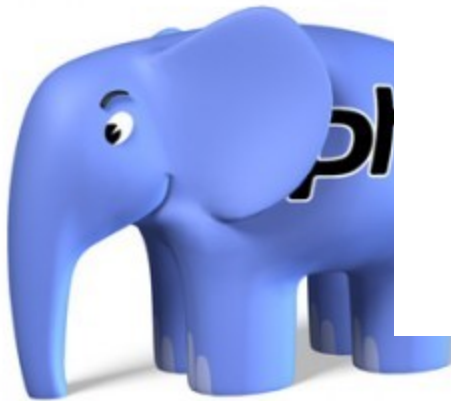


Server side scripting languages

Εισαγωγή στη PHP



Λεμεσού

17 – 2018

Άριστος Πασιάς

Παράδειγμα κώδικα PHP ...



```
public function setJpegQuality($number)
{
    if (!is_numeric($number) || $number < 0 || $number
        > 100) {
        throw new \Exception('JPG Quality
            number in the range 0-100.');
```



```
public function setPngCompression($number)
{
    if (!is_numeric($number) || $number < 0 || $number
        > 9) {
        throw new \Exception('PNG Compression n
```

Τι είναι η PHP



- Η PHP (PHP: Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού ανοικτού λογισμικού ειδικά σχεδιασμένη για δημιουργία σελίδων Web με δυναμικό περιεχόμενο.
- Ο κώδικας PHP ενσωματώνεται μέσα σε HTML σελίδες και εκτελείται από τον Server (που είναι αποθηκευμένες) κάθε φορά που ο χρήστης επισκέπτεται τις σελίδες αυτές.
- Η εκτέλεση ενός κώδικα PHP έχει ως αποτέλεσμα τη παραγωγή κώδικα HTML ο οποίος στέλνεται τελικά στο φυλλομετρητή του χρήστη.

Τι Χρειαζόμαστε για δημιουργήσουμε εφαρμογές σε PHP



Το όνομα ενός PHP αρχείου έχει επέκταση (extension) **.php** και τρέχει στον **Server** που είναι αποθηκευμένο. Εννοείται ότι στο server έχει ενεργοποιηθεί η υποστήριξη για την PHP και ότι όλα τα αρχεία που τελειώνουν στο .php αντιμετωπίζονται από την PHP. Εάν ο server σας υποστηρίζει PHP, τότε δεν χρειάζεται να κάνετε τίποτα. Απλά δημιουργήστε τα αρχεία .php με ένα απλό κειμενογράφο, τα βάζετε στον κατάλόγό σας στον ιστό και ο server θα τα αναλύσει αυτόματα για εσάς. Δεν χρειάζεται να εγκαταστήσετε επιπλέον εργαλεία. Σκεφτείτε αυτά τα αρχεία με δυνατότητα PHP ως απλά αρχεία HTML με μια εντελώς νέα οικογένεια μαγικών ετικετών που σας επιτρέπουν να κάνετε πολλά περισσότερα πράγματα.

Τι Χρειαζόμαστε για δημιουργήσουμε εφαρμογές σε PHP



Μπορεί όμως να θέλουμε να αναπτύξουμε μια εφαρμογή τοπικά. Είτε για να εξοικονομήσουμε πολύτιμο εύρος ζώνης, είτε για άλλο λόγο.

Σε αυτή την περίπτωση, θα πρέπει να εγκαταστήσουμε:

- Έναν Server ιστού, όπως «**Apache**»,
- «**PHP**»,
- Μια βάση δεδομένων, όπως «**MySQL**».
- Έναν απλό επεξεργαστή κειμένου π.χ. «**Notepad++**»

Τι Χρειαζόμαστε για δημιουργήσουμε εφαρμογές σε PHP



Μπορούμε επίσης να εντοπίσουμε κάποιο προ-ρυθμισμένο πακέτο (ανάλογα με το λειτουργικό μας σύστημα), το οποίο εγκαθιστά αυτόματα όλα τα απαραίτητα με μερικά μόνο κλικ.:

- [XAMPP](#) - [Apache](#), [MariaDB](#), [Perl](#) and [PHP](#) on Windows, macOS or Linux

Bitnami WAMP Stack	Windows
IZ-WAMP	Windows
Wamp.NET	Windows
WampServer	Windows
MAMP	Windows, macOS
AMPPS	Windows, macOS, Linux, BSD
Zend Server Community Edition	Windows, macOS, Linux, IBMi

Τι Χρειαζόμαστε για δημιουργήσουμε εφαρμογές σε PHP



Εναλλακτική λύση παρέχουν και οι on line πλατφόρμες ελέγχου κώδικα PHP σε περίπτωση που δεν θέλουμε να εγκαταστήσουμε λογισμικό.

Π.χ.

http://www.compileonline.com/execute_php_online.php

<http://sandbox.onlinephpfunctions.com/>

Ιστορία της PHP

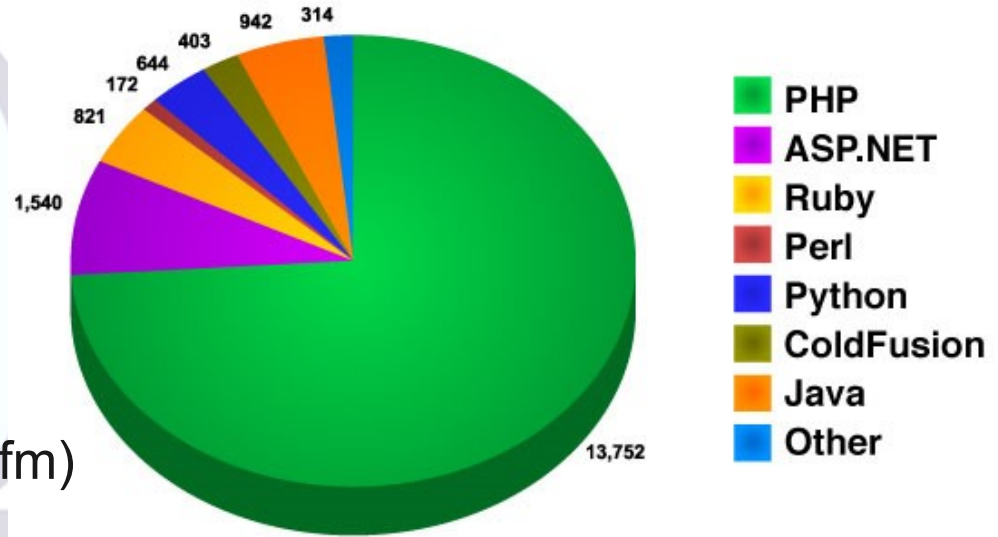


- Ο Rasmus Lerdorf το 1995 έγραψε ένα script με σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα.
- Η γλώσσα αρχικά ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter.
- Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους.
- Οι Andi Gutmans και Zeev Suraski το 1997 ξαναέγραψαν τη γλώσσα από την αρχή δημιουργώντας την εταιρεία Zend, η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP.
- Τρέχουσα έκδοση: **PHP 7.2.0** [Ιανουάριος 2018]

