



# Προγραμματισμός Διαδικτύου

Δρ. Μηνάς Δασυγένης  
[mdasygenis@uowm.gr](mailto:mdasygenis@uowm.gr)

Εργαστήριο Ρομποτικής, Ενσωματωμένων και Ολοκληρωμένων Συστημάτων

---



# Άδειες Χρήσης

---

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



# Σκοπός ενότητας

---

- ✓ Γνωριμία με την PHP
- ✓ Λειτουργία PHP script στο web και στο webserver
- ✓ Εκμάθηση λογισμικού και εξαρτήματος για την εκκίνηση της PHP
- ✓ Δημιουργία και εκτέλεση ενός απλού PHP script
- ✓ Κατανόηση βασικών συναρτήσεων όπως οι `strlen()`, `trim()`, `substr()`



# Προγραμματισμός διαδικτύου

---

## PHP

### Περιεχόμενα

- 4.1 Εισαγωγή
- 4.2 Μεταβλητές
- 4.3 Εντολές Ελέγχου
- 4.4 Δομές Επανάληψης
- 4.5 Τελεστές
- 4.6 Πίνακες



# Εισαγωγικά PHP

---

- PHP σημαίνει:
  - PHP: Hypertext Preprocessor
- Το PHP είναι μία γλώσσα Script η οποία τρέχει στον Server και στον Web Browser επιστρέφει script σε μορφή HTML.
- Το PHP συνεργάζεται με διάφορες βάσεις δεδομένων όπως:
  - MySQL, Oracle, Sybase, Generic ODBC κ.α.
- Το PHP είναι Open Source λογισμικό και είναι δωρεάν
- Τα αρχεία PHP έχουν extensión (π.χ. "script.php")
- Ένα αρχείο PHP μπορεί να περιέχει και HTML tags και scripts
- Η σύνταξη του PHP είναι σαν της γλώσσας C++



# HTML – PHP

---

- Η Hypertext Markup Language (HTML) χρησιμοποιεί κωδικοποιημένες εντολές που ονομάζονται ετικέτες HTML οι οποίες παρέχουν οδηγίες σε προγράμματα περιήγησης στο Web και δείχνουν πώς εμφανίζεται κάθε σελίδα.
- Η PHP είναι μια γλώσσα προγραμματισμού που χρησιμοποιείται για την επέκταση των δυνατοτήτων των HTML εγγράφων και τη δημιουργία δυναμικών εφαρμογών Web.



# Η PHP είναι δημοφιλής γλώσσα προγραμματισμού

---

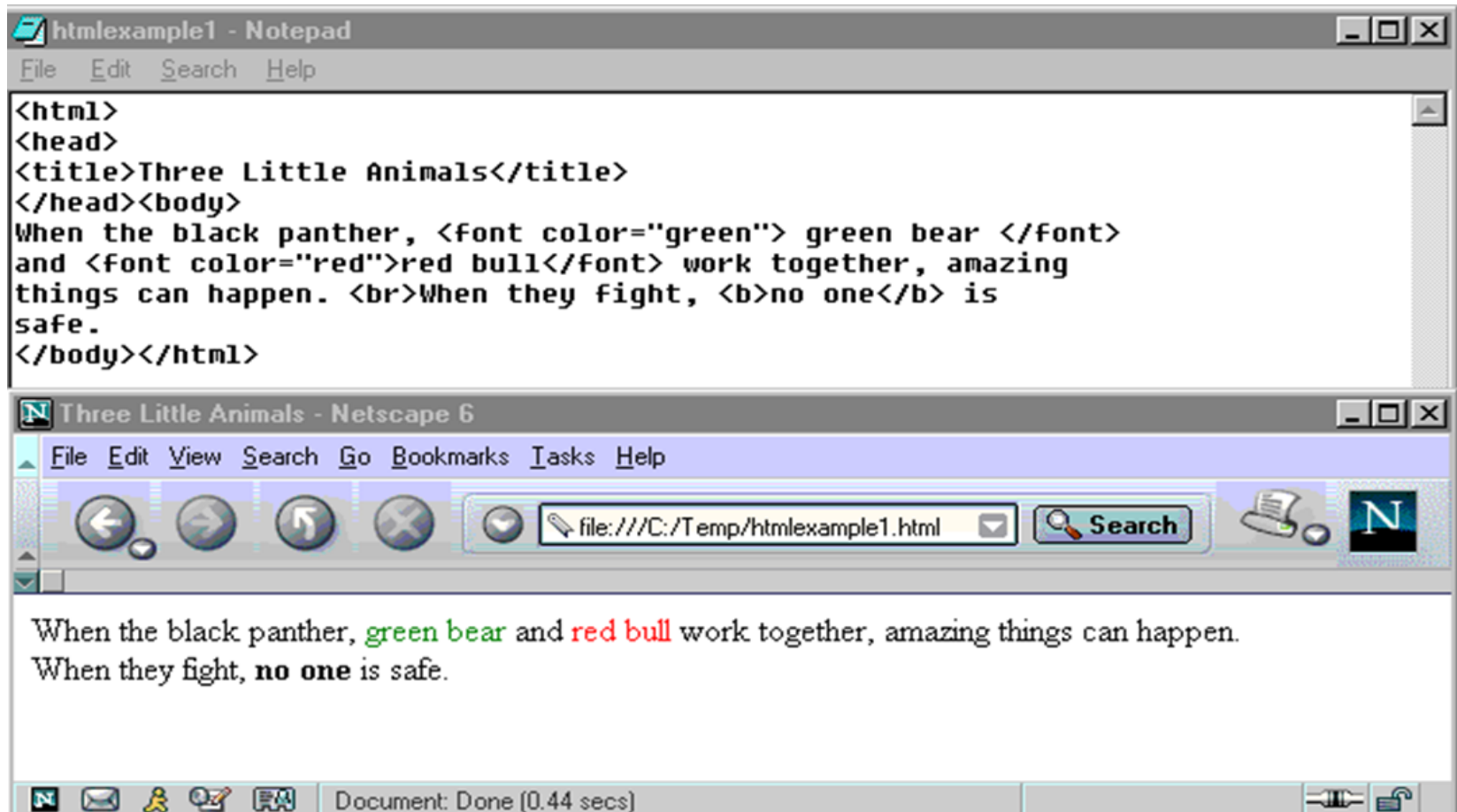
- Η PHP τρέχει κάτω από διάφορα λειτουργικά συστήματα (Windows. Linux. Unix. κτλ.)
- Η PHP συνεργάζεται με τους περισσότερους server (Apache. IIS. κτλ)
- Τα PHP αρχεία επιστρέφονται στον browser σαν αρχεία απλής HTML.
- Τα PHP αρχεία μπορούν να έχουν κατάληξη ".php". ".php3". ή ".phtml"
- Μπορούμε να κατεβάσουμε την PHP ελεύθερα από τη διεύθυνση:

[www.php.net](http://www.php.net)

- Η PHP είναι εύκολη στην εκμάθηση και ιδιαίτερα αποδοτική.



