



# Προγραμματισμός Διαδικτύου

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

Τμήμα της παρουσίασης δημιουργήθηκε από τον κ. Παναγιώτη Συμεωνίδη (ICTE/UOWM, 2008-2009)

---



# 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

---

- ✓ Εκμάθηση διάφορων λειτουργιών PHP
- ✓ Καλός χειρισμός συναρτήσεων
- ✓ Χρήση οπών δήλωσης για σύγκριση τιμών αριθμητικών και string δεδομένων
- ✓ Χρήση δηλώσεων επανάληψης
- ✓ Χρήση λογικών τελεστών για δημιουργία com-round υποθετικών δοκιμαστικών δηλώσεων



# Προγραμματισμός διαδικτύου

---

## Συναρτήσεις, Αποθήκευση & Ανάκληση, Συμβολοσειρές

### Περιεχόμενα

5.1 Βασικές λειτουργίες PHP

5.2 Συναρτήσεις `abs()`, `sqrt()`, `round()`, `rand()`, `print()`, `date()`

5.3 Η εντολή `include`

5.4 Η εντολή `get`

5.5 Η συνάρτηση `isset()`

5.6 Αρχεία

5.7 Ρήτρα `else if`, `else`, `switch`

5.8 Δήλωση επανάληψης `for`, `while`

5.9 Λογικός τελεστής `OR`, `NOT`



# Χρησιμοποιώντας ορισμένες λειτουργίες Βασικές PHP

---

Έχουμε προηγουμένως συζητήσει για συναρτήσεις όπως οι `strlen()`, `trim()`, `strtolower()`, `strtoupper()`, and `substr()`.

- Στην ενότητα αυτή εξετάζουμε πολλές άλλες χρήσιμες συναρτήσεις μερικές βασικές αριθμητικές PHP λειτουργίες-Π.χ., η απόλυτη τιμή [`abs ()`], τα ετραγωνική ρίζα [`sqrt ()`], στρογγυλό [`round ()`], ακέραιος ελεγκτής [`is_numeric ()`] και τη δημιουργία τυχαίων αριθμών [`rand ()`].
- Τη συνάρτηση `print()` -Θα καλύψουμε με περισσότερες λεπτομέρειες.
- Τη συνάρτηση `date` - Σήμερα θα συζητήσουμε τη χρήση της συνάρτησης `date()` που καθορίζει τις πληροφορίες ημερομηνίας και ώρας.



# Ορισμένες Βασικές Λειτουργίες PHP

---

- Απόλυτη τιμή
- Τετραγωνική ρίζα
- Κύκλος
- Ακέραιος πλέγμα
- Τυχαία γενιά αριθμό



# Η συνάρτηση *abs* ()

---

Η συνάρτηση απόλυτης τιμής παίρνει ένα ενιαίο αριθμητικό όρισμα και επιστρέφει την απόλυτη τιμή.

Για παράδειγμα :

$x = \text{abs} (-5)$ .

$y = \text{abs} (42)$ .

Εκτύπωση " $x = x \quad y = y$ ";

Θα έχει εξοδο:

$x = 5 \quad y = 42$



# Η συνάρτηση `sqrt ()`

---

Η συνάρτηση τετραγωνικής ρίζας λαμβάνει ένα ενιαίο αριθμητικό όρισμα και επιστρέφει την τετραγωνική ρίζα του.

Για παράδειγμα :

```
$ x = sqrt (25).  
$ y = sqrt (24).  
print "x=$x y=$y";
```

Θα έχει έξοδο :

```
x = 5 y = 4.898979485566
```





# Η συνάρτηση *round()*

---

Η συνάρτηση `round()` παίρνει ένα ενιαίο αριθμητικό επιχείρημα και επιστρέφει τον αριθμό στρογγυλοποιημένο προς τα πάνω ή προς τον πλησιέστερο ακέραιο αριθμό.

Για παράδειγμα:

```
$x=round(-5.456);  
$y=round(3.7342);  
print "x=$x y=$y";
```

Θα έχει έξοδο: `x=-5 y=4`



# Η συνάρτηση *round()*

---

Μπορείτε να συμπεριλάβετε δεύτερο επιχείρημα για να καθορίσει τον αριθμό των ψηφίων μετά το δεκαδικό σημείο προς στρογγυλοποίηση.

Για παράδειγμα:

```
$x=round(-5.456,2);  
$y=round(3.7342,3);  
print "x=$x y=$y";
```

Θα έχει έξοδο:  $x=-5.46$   $y=3.734$



# Η συνάρτηση `is_numeric()`

---

Η συνάρτηση `is_numeric()` είναι χρήσιμη για τον καθορισμό, αν η μεταβλητή είναι ένας έγκυρος αριθμός ή μια αριθμητική συμβολοσειρά.

- Επιστρέφει *true* ή *false*.
- Εξετάστε το ακόλουθο παράδειγμα

```
if (is_numeric($input)) {  
    print "Got Valid Number=$input";  
} else {  
    print "Not Valid Number=$input";  
}
```
- Αν το `$input` ήταν “6” τότε θα είχαμε : Got Valid Number=6
- Αν το `$input` ήταν “Happy” τότε θα είχαμε έξοδο : Not Valid Number=Happy



# Η συνάρτηση *rand()*

---

Χρησιμοποιήστε τη `rand ()` για να δημιουργήσετε ένα τυχαίο αριθμό.

Μπορείτε να χρησιμοποιήσετε τυχαίους αριθμούς για να προσομοιώσετε ένα ρολό ζαριών ή να πετάξετε ένα κέρμα ή να επιλέξετε τυχαία ένα διαφημιστικό banner για εμφάνιση.

Η `rand ()` χρησιμοποιεί συνήθως 2 επιχειρήματα για να καθορίσει το εύρος αριθμών που πρέπει να επιστρέψει (ελάχιστο και μέγιστο όριο).

Για παράδειγμα, τα ακόλουθα επιστρέφουν έναν αριθμό 1 – 15  
`$num = rand(1, 15);`



# Η συνάρτηση *rand()* Μέρος II

---

Χρησιμοποιήστε το `srand` και το `microtime` να σπείρουν τη `rand ()` και βεβαιωθείτε ότι επιστρέφει έναν τυχαίο αριθμό, για παράδειγμα

- `srand ((double) microtime() * 10000000);`
- `$dice = rand(1, 6);`
- `print "Your random dice toss is $dice";`

Ο τυχαίος αριθμός που παράγεται σε αυτή την περίπτωση μπορεί να είναι 1, 2, 3, 4, 5 ή 6

Καλέστε τη `srand ()` μόνο μία φορά ανά Script!



# Περισσότερες πληροφορίες σχετικά με τα συνάρτηση `print()`

---

Δεν χρειάζεται να χρησιμοποιήσετε παρένθεση με την `print ()`

- Διπλά εισαγωγικά σημαίνουν, έξοδος της τιμής της κάθε μεταβλητής:

```
$ x = 10;  
print ("Mom, please send $x dollars");
```

- Μεμονωμένα εισαγωγικά σημαίνουν, έξοδο του πραγματικού ονόματος μεταβλητής

```
$x = 10;  
print ('Mom, please send $x dollars');
```

- Για την έξοδο μιας απλής τιμής ή την έκφραση μεταβλητής παραλείψετε τα εισαγωγικά.

```
$x=5;  
print $x*3;
```



# Δημιουργία HTML Tags με print()

---

Τα μονά ή διπλά εισαγωγικά μπορεί να είναι χρήσιμα κατά τη δημιουργία ετικετών HTML

```
print '<font color="blue">';
```

Το παραπάνω είναι εύκολο στην κατανόηση και στην πραγματικότητα τρέχει ελαφρώς ταχύτερα από τα εισαγωγικά και την ανάστροφη πλαγιοκάθετο (\)

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



# Ένα πλήρες παράδειγμα

---

Εξετάστε την ακόλουθη εφαρμογή:

Χρησιμοποιεί μια φόρμα HTML για να ζητήσει από τον τελικό χρήστη να μαντέψει τα αποτελέσματα ενός νομίσματος:

- `<input type="radio" name="pick" value="0"> Heads`
- `<input type="radio" name="pick" value="1"> Tails`





# Λήψη κώδικα

---

```
1. <html>
2. <head><title> Coin Flip Results </title></head> <body> <?php
3. srand ((double) microtime() * 10000000);
4. $flip = rand( 0, 1 );
5. if ( $flip == 0 && $pick == 0 ) {
6.     print "The flip=$flip, which is heads! <br> ";
7.     print '<font color="blue"> You got it right!</font>';
8. } elseif ( $flip == 0 && $pick == 1 ) {
9.     print "The flip=$flip, which is heads! <br> ";
10.    print '<font color="red"> You got it wrong!</font>';
11. } elseif ( $flip == 1 && $pick == 1 ) {
12.    print "The flip=$flip, which is tails! <br>";
13.    print '<font color="blue"> You got it right!</font>';
```

Ελέγξτε την κατάσταση και στα δύο νομίσματα και μαντέψτε την κορώνα

Ελέγξτε το νόμισμα που είναι κορώνα, αλλά η μαντεψιά ήταν γράμματα

Ελέγξτε την κατάσταση και των δυο νομισμάτων και μαντέψτε τα γράμματα



# Λήψη κώδικα (συνέχεια)

---

```
14. } elseif ( $flip == 1 && $pick == 0 ) {  
15.   print "The flip=$flip, which is tails! <br>";  
16.   print '<font color="red"> You got it wrong!</font>';  
17. } else {  
18.   print "<br>Illegal state error!";  
19. }  
20. ?> </body></html>
```

Ελέγξτε το νόμισμα που  
είναι γράμματα, αλλά η  
μαντεψιά ήταν κορώνα



# Λήψη κώδικα με *REGISTER\_GLOBALS* απενεργοποιημένα

---

```
1. <html>
2. <head><title> Coin Flip Results </title></head> <body> <?php
3. srand ((double) microtime() * 10000000);
4. $pick = $_POST["PICK"];
5. $flip = rand( 0, 1 );
6. if ( $flip == 0 && $pick == 0 ) {
7.     print "The flip=$flip, which is heads! <br> ";
8.     print '<font color="blue"> You got it right!</font>';
9. } elseif ( $flip == 0 && $pick == 1 ) {
10.     print "The flip=$flip, which is heads! <br> ";
11.     print '<font color="red"> You got it wrong!</font>';
12. } elseif ( $flip == 1 && $pick == 1 ) {
13.     print "The flip=$flip, which is tails! <br>";
```