Server side scripting languages



- ASP/ASP.NET (*.asp/*.aspx)
- Java via JavaServer Pages (*.jsp)
- PHP (*.php)
- Python
- Ruby
- Perl CGI (*.cgi, *.ipl, *.pl)
- ColdFusion Markup Language (*.cfm)
- και άλλες όχι τόσο διαδεδομένες



	usage	change since 1 December 2017
1. PHP	83.1%	+0.1%
2. ASP.NET	14.1%	-0.1%
3. Java	2.5%	
4. static files	1.4%	
5. ColdFusion	0.6%	

percentages of sites

Πλεονεκτήματα της PHP



- Υψηλή απόδοση.
- Διασυνδέσεις με πολλά διαφορετικά συστήματα βάσεων δεδομένων (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, κ.ά.).
- Ενσωματωμένες βιβλιοθήκες για πολλές συνηθισμένες Web διαδικασίες.
- Χαμηλό κόστος.
- Ευκολία μάθησης και χρήσης.
- Μεταφερσιμότητα.
- Διαθεσιμότητα του πηγαίου κώδικα.
- Αρκετά frameworks με καλό documentation

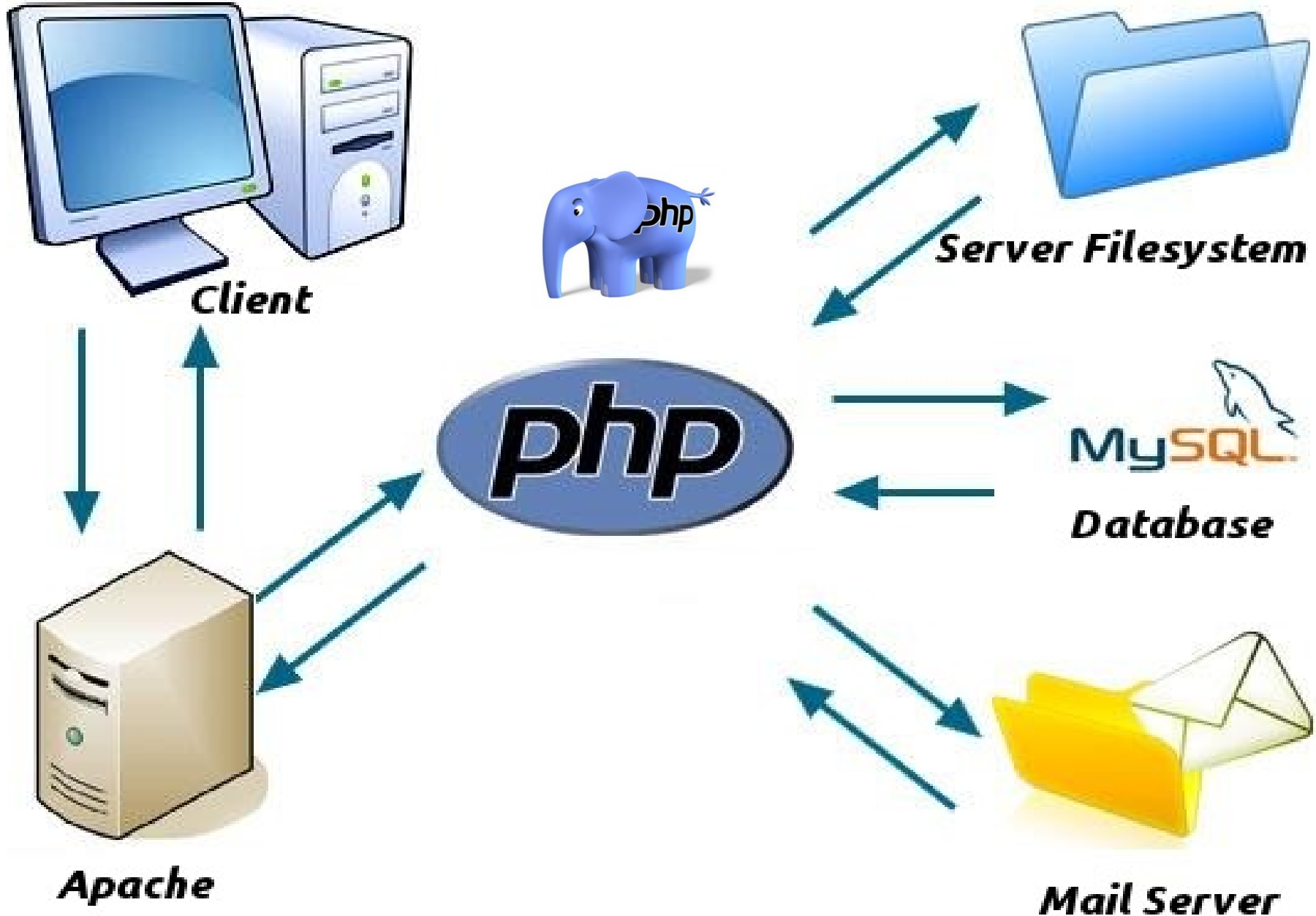


Βασική αρχή λειτουργίας



- Η PHP τρέχει σε ένα διακομιστή (Server).
- Ο Web browser ενός χρήστη κάνει μία `http://` αίτηση για μια συγκεκριμένη php Web σελίδα (πχ. `http://www.hua.gr/index.php`)
- Ο Web Server λαμβάνει την αίτηση για την σελίδα php, ανακαλεί το αρχείο και το περνά στην μηχανή php για επεξεργασία.
- Η μηχανή php αρχίζει την ανάλυση του php κώδικα επικοινωνώντας αν υπάρχει ανάγκη με τον Server της βάσης δεδομένων.
- Η μηχανή php σταματά την εκτέλεση του προγράμματος επιστρέφοντας στον Web Server την τελική HTML σελίδα.
- Ο Web Server περνά την html σελίδα ξανά στο Web browser όπου ο χρήστης μπορεί να διαβάσει το αποτέλεσμα της php σελίδας.