# Ιστορικό τεχνολογίας Ιστού





# *Τι είναι η PHP;*

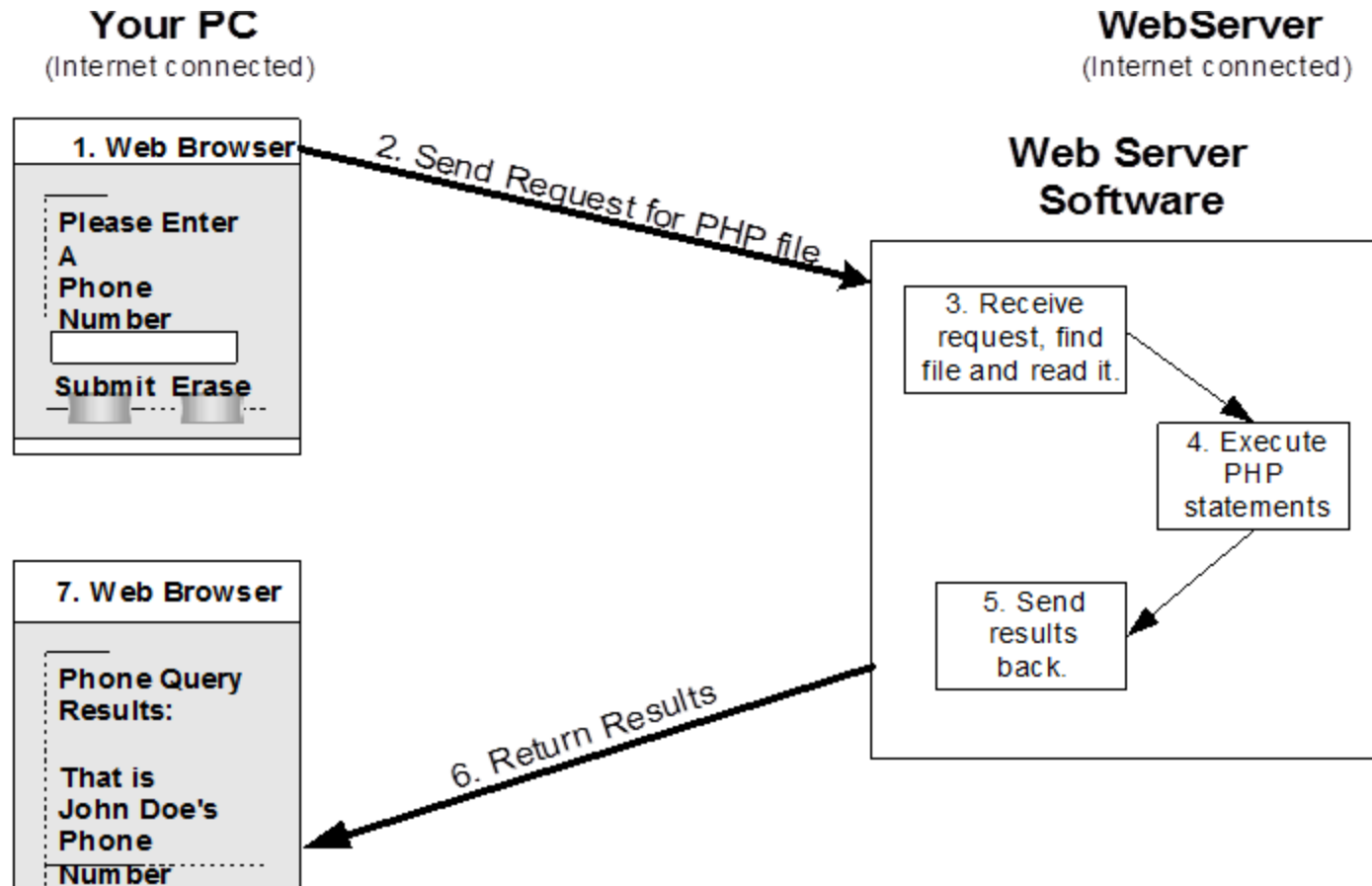
---

Πλεονεκτήματα της χρήσης της PHP για την ενίσχυση ιστοσελίδων:

- Εύκολο στη χρήση
- Ανοιχτός κώδικας
- Πολλαπλές πλατφόρμες
- Performance
- Υποστήριξη γλωσσών για βάσεις δεδομένων



# Πώς γίνονται προσβάσιμες και πως ερμηνεύονται οι σελίδες PHP



# Ξεκινώντας με την PHP

---

Για να αναπτύξετε και να δημοσιεύσετε PHP scripts, όλα όσα χρειάζεστε είναι :

- Ένας διακομιστής Web με την PHP χτισμένη σε αυτό
- Ένα μηχάνημα πελάτη με ένα βασικό πρόγραμμα επεξεργασίας κειμένου και σύνδεσης στο Internet.
- FTP ή Telnet λογισμικό



# *Το πρώτο script σε PHP*

---

```
<?php  
κώδικας PHP  
?>  
κώδικας HTML  
<?php  
κώδικας PHP  
?>
```



# Hello World στη PHP

---

- Αρχείο "myfirst.php"

- `<?php`

- `echo "Hello World";`

- `?>`

- Η εντολή `echo` στέλνει στον Web Browser κείμενο

- Οι εντολές στην PHP τελειώνουν με `;` όπως και οι εντολές στην C++

- Σχόλια στην PHP

- `<?php`

- `// Αυτό είναι σχόλιο`

- `/*`

- Σχόλια Πολλών Γραμμών

- `*/`

- `?>`

---

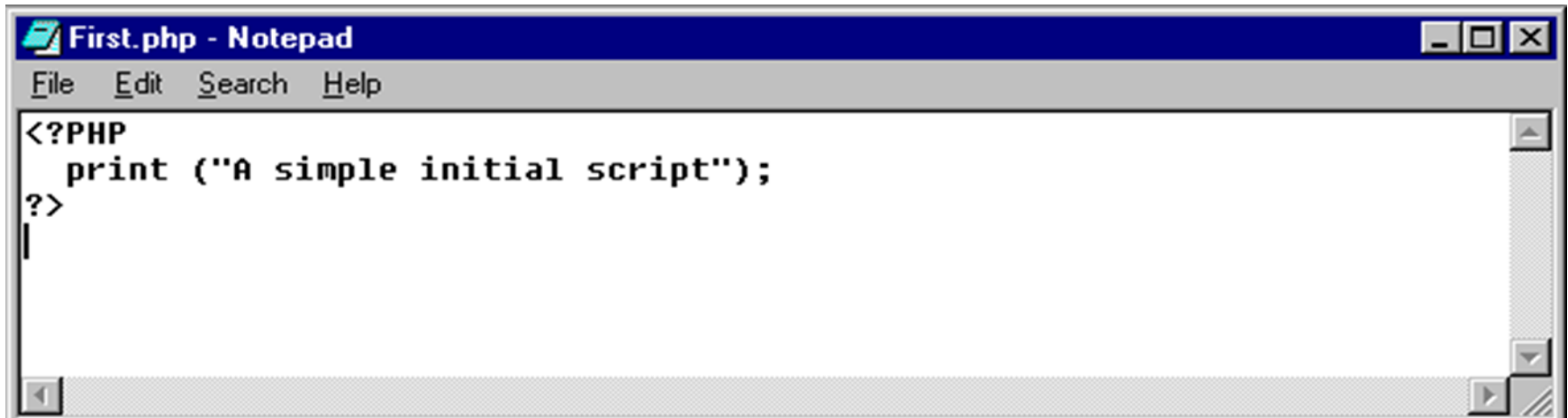


# Δημιουργία αρχείου PHP Script και αποθήκευσης του σε τοπικό δίσκο

---

Μπορείτε να χρησιμοποιήσετε έναν αριθμό διαφορετικών εκδόσεων για να δημιουργήσετε αρχεία PHP script.

- Το σενάριο PHP αρχίζει με μια ετικέτα `<? Php` και τελιώνει με `?>`.
- Μεταξύ αυτών των ετικετών είναι μια δήλωση εκτύπωσης `singlePHP`.



```
File Edit Search Help
<?PHP
    print ("A simple initial script");
?>
```



# Εναλλακτικοί χωριστές PHP

---

- Μπορείτε εναλλακτικά να ξεκινήσετε scripts PHP σας με την ετικέτα `<script>` ως εξής:

```
<script language="PHP">  
print ("A simple initial script");  
</script>
```

- Εάν οι `short_open_tag` ενεργοποιούν τη διαμόρφωση του αρχείου, μπορείτε να χρησιμοποιήσετε το `<? και ?>`
- Αν `asp_tags` είναι ενεργοποιημένες στις ρυθμίσεις της PHP μπορείτε να χρησιμοποιήσετε `<% και %>` ως οριοθέτες.

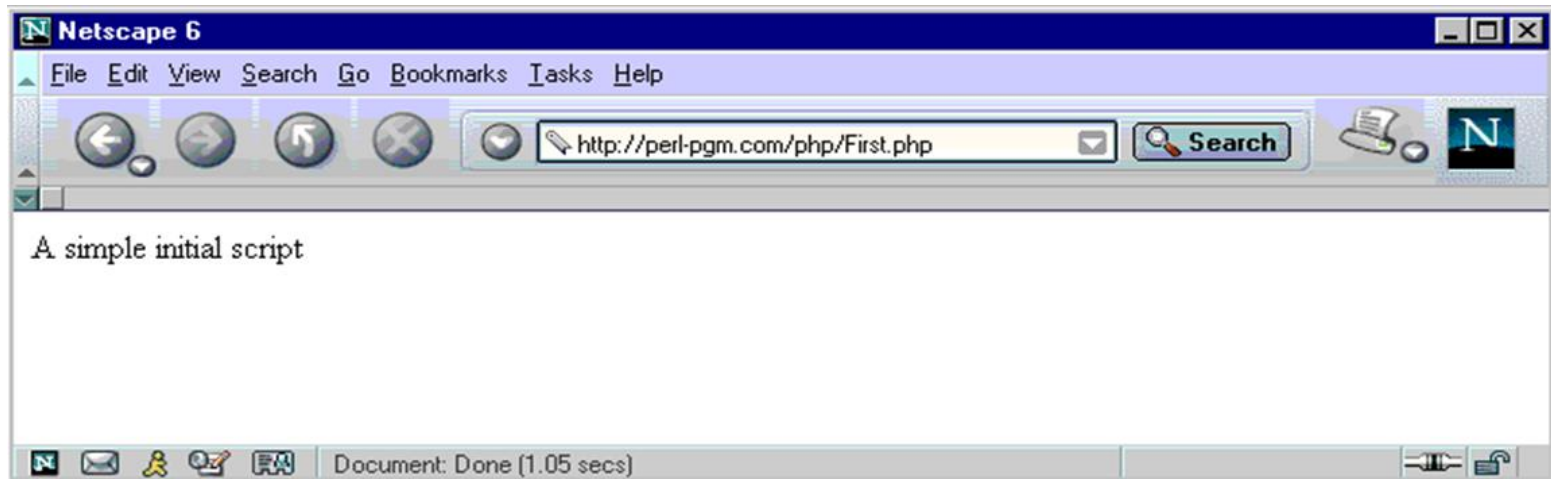
**Για μέγιστη συμβατότητα χρησιμοποιείτε πάντα**

```
<?php  
?>
```