Ελέγξτε την κατάσταση και στα δύο νομίσματα και μαντέψτε την κορώνα

Ελέγξτε την κατάσταση του νομίσματος στην κορώνα αλλά η μαντεψιά είναι γράμματα

Ελέγξτε την κατάσταση και στα δύο νομίσματα και μαντέψτε τα γράμματα



# Λήψη κώδικα με *REGISTER\_GLOBALS* απενεργοποιημένα (συνέχεια)

---

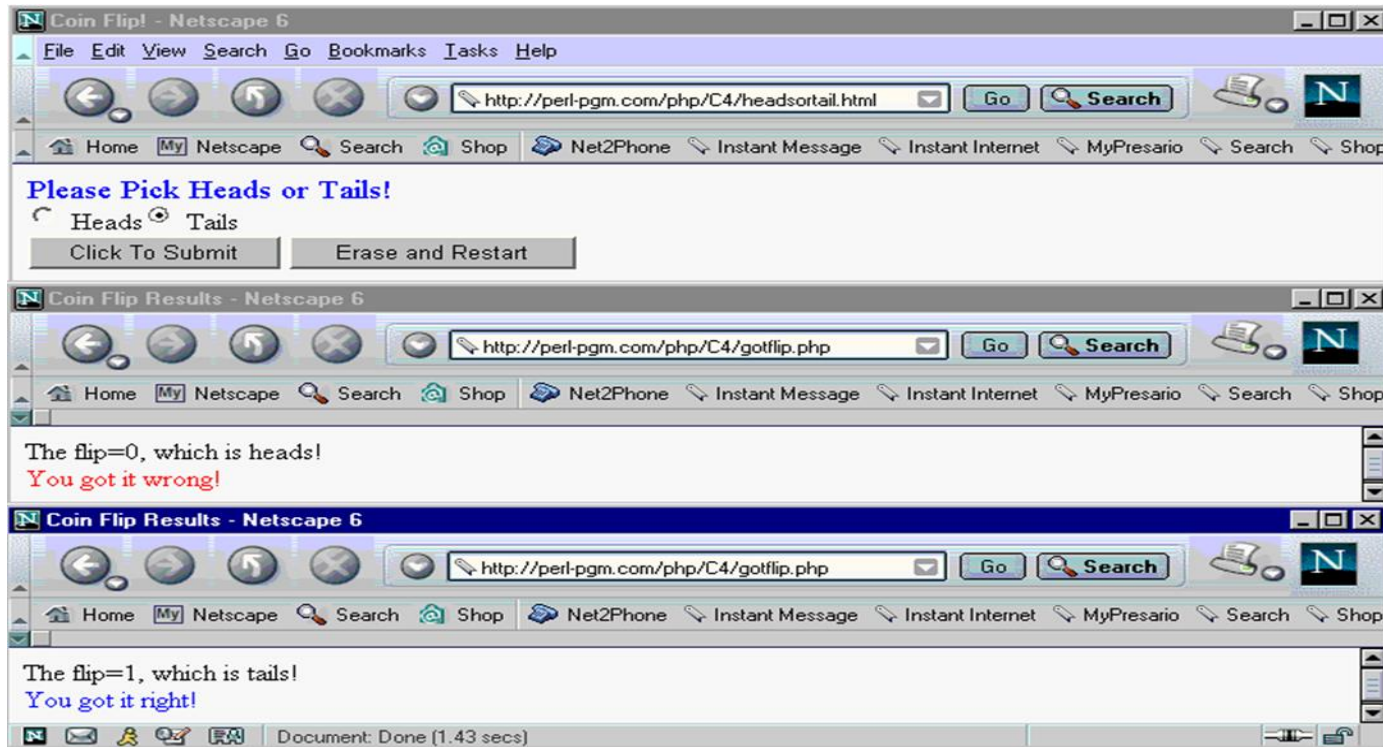
```
14.  print '<font color="blue"> You got it right!</font>';
15. } elseif ( $flip == 1 && $pick == 0 ) {
16.  print "The flip=$flip, which is tails! <br>";
17.  print '<font color="red"> You got it wrong!</font>';
18. } else {
19.  print "<br>Illegal state error!";
20. }
21. ?> </body></html>
```

Ελέγξτε την κατάσταση του νομίσματος στα γράμματα αλλά η μαντεψιά είναι κορώνα



# Η έξοδος του προηγούμενου παραδείγματος

Ο προηγούμενος κώδικας μπορεί να εκτελεστεί σε <http://webwizard.aw.com/~phppgm/C3/headsortail.html>



# Η συνάρτηση date()

Η συνάρτηση date() είναι μία χρήσιμη λειτουργία για τον καθορισμό της τρέχουσας ημερομηνίας και ώρας.

Μια σειρά από ένα ή περισσότερους χαρακτήρες που ορίζει σε ποια μορφή πρέπει να γίνει η έξοδος

```
$x = date('format string');
```

Λαμβάνει πληροφορίες ημερομηνίας () στην απαιτούμενη μορφή

Κλήση της συνάρτησης date()

Η **συμβολοσειρά μορφοποίησης** που καθορίζει τη μορφή της εξόδου της συνάρτησης date() είναι:

```
$day = date('d');  
print "day=$day";
```

Εάν εκτελεστεί στις 27 Δεκεμβρίου 2001, τότε θα έχουμε έξοδο “day=27”.



# Επιλεγμένα μορφές χαρακτήρα για την *date()*

---

Μορφή συμβολοσειράς	Σημασία
D	Ένδειξη τριών γραμμάτων των ημερών της εβδομάδας (πχ. Mon, Tue)
d	Αριθμητική ημέρα του μήνα που επιστρέφεται ως δυο ψηφία (πχ. 01, 02)
F	Τρέχων μήνας ολογράφως (πχ. January, February)
h	Τρέχουσα ώρα της ημέρας από 01 έως 12 (πχ. 02, 11)
H	Τρέχουσα ώρα της ημέρας από 00 έως 23 (πχ. 01, 18)
i	Τρέχοντα λεπτά από 00 έως 59 (πχ. 05, 46)
l	Τρέχουσα μέρα της εβδομάδας ολογράφως (πχ. Monday , Sunday)
L	Επιστέφει 1 αν είναι δίσεκτος χρόνος αλλιώς 0
m	Τρέχων μήνα του χρόνου από 01 έως 12



# Περισσότερα για την `date()`

---

Μπορείτε να συνδυάσετε πολλαπλές μορφές χαρακτήρων που επιστρέφουν περισσότερες από μία μορφές από την `date()`.

Για παράδειγμα

- `$today = date( 'l, F d, Y' );`
- `print "Today=$today";`

Στις 27 Δεκεμβρίου 2001, έχει έξοδο: “Today=Thursday, December 27, 2001”.





# Ένα πλήρες παράδειγμα

---

Εξετάστε την ακόλουθη διαδικτυακή εφαρμογή που χρησιμοποιεί την `date()` για να καθορίσει την τρέχουσα ημερομηνία και τον αριθμό των ημερών που απομένουν για ένα event εκπτώσεων σε ένα κατάστημα.



# Λήψη κώδικα

---

```
1. <html> <head><title> Our Shop </title> </head>
2. <body> <font size=4 color="blue">
3. <?php
4. $today = date( 'l, F d, Y' );
5. print "Welcome on $today to our huge blowout sale! </font>";
6. $month = date('m');
7. $year = date('Y');
8. $dayofyear = date('z');
9. if ( $month == 12 && $year == 2001 ) {
10.  $daysleft = (365 - $dayofyear + 10);
11.  print "<br> There are $daysleft sales days left";
12. } elseif ( $month == 01 && $year == 2002 ) {
13.  if ( $dayofyear <= 10 ) {
14.   $daysleft = (10 - $dayofyear);
15.   print "<br> There are $daysleft sales days left";
16.  } else {
19.   print "<br>Sorry, our sale is over.";
20.  }
21. } else {
22.  print "<br>Sorry, our sale is over.";
23. }
24. print "<br>Our Sale Ends January 10, 2002";
25. ?> </body></html>
```

---

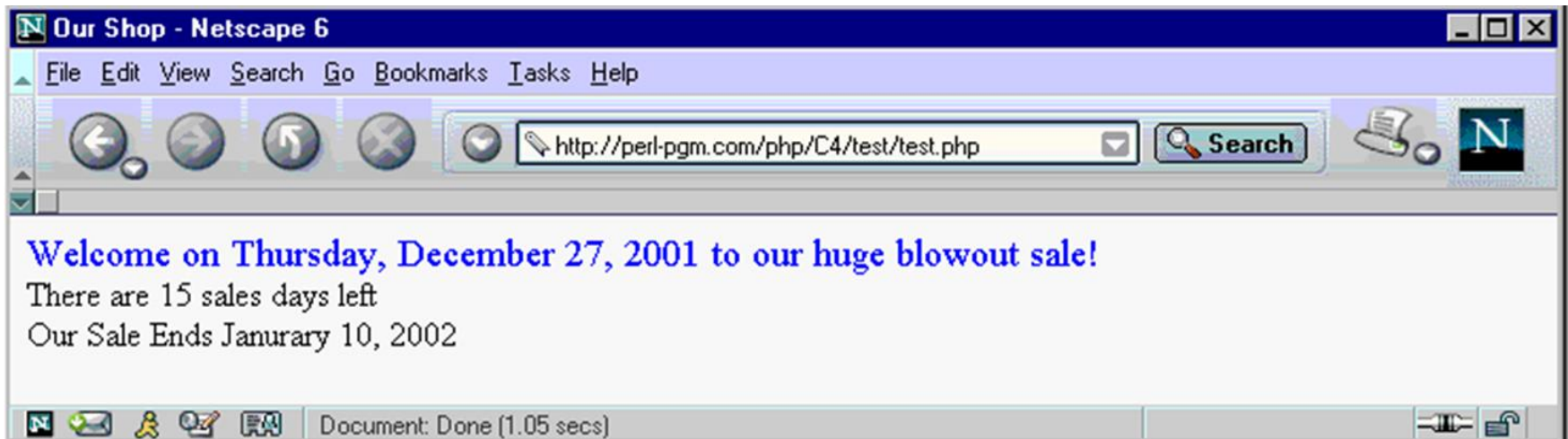


# Η έξοδος του παραδείγματος με τη χρήση *date()*

---

Ο προηγούμενος κώδικας μπορεί να εκτελεστεί σε

<http://webwizard.aw.com/~phppgm/C3/date.php>



# Συναρτήσεις

---

Οι προγραμματιστικές-καθορισμένες συναρτήσεις παρέχουν έναν τρόπο για να ομαδοποιήσουν μια σειρά δηλώσεων, να τα αφήσουν κατά μέρος, και να τα μετατρέπουν σε mini-scripts στα πλαίσια ενός μεγαλύτερου σεναρίου.