Βασική αρχή λειτουργίας



Γράφοντας PHP



Για ν' αναγνωρίσει η PHP ότι ένα κομμάτι κώδικα μέσα σε μία σελίδα HTML είναι PHP πρέπει να περικλείεται ανάμεσα σε εισαγωγικά της μορφής `<?php ... ?>`.

```
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <?php
      echo '<p>Hello World</p>';
    ?>
  </body>
</html>
```

Ο Server θα μου επιστρέψει το HTML αρχείο που φαίνεται στα δεξιά.

Αν δημιουργήσω και τρέξω το αρχείο στα αριστερά με όνομα Hello.PHP ,

```
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <p>Hello World</p>
  </body>
</html>
```

Γράφοντας PHP

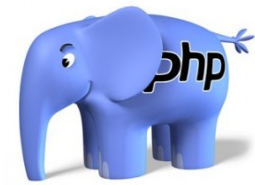


Μπορείτε να μαντέψτε ποια ακριβώς λειτουργία εκτελεί ο πιο κάτω κώδικας σε PHP.;

```
<html>
  <head>
    <title>PHP Test</title>
  </head>
  <body>
    <?php
      echo 5*7;
    ?>
  </body>
</html>
```

Σωστά μαντέψατε θα εμφανίσει το αποτέλεσμα της αριθμητικής πράξης «5*7» και θα εμφανίσει «35»

Σχόλια PHP



- Στην PHP για να εισάγουμε σχόλιο σε μια γραμμή χρησιμοποιούμε `//` και
- `/*` και `*/` για να εισάγουμε ένα μεγάλο μπλοκ σχολίων.

```
<html>
<body>
<?php
// This is a comment
/*
This is
a comment block
*/
?>
</body>
</html>
```

Μεταβλητές PHP



Οι μεταβλητές χρησιμοποιούνται για αποθήκευση τιμών, όπως αλφαριθμητικά, αριθμοί ή πίνακες. Όλες οι μεταβλητές στην PHP ξεκινούν με το σύμβολο `$`. Ο σωστός τρόπος για να ορίσουμε μια μεταβλητή είναι:

```
$var_name = value ;
```

```
$myAge = 32
```

- Τα ονόματα των μεταβλητών είναι case-sensitive
- Μια μεταβλητή δεν χρειάζεται να δηλωθεί πριν χρησιμοποιηθεί.
- Δεν χρειάζεται να δηλώσουμε τον τύπο της μεταβλητής.
- Οι τύποι δεδομένων που υποστηρίζονται από την PHP είναι οι εξής: String, Integer, Float (λέγεται και double), Boolean, Array, Object, NULL και Resource.
- Με τη συνάρτηση `print_r()` τυπώνει την τιμή μιας μεταβλητής. Με τη `var_dump()` μπορούμε να τυπώσουμε τόσο την τιμή μιας μεταβλητής όσο και τον τύπο της.

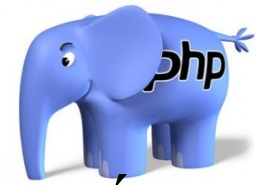
Άσκηση



Στον παρακάτω κώδικα δημιουργήστε δύο μεταβλητές με το όνομα `$myName` και `$myAge` και εκχωρήστε σε αυτές το όνομά σας και την ηλικία σας αντίστοιχα. Εμφανίστε την τιμή των παραπάνω μεταβλητών χρησιμοποιώντας την εντολή `echo`.

```
1. <html>
2.   <head>
3.     </head>
4.     <body>
5.       <p>
6.         <?php
7.           ?>
8.       </p>
9.     </body>
10. </html>
```

Άσκηση



Εκτελέσετε τον παρακάτω κώδικα για να μάθετε το όνομα και το τηλέφωνο που έχει ο καθηγητής σας. Στη συνέχεια να αλλάξετε το όνομα και το τηλέφωνο με τα δικά σας. Τι καινούριο μάθατε από την εκτέλεση αυτού του κώδικα;

1. `<?php`
2. `$identity = 'Άριστος Πασιάς';`
3. `$telephone= 'HTC';`
4. `$sentence = "Ο $identity έχει ένα $telephone";`
5. `echo $sentence;`
6. `?>>`

Άσκηση



Εκτελέσετε τον παρακάτω κώδικα για να μάθετε δυο τρόπους να γράψετε την ίδια εντολή.

```
1. <?php
2. // αυτό...
3. echo "Hello, world!";
4. // ... είναι ίδιο με αυτό
5. echo "Hello," . " " . "world" . "!!!";
6. ?>
```

Άσκηση



Εκτελέσετε τον παρακάτω κώδικα για να μάθετε δυο τρόπους να γράψετε την ίδια εντολή.

1. `<?php`
2. `// αυτό...`
3. `$a = 5;`
4. `$a = $a + 10;`
5. `// ... είναι ίδιο με αυτό`
6. `$a = 5;`
7. `$a += 10;`
8. `?>`

Concatenation στη PHP



Για να ενώσουμε δύο Αλφαριθμητικές μεταβλητές στην PHP χρησιμοποιούμε την τελεία (.) Ο πιο κάτω κώδικας θα έχει ως αποτέλεσμα το:

Hello World 1234

```
<?php
$txt1= "Hello World";
$txt2= "1234";
echo $txt1 . " " . $txt2;
?>
```

Άλλες συναρτήσεις για αλφαριθμητικά (string) είναι:

- strlen() – Μήκος αλφαριθμητικού
- strpos() – Θέση αλφαριθμητικού
- strcmp() – Συγκρίνει δύο αλφ.και άλλες!