# Πρόσβαση στο αρχείο χρησιμοποιώντας πρόγραμμα περιήγησης

---

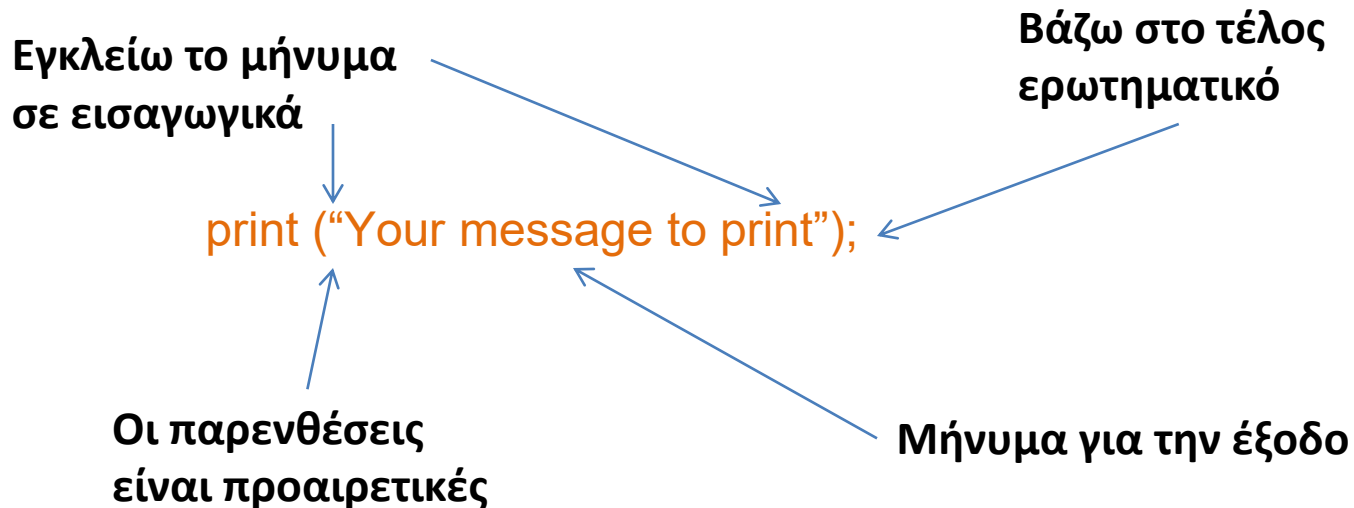




# Η σωστή σύνταξη

---

- Εάν έχετε ένα **συντακτικό λάθος** τότε έχετε γράψει μία ή περισσότερες δηλώσεις PHP που είναι γραμματικά λανθασμένη στη γλώσσα PHP.
- Η σύνταξη της πρότασης εκτύπωσης:



# Αν χρησιμοποιείτε εσφαλμένη σύνταξη

---

Ας υποθέσουμε ότι χρησιμοποιείτε τη λάθος σύνταξη:

1. `<?php`
2. `print ( "A simple initial script);`
3. `?>`



# Σχετικά με τη σύνταξη της PHP

---

Ορισμένα θέματα PHP σύνταξης:

- Προσέξτε να χρησιμοποιείτε τα εισαγωγικά, τις παρενθέσεις και τις αγκύλες σε ζεύγη.
- Οι περισσότερες εντολές PHP τελειώνουν με ερωτηματικό (;).
- Να είστε προσεκτικοί στις περιπτώσεις.
- Η PHP αγνοεί κενές θέσεις



# Ενσωμάτωση δηλώσεων PHP σε έγγραφο html

---

Ένας τρόπος για να χρησιμοποιήσετε την PHP είναι να ενσωματώσετε PHP scripts μέσα σε ετικέτες HTML σε έγγραφο HTML.

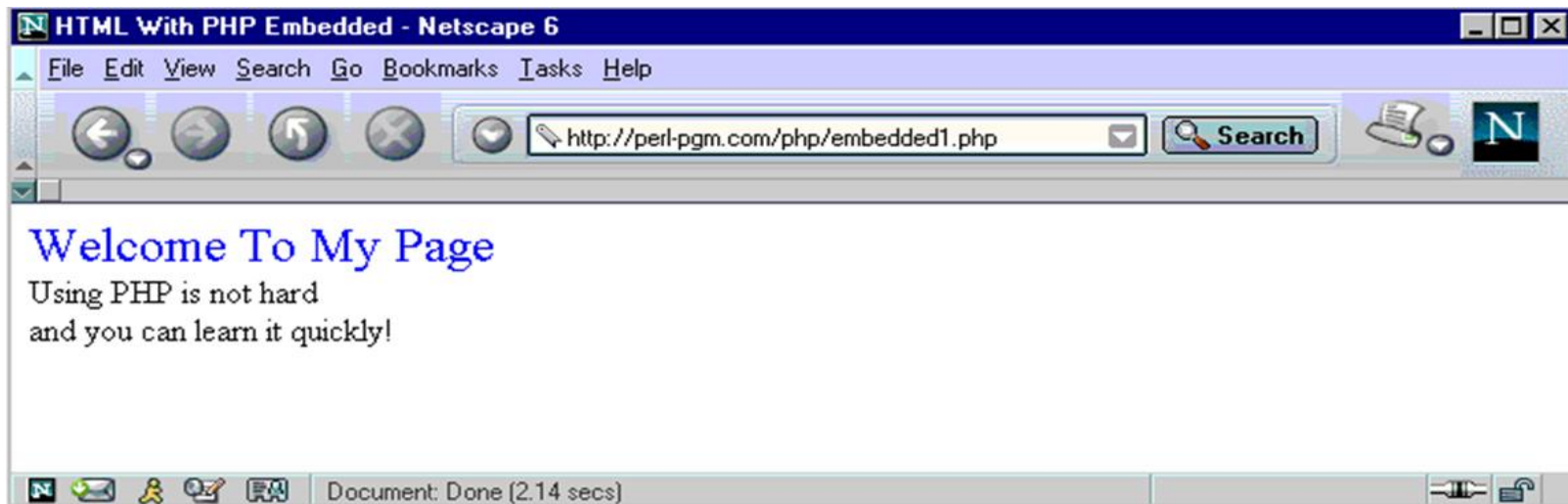
```
<html>
<head>
<title>HTML With PHP Embedded</title> </head>
<body>
<font size=5 color="blue">Welcome To My Page</font>
<?php
    print ("<br> Using PHP is not hard<br>");
?>
and you can learn to use it quickly!
</body></html>
```



# Θα εξάγει τα ακόλουθα ...

---

Όταν οι δηλώσεις PHP είναι ενσωματωμένες σε ένα έγγραφο HTML, οποιαδήποτε έξοδο από την PHP χρησιμοποιείται **ως μέρος του εγγράφου HTML**



# Χρησιμοποιήστε την ανάστροφη κάθετο («\») για να δημιουργήσετε HTML tags με print()

---

Μερικές φορές θέλετε να εμφανίσετε μια ετικέτα HTML που απαιτεί διπλά εισαγωγικά.

- Χρησιμοποιήστε την ανάστροφη κάθετο («\») για να επισημάνει ότι τα διπλά εισαγωγικά θα πρέπει να εξάγονται :

```
print("<font color=\"blue\">");
```

- Η παραπάνω δήλωση θα αποφέρει :

```
<font color="blue">
```



# Χρησιμοποιώντας σχόλια με *PHP scripts*

---

- Τα σχόλια, σας δίνουν τη δυνατότητα να συμπεριλάβετε περιγραφικό κείμενο μαζί με το PHP script.
- Οι γραμμές με τα σχόλια αγνοούνται όταν το σενάριο τρέχει.  
Δεν επιβραδύνουν το χρόνο εκτέλεσης.
- Τα σχόλια έχουν δύο κοινές χρήσεις.
  - Περιγράφουν το γενικό σκοπό σενάριο.
  - Περιγράφουν τις ιδιαίτερα δύσκολες γραμμές στο σενάριο



# Χρησιμοποιώντας σχόλια με *PHP scripts*

---

- Σχόλιο Σύνταξη - Χρήση //

```
<?php  
// This is a comment  
?>
```

- Μπορούν να μπουν στην ίδια γραμμή με μια δήλωση:

```
<?php  
print ("A simple initial script"); //Output a line  
?>
```