- Σενάρια που είναι πιο κατανοητά και αλλάζουν.
- Επαναχρησιμοποιούμενες ενότητες script.
- Μικρότερο μέγεθος προγράμματος




# Γράφοντας τη δική σας συνάρτηση

---

Χρησιμοποιήστε την ακόλουθη γενική μορφή

```
function function_name() {  
set of statements  
}
```


Συμπεριλάβετε παρενθέσεις  
στο τέλος του ονόματος της  
συνάρτησης



Η συνάρτηση εκτελεί  
αυτές τις δηλώσεις  
όταν καλείται



Εγκλείστε της αγκύλες



# Για παράδειγμα

---

Σκεφτείτε το εξής:

```
function OutputTableRow() {  
    print '<tr><td>One</td><td>Two</td></tr>';  
}
```

Μπορείτε να εκτελέσετε τη συνάρτηση εκτελώντας

```
OutputTableRow();
```



# Ένα πλήρες παράδειγμα

---

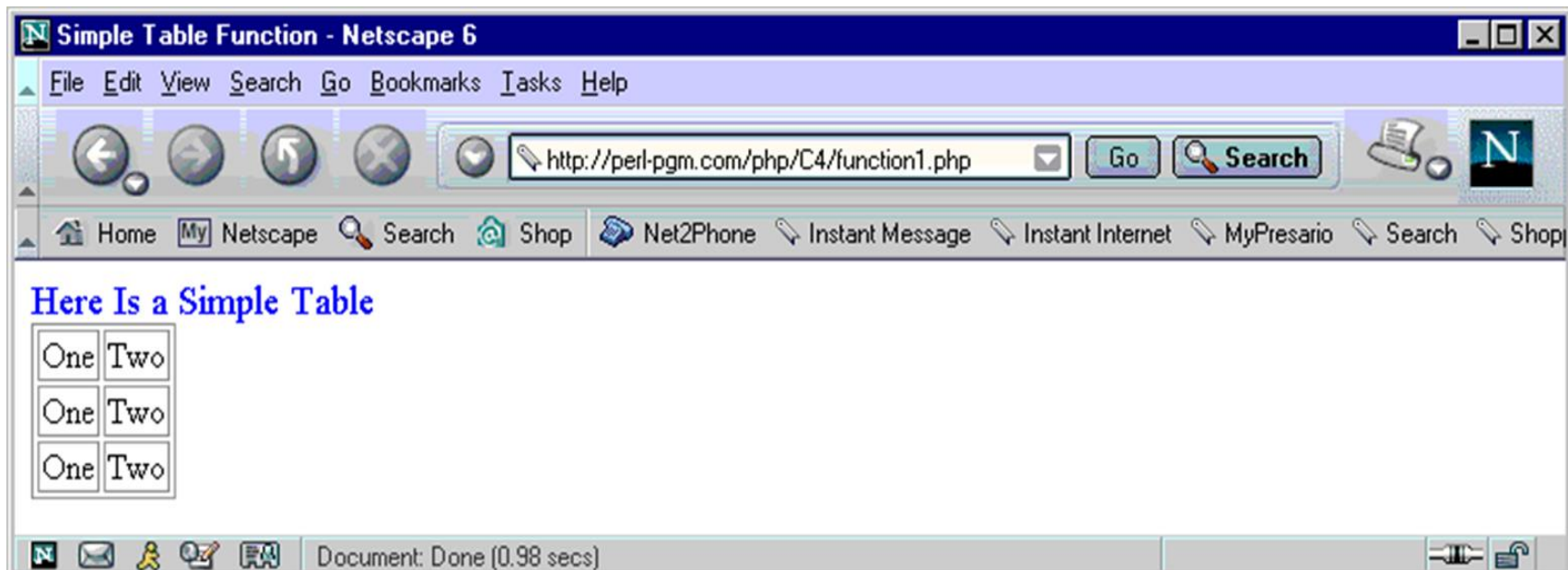
```
1. <html>
2. <head><title> Simple Table Function </title> </head> <body>
3. <font color="blue" size="4"> Here Is a Simple Table <table border=1>
4. <?php
5.     function OutputTableRow() {
6.         print '<tr><td>One</td><td>Two</td></tr>';
7.     }
8.     OutputTableRow();
9.     OutputTableRow();
10.    OutputTableRow();
11. ?>
12. </table></body></html>
```

← OutputTableRow ()  
ορισμός συνάρτησης

← Τρεις συνεχόμενες κλήσεις  
στην OutputTableRow ()  
συνάρτηση



# Έξοδος παραδείγματος με πίνακα





# Συμβουλή: Χρησιμοποιήστε Σχόλια κατά τη έναρξη της συνάρτησης

---

Είναι καλή πρακτική να τοποθετήσει τα σχόλια στην αρχή μιας συνάρτησης.

Για παράδειγμα

```
function OutputTableRow() {  
  // Simple function that outputs 2 table cells  
  print '<tr><td>One</td><td>Two</td></tr>';  
}
```



# Περνώντας επιχειρήματα στη συνάρτηση

---

Οι μεταβλητές εισόδου για τις συναρτήσεις ονομάζονται **επιχειρήματα στη συνάρτηση**.

Για παράδειγμα, η ακόλουθη στέλνει 2 επιχειρήματα  
`OutputTableRow("A First Cell", "A Second Cell");`

Μέσα στον ορισμό της συνάρτησης μπορεί να έχουμε πρόσβαση σε τιμές

```
function OutputTableRow($col1, $col2) {  
    print "<tr><td>$col1 </td><td>$col2</td></tr>";  
}
```



# Εξετάστε τον ακόλουθο κώδικα

---

```
1. <html>
2. <head><title> Simple Table Function </title> </head> <body>
3. <font color="blue" size=4> Revised Simple Table <table border=1>
4. <?php
5. function OutputTableRow( $col1, $col2 ) {
6.     print "<tr><td>$col1</td><td>$col2</td></tr>";
7. }
8. for ( $i=1; $i<=4; $i++ ) {
9.     $message1="Row $i Col 1";
10.    $message2="Row $i Col 2";
11.    OutputTableRow( $message1, $message2 );
12. }
13. ?>
14. </table></body></html>
```

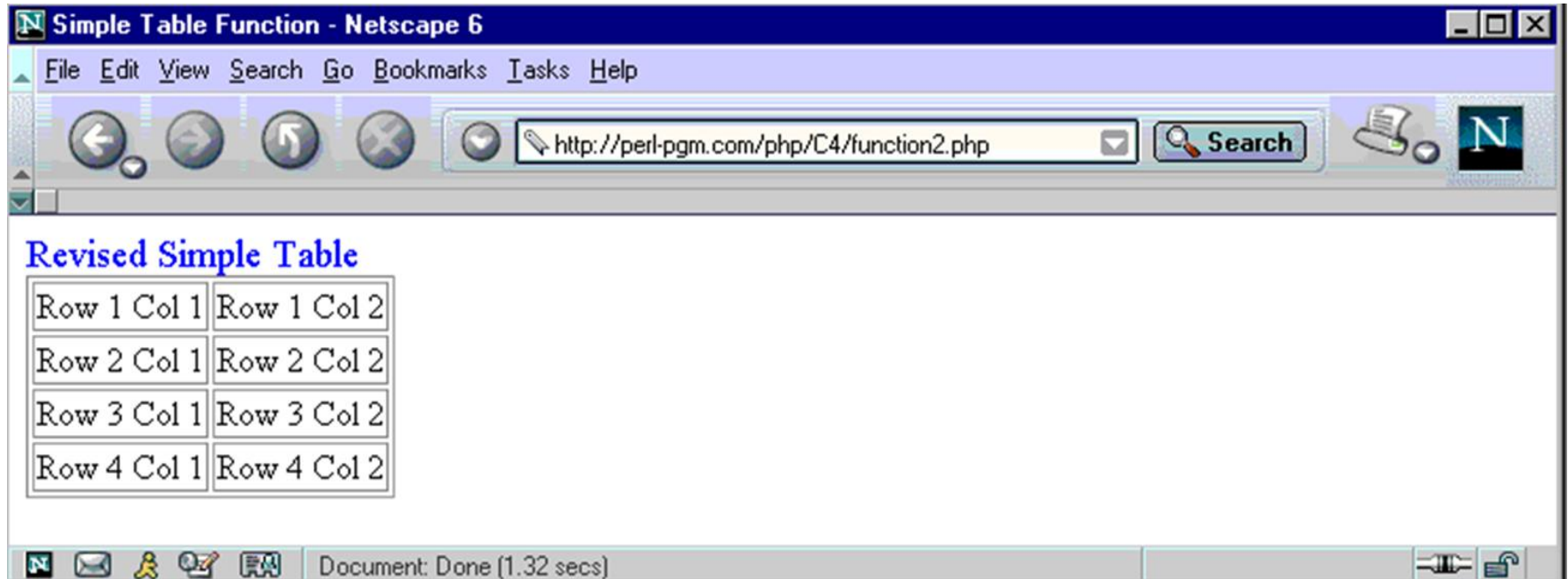
OutputTableRow ()  
Ορισμό της συνάρτησης.