Δομή Επιλογής: If/else στη PHP



Ας θεωρήσουμε ότι θέλουμε να γράψουμε ένα πρόγραμμα το οποίο να ρωτάει αν η ηλικία σου είναι μεγαλύτερη ή ίση του 18. Αν η απάντηση είναι ναι, θέλουμε να εμφανίζεται ένα μήνυμα που να λέει «Μπορείς να ψηφίσεις!». Διαφορετικά να εμφανίζει το μήνυμα «Είσαι ακόμα μικρός!»

```
<?php
    $age = 19;
    if( $age > = 18 ) {
        echo " Μπορείς να ψηφίσεις!"; }
    else {
        echo " Είσαι ακόμα μικρός!"; }
?>
```

Στο παραπάνω παράδειγμα, αν η συνθήκη στη παρένθεση της εντολής If είναι **Αληθής** εκτελείται η εντολή ανάμεσα στα { ...} διαφορετικά εκτελείται η εντολή στα { ...} μετά το else.

Δομή Επιλογής if - Παράδειγμα



```
<?php
date_default_timezone_set ("Europe/Athens");
$t=date("H");
if ($t<"10")
{
echo "Have a good morning!";
}
elseif ($t<"20")
{
echo "Have a good day!";
}
else
{
echo "Have a good night!" ;
}
?>
```

Δομή Επανάληψης For

Αν θέλω να εμφανίσω όλους τους αριθμούς από το 1 ως το 100 μπορώ να χρησιμοποιήσω τον πιο κάτω κώδικα.

```
<?php  
echo 1;  
echo 2;  
echo 3;  
// ... και τα λοιπά και τα λοιπά ...  
?>
```

Ή πιο απλά να χρησιμοποιήσω αυτόν τον κώδικα.



```
<?php  
// Εμφανίστε τους αριθμούς από το 1 μέχρι και το 100  
for ($i = 1; $i < 101; $i = $i + 1) {  
echo $i;  
}  
?>
```

Δομή Επανάληψης For



Η δομή επανάληψης «for» ξεκινάει με την δεσμευμένη λέξη «for». Είναι σαν να λέμε στην PHP ετοιμάσου να εκτελέσεις αυτό που θα σου πούμε πολλές φορές. Στη συνέχεια προσθέτουμε ένα ζευγάρι από παρενθέσεις «()». Μέσα στις οποίες λέμε στην PHP τρία πράγματα που χωρίζονται μεταξύ τους με το σύμβολο «;» και λένε αντίστοιχα τα εξής: Πώς αρχίζει η επανάληψη, πότε τελειώνει η επανάληψη και τι κάνουμε κάθε φορά πριν πάμε στην επόμενη επανάληψη. Στη συνέχεια των παρενθέσεων ακολουθούν τα σύμβολα «{}». Μέσα στα σύμβολα «{}» λέμε στην PHP ποιές εντολές να εκτελεί σε κάθε επανάληψη.

Δομή Επανάληψης foreach



Μπορείτε να μαντέψετε την λειτουργία της πιο κάτω εφαρμογής;

```
<html>
<body>
<?php
$x=array ("one", "two", "three");
foreach ( $x as $value )
{
echo $value . "\n";
}
?>
</body>
</html>
```

Άσκηση



Στη παρακάτω εφαρμογή προσπαθώ να εμφανίσω τους πέντε πρώτους άρτιους αριθμούς. Δηλαδή τους 2, 4, 6, 8, και 10. Κάτι όμως δεν κάνω καλά. Μπορείτε να βρείτε τι;

```
<html>
<body>
<?php
for ($i = 2; $i < 10; $i = $i + 1)
{
echo $i;
}
?>
</body>
</html>
```

Δομή Επανάληψης: While

Όταν θέλουμε να επαναλάβουμε ένα σύνολο εντολών πολλές φορές χωρίς όμως να γνωρίζουμε εκ των προτέρων τον αριθμό των επαναλήψεων χρησιμοποιούμε την εντολή **while**.

```
<?php
$loopCond = true;
while ($loopCond)
{
echo "<p> Είμαστε μέσα στην επανάληψη </p>";
$loopCond = false;
}
echo "<p> Βγήκαμε από την επανάληψη </p>";
?>
```

Δομή Επανάληψης: Do While

Μία εναλλακτική περίπτωση της while αποτελεί η **do/while** που κάνει ακριβώς το αντίθετο. Δηλαδή, πρώτα εκτελείται η επανάληψη και στη συνέχεια ελέγχεται η συνθήκη.

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

Πόσες φορές θα εκτελεστεί η εντολή «echo \$i» στο πιο πάνω παράδειγμα;

Δομή Επανάληψης: While – Do While

Να συγκρίνετε τα δυο Παραδείγματα

```
<html>
<body>
<?php
$i=0;
while ($i <= 5)
{ echo "Number : " .
$i . "\n";
$i++;
}
?>
</body>
</html>
```

```
Number : 0
Number : 1
Number : 2
Number : 3
Number : 4
Number : 5
```



```
<html>
<body>
<?php
$i=0;
do
{
$i++;
echo "The number is " . $i
. "\n";
}
while ($i <= 5);
?>
</body>
</html>
```

```
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
```

Δομή Επανάληψης: Switch

```
<?php
$favcolor= "blue" ;
switch ($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 , or
green!";
}
?>
```

Πίνακες στη PHP



Ένας πίνακας είναι μια λίστα από αντικείμενα. Σας επιτρέπει να αποθηκεύσετε περισσότερα από ένα αντικείμενα σε μία μόνο μεταβλητή.

Οι πίνακες πάντα έχουν: **key** και **value**. Κάθε key αναφέρεται σε ένα συγκεκριμένο value

```
<?php
$languages = array("HTML/CSS",
"JavaScript", "I don't know", "Python",
"Ruby");
$languages[2] = "PHP";
echo $languages[2] ;
?>
```

Τι θα εμφανίσει η εντολή echo στο πιο πάνω παράδειγμα;

Πίνακες στη PHP



```
\\ index automatically assigned  
$cars= array ( " Saab " , " Volvo " , "BMW" , " Toyota " ) ;
```

```
\\ index manually assigned
```

```
<?php  
$cars[0] = "Saab";  
$cars[1] = "Volvo";  
$cars[2] = "BMW";  
$cars[3] = "Toyota";  
echo $cars[0] . " and " . $cars[1] . " are Swedish cars . ";  
?>
```

```
\\Associative Arrays
```

```
$ages = array("Peter"=>32, "Quagmire"=>30, "Joe"=>34);
```

```
<?php  
$ages['Peter'] = "32";  
$ages['Quagmire'] = "30";  
$ages['Joe'] = "34";  
echo
```