# Παράδειγμα *script* με σχόλια

---

```
<html> <head>
<title> Generating HTML From PHP</title> </head>
<body> <h1> Generating HTML From PHP</h1>
<?php
//
// Example script to output HTML tags
//
print ("Using PHP has <i>some advantages:</i>");
print ("<ul><li>Speed</li><li>Ease of use</li>
<li>Functionality</li></ul>"); //Output bullet list
print ("</body></html>");
?>
```



# Εναλλακτική σύνταξη σχολίων

---

- Η PHP επιτρέπει μερικούς επιπλέον τρόπους δημιουργίας σχολίων.

```
<?php  
phpinfo(); # This is a built-in function  

```

- Πολλαπλά σχόλια γραμμής.

```
<?php  
/*  
A script that gets information about the  
PHP version being used.  
*/  

```



# Περίληψη της PHP Intro

---

- Μπορείτε να ενσωματώσετε ένα PHP script μέσα σε ένα HTML έγγραφο ή να το τρέξετε ως stand-alone σενάριο.
- Για να αρχίσετε να εργάζεστε με την PHP θα πρέπει να έχετε ένα Web server με ενσωματωμένο PHP, ένα μηχάνημα πελάτη με βασικό πρόγραμμα επεξεργασίας κειμένου και λογισμικό FTP ή Telnet.
- Διαδικασία PHP script : γράψτε το PHP script, αντιγράψτε αυτό το αρχείο στο Web server και αποκτήστε πρόσβαση στο αρχείο μέσω Web browser.
- Τα σχόλια μπορούν να προχωρήσουν δύο προς τα εμπρός slashes (//).



# Στόχοι

---

- Να μάθετε πώς να αποθηκεύσετε και πώς να έχετε πρόσβαση σε δεδομένα σε PHP μεταβλητές.
- Να καταλάβετε πώς να δημιουργήσετε και να χειριστείτε αριθμητικές και μεταβλητές συμβολοσειρές.
- Να δείτε πώς μπορείτε να δημιουργήσετε φόρμες εισαγωγής HTML.
- Να μάθετε πώς να περνάνε τα δεδομένα από φόρμες HTML σε PHP scripts.



# Μεταβλητές στη PHP

---

- ❖ Όλες οι μεταβλητές στην PHP ξεκινούν με το σύμβολο \$. Οι μεταβλητές μπορούν να περιέχουν αλφαριθμητικά, αριθμούς πίνακες και τιμές τύπου true- false.

Πχ \$auth = true; \$num = 20; \$price = 14.95; \$size = "large";  
Λογική Ακέραια Πραγματική Συμβολοσειρά

- ❖ Κανόνες ονοματολογίας:

Ένα όνομα μεταβλητής

- Ξεκινά από γράμμα ή κάτω παύλα " "
- Περιέχει μόνο αλφαριθμητικούς χαρακτήρες και κάτω παύλα !!! (α-Z, 0-9, και \_)

- Δεν περιέχει κενά

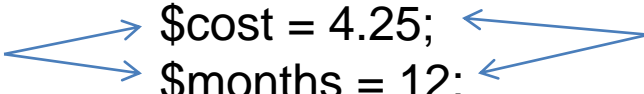

π.χ. \$my\_string "Η \$myString όχι \$my String



# Χρησιμοποιώντας μεταβλητές PHP

---

- Οι μεταβλητές χρησιμοποιούνται για την αποθήκευση και την πρόσβαση των δεδομένων στη μνήμη του υπολογιστή.
- Μια μεταβλητή **name** είναι μια ετικέτα που χρησιμοποιείται σε ένα σενάριο και αφορά τα δεδομένα.

Ονόματα μεταβλητών  `$cost = 4.25;`  
`$months = 12;`  Μεταβλητές νέες τιμές



# Αντιστοίχιση νέων τιμών σε μεταβλητές

---

Μπορείτε να ορίσετε νέες τιμές στις μεταβλητές:

```
$days = 3;
```

```
$newdays = 100;
```

```
$days = $newdays;
```

Στο τέλος αυτών των τριών γραμμών , `$days` και `$newdays` και οι δύο έχουν τιμές 100.



# Επιλογή μεταβλητών ονομάτων

---

Μπορείτε να επιλέξετε για οποιοδήποτε σύνολο χαρακτήρων ένα μεταβλητό όνομα στην PHP, αλλά πρέπει:

- Να χρησιμοποιήσετε το σύμβολο του δολαρίου (\$) σαν πρώτο χαρακτήρα.
- Να χρησιμοποιήσετε ένα γράμμα ή ένα χαρακτήρα υπογράμμισης (\_), σαν δεύτερο χαρακτήρα.

Σημείωση: Προσπαθήστε να επιλέξετε τα ονόματα των μεταβλητών που συμβάλλουν περιγραφή της λειτουργίας τους. Για παράδειγμα, ο μετρητής \$ είναι πιο περιγραφικός από το \$ c ή το \$ ctr.





# Συνδυάζοντας μεταβλητές και δήλωση εκτύπωσης

---

Δηλαδή, για να εκτυπώσετε την τιμή του \$ x, γράψτε την ακόλουθη πρόταση PHP.

```
print ("$x");
```

Ο ακόλουθος κώδικας θα εμφανίσει “ Bryant is 6 years old”.

```
$age=6;  
print ("Bryant is $age years old.");
```



# Ένα ολοκληρωμένο παράδειγμα

---

```
<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>
```

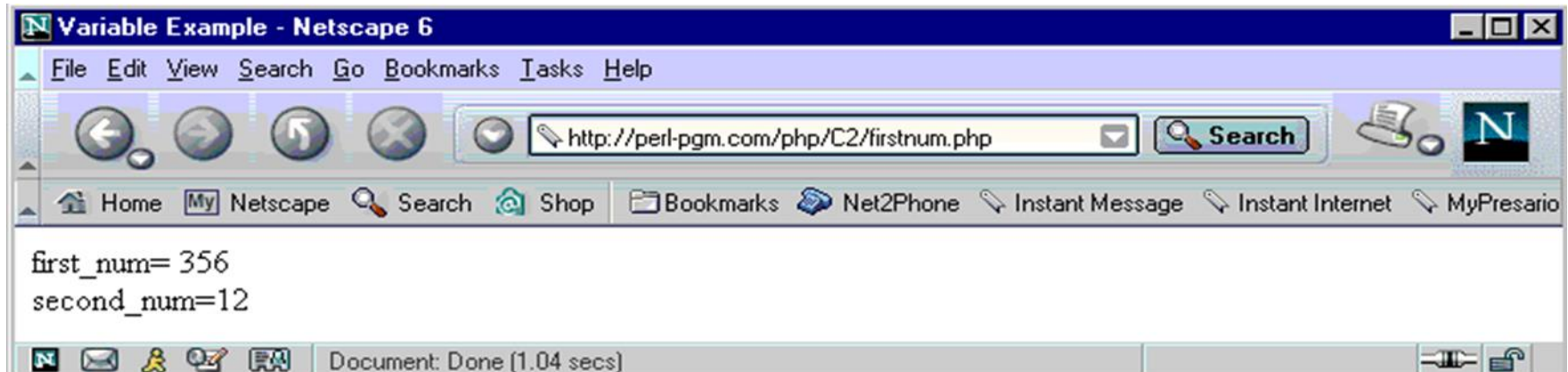


# Ένα ολοκληρωμένο παράδειγμα

---

Ο προηγούμενος κώδικας μπορεί να εκτελεστεί σε:

<http://webwizard.aw.com/~phppgm/C2/firstnum.php>




# Τα αλφαριθμητικά μπορούν να συνδεθούν

---

```
<html>
<body>
<?php
$txt="Hello World";
echo $txt;
?>
</body>
</html>
```

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



Ο τελεστής . < τελεία >  
χρησιμοποιείται για την  
συνένωση αλφαριθμητικών

# Η αρχική τιμή μιας μεταβλητής δηλώνει το είδος της

---

```
$a = "Hello ";  
$b = $a . "World!";  
// το $b περιέχει το "Hello World!"
```

```
$a = "Hello ";  
$a .= "World!";  
// το $a περιέχει το 'Hello World!'
```

```
<?php  
$foo = "0";  
// $foo είναι string (ASCII 48)
```

```
$foo += 2;  
// $foo integer (2)
```

```
$foo = $foo + 1.3;  
// $foo float (3.3)
```

```
$foo = 5 + "10 Little Piggies";  
// $foo integer (15)
```

```
$foo = 5 + "ten Small Pigs";  
// $foo integer (5)
```

```
?>
```



# Διπλά ή μονά εισαγωγικά

---

```
<?php
$identity = " James Bond ";
$car = 'BMW';
/* Θα περιέχει τη συμβολοσειρά "James Bond drives a BMW" */
$sentence = "$identity drives a $car";
/* Θα περιέχει τη συμβολοσειρά
"$identity drives a $car" */
$sentence = "$identity drives a $car";
$statement = "It's hot outside"; //OXI
$statement = "It\'s hot outside"; //NAI
?>
```

Όταν χρησιμοποιούμε διπλά εισαγωγικά οι μεταβλητές αντικαθίστανται από την τιμή τους.