Τέσσερις κλήσεις προς  
OutputTableRow ()



# Έξοδος παραδείγματος με πίνακα

---



The screenshot shows a Netscape 6 browser window titled "Simple Table Function - Netscape 6". The address bar contains the URL "http://perl-pgm.com/php/C4/function2.php". The main content area displays a table with the title "Revised Simple Table". The table has 4 rows and 2 columns, with each cell containing text indicating its row and column number. The status bar at the bottom shows "Document: Done (1.32 secs)".

Row 1 Col 1	Row 1 Col 2
Row 2 Col 1	Row 2 Col 2
Row 3 Col 1	Row 3 Col 2
Row 4 Col 1	Row 4 Col 2



# Επιστρέφοντας Τιμές

---

- Οι συναρτήσεις σας μπορούν να επιστρέψουν δεδομένα στο καλούμενο script.
- Για παράδειγμα, οι συναρτήσεις σας μπορούν να επιστρέψουν τα αποτελέσματα ενός υπολογισμού.
- Μπορείτε να χρησιμοποιήσετε τη δήλωση PHP επιστροφής για να επιστρέψετε μια τιμή στην εντολή δέσμης εντολών:

```
return $result;
```



Αυτή η μεταβλητή  
αξία θα επιστραφεί με την  
κλήση του script



# Παράδειγμα συνάρτησης

---

```
1. function Simple_calc( $num1, $num2 ) {  
2.   // PURPOSE: returns largest of 2 numbers  
3.   // ARGUMENTS: $num1 -- 1st number, $num2 -- 2nd number  
4.   if ($num1 > $num2) {  
5.     return($num1);  
6.   } else {  
7.     return($num2);  
8.   }  
9. }
```

Επιστρέψτε \$num1 όταν είναι η μεγαλύτερη τιμή

Επιστρέψτε \$num2 όταν είναι η μεγαλύτερη τιμή.

What is output if called as follows:

```
$largest = Simple_calc(15, -22);
```



# Ένα πλήρες παράδειγμα

---

- Σκεφτείτε ένα σενάριο που υπολογίζει την επί ποσοστό αλλαγή από μια αρχική σε μια τελική τιμή.
- Χρησιμοποιείστε την ακόλουθη front-end μορφή:

Starting Value: `<input type="text" size="15"  
maxlength="20" name="start">`

Ending Value: `<input type="text" size="15"  
maxlength="20" name="end">`



# Ο πηγαίος κώδικας

---

```
1. <html>
2. <head><title> Your Percentage Calculation </title></head><body>
3. <font color="blue" size=4> Percentage Calculator </font>
4. <?php
5. function Calc_perc($buy, $sell) {
6.     $per = (($sell - $buy) / $buy) *100;
7.     return($per);
8. }
9. print "<br>Your starting value was $start.";
10. print "<br>Your ending value was $end.";
11. if (is_numeric($start) && is_numeric($end) ) {
12.     if ($start != 0) {
13.         $per = Calc_perc($start, $end);
14.         print "<br> Your percentage change was $per %.";
15.     } else { print "<br> Error! Starting values cannot be zero "; }
16. } else {
17.     print "<br> Error! You must have valid numbers for start and end ";
18. }
19. ?> </body></html>
```

Υπολογίστε το ποσοστό αλλαγής από την αρχική στην τελική τιμή

Η κλήση της Calc\_perc () επιστρέφει το ποσοστική αλλαγή σε \$per.





# Ο πηγαίος κώδικας με τα REGISTER\_GLOABLS απενεργοποιημένα

---

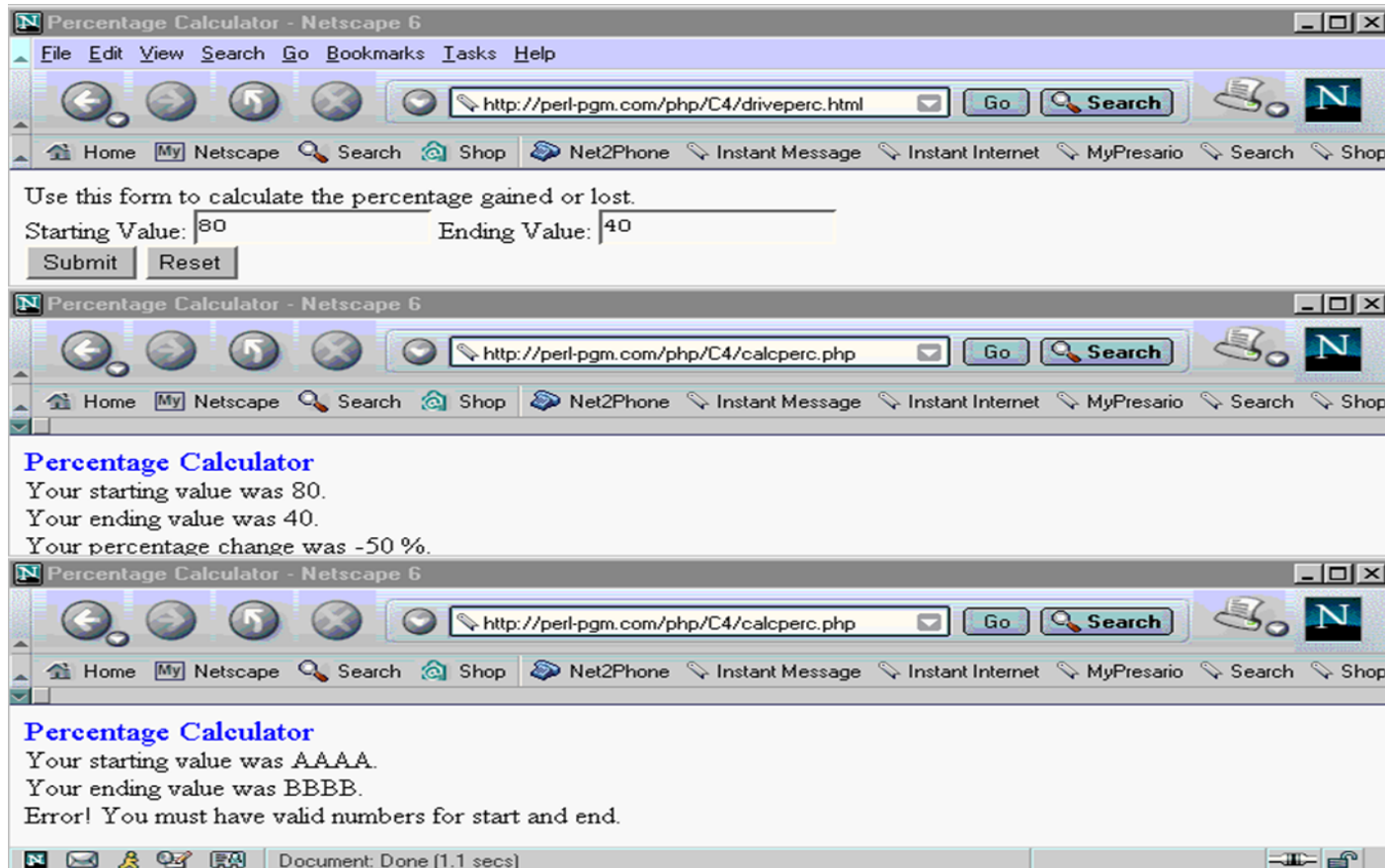
```
1. <html>
2. <head><title> Your Percentage Calculation </title></head><body>
3. <font color="blue" size=4> Percentage Calculator </font>
4. <?php
5. function Calc_perc($buy, $sell) {
6.     $per = (($sell - $buy) / $buy) *100;
7.     return($per);
8. }
9. $start = $_POST["start"]; $end = $_POST["end"];
10. print "<br>Your starting value was $start.";
11. print "<br>Your ending value was $end.";
12. if (is_numeric($start) && is_numeric($end) ) {
13.     if ($start != 0) {
14.         $per = Calc_perc($start, $end);
15.         print "<br> Your percentage change was $per %.";
16.     } else { print "<br> Error! Starting values cannot be zero "; }
17. } else {
18.     print "<br> Error! You must have valid numbers for start and end ";
19. }
20. ?> </body></html>
```