Συναρτήσεις στη PHP



Οι συναρτήσεις είναι επαναχρησιμοποιήσιμα κομμάτια κώδικα που μπορείτε να χρησιμοποιήσετε μέσα στο πρόγραμμά σας. Η PHP έχει πολλές ενσωματωμένες συναρτήσεις. Μία από αυτές είναι η **strlen()** που επιστρέφει τον αριθμό των χαρακτήρων που υπάρχουν σε ένα αλφαριθμητικό. Στο παρακάτω παράδειγμα εμφανίζεται ο αριθμός 7, όσοι είναι και οι χαρακτήρες της λέξης «Άριστος».

```
<?php
$length = strlen(«Άριστος»);
print $length;
?>
```

Συναρτήσεις στη PHP



Εκτός από τις ενσωματωμένες συναρτήσεις PHP που μας παρέχονται μπορούμε να δημιουργήσουμε και τις δικές μας, όπως φαίνεται στο παρακάτω παράδειγμα, όπου εμφανίζουμε το όνομά μας.

```
<?php
function displayname() {
echo "Το όνομά μου είναι
Πέτρος"; }
displayname();
?>
```

Συναρτήσεις στη PHP



Στο παρακάτω κώδικα η συνάρτηση δέχεται μία παράμετρο, την οποία την πολλαπλασιάζει με τον εαυτό της και στη συνέχεια εμφανίζει το αποτέλεσμα της αριθμητικής πράξης.

```
<?php
function squareValue($number) {
echo $number * $number; }
$n = 6;
squareValue($n);
?>
```

Οπότε θα εμφανίσει τον αριθμό 36

Συναρτήσεις στη PHP



Μπορείτε να μαντέψετε τι κάνει η παρακάτω εφαρμογή;

```
<?php
function aboutMe($name, $age)
{
echo "Hello! My name is " . $name
. " and I am " . $age . " years
old.";
}
aboutMe( "Maria", 17);
?>
```

Συναρτήσεις στη PHP



Μπορείτε να μαντέψετε τι εμφανίζει η εντολή `echo` στη παρακάτω εφαρμογή;

```
<?php
function sum($a, $b)
{
    $c = $a + $b;
    return $c;
}
echo sum(2,13) ;
?>
```

Συναρτήσεις Παράδειγμα



```
<html>
<body>
<?php
function div($x,$y)
{
$result = $x/$y ;
return $result;
}
echo " 24 / 3 = " . div(24,3);
?>
</body>
</html>
```

Πρόσβαση και διαχείριση Βάσεων Δεδομένων

- Ο web Server λαμβάνει ένα αίτημα για το results.php. Βρίσκει το αρχείο αυτό και το δίνει στην PHP για επεξεργασία.
- Η PHP αρχίζει να αναλύει το σενάριο. Μέσα στο σενάριο περιέχονται εντολές σύνδεσης με την βάση δεδομένων και εκτέλεσης του ερωτήματος (αναζήτηση των βιβλίων). Η PHP ανοίγει μία σύνδεση με τον MySQL Server και στέλνει το ερώτημα.
- Ο MySQL Server λαμβάνει το ερώτημα, το επεξεργάζεται και στέλνει πίσω στην PHP τα αποτελέσματα, δηλαδή την λίστα με τα βιβλία.
- Η PHP μορφοποιεί κατάλληλα τα αποτελέσματα σε μορφή κώδικα HTML και τα επιστρέφει στον web Server .
- Ο web Server τα στέλνει στη συνέχεια στο πρόγραμμα περιήγησης, όπου ο χρήστης μπορεί να δει την λίστα με τα βιβλία που ζήτησε.

Πρόσβαση και διαχείριση Βάσεων Δεδομένων

Οι βασικές λειτουργίες ενός σεναρίου .rhp που ανακτά αποτελέσματα αναζήτησης από την βάση δεδομένων και τα μορφοποιεί για παρουσίαση είναι:

Ελεγχος και φιλτράρισμα δεδομένων φόρμας

Δημιουργία σύνδεσης με την βάση δεδομένων

Ερώτημα προς την βάση δεδομένων

Ανάκτηση αποτελεσμάτων

Αποσύνδεση από την βάση

Προβολή δεδομένων βάσης MySQL σε HTML

Βασική λειτουργία της PHP αποτελεί η διασύνδεση της με βάση δεδομένων (MySQL) με σκοπό την προβολή τους σε ιστοσελίδα HTML. Στο επόμενο παράδειγμα, στην απομακρυσμένη βάση δεδομένων **library** υπάρχει ο πίνακας **books** με τα πεδία **btitle**, **isbn** και **bauthor**. Σκοπός του παραδείγματος είναι η προβολή των στοιχείων όλων των βιβλίων που είναι αποθηκευμένα στην βάση σε ιστοσελίδα HTML.

```
<?php
$hostname="localhost";
$username="root";
$password="yourpassword";
$dbname="library";
    $conn=mysql_connect($hostname, $username $password);
    mysql_select_db('dbname', $conn);
        echo '<h3>Τα βιβλία μας, '</h3>';
$result = mysql_query('SELECT * FROM books',$conn) or die('cannot show
tables');
    if(mysql_num_rows($result)) {
        echo '<table cellpadding="0" cellspacing="0" class="db-
table">';
        echo
'<tr><th>Τίτλος</th><th>ISBN</th><th>Συγγραφέας</th><tr>';
        while($row=mysql_fetch_row($result)) {
            echo '<tr>';
            for (i=0; i<3; i++) {
echo '<td>', $row[i], '</td>';        } echo '</tr>';        } echo '</table> \n';        }
?>
```

Φόρμες



```
<html>
<body>
<form action="welcome.php" method="get">
Name: <input type="text" name="name"/>
Age : <input type="text" name="age"/>
<input type="submit"/ >
</form>
</body>
</html>
```

```
<html>
<body>
Welcome <?php echo $_POST["name"]; ?> . <br/>
You are <?php echo $_POST["age"]; ?> years old.
</body>
</html>
```

Οι PHP μεταβλητές **\$_GET** και **\$_POST** χρησιμοποιούνται για να ανακτούν δεδομένα από φόρμες που συμπληρώνονται από χρήστες.