Δε συμβαίνει το ίδιο στα μονά εισαγωγικά.

Όταν στη συμβολοσειρά έχουμε ειδικούς χαρακτήρες πρέπει να χρησιμοποιούμε “\”



# Χρησιμοποιώντας αριθμητικούς χειριστές

---

Μπορείτε να χρησιμοποιήσετε **φορείς**, όπως ένα σύμβολο συν (+) για προσθήκη και σύμβολο μείον (-) για αφαίρεση για να δημιουργήσετε μαθηματικές εκφράσεις.

Για παράδειγμα

```
<?php
$apples = 12;
$oranges = 14;
$total_fruit = $apples + $oranges;
print ("The total number of fruit is $total_fruit");
?>
```

Οι δηλώσεις PHP αυτές θα έχουν έξοδο « The total number of fruit is 26.»



# Κοινή Αριθμητική ΡΗΡ

---

Τελεστές	Πράξη	Παράδειγμα	Αποτέλεσμα
+	Πρόσθεση	$\$x = 2 + 2;$	$\$x$ έχει εκχωρηθεί 4
-	Αφαίρεση	$\$y = 3;$ $\$y = \$y - 1;$	$\$y$ έχει εκχωρηθεί 2
/	Διαίρεση	$\$y = 14 / 2;$	$\$y$ έχει εκχωρηθεί 7
*	Πολλαπλασιασμός	$\$z = 4;$ $\$y = \$z * 4;$	$\$y$ έχει εκχωρηθεί 16
%	Υπόλοιπο	$\$y = 14 \% 3;$	$\$y$ έχει εκχωρηθεί 2





# Κοινή Αριθμητική PHP

---

Τελεστής	Παράδειγμα	Είναι το ίδιο με
=	$X = Y$	$X = Y$
+=	$X += Y$	$X = X+Y$
-=	$X -= Y$	$X = X-Y$
*=	$X *= Y$	$X = X*Y$
/=	$X /= Y$	$X = X/Y$
%=	$X \% = Y$	$X = X\%Y$



# Σύγκριση τελεστών

---

Τελεστής	Περιγραφή	Παράδειγμα
==	Είναι ίσο με	4 == 2 (false)
!=	Δεν είναι ίσο με	4 != 2 (true)
>	Είναι μεγαλύτερο από	4 > 3 (true)
<	Είναι μικρότερο από	4 < 4 (false)
>=	Είναι μεγαλύτερο από	4 >= 4 (true)
<=	Είναι μικρότερο από	4 <= 3 (false)



# Λογικοί τελεστές

---

Τελεστής	Περιγραφή	Παράδειγμα
&&	and	X=6 Y=3 (x<10 && y>1) Αποτέλεσμα true
	or	X=6 Y=3 (x==5    y==5) Αποτελέσματα false
!	not	X=6 Y=3 !(x==y) Αποτελέσματα true



# Ένα ολοκληρωμένο παράδειγμα

---

```
<html>
<head> <title>Variable Example </title> </head>
<body>
<?php
$columns = 20;
$rows = 12;
$total_seats = $rows * $columns;

$ticket_cost = 3.75;
$total_revenue = $total_seats * $ticket_cost;

$building_cost = 300;
$profit = $total_revenue - $building_cost;

print ("Total Seats are $total_seats <br>");
print ("Total Revenue is $total_revenue <br>");
print ("Total Profit is $profit");
?> </body> </html>
```

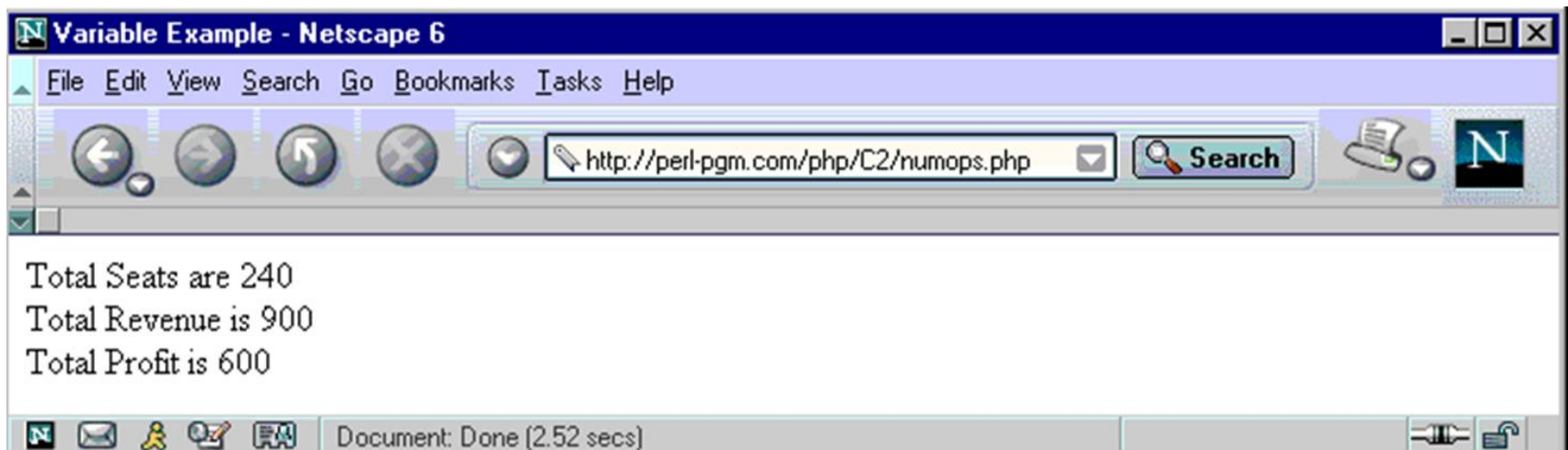


# Ένα ολοκληρωμένο παράδειγμα

---

Ο προηγούμενος κώδικας μπορεί να εκτελεστεί σε:

<http://webwizard.aw.com/~phppgm/C2/numops.php>



# Προσοχή : Στη χρήση μεταβλητών με αόριστες τιμές

---

Εάν χρησιμοποιήσετε κατά λάθος μια μεταβλητή που δεν έχει τιμή που έχει εκχωρηθεί σε αυτήν, δεν θα έχει αξία (ονομάζεται μηδενική τιμή) Όταν μια μεταβλητή με τιμή null χρησιμοποιείται σε μια έκφραση PHP, δεν μπορεί να δημιουργήσει σφάλμα και μπορεί να ολοκληρώσει την έκφραση.

Για παράδειγμα, το ακόλουθο script PHP θα έχει έξοδο  $x = y = 4$

```
<?php  
$y = 3;  
$y=$y + $x + 1; // $x has a null value  
print ("x=$x y=$y");  
?>
```



# Σύνταξη σύνθετων εκφράσεων

---

**Οι κανόνες προτεραιότητας τελεστών** καθορίζουν τη σειρά με την οποία την αξιολογούν οι φορείς εκμετάλλευσης.

Για παράδειγμα  
 $\$x = 5 + 2 * 6;$

Η αξία των  $\$ x$  είναι ή 42 ή 17, ανάλογα με τη σειρά αξιολόγησης.

Ο πολλαπλασιασμός αξιολογείται πριν την πρόσθεση αυτή η έκφραση αξιολογείται ως 17



# ***Κανόνες Προτεραιότητας ΡΗΡ***

---

Η ΡΗΡ ακολουθεί τους κανόνες προτεραιότητας που αναφέρονται παρακάτω:

- Πρώτον, αξιολογεί τους φορείς εντός παρενθέσεων.
- Στη συνέχεια αξιολογεί τους τελεστές πολλαπλασιασμού και διαίρεσης
- Τέλος, αξιολογεί τους τελεστές πρόσθεσης και αφαίρεσης





# Κανόνες Προτεραιότητας PHP

---

Για παράδειγμα, οι πρώτες 2 δηλώσεις αξιολογούνται σε 80 ενώ θα διαρκέσει μέχρι το 180

- $x = 100 - 10 * 2;$
- $y = 100 - (10 * 2);$
- $z = (100 - 10) * 2;$



# Ένα ολοκληρωμένο παράδειγμα

---

```
<html>
<head> <title>Expression Example </title> </head>
<body>
<?php
$grade1 = 50;
$grade2 = 100;
$grade3 = 75;
$average = ($grade1 + $grade2 + $grade3) / 3;
print ("The average is $average");
?> </body> </html>
```