Υπολογίστε το ποσοστό αλλαγής από την αρχική στην τελική τιμή

Η κλήση της Calc\_perc () επιστρέφει το ποσοστική αλλαγή σε \$per



# Έξοδος παραδείγματος με % κέρδος ή απώλεια



# Ένα ολοκληρωμένο παράδειγμα Script με τα REGISTER\_GLOBALS απενεργοποιημένα

---

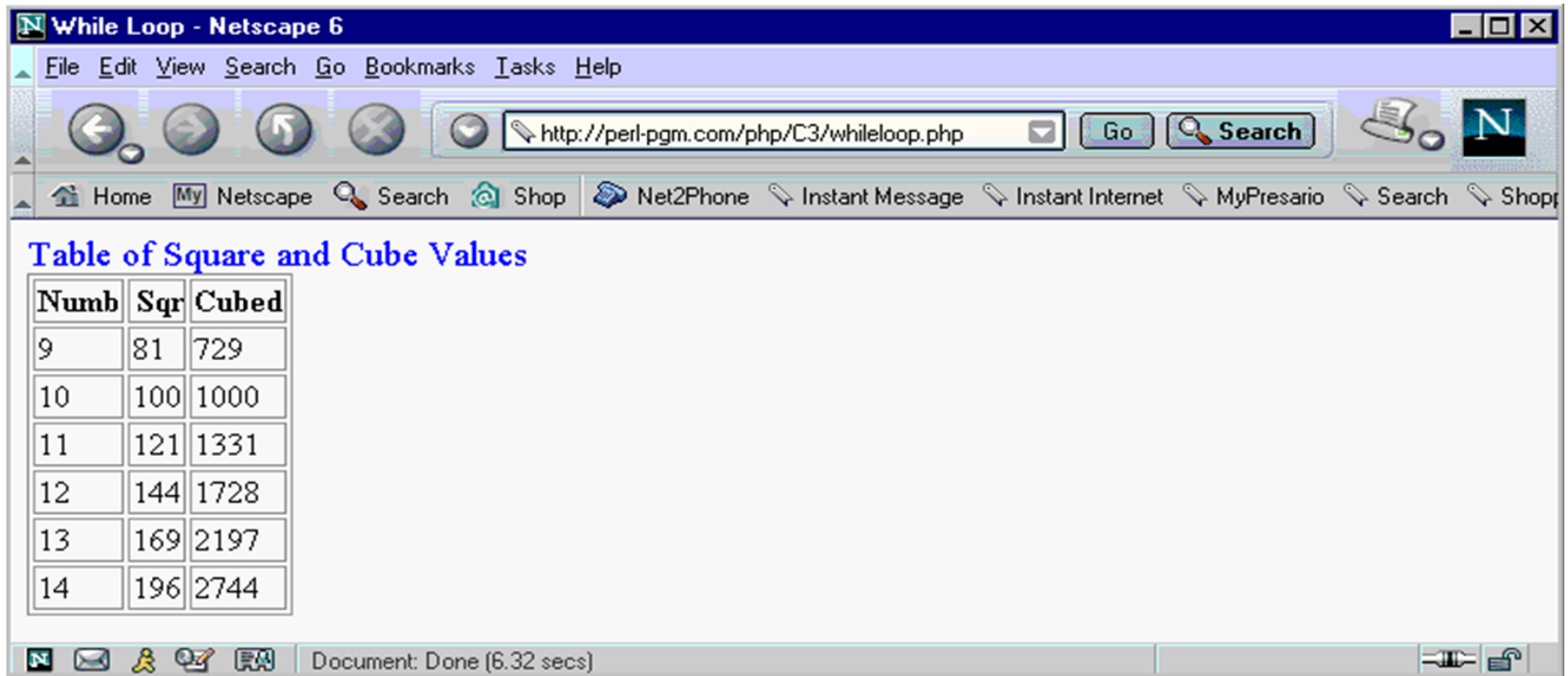
```
1. <html>
2. <head><title>While Loop</title></head>
3. <body>
4. <font size="4" color="blue"> Table of Square and Cube Values </font>
5. <table border=1>
6. <th> Numb </th> <th> Sqr </th> <th> Cubed </th>
7. <?php
8.   $start = $_POST["start"]; $end = $_POST["end"];
9.   $i = $start;
10.  while ($i <= $end) {
11.    $sqr=$i*$i;
12.    $cubed=$i*$i*$i;
13.    print ("<tr><td>$i</td><td>$sqr</td><td>$cubed</td></tr>");
14.    $i = $i + 1;
15.  }
16. ?></table></body></html>
```



# Έξοδος παραδείγματος με πίνακα

Ο προηγούμενος κώδικας μπορεί να εκτελεστεί στο

<http://webwizard.aw.com/~phppgm/C3/whileloop.php>



The screenshot shows a Netscape 6 browser window titled "While Loop - Netscape 6". The address bar contains the URL "http://perl-pgm.com/php/C3/whileloop.php". The page content displays a table titled "Table of Square and Cube Values". The table has three columns: "Numb", "Sqr", and "Cubed". The data rows are as follows:

Numb	Sqr	Cubed
9	81	729
10	100	1000
11	121	1331
12	144	1728
13	169	2197
14	196	2744

The browser status bar at the bottom indicates "Document: Done (6.32 secs)".



# Χρήση Εξωτερικών αρχείων Script

---

Κάποια στιγμή θα θελήσετε να χρησιμοποιήσετε Scripts από εξωτερικά αρχεία  
Η PHP υποστηρίζει 2 λειτουργίες που σχετίζονται με:

```
require ("header.php");  
include ("trailer.php");
```

→ Η συνάρτηση include ()  
δημιουργεί μια προειδοποίηση  
οτι δεν μπορείτε να  
εισαγάγετε το καθορισμένο αρχείο

← Η συνάρτηση require ()  
παράγει ένα θανατηφόρο  
σφάλμα εάν δεν μπορείτε να  
εισαγάγετε το καθορισμένο  
αρχείο

Και τα δυο είναι αναζητήσεις για το όνομα του αρχείου μέσα στα διπλά  
εισαγωγικά και την εισαγωγή PHP, HTML, ή κώδικα JavaScript στο τρέχον  
αρχείο.



# Η εντολή *include*

---

- Με την εντολή `include` μπορούμε να γράψουμε ένα μεγάλο πρόγραμμα σε πολλά αρχεία και μετά να τα κάνουμε όλα `include` από ένα πρόγραμμα.
- Με το `include`, είναι σαν να κάναμε copy-paste το αρχείο που κάναμε `include` εκεί που γίνεται η κλήση του `include`.
- `Menu.php`
  - `<html>`
  - `<body>`
  - `<a href=http://www.mypage.com>Home</a>`
  - `<a href=http://www.yahoo.com>Yahoo</a>`
- Κύριο Αρχείο
  - `<? Php include ("menu.php"); ?>`
  - `<h1> Hello </h1>`
  - ...



# Η εντολή GET

---

- Η μεταβλητή `$_GET` είναι ένας πίνακας ο οποίος περιέχει τιμές που έχουν σταλεί με την μέθοδο HTTP GET.
- Όταν χρησιμοποιείτε η μέθοδος `$_GET` οι τιμές οι οποίες στέλνονται φαίνονται στην μπάρα διεύθυνσης (URL).
  - Οπότε αυτή η μέθοδος δεν είναι καλή για την αποστολή κωδικών χρήσης.
  - Είναι χρήσιμη όμως στην περίπτωση που ο χρήστης επιθυμεί να βάλει την σελίδα στα 'Αγαπημένα (Bookmarks) Π.χ. στην περίπτωση που έχει κάνει κάποια αναζήτηση.
- Η μέθοδος `$_GET` μπορεί να στείλει μεταβλητές που το περιεχόμενό τους δεν ξεπερνάει τους 100 χαρακτήρες.
- Σημείωση: Η μεταβλητή `$_REQUEST` περιέχει τα περιεχόμενα της `$_GET` και της `$_POST`



# Παραδείγματα GET

---

## ➤ Αρχείο HTML (Φόρμα)

- `<form action="php_GET01.php" method="get">`
- Όνομα:
- `<input type="text" name="firstname" value="" size=20>`
- `<br>`
- Επίθετο:
- `<input type="text" name="lastname" value="" size=20>`
- `<br>`
- `<input type="reset">`
- `<input type="submit">`
- `</form>`

## ❖ Αρχείο PHP

- `<?php`
- `echo "Το όνομα είναι <i> " . $_GET["firstname"];`
- `echo "</i> και το επίθετο είναι " . $_GET["lastname"];`
- `?>`





# Παραδείγματα *POST*

---

- Η μεταβλητή `$_POST` είναι ένας πίνακας ο οποίος περιέχει τιμές που έχουν σταλεί με την μέθοδο HTTP POST.
- Όταν χρησιμοποιείτε η μέθοδος `$_POST` οι τιμές οι οποίες στέλνονται ΔΕΝ φαίνονται στην μπάρα διεύθυνσης (URL).
  - Οπότε αυτή η μέθοδος είναι καλή για την αποστολή κωδικών χρήσης.
- Η μέθοδος `$_POST` δεν έχει περιορισμό στο μέγεθος των δεδομένων που πρέπει να στείλει.
- Σημείωση: Η μεταβλητή `$_REQUEST` περιέχει τα περιεχόμενα της `$_GET` και της `$_POST`



# Παραδείγματα POST

---

- Αρχείο HTML (Φόρμα)

```
<form action="php_POST01.php" method="post">
```

Όνομα:

```
<input type="text" name="firstname" value="" size=20>  
<br>
```

Επίθετο:

```
<input type="text" name="lastname" value="" size=20>  
<br>
```

```
<input type="reset">
```

```
<input type="submit">
```

```
</form>
```

- Αρχείο PHP

- <?php

- echo "Το όνομα είναι <i> " . \$\_POST["firstname"];</i>

- echo "<i> και το επίθετο είναι". \$\_POST["firstname"];</i>";

- ?>



# Η συνάρτηση *isset()*

---

- `isset (<όνομα μεταβλητής>);`  
-Επιστρέφει `true/false` αναλόγως με το αν έχει δηλωθεί η μεταβλητή.
- `if (!isset ($_REQUEST["name"]))`
  - {
    - `<form action="php—POST01.php" method="post">`
    - Όνομα:
      - `<input type="text" name="name" value="" size=20>`
      - `<br>`
    - Επίθετο:
      - `<input type="text" name="lastname" value="" size=20>`
      - `<br>`
    - `<input type="reset">`
    - `<input type="submit">`
    - `</form>`
  - }
  - `else`
    - {
      - `echo "Το όνομα είναι <i> " . $_REQUEST["name"];`
      - `echo "</i> και το επίθετο είναι " . $_REQUEST["firstname"];`
      - }



# Εξετάστε το ακόλουθο παράδειγμα

---

```
1. <font size=4 color="blue">
2. Welcome to Harry's Hardware Heaven!
3. </font><br> We sell it all for you!<br>
4. <?php
5.     $time = date('H:i');
6.     function Calc_perc($buy, $sell) {
7.         $per = (($sell - $buy ) / $buy) * 100;
8.         return($per);
9.     }
10. ?>
```

← Το script θα εξάγει αυτές τις γραμμές όταν περιλαμβάνει το αρχείο

← Θα οριστεί η τιμή \$time όταν περιλαμβάνει το αρχείο

← Αυτή η συνάρτηση θα είναι διαθέσιμη για χρήση όταν περιλαμβάνει το αρχείο



# header.php

---

Εάν το προηγούμενο script τοποθετηθεί σε ένα αρχείο που ονομάζεται header.php :

1. <html><head><title> Hardware Heaven </title></head> <body>
2. <?php
3. include("header.php");
4. \$buy = 2.50;
5. \$sell = 10.00;
6. print "<br>It is \$time.";
7. print "We have hammers on special for \\$\$sell!";
8. \$markup = Calc\_perc(\$buy, \$sell);
9. print "<br>Our markup is only \$markup%!";
10. ?>
11. </body></html>

