
}

mysql_close($con);

?>
```

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

Η εντολή **mysql_query()** της php είναι η πιο σημαντική από τις εντολές που αφορούν στην επικοινωνία με τη MySQL και χρησιμοποιείται όποτε θέλουμε να κάνουμε μία ερώτηση στη βάση. Ως είσοδο δέχεται ένα **SQL query** και στην έξοδο είτε επιστρέφει το αποτέλεσμα του query σε ένα πίνακα, είτε επιστρέφει ότι εκτελέστηκε επιτυχώς.

Η σύνδεση με μία βάση δεδομένων τερματίζει με την εντολή:

```
mysql_close(mysql_connect(servername,username,password));
```

Εισαγωγή δεδομένων σε πίνακα

Η εισαγωγή δεδομένων σε έναν πίνακα που έχουμε ήδη δημιουργήσει γίνεται με το ακόλουθο query της SQL:

```
INSERT INTO table_name (column1, column2, ..., columnN)
VALUES (value1, value2,... ,valueN)
```

Ο αριθμός των τιμών value1, value2, ... που εισάγουμε θα πρέπει να ίσος με το πλήθος των πεδίων του πίνακα.

Για παράδειγμα, αν θέλουμε να εισάγουμε τα δεδομένα της πρώτης γραμμής του πίνακα **phone_book** θα πρέπει να γράψουμε την ακόλουθη php εντολή:

```
<?php
mysql_query("
INSERT INTO phone_book (Id, LastName, FirstName, Address,
Age, Phone)
VALUES ('0', 'Αλεξάνδρου', 'Μιχαήλ', 'Παλαιολόγου 3,
Αλεξανδρούπολη', '30', '25510 99999')
")
?>
```

Επιλογή Δεδομένων από πίνακα

Η επιλογή δεδομένων από έναν ή περισσότερους πίνακες γίνεται με το ακόλουθο SQL query:

**SELECT (column1, column2, ..., columnN)
FROM (table1, table2,... ,tableM)**

Στο παραδειγμά μας, αν θέλουμε να επιλέξουμε τα δεδομένα από όλα τα πεδία του πίνακα *phone_book* τότε θα πρέπει να εισάγουμε το ακόλουθο κώδικα PHP:

```
<?php

// Select the database
mysql_select_db("my_db",
mysql_connect("localhost","username","password"));

$result = mysql_query("SELECT * FROM phone_book");

while($row = mysql_fetch_array($result))
{
echo $row['LastName'] . " " . $row['FirstName'] . " " .
$row['Address'] . " " . $row['Age'] . " " . $row['Phone'] . "<br>";
}

?>
```

ενώ αν θέλουμε να επιλέξουμε όλα τα άτομα που η ηλικία τους είναι πάνω από 30 ετών τότε κάνουμε την ακόλουθη ερώτηση:

```
<?php

// Select the database
mysql_select_db("my_db",
mysql_connect("localhost","username","password"));

$result = mysql_query("SELECT * FROM phone_book WHERE age>30");

while($row = mysql_fetch_array($result))
{
echo $row['LastName'] . " " . $row['FirstName'] . " " . $row['Age'] .
"<br>";
}

?>
```

Από τα παραπάνω παραδείγματα είναι εμφανές ότι η **php** εκτελώντας την εντολή **mysql_fetch()** δέχεται ως **input** ένα **sql** ερώτημα και επιστρέφει μία εγγραφή του πίνακα που απαντά στο ερώτημα σε μορφή πίνακα. Κάθε φορά που εκτελείται η **mysql_fetch_array()** η **php** ρωτά τη βάση και παίρνει την επόμενη εγγραφή μέχρι να μην υπάρχει άλλη.