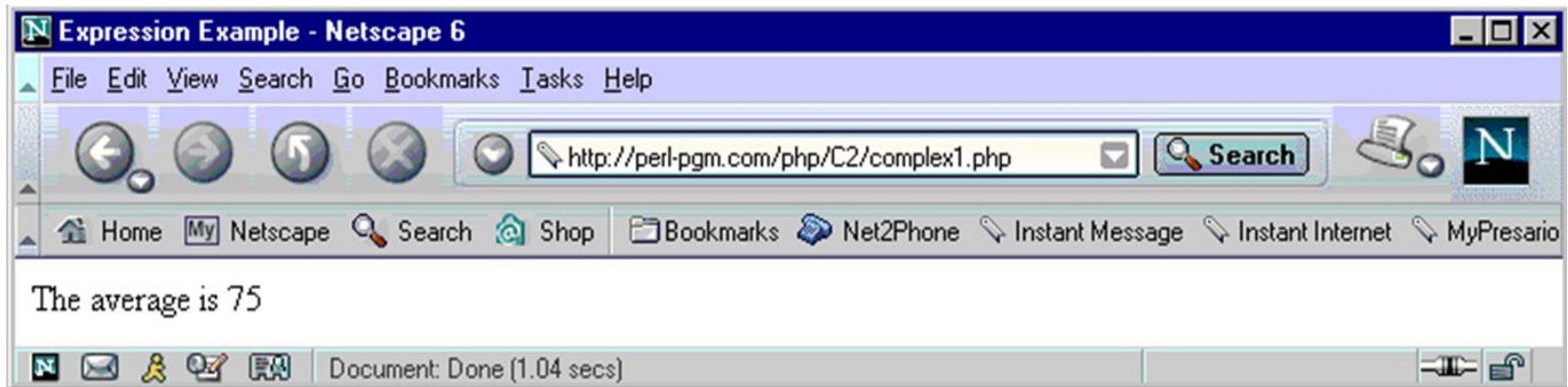


# Ένα ολοκληρωμένο παράδειγμα

---

Ο προηγούμενος κώδικας μπορεί να εκτελεστεί σε :

<http://webwizard.aw.com/~phppgm/C2/complex1.php>



# Δουλεύοντας με μεταβλητές συμβολοσειρών

---

Τα strings χαρακτήρων χρησιμοποιούνται σε scripts για να κρατήσουν τα δεδομένα, όπως ονόματα πελατών, διευθύνσεις, προϊόντα ονόματα και περιγραφές.

Εξετάστε το ακόλουθο παράδειγμα

- `$name="Christopher";`
  - `$preference="Milk Shake";`
- 
- Στη μεταβλητή `$ name` έχει εκχωρηθεί “Christopher” και στη μεταβλητή `$preference` έχει εκχωρηθεί “Milk Shake”



# Προσοχή : μην αναμιγνύετε τύπους αριθμητικών μεταβλητών

---

- Να είστε προσεκτικοί στο να μην αναμειγνύετε string και τύπους αριθμητικών μεταβλητών.
- Για παράδειγμα, μπορείτε να φανταστείτε τις ακόλουθες δηλώσεις που δημιουργούν ένα μήνυμα σφάλματος , αλλά δεν θα το κάνουν. Αντ 'αυτού, θα εξάγει "y = 1"

```
<?php  
$x ="banana";  
$sum = 1 + $x;  
print ("y=$sum");  
?>
```



# Χρησιμοποιώντας τον τελεστή *concatenate*

---

Ο τελεστής concatenate συνδυάζει δύο ξεχωριστές μεταβλητές συμβολοσειράς σε μία.

Για παράδειγμα

```
$fullname = $firstname . $lastname;
```

\$fullname θα λάβουν τις τιμές συμβολοσειράς των \$firstname και \$lastname συνδέονται μεταξύ τους.

Για παράδειγμα

```
$firstname = "John";
```

```
$lastname = "Smith";
```

```
$fullname = $firstname . $lastname;
```

```
print ("Fullname=$fullname");
```



# TIP : Ένας ευκολότερος τρόπος να συγκολλήσετε τα Strings

---

Μπορείτε επίσης να χρησιμοποιήσετε διπλά εισαγωγικά για να δημιουργήσετε απευθείας συγκόλληση (concatenation).

Για παράδειγμα

- `$FullName2 = "$FirstName $LastName";`
- This statement has the same effect as
- `$FullName2 = $FirstName . " " . $LastName;`



# Η συνάρτηση *strlen()*

---

- Οι περισσότερες string συναρτήσεις χρειάζονται να στέλνετε ένα ή περισσότερα επιχειρήματα.
- Τα επιχειρήματα είναι τιμές εισόδου που οι συναρτήσεις τα χρησιμοποιούν στην επεξεργασία που κάνουν.
- Συχνά οι συναρτήσεις επιστρέφουν μια τιμή στο script με βάση τα επιχειρήματα εισαγωγής. Για παράδειγμα

`$len = strlen ($name);`

Παίρνει τον αριθμό των χαρακτήρων στο \$name

Μεταβλητές ή τιμές που δουλεύουν μαζί

Όνομα συνάρτησης





# Παράδειγμα της συνάρτησης `strlen()`

---

```
<?php
    $comments = "Good Job";
    $len = strlen($comments);
    print ("Length=$len");
?>
```

**Αυτό το σενάριο PHP θα εξάγει “Length=8”.**



# Η συνάρτηση *trim()*

---

Η λειτουργία αυτή αφαιρεί κενούς χαρακτήρες από την αρχή και το τέλος μιας συμβολοσειράς. Για την παράδειγμα, εξετάστε το ακόλουθο σενάριο

- `<?php`
- `$in_name = " Joe Jackson ";`
- `$name = trim($in_name);`
- `print ("name=$name$name");`
- `?>`



# Οι συναρτήσεις *strtolower()* και *strtoupper()*

---

Οι συναρτήσεις αυτές επιστρέφουν το string εισόδου σε όλα κεφαλαία ή όλα πεζά γράμματα, αντίστοιχα.

Για παράδειγμα

```
<?php
$inquote = "Now Is The Time";
$lower = strtolower($inquote);
$upper = strtoupper($inquote);
print ("upper=$upper lower=$lower");
?>
```

Τα παραπάνω θα βγάλουν έξοδο “upper=NOW IS THE TIME lower=now is the time”.



# Η συνάρτηση *strtoupper()*

---

Η Substr έχει την ακόλουθη γενική μορφή:

Εκχωρήστε το  
εξαγόμενο sub-  
string σε αυτή τη  
μεταβλητή

```
$part = substr ( $name, 0, 5);
```

Απόσπασμα από το  
string της μεταβλητής

Αρχική θέση για να  
ξεκινήσετε την  
εξαγωγή από δω

Αριθμός χαρακτήρων  
για να εξάγετε. (Αν παραλειφθεί θα  
συνεχίστε να εξάγει μέχρι το τέλος  
της συμβολοσειράς.



# Η συνάρτηση *substr()*

---

- Η *substr()* συνάρτηση απαριθμεί θέσεις χαρακτήρων ξεκινώντας με το 0 (όχι με 1).
- Για παράδειγμα, στη συμβολοσειρά "Homer", το «H» θα είναι θέση 0, το «o» θα είναι στη θέση 1, στη θέση 2 το "m" και ούτω καθεξής.
- Για παράδειγμα, τα ακόλουθα θα έχουν έξοδο "Month=12 Day=25".

```
<?php
$date = "12/25/2002";
$month = substr($date, 0, 2);
$day = substr($date, 3, 2);
print ("Month=$month Day=$day");
?>
```



# Η συνάρτηση *substr()*

---

- Ως ένα άλλο παράδειγμα, εξετάστε την ακόλουθη χρήση της συνάρτησης `substr ()`.
- Δεν περιλαμβάνει το τρίτο επιχείρημα (και έτσι επιστρέφει μια συμβολοσειρά από την αρχική θέση μέχρι το τέλος της συμβολοσειράς αναζήτησης).

```
<?php
$date = "12/25/2002";
$year = substr($date, 6);
print ("Year=$year");
?>
```

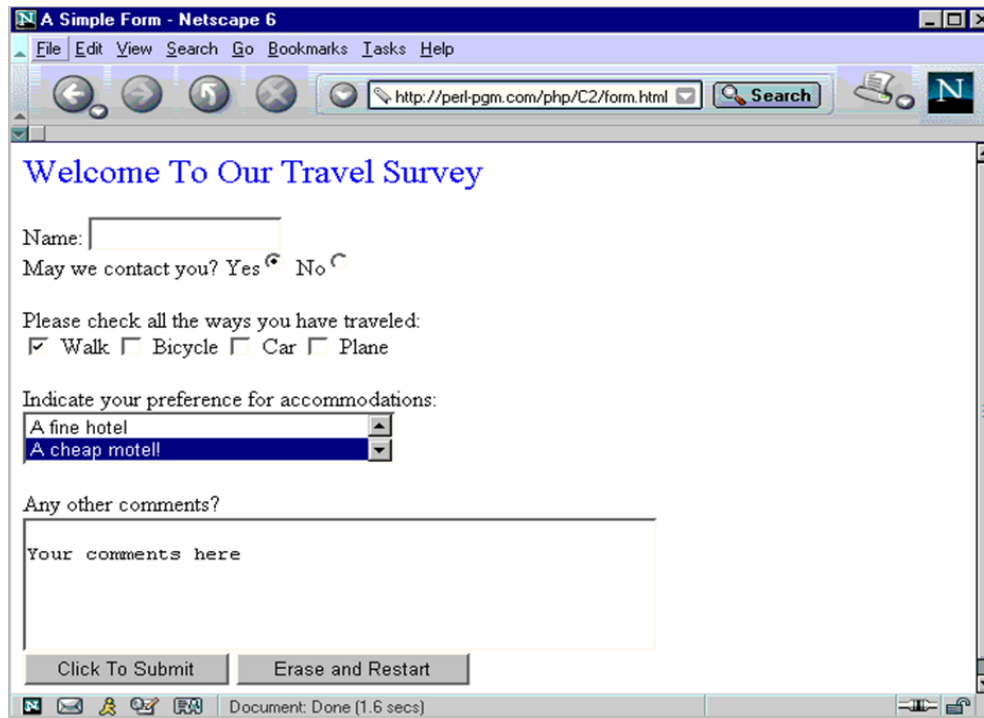
- Το παραπάνω τμήμα script θα έχει έξοδο “Year=2002”.