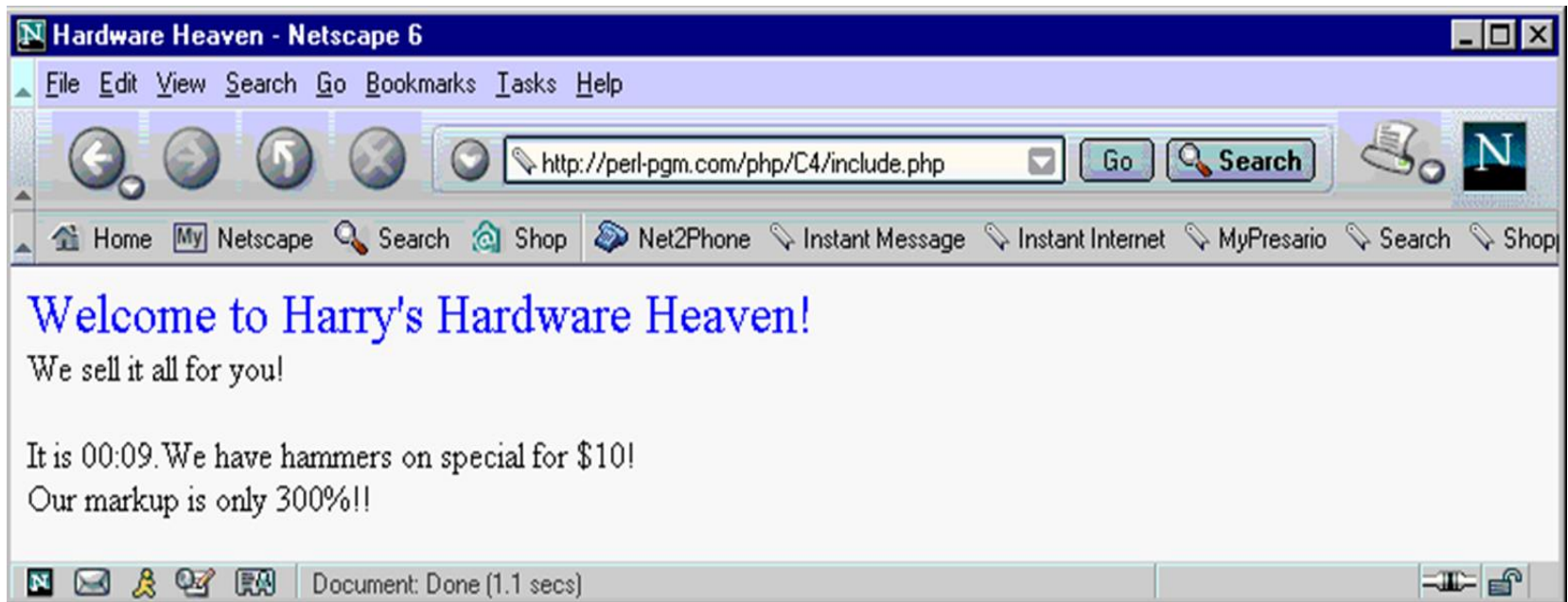
← Συμπεριλάβετε το αρχείο header.php

← Το Calc\_perc () ορίζεται στο header.php



# Έξοδος παραδείγματος «Hardware Heaven»

---



# Πιο τυπική χρήση του Εξωτερικού Κώδικα Αρχεία

---

- Πιο τυπικά θα μπορούσαμε να χρησιμοποιήσουμε ένα ή περισσότερα αρχεία με μόνο συναρτήσεις και άλλα αρχεία που περιέχουν HTML.
- Για παράδειγμα, θα μπορούσαμε να χρησιμοποιήσουμε τα ακόλουθα σαν footer.php

```
<hr>
```

```
Hardware Harry's is located in beautiful downtown Hardwareville.
```

```
<br>We are open every day from 9 A.M. to midnight, 365 days a year.
```

```
<br>Call 476-123-4325. Just ask for Harry.
```

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

Μπορεί να περιλαμβάνει τη χρήση:

```
<?php include("footer.php"); ?>
```



# Παράδειγμα συνάρτησης χωρίς ορίσματα

---

```
<?php
```

```
function writeMyName()  
{  
    echo"Michael";  
}
```

```
writeMyName(); // κλήση συνάρτησης
```

```
?>
```





# Παράδειγμα συνάρτησης με ένα όρισμα

---

```
<?php
```

```
//Συνάρτηση με μία παράμετρο
```

```
function writeMyName($fname)
{
    echo $fname;
}
```

```
//Η μεταβλητή $fname Θα πάρει ότι τιμή βάλουμε
// εδώ. Στην συνάρτηση δεν δηλώνουμε τον τύπο
//της παραμέτρου.
```

```
writeMyName("Nick");
?>
```



# Παράδειγμα συνάρτησης με δυο ορίσματα

---

```
<?php
```

```
//Συνάρτηση με δύο παραμέτρους
```

```
function writeMyName($fname, $fage)
{
    echo $fname . " είναι " . $fage . " χρονών";
}
writeMyName("Nick", 54);
?>
```

Αποτέλεσμα :

Nick είναι 54 χρονών χρονών



# Η συνάρτηση μπορεί να κληθεί μόνη της ή να εμφωλιαστεί

---

```
<?php
```

```
//Επιστρέφοντας τιμές
```

```
function add($num1, $num2)
{
$result = $num1 + $num2;
return $result;
}
```

```
//Κλήση συνάρτησης. Το $x θα γίνει 10.
```

```
$x = add(3, 7);
```

```
//Θα τυπωθεί το αποτέλεσμα της συνάρτησης με την echo.
//Η συνάρτηση add καλείτε πριν τυπωθεί οτιδήποτε.
```

```
echo "5 + 15 μας κάνει " . add(5, 15);
?>
```



# Αρχεία

---

- `fopen(<όνομα αρχείου>, <mode>);`
  - Ανοίγει ένα αρχείο σύμφωνα με το `mode`.
  - Π.χ. `$file = fopen("a.txt", "r");`
- `fclose (<αρχείο>);`
  - Κλείνει ένα ανοιγμένο αρχείο.
- `feof (<αρχείο>);`
  - Επιστρέφει `true` εάν το αρχείο έφτασε στο τέλος του.
- `fgets(<αρχείο>);`
  - Επιστρέφει μια γραμμή από ένα ανοιχτό αρχείο.



# Αρχεία

---

- `fgetc(<αρχείο>);`
    - Επιστρέφει έναν χαρακτήρα από ένα ανοιχτό αρχείο.
  - `fwrite (<αρχείο>, <string δεδομένων>);`
    - Γράφει σε ένα ανοιχτό αρχείο το `string` δεδομένων και επιστρέφει το μέγεθος δεδομένων που γράφτηκαν.
  - Παράδειγμα:
    - `$fl = fopen ("file.txt". "w");`
    - `echo fwrite ($fl. "File Testing!");`
- // Θα τυπώσει τον αρ. 14
- `fclose($fl)`
  - `rewind(<αρχείο>);`
    - Μετακινεί τον δείκτη ενός ανοιχτού στην αρχή (στο 0 δηλαδή).
- 



# Χρήσιμες συναρτήσεις

---

- die (<μήνυμα>);
  - Π.χ. die ("error");
- exit σαν την die
- file\_exists(<όνομα αρχείου>);
  - Επιστρέφει true εάν το αρχείο υπάρχει.
- copy (<αρχείο 1>, <αρχείο 2>);
  - Αντιγράφει το αρχείο 1 στην διαδρομή/όνομα αρχείου του αρχείου 2.
- file\_size(<αρχείο>);
  - Επιστρέφει μέγεθος του αρχείου.
- diskfreespace(<διαδρομή>);
  - Επιστρέφει τον διαθέσιμο χώρο στον δίσκο.
  - Π.χ. echo diskfreespace("c:");



# Σημαίες για πρόσβαση σε αρχεία

---

Mode	Περιγραφή
r	Μόνο ανάγνωση. Τοποθετεί τον δείκτη στην αρχή του αρχείου
r+	Ανάγνωση και εγγραφή. Τοποθετεί τον δείκτη στην αρχή του αρχείου
w	Μόνο εγγραφή. Σβήνει ότι προηγούμενα δεδομένα υπήρχαν και εάν δεν υπάρχει το αρχείο το δημιουργεί.
w+	Ανάγνωση και εγγραφή. Σβήνει ότι προηγούμενα δεδομένα υπήρχαν και εάν δεν υπάρχει το αρχείο το δημιουργεί.
a	Προσθήκη. Ανοίγει και προσθέτει στο τέλος του αρχείου ή δημιουργεί καινούργιο αρχείο εάν δεν υπάρχει
a+	Προσθήκη και ανάγνωση. Ανοίγει και προσθέτει στο τέλος του αρχείου ή δημιουργεί καινούργιο αρχείο εάν δεν υπάρχει
x	Μόνο εγγραφή. Επιστρέφει FALSE εάν το αρχείο υπάρχει.



# Περίληψη

---

- Η PHP παρέχει αρκετές χρήσιμες συναρτήσεις συμπεριλαμβανομένων των `abs()`, `round()`, `is_numeric()`, `rand()`, `date()`
- Οι προγραμματισμένες - καθορισμένες συναρτήσεις σας επιτρέπουν να ομαδοποιήσετε μια σειρά δηλώσεων, να τις αφήσετε κατά μέρος, και να μετατρέψτε αυτές τις ομαδοποιημένες δηλώσεις σε mini-scripts.





# Χρησιμοποιώντας *test* όρους καταστάσεων

---

- Η δήλωση συνθήκης παρέχει έναν τρόπο για `scripts`, για να ελέγξετε ορισμένες τιμές δεδομένων και στη συνέχεια να αντιδράσετε διαφορετικά ανάλογα με την τιμή που βρέθηκε.