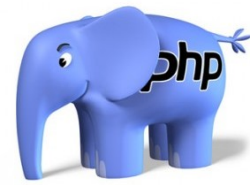
Φόρμες PHP \$_GET



Η μεταβλητή `$_GET` είναι ένας πίνακας από ονόματα μεταβλητών και τιμές που στέλνονται μέσω της μεθόδου HTTP GET. Χρησιμοποιείται για να συλλέξει τιμές από μια φόρμα που χρησιμοποιεί τη μέθοδο `get`. Η πληροφορία που στέλνεται μέσω της HTTP GET μεθόδου είναι ορατή στον καθένα (φαίνεται στο address bar του browser!) και υπάρχει περιορισμός στην ποσότητα πληροφορίας που μπορεί να αποσταλεί (max 2000 χαρακτήρες).

Προσοχή! Δεν μπορούμε να στέλνουμε passwords ή άλλη ευαίσθητη πληροφορία με τη GET. Όμως όταν θέλουμε να θέσουμε έναν σελιδοδείκτη είναι χρήσιμη η GET.

Φόρμες PHP \$_POST



Η μεταβλητή `$_POST` είναι ένας πίνακας από ονόματα μεταβλητών και τιμές που στέλνονται μέσω της μεθόδου HTTP POST.

Χρησιμοποιείται για να συλλέξει τιμές από μια φόρμα που χρησιμοποιεί τη μέθοδο `post`. Η πληροφορία που στέλνεται μέσω της POST μεθόδου είναι αόρατη και δεν έχει όριο μεγέθους για την πληροφορία που αποστέλλεται. (Καθορίζεται μέσω της μεταβλητής `post_max_size` στο αρχείο `php.ini`)

Φόρμες PHP \$_POST



```
<html>
<body>
<form action="welcome.php" method="post">
Name: <input type="text" name="name"/>
Age : <input type="text" name="age"/>
<input type="submit"/>
</form>
</body>
</html>
```

```
<html>
<body>
Welcome <?php echo $_POST["name"]; ?> . <br/>
You are <?php echo $_POST["age"]; ?> years old.
</body>
</html>
```

PHP \$_REQUEST



- Η μεταβλητή `$_REQUEST` είναι ένας πίνακας που κρατά τα περιεχόμενα των πινάκων `$_GET`, `$_POST` και `$_COOKIE`.
- Χρησιμοποιείται για να συλλέξει τιμές από μια φόρμα που χρησιμοποιεί τη μέθοδο `get` και `post`.

```
<html>
<body>
Welcome <?php echo $_REQUEST["name"]; ?>!<br/>
You are <?php echo $_REQUEST["age"]; ?> years old.
</body>
</html>
```

Έλεγχος τιμών των μεταβλητών



Για να ελέγξετε αν μία μεταβλητή έχει πάρει τιμή κάντε χρήση της συνάρτησης `empty()`. Η `empty()` ελέγχει αν η μεταβλητή έχει σαν τιμή (`empty string` , `0`, `NULL` ή `False`) . Επιστρέφει `FALSE` αν η μεταβλητή έχει `non-empty` ή `non-zero` τιμή. Η `isset()` ελέγχει αν η μεταβλητή έχει πάρει τιμή, η οποία μπορεί να είναι ακόμη και (`False` , `0` ή `Empty string`) , αλλά όχι `NULL`. Επιστρέφει `TRUE` αν η μεταβλητή υπάρχει και `FALSE` σε οποιαδήποτε άλλη περίπτωση.

Με την εντολή `unset()` μπορείτε να ελευθερώσετε μια μεταβλητή `$_POST` από την τιμή που έχει πάρει.

Έλεγχος τιμών των μεταβλητών



```
<?php
$var = 0 ;
//Evaluates to true because $var is empty
if (empty($var)) {
echo '$var is either 0, empty, or not set at all'; }
// Evaluates as true because $var is set
if (isset($var)) {
echo ' $var is set even though it is empty'; }
?>
```

```
unset ( $_POST['variable_name']);
```

Cookies



Τα cookies χρησιμοποιούνται για να αναγνωρίζουμε χρήστες. Είναι ένα μικρό αρχείο το οποίο ο server ενσωματώνει στον υπολογιστή του χρήστη. Κάθε φορά που ο ίδιος υπολογιστής αιτείται μια σελίδα μέσω του browser, στέλνεται και το cookie μαζί. Με την php μπορούμε να δημιουργούμε και να ανακτούμε τις τιμές ενός cookie. Πρέπει να μπει πριν το `<html>` tag

```
setcookie(name,value,expire,path,domain);
```

```
<?php  
setcookie("user","Alex Porter",time () + 3600);  
?>  
<html>  
.....
```



Ανάκτηση τιμής ενός cookie: \$_COOKIE



```
<html>
<body>
<?php
if (isset($_COOKIE["user"]))
echo "Welcome" . $_COOKIE["user"] . "!<br/>";
else
echo "Welcome guest!<br/>";
// A way to view all cookies
print_r($_COOKIE);
?>
</body>
</html>
```

Διαγραφή cookie:

```
setcookie("user","",time() -3600);
```

Sessions



Η session (σύνοδος) αποτελεί τον τρόπο με τον οποίο μπορούμε να αποθηκεύσουμε πληροφορία (με τη μορφή μεταβλητών) ώστε να μπορεί να χρησιμοποιηθεί μεταξύ πολλών σελίδων (username, shopping items κτλ). Αντίθετα με τα cookies, η σχετική πληροφορία των μεταβλητών δεν αποθηκεύεται στον υπολογιστή των χρηστών. Επίσης διαφέρει από τις υπόλοιπες μεταβλητές με την έννοια ότι δεν τις περνάμε ξεχωριστά σε κάθε σελίδα, αλλά τις ανακτούμε από τη σύνοδο (session), την οποία ανοίγουμε στην αρχή κάθε σελίδας. Ωστόσο, αυτού του είδους οι πληροφορίες είναι προσωρινές και συνήθως διαγράφονται πολύ σύντομα μόλις ο χρήστης αφήσει το website που χρησιμοποιεί sessions.

session start



Πριν αποθηκεύσουμε πληροφορίες στη σύνοδο, πρέπει πρώτα να την ξεκινήσουμε. Αυτό γίνεται όσο πιο νωρίς είναι δυνατό στον κώδικα, πριν οποιοδήποτε HTML κείμενο σταλεί. Το αποτέλεσμα του παρακάτω κώδικα είναι:

Pageviews = 1

```
<?php
session_start ( ) ;
$_SESSION['views']=1; // store session data
echo "Pageviews = " . $_SESSION['views']; //retrieve data
?>
```

```
<?php
session_start();
if (isset($_SESSION['views']))
$_SESSION['views']=$_SESSION['views'] + 1;
else
$_SESSION['views']=1;
echo "views = " . $_SESSION['views'];
?>
```

Προκειμένου να πάρουμε την τιμή μιας μεταβλητής συνόδου, καλό θα ήταν να δούμε πρώτα αν έχει πάρει τιμή με τη συνάρτηση `isset`

session destroy



Αν θέλουμε να διαγράψουμε ορισμένα δεδομένα από τη σύνοδο, μπορούμε να χρησιμοποιήσουμε την `unset()`

```
<?php
unset($_SESSION['views']);
?>
```

Μπορούμε να τελειώσουμε μια σύνοδο, χρησιμοποιώντας τη συνάρτηση `session_destroy`. Όταν γίνει αυτό, σημαίνει ότι όλα τα δεδομένα που αποθηκεύτηκαν στη σύνοδο θα χαθούν!

```
<?php
session_start();
session_destroy();
?>
```

include - require



Μπορούμε να εισάγουμε τα περιεχόμενα ενός PHP αρχείου σε ένα άλλο, πριν ο server ξεκινήσει να τα εκτελεί. Αυτό γίνεται με τις:

- `include()` σε περίπτωση σφάλματος δημιουργεί μια προειδοποίηση, αλλά ο κώδικας συνεχίζεται να εκτελείται
- `require()` σε περίπτωση σφάλματος δημιουργεί ένα fatal error και ο κώδικας σταματάει να εκτελείται.

```
<?php include("header.php");?>
<h1>Welcome to my home page!</h1>
<p>Some text.</p>
```

Error Handling



Η διαχείριση σφαλμάτων είναι σημαντικό μέρος της ανάπτυξης web εφαρμογών. Αν ο κώδικας δεν ελέγχει για τυχόν σφάλματα, υπάρχει πιθανότητα να προκύψουν θέματα ασφάλειας δεδομένων.

Μέθοδοι αντιμετώπισης σφαλμάτων:

- δηλώσεις τύπου "die()"
- αναφορά σφάλματος
- καθορισμός τύπων σφαλμάτων

Error Handling



```
<?php
$file=fopen("welcome.txt","r");
?>
Warning : fopen (welcome.txt) [function.fopen]:failed to open
stream
:
No such file or directory in C:\webfolder\test.php on line 2
```

```
<?php
if (!file_exists("welcome.txt")) { die ("File not found"); }
else
{ $file=fopen("welcome.txt","r");} ?>
File not found
```

Debugging



- Κοιτάμε τον κώδικα!
- Κοιτάμε το log του server (σε unix συνήθως είναι στο `/var/log/apache2/error.log`).
- Χρησιμοποιούμε `echo`, `print_r`, `var_dump` για να δούμε αν μεταβλητές ή αντικείμενα έχουν τις τιμές που θέλουμε.
- Ελέγχουμε αν όλα τα απαραίτητα πακέτα είναι εγκατεστημένα στον server.
- Χρησιμοποιούμε το logging σύστημα του framework.

PHP Class



Οι κλάσεις αποτελούν τη δομή των δεδομένων και των ενεργειών και χρησιμοποιούν αυτές τις πληροφορίες για να χτίσουν αντικείμενα. Περισσότερα από ένα ανεξάρτητα αντικείμενα μπορεί να δημιουργηθούν από μια κλάση.

```
<?php
class MyClass
{
public $prop1 = "I'm a class property!";
public function setProperty($newval)
{ $this->prop1 = $newval; }
public function getProperty()
{ return $this->prop1 . "<br/>"; }
}
$obj = new MyClass;
echo $obj->prop1; ?>
```

<http://code.tutsplus.com/tutorials/object-oriented-php-forbeginners--net-12762>

Constructors- Destructors



Όταν ένα αντικείμενο δημιουργείται, πολλές φορές χρειάζεται να γίνουν κάποια πράγματα. Για το σκοπό αυτό η PHP παρέχει τη μέθοδο **__construct()** η οποία καλείται αυτόματα κάθε φορά που δημιουργείται ένα αντικείμενο.

Επίσης όταν «καταστρέφεται» ένα αντικείμενο μπορούμε να «τακτοποιούμε» (π.χ. να κλείνουμε μια σύνδεση με τη βάση) πράγματα χρησιμοποιώντας τη μέθοδο **__destruct()**.

Constructors- Destructors



```
<?php
class MyClass
{
public $prop1 = "I'm a class property!";
public function __construct()
{ echo 'The class "', __CLASS__, "' was
initiated!<br />'; }
public function setProperty($newval)
{ $this->prop1 = $newval; }
public function getProperty()
{ return $this->prop1 . "<br />"; }
}
// Create a new object
$obj = new MyClass;
// Get the value of $prop1
echo $obj->getProperty();
// Output a message at the end of the file
echo "End of file.<br />";
?>
```

Σύνδεση με βάση δεδομένων



Σε περίπτωση αποτυχημένης σύνδεσης ο χρήστης πρέπει να ειδοποιείται.

```
<?php
$con =
mysql_connect("localhost", "user", "password");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
?>
```

Δημιουργία βάσης – εισαγωγή πίνακα



```
<?php
$con = mysql_connect("localhost","user","
password");
// Select the database
mysql_select_db("my_db", $con);
// Create the table
$sql = "CREATE TABLE phone_book
(
personID int NOT NULL,
PRIMARY KEY(personID),
LastName varchar(20) NOT NULL,
FirstName varchar(20),
Address varchar(30),
Age int,
Phone varchar(10) )";
mysql_query($sql,$con);
mysql_close($con);
?>
```