# Δημιουργία εντύπων εισαγωγής HTML

---

Οι HTML Forms δεν αποτελούν μέρος της γλώσσας PHP, αλλά είναι σημαντικός τρόπος για την αποστολή δεδομένων στο scripts



The screenshot shows a Netscape 6 browser window titled "A Simple Form - Netscape 6". The address bar displays "http://perl-pgm.com/php/C2/form.html". The page content includes a heading "Welcome To Our Travel Survey" in blue. Below the heading, there is a form with the following elements:

- A text input field labeled "Name:".
- Radio buttons for "May we contact you? Yes" (selected) and "No".
- A section titled "Please check all the ways you have traveled:" with checkboxes for "Walk" (checked), "Bicycle", "Car", and "Plane".
- A section titled "Indicate your preference for accommodations:" with a dropdown menu showing "A fine hotel" and "A cheap motel" (selected).
- A text area labeled "Any other comments?" with the placeholder text "Your comments here".
- Two buttons at the bottom: "Click To Submit" and "Erase and Restart".

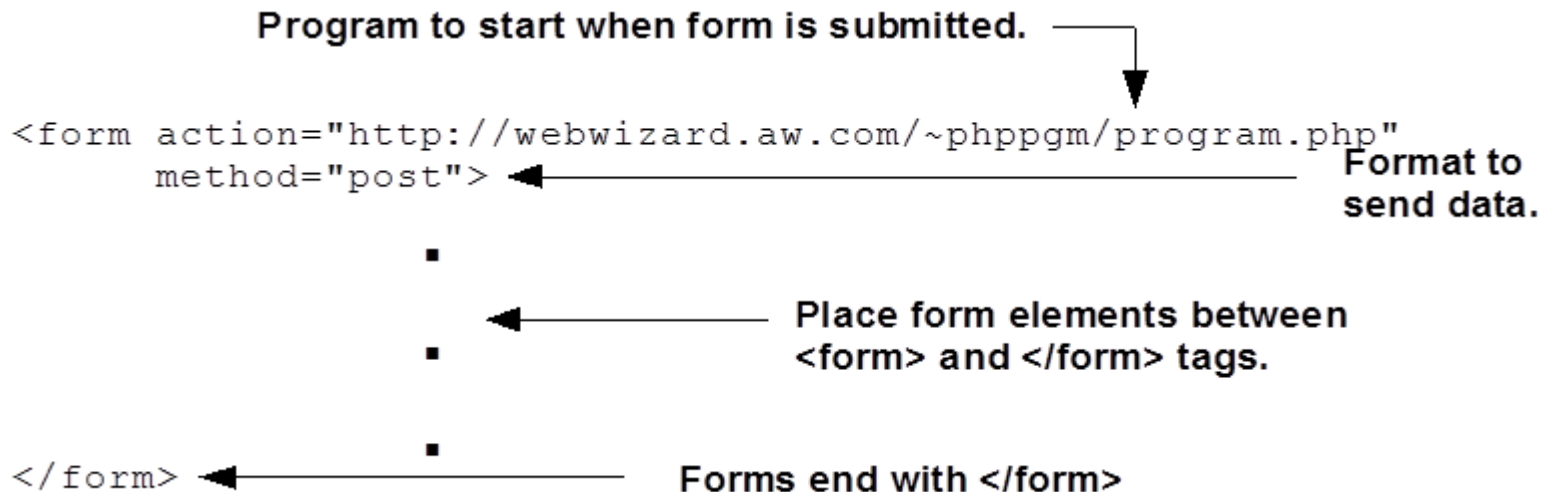
The browser's status bar at the bottom indicates "Document: Done (1.6 secs)".



# Αρχίζοντας και τελειώνοντας τις HTML φόρμες

---

Μπορείτε να δημιουργήσετε φόρμες HTML, χρησιμοποιώντας τις HTML `<form>` και `</form>` ετικέτες.





# Προσοχή : τα *password boxes* δεν είναι ασφαλή

---

- Όταν ο χρήστης υποβάλλει τη φόρμα, κάθε εισαγωγή δεδομένων αποστέλλεται σε καθαρό κείμενο (μη κρυπτογραφημένο) όπως και κάθε άλλο πεδίο φόρμας HTML.
- Κάποιος με πρόσβαση στο δίκτυο θα μπορούσε, ως εκ τούτου, να διαβάσει τον κωδικό πρόσβασης που μεταφέρεται.
- Για το λόγο αυτό, οι περισσότερες εφαρμογές Web δεν χρησιμοποιούν αυτή την προσέγγιση για να λαμβάνουν και να μεταδίδουν κωδικούς πρόσβασης.



# Η ασφάλεια της PHP βελτιώθηκε με το *Register\_Globals*

---

- Από την PHP 4.2.1, η προεπιλεγμένη ρύθμιση PHP απαιτεί ένα διαφορετικό μηχανισμό για να λαμβάνει την είσοδο για λόγους ασφαλείας (από αυτόν που μόλις δείξαμε).
- Τεχνικές λεπτομέρειες: είναι μια επιλογή διαμόρφωσης PHP για να στρέψετε το REGISTER\_GLOBALS OFF (νέα προεπιλογή) ή ON στο αρχείο ρυθμίσεων php.ini.
- Εάν η περιοχή σας έχει register\_globals OFF πρέπει να χρησιμοποιήσετε διαφορετικό μηχανισμό για να λαμβάνετε μεταβλητές φόρμας HTML.



# Πως μπορώ να πω ότι το *Register\_Global* είναι απενεργοποιημένο

---

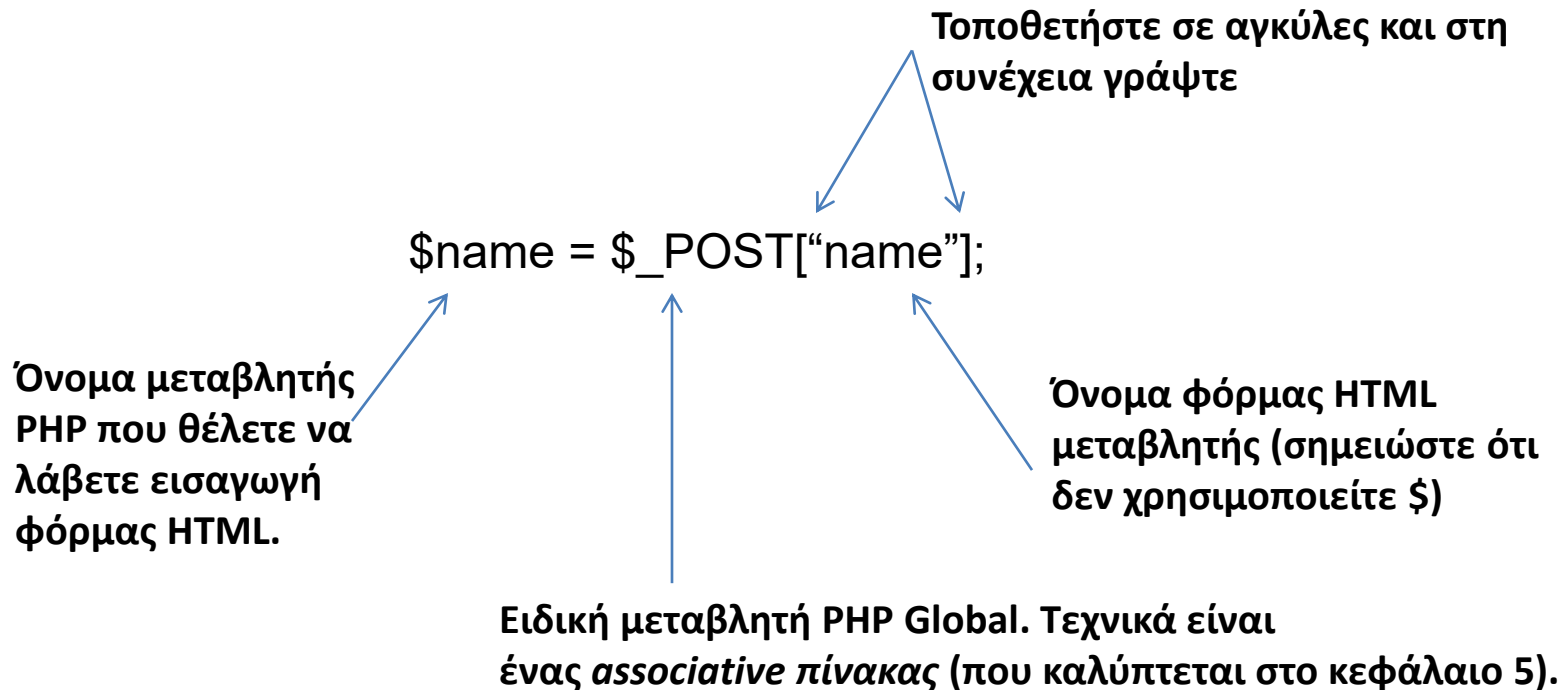
- Εισάγετε το παρακάτω script PHP και εκτελέστε το.
- `<?PHP phpinfo(); ?>`
- Αναζητήστε μέσω της εξόδου για REGISTER\_GLOBALS και δείτε αν έχει ρυθμιστεί σε OFF ή ON.
- Αν είναι μακριά πρέπει να χρησιμοποιήσετε τον ακόλουθο τρόπο για να λάβετε δεδομένα εισόδου.