➤ Θα εξετάσει :

- Η εντολή `if`,
- Η ρήτρα `elseif`,
- Η ρήτρα `else`,
- Και τη δήλωση διακόπτη




# Χρησιμοποιώντας τη δήλωση *if*

---

Χρησιμοποιήστε μια εντολή *if* για να καθορίσετε μια συνθήκη ελέγχου και ένα σύνολο δηλώσεων που θα εκτελούνται κατά τη διάρκεια μιας δοκιμαστικής κατάσταση *true*.

```
if ($average > 69) {  
    $Grade="Pass";  
    print "Grade=$Grade ";  
}  
print "Your average was $average";
```

Όταν το *\$average* είναι μεγαλύτερο από το 69 εκτελούνται αυτές οι δηλώσεις



Αν η *\$average* ήταν ίση με το 70 τότε το παραπάνω θα είχε έξοδο :  
Your average was 70



# Εκφράσεις Test

---

- Οι δοκιμαστικές εκφράσεις χρησιμοποιούν δοκιμαστικούς τελεστές στο πλαίσιο των εκφράσεων.
- Οι δοκιμαστικοί τελεστές λειτουργούν σαν τους εκφραστικούς τελεστές.
- Η παραπάνω εντολή if χρησιμοποιεί το μεγαλύτερο από (>) τους τελεστές για να ελέγξει αν το \$average είναι μεγαλύτερο από το 69.
- Οι δοκιμαστικοί τελεστές test αξιολογούν για true ή false.



# PHP Operators Test

Τελεστές	Πράξη	Παράδειγμα	Αποτέλεσμα
=	Ίσο με	<pre>If (\$x == 6){     \$x=\$y+1;     \$y=\$x+1; }</pre>	Τρέχει την δεύτερη και την τρίτη δήλωση αν η τιμή του \$x είναι ίση με 6.
!=	Διάφορο με	<pre>If (\$x != \$y){     \$x=5+1; }</pre>	Τρέχει την δεύτερη δήλωση αν η τιμή του \$x δεν είναι ίση με την τιμή του \$y.
<	Μικρότερο απο	<pre>If (\$x &lt; 100){     \$y=5; }</pre>	Τρέχει τη δεύτερη δήλωση αν η τιμή του \$x είναι μικρότερη από 100.
>	Μεγαλύτερο απο	<pre>If (\$x &gt; 51){     print "OK"; }</pre>	Τρέχει τη δεύτερη δήλωση αν η τιμή του \$x είναι μεγαλύτερη από 51.
>=	Μεγαλύτερο ή ίσο με	<pre>If (16 &gt;= \$x){     print "x=\$x"; }</pre>	Τρέχει τη δεύτερη δήλωση αν το 16 είναι μεγαλύτερο ή ίσο με την τιμή του \$x.
<=	Μικρότερο ή ίσο με	<pre>If (\$x &lt;= \$y){     print "y=\$y";     print "x=\$x"; }</pre>	Τρέχει τη δεύτερη και την τρίτη δήλωση αν η τιμή του \$x είναι μικρότερη ή ίση με την τιμή του \$y.



# Ένα πλήρες παράδειγμα

---

- Εξετάστε την ακόλουθη εφαρμογή:
- Λαμβάνει δύο βαθμούς ως είσοδο και καθορίζει αν ο μέσος όρος τους είναι μεγαλύτερος από 89.
- Χρησιμοποιεί μια φόρμα HTML για τους βαθμούς εισόδου:

Enter First Score `<input type="text" size="4" maxlength="7" name="grade1">` ← Θέτει \$grade1

Enter Second Score `<input type="text" size="4" maxlength="7" name="grade2">` ← Θέτει \$grade2



# Λήψη κώδικα

```
1. <html>
2. <head><title>Decisions</title></head>
3. <body>
4. <?php
5.     $average = ($grade1 + $grade2) / 2;
6.     if ( $average > 89 ) {
7.         print "Average score: $average You got an A! <br>";
8.     }
9.     $max=$grade1;
10.    if ($grade1 < $grade2) {
11.        $max = $grade2;
12.    }
13.    print ("Your max score was $max");
14. ?>
15. </body></html>
```

Υπολογίστε τον μέσο όρο

Η έξοδος εάν είναι το \$average είναι περισσότερο από 89

Ορίστε όταν το \$grade2 είναι περισσότερο από το \$grade1



# Λήψη κώδικα με REGISTER\_GLOBALS απενεργοποιημένα

---

```
1. <html>
2. <head><title>Decisions</title></head>
3. <body>
4. <?php
5.   $grade1= $_POST["grade1"];
6.   $grade2= $_POST["grade2"];
5.   $average = ($grade1 + $grade2) / 2;
6.   if ( $average > 89 ) {
7.     print "Average score: $average You got an A! <br>";
8.   }
9.   $max=$grade1;
10.  if ($grade1 < $grade2) {
11.    $max = $grade2;
12.  }
13.  print ("Your max score was $max");
14. ?>
15. </body></html>
```

Αποκτήστε grade1 και grade2  
από τη φόρμα HTML

Υπολογίστε τον μέσο όρο

Έξοδος αν το \$average  
είναι περισσότερο από 89

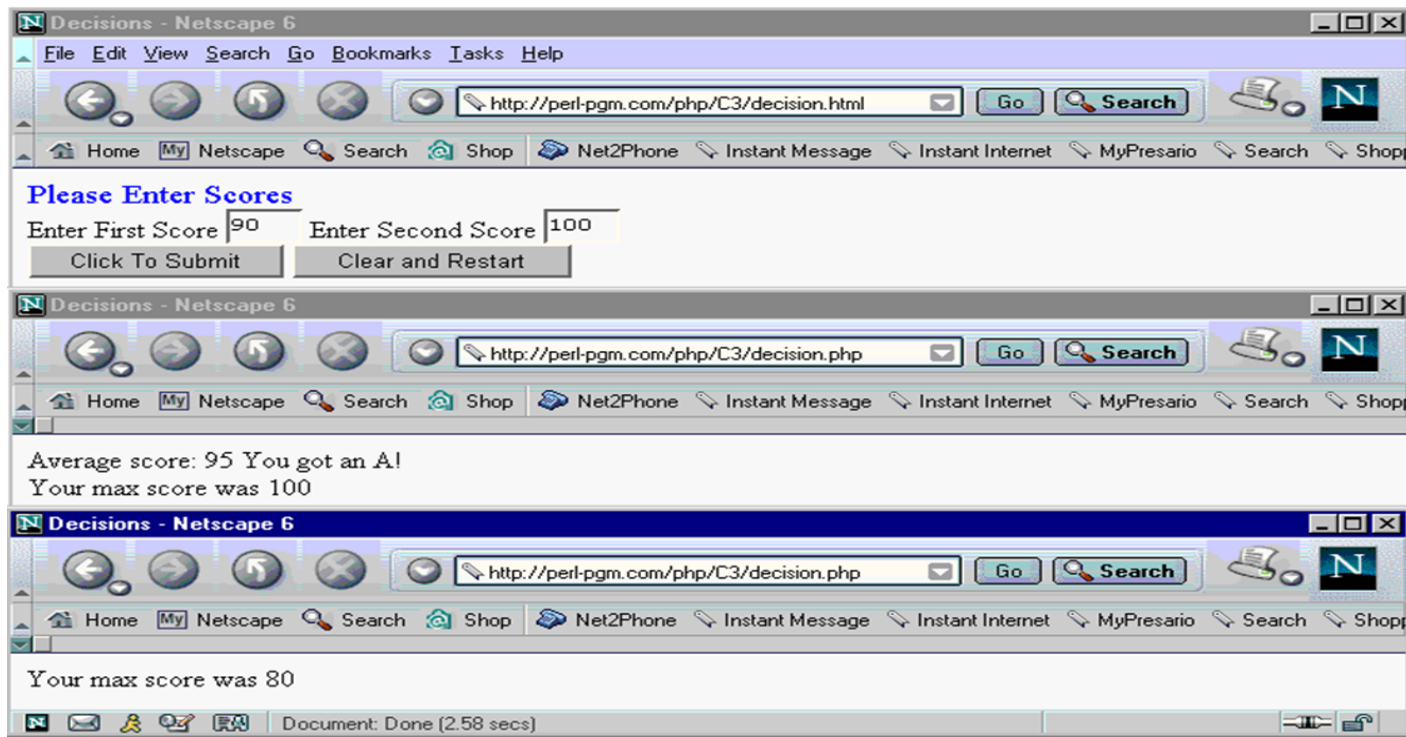
Ορίστε όταν το \$grade2 είναι  
περισσότερο από \$grade1



# Έξοδος παραδείγματος «max score»

Ο προηγούμενος κώδικας μπορεί να εκτελεστεί σε:

<http://webwizard.aw.com/~phppgm/C3/decision.html>





# Συγκρίνοντας *Strings*

---

- ❖ Η PHP αντιπροσωπεύει Strings χρησιμοποιώντας τις **ASCII** τιμές κώδικα. (American Standard Code for Information Interchange).
- Ο ASCII παρέχει έναν τυπικό, αριθμητικό τρόπο για να εκπροσωπεί τους χαρακτήρες σε ένα υπολογιστή.
- Κάθε γράμμα, αριθμός και σύμβολο μεταφράζεται σε έναν κωδικό αριθμό.
- Το “A” είναι στον ASCII code το 65, το “B” είναι το 66, το “C” είναι το 67, και πάει λέγοντας.
- Το “a” στον ASCII code είναι το 97, το “b” είναι το 98, το “c” είναι το 99.
- Στον ASCII το “A” είναι μικρότερο από το “a”, το “B” είναι μικρότερο από το “b”, και το “c” είναι μικρότερο από το “d”.
- Οι ASCII χαρακτήρες έχουν μεταβλητές χαμηλότερες από γράμματα. Ο ASCII χαρακτήρας “1” είναι μικρότερος από το “a” ή το “A”.



# Συγκρίνοντας *Strings*

---

Μπορείτε να χρησιμοποιήσετε τον τελεστή `==` για να ελέγχει αν μία χορδή είναι ίση με την άλλη. Για παράδειγμα:

```
$name1 = "George"; $name2 = "Martha";  
if ($name1 == $name2) {  
    print ("$name1 is equal to $name2" );  
} else {  
    print ("$name1 is not equal to $name2");  
}
```

Θα έχει έξοδο : “George is not equal to Martha”.