Δημιουργία βάσης – εισαγωγή πίνακα

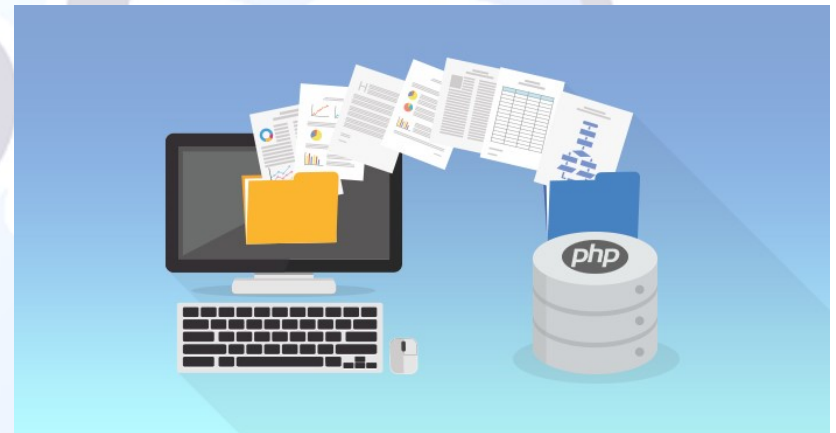


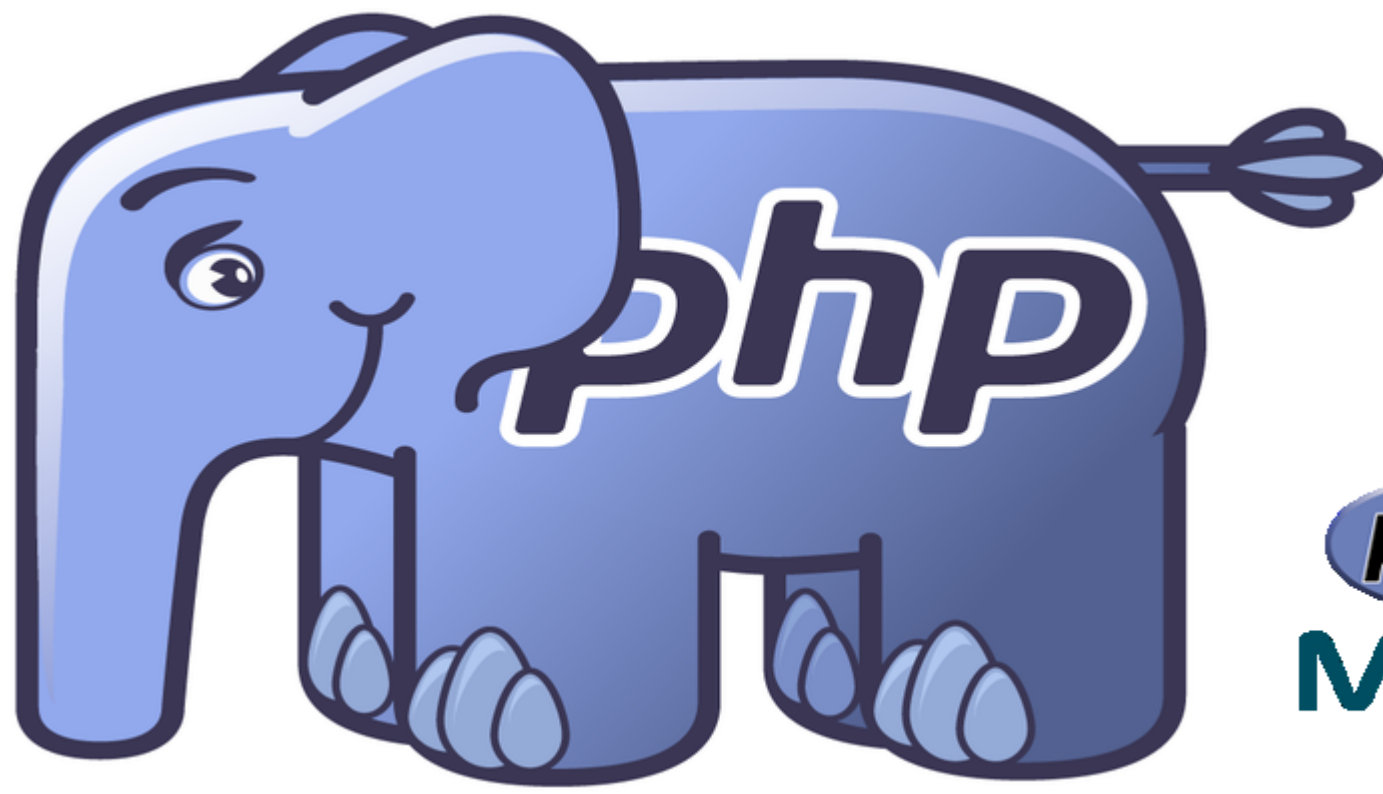
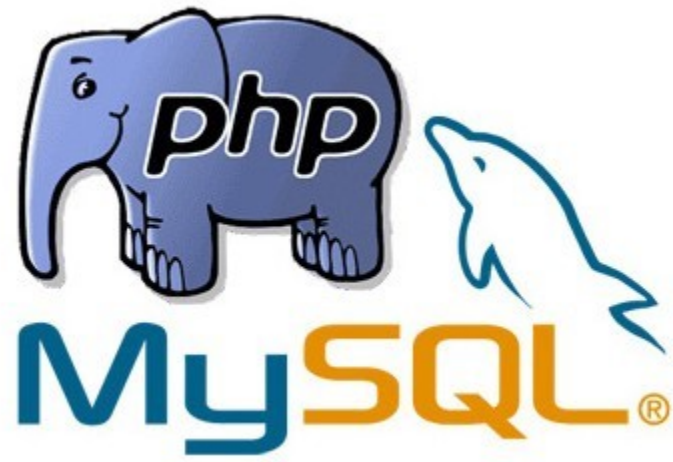
```
<?php
$con = mysql_connect("localhost","user","
password");
if (!$con)
{
die('Could not connect: ' . mysql_error());
}
if (mysql_query("CREATE DATABASE my_db",$con))
{
echo "Database created";
}
else
{
echo "Error creating database: ".mysql_error();
}
mysql_close($con);
?>
```

Επιλογή δεδομένων από πίνακα



```
<?php
// Select the database
mysql_select_db("my_db", mysql_connect
("localhost", "user", "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>";
}
?>
```





```
<?php
date_default_timezone_set ("Europe/ Athens");
$t=date("H") ;
if ($t<"10")
{
echo "Have a good morning!";
}
elseif ($t<"20")
{
echo "Have a good day!";
}
else
{
echo "Have a good night!" ;
}
?>
```