# Λήψη δεδομένων εισόδου με *Register\_Globals* απενεργοποιημένο

---

Για να λάβετε δεδομένα με REGISTER\_GOBALS OFF χρησιμοποιείτε μια ειδική μεταβλητή που ονομάζεται \$ POST



# Ολοκληρωμένο παράδειγμα όταν το **REGISTER\_GLOBALS** είναι απενεργοποιημένο

---

Ας υποθέσουμε ότι η μορφή HTML χρησιμοποιεί τα εξής:

Εισάγετε τη διεύθυνση ηλεκτρονικού ταχυδρομείου:

```
<input type="text" size="16" maxlength="20" name="email">
```

Στη συνέχεια, μπορείτε να λάβετε είσοδο ως εξής:

```
<html>
<head><title> Receiving Input </title> </head>
<body>
$email = $_POST["email"];
$contact = $_POST["contact"];
<font size=5>Thank You: Got Your Input.</font>
<?php
    print ("<br>Your email address is $email");
    print ("<br> Contact preference is $contact");
?>
```

**Ο ενδεικτικός τρόπος είναι μέσω `$_POST` , `$_GET`**

---



# Σύνοψη μεταβλητών

---

- Οι μεταβλητές χρησιμοποιούνται για αποθήκευση και πρόσβαση σε δεδομένα στη μνήμη του υπολογιστή. Μπορείτε να συσχετίσετε μια τιμή με μια μεταβλητή, να αλλάξετε την τιμή, να την εκτυπώσετε, και να εκτελέσετε πολλές διαφορετικές λειτουργίες σε αυτό.
- Η PHP υποστηρίζει τόσο αριθμητικές όσο και string μεταβλητές. Οι string μεταβλητές χρησιμοποιούν διαφορετικές μεθόδους χειρισμού τιμών (για παράδειγμα συσχέτισης) από τις αριθμητικές μεταβλητές.



# Περίληψη *Forms*

---

- Μπορείτε να χρησιμοποιήσετε HTML φόρμες για να περάσετε δεδομένα σε PHP scripts. Τα στοιχεία της μορφής HTML περιλαμβάνουν πλαίσια κειμένου, περιοχές κειμένου, πλαίσια κωδικού πρόσβασης, πλαίσια ελέγχου, κουμπιά επιλογής και λίστες επιλογών.
- Τα PHP scripts μπορούν να λάβουν τιμές εισόδου μορφής στοιχείου με τη χρήση ενός PHP ονόματος μεταβλητής που ταιριάζει με αυτό που καθορίζεται στο όνομα του στοιχείου φόρμας.
- Αν οι `register_globals` είναι απενεργοποιημένοι στην εγκατάστασή σας θα πρέπει να πάρετε είσοδο δεδομένα με χρήση του `POST ["var_name"]`;

**Εμείς θα χρησιμοποιούμε πάντα τον πίνακα  
`$_POST`  
ή τον πίνακα  
`$_GET`**



# Δομή Ελέγχου *if*

---

```
<?php  
if (Συνθήκη)  
{  
  εντολές;  
}  
?>
```

```
<? php  
if ($grade <= 5)  
{  
  echo "Next Time!";  
}  
?>
```





# Δομή Ελέγχου *if/else*

---

```
<? php
if (συνθήκη)
{
Εντολές;
}
else
{
Εντολές ;
}
?>
```

```
<? php
$d=date("D");
if ($d=="Fri")
{
echo "Hello! <br>";
echo "Have a nice weekend!";
echo "See you on <l>Monday!</l>";
}
else
echo "Have a nice day";
?>
```



# Δομή πολλαπλών ελέγχων με *if/elseif/else*

```
<? php
if (συνθήκη #1)
{
Εντολές;
}
else if (συνθήκη #2)
{
Εντολές;
}
...
elseif (συνθήκη #n)
{
Εντολές;
}
else
{
Εντολές;
}
```

```
<? php
if ($country == "UK")
{
$capital = "London";
}
elseif ($country == "US")
{
$capital = "Washington";
}
elseif ($country == "FR")
{
$capital = "Paris";
}
else
{
$capital = "Unknown"
}
?>
```



# Δομή ελέγχου *switch/case*

---

```
<? php
switch (Μεταβλητή)
{
Περίπτωση #1:
Εντολές;
break;
Περίπτωση #2:
Εντολές;
break;
...
Περίπτωση #n:
Εντολές;
break;
default;
Εντολές;
?>
```

```
<? php
switch ($country)
{
case "UK":
$capital = "London";
break;
case "US":
$capital = "Washington";
break;
case "FR":
$capital = "Paris";
break;
default:
$capital = "Unknown";
}
?>
```



# Ο υποθετικός τελεστής (συνθήκη)?

---

```
<?php
if ($grade <5)
{
$msg = "you have failed";
}
else
{
$msg = "you have passed"
}
?>
```

```
<? php
$msg = $grade < 5?
"you have failed":
"you have passed";
?>
```



# Η επαναληπτική δομή while()

---

```
<? php
while (συνθήκη)
{
Εντολές;
}
?>
```

```
<html>
<body>
<?php
$i=1;
while ($i<=5)
{
echo "the number is" .$i. "<br>";
$i++;
}
?>
</body>
</html>
```



# Η επαναληπτική δομή *do...while()*

---

```
<?php
do
{
Εντολές;
} while (συνθήκη);
?>
```

```
<? php
$upperLimit = 10;
$lowerLimit = 1;
do
{
echo "$lowerLimit x $upperLimit =". (lowerLimit *$ upperLimit). "<br>";
$lowerLimit++;
}while ($lowerLimit <= $upperLimit);
?>
```



# Η επαναληπτική δομή *for()*

---

```
for (αρχικοποίηση;  
    Συνθήκη; αύξηση)  
{  
    Εντολές;  
}
```

```
<html>  
<body>  
<? php  
for ($i=1; $i<=5; $i++)  
{  
    echo "Hello World!<br>";  
}  
?>  
</body>  
</html>
```



# Πίνακες στη PHP

---

## Παράδειγμα 1

```
$pin[0] = "Γιάννης";  
$pin[1] = "Νίκος";  
$pin[2] = "Ελένη";
```

Δεν χρειάζεται να  
ορίσουμε μέγεθος

## Παράδειγμα 2

```
$pin = array ("Γιάννης",  
             "Νίκος", "Ελένη");
```

```
// Τυπώνει το περιεχόμενο  
// στην θέση 1 echo $pin[1];
```





# Hash Πίνακες (χωρίς αριθμητικό *index*) στη PHP

---

## Παράδειγμα 1

```
$age["Γιάννης"]=20;  
$age["Ελένη"]=25;  
$age["Μαρία"]=37;
```

## Παράδειγμα 2

```
$age = array ("Γιάννης"=>20,  
             "Ελένη" =>25,  
             "Μαρία"=> 37);
```

Αντί για INDEX χρησιμοποιούμε ένα όνομα

```
// Τυπώνει την ηλικία της Μαρίας echo $age["Μαρία"];
```



# Επαναληπτική Δομή πρόσβασης Πίνακα *foreach()*

---

```
foreach (πίνακας as μεταβλ.)  
{  
  // εντολές;  
}
```

```
<? php  
$pin[0]="Γιάννης";  
$pin[1]="Νίκος";  
$pin[2]="Ελένη";
```

```
foreach ($pin as $v)  
{  
  Echo $v;  
}  
?>
```