# Συγκρίνοντας *Strings*

---

- Επίσης μπορείτε να χρησιμοποιήσετε τους τελεστές `<`, `>`, `<=`, και `>=` για να συγκρίνετε τιμές συμβολοσειράς χρησιμοποιώντας τιμές κώδικα ASCII.

Για παράδειγμα:

```
$name1 = "George"; $name2 = "Martha";  
if ($name1 < $name2) {  
    print ("$name1 is less than $name2");  
} else {  
    print ("$name1 is not less than $name2");  
}
```

Θα έχει έξοδο: "George is less than Martha".



# Ένα πλήρες παράδειγμα

---

Εξετάστε την ακόλουθη εφαρμογή:

- Συγκρίνει δύο συμβολοσειρές εισόδου.
- Χρησιμοποιεί το στοιχείο της φόρμας HTML που ρυθμίζει τις μεταβλητές \$first και \$second.

First Name: maxlength="15" name="first">

Θέτει \$first

Second Name: maxlength="15" name="second">

Θέτει \$second



# Λήψη κώδικα

---

```
1. <html>
2. <head><title>String Comparison Results</title></head>
3. <body>
4. <?php
5. print ("First=$first Second=$second<br>");
6. if ($first == $second) {
7.   print ("$first and $second are equal");
8. }
9. if ($first < $second) {
10.  print ("$first is less than $second");
11. }
12. if ($first > $second) {
13.  print ("$first is greater than $second");
14. }
15. ?></body></html>
```

Έξοδος, αν είναι το \$first  
ίσο με \$second

Ορίστε, όταν το \$second  
είναι μικρότερο από το \$first

Ορίστε, όταν είναι το \$first  
μεγαλύτερο από το \$second



# Λήψη κώδικα με τα *REGISTER\_GLOBALS* απενεργοποιημένα

---

```
1. <html>
2. <head><title>String Comparison Results</title></head>
3. <body>
4. <?php
5. $first = $_POST["first"];
6. $second = $_POST["second"];
7. print ("First=$first Second=$second<br>");
8. if ($first == $second) {
9.   print ("$first and $second are equal");
10. }
11. if ($first < $second) {
12.   print ("$first is less than $second");
13. }
14. if ($first > $second) {
15.   print ("$first is greater than $second");
16. }
17. ?></body></html>
```

← Αποκτήστε τις τιμές του \$first και του \$second

← Έξοδος, αν το \$first είναι ίσο με το \$second

← Ορίστε, όταν το \$second είναι μικρότερο από το \$first

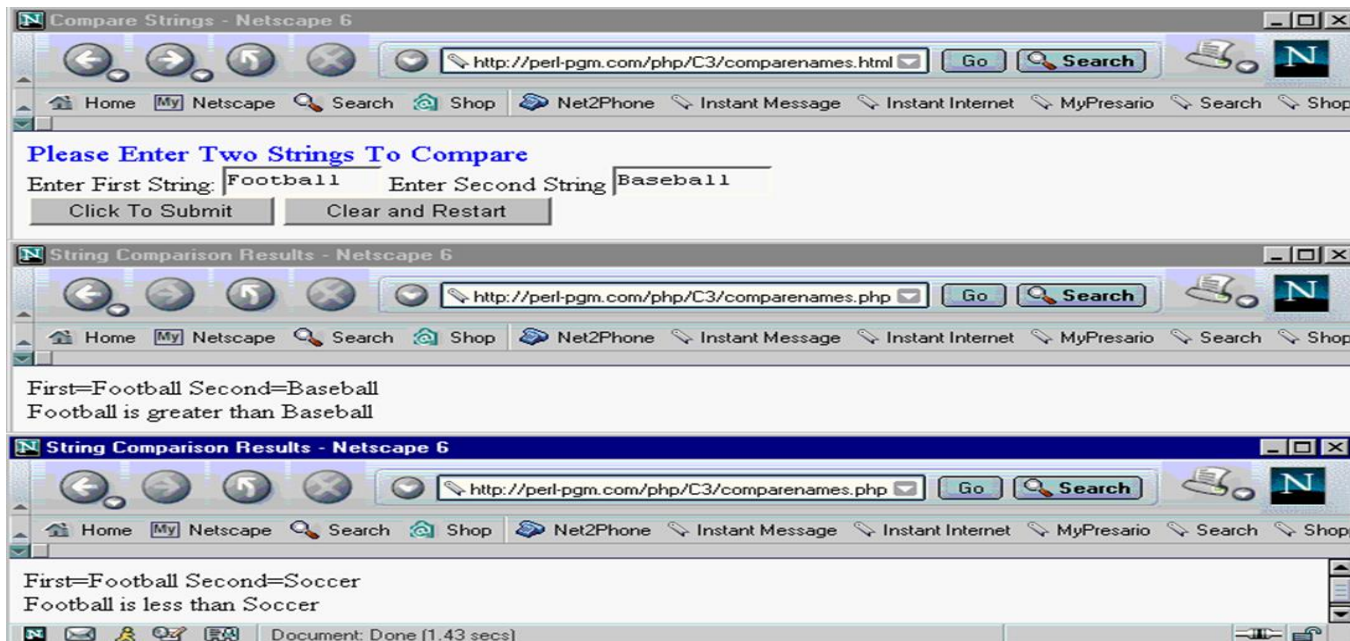
← Ορίστε, πότε το \$first είναι μεγαλύτερο από το \$second



# Έξοδος παραδείγματος σύγκρισης

Ο προηγούμενος κώδικας μπορεί να εκτελεστεί σε:

<http://webwizard.aw.com/~phppgm/C3/comparenames.html>



# Χρησιμοποιώντας τη ρήτρα *elseif*

---

Χρησιμοποιήστε μια *elseif* ρήτρα με μια δήλωση *if* για να της καθορίσετε μια πρόσθετη προϋπόθεση δοκιμής.

```
if (test expression) {  
    one or more PHP statements  
} elseif (test expression)  
    one or more PHP statements  
}
```

Το παραπάνω σενάριο ελέγχει την *elseif* έκφραση όταν η συνθήκη δοκιμής για το *if* όταν είναι *false*.





# Χρησιμοποιώντας τη ρήτρα *elseif*

---

Ένα ή περισσότερα *elseif* μπορούν να χρησιμοποιηθούν με μια εντολή *if*

```
if ($hour < 9) {  
    print "Sorry, it is too early.";  
} elseif ($hour < 12) {  
    print "Good morning. The hour is $hour. ";  
    print "How can we help you?";  
} elseif ($hour < 13) {  
    print "Sorry, we are out to lunch. ";  
} elseif ($hour < 17) {  
    print "Good afternoon. The hour is $hour. ";  
    print "How can we help you?";  
} elseif ($hour <= 23) {  
    print "Sorry, we have gone home already.";  
}
```

Ελέγξτε αυτήν τη δοκιμαστική έκφραση όταν η πρώτη προϋπόθεση είναι *false*.

Ελέγξτε αυτήν δοκιμαστική έκφραση όταν οι πρώτες δύο Προϋποθέσεις είναι όλες *false*

Ελέγξτε αυτήν τη δοκιμαστική έκφραση όταν οι τρεις πρώτες προϋποθέσεις είναι όλες *false*



# Χρησιμοποιώντας τη ρήτρα *else*

---

- Χρησιμοποιήστε μια ρήτρα *else* με ένα *if* και ενδεχομένως, μια ή περισσότερες άλλες ρήτρες.
- Καθορίστε το σύνολο των δηλώσεων που θα εκτελούνται όταν όλες οι προηγούμενες συνθήκες δοκιμής είναι *false*.
- Έχει την ακόλουθη γενική μορφή που εμφανίζεται στο παράδειγμα:

```
if (test expression) {  
one or more PHP statements  
} else {  
one or more PHP statements  
}
```



# Χρησιμοποιώντας τη ρήτρα *else*

---

Για παράδειγμα, αν η `$count` είχε τιμή `-75`, τότε η έξοδος θα ήταν:  
“Illegal value for count = `-75`”

```
if ( $count == 0 ) {  
    print ("Time to reorder.");  
    $reorder=1;  
} elseif ( $count == 1 ) {  
    $reorder=1;  
    print ("Warning: we need to start reordering.");  
} elseif ( $count > 1 ) {  
    $reorder = 0;  
    print ("We are OK for now.");  
} else {  
    print ("Illegal value for count = $count");  
}
```



# Ένα πλήρες παράδειγμα

---

Πλήρες παράδειγμα που επεκτείνει την τάξη, κατά μέσο όρο, για να καθορίσετε έναν βαθμό γράμματος (A, B, C, D, ή F) και να αλιευτούν παράνομες εισροές.

Χρησιμοποιήστε την παρακάτω φόρμα HTML για την είσοδο:

Enter First Score `<input type="text" size="4"  
maxlength="7" name="grade1">`

← Θέτει \$grade1

Enter Second Score `<input type="text" size="4"  
maxlength="7" name="grade2">`

← Θέτει \$grade2



# Λήψη κώδικα

```
1. <html>
2. <head><title>Grade Calculation</title></head>
3. <body>
4. <?php
5. $average = ($grade1 + $grade2) / 2;
6. if ($average > 89) {
7.     print ("Average=$average You got an A");
8. } elseif ($average > 79) {
9.     print ("Average=$average You got a B");
10. } elseif ($average > 69) {
11.     print ("Average=$average You got a C");
12. } elseif ($average > 59) {
13.     print ("Average=$average You got a D");
14. } elseif ($average >= 0) {
15.     print (" Average=$average You got an F");
16. } else {
17.     print ("Illegal average less than 0 average=$average");
18. }
19. $max=$grade1;
20. if ($grade1 < $grade2) {
21.     $max = $grade2;
22. }
23. print ("<br>Your max score was $max");
24. ?> </body></html>
```

Υπολογίστε τον μέσο όρο του  
\$grade1 και \$grade2

Ελέγξτε αν το \$average  
είναι "A", "B", "C", "D"  
ή "F"



# Λήψη κώδικα με τα REGISTER\_GLOBALS απενεργοποιημένα

```
1. <html> <head><title>Grade Calculation</title></head>
2. <body>
3. <?php
4. $grade1 = $_POST["grade1"]; $grade2 = $_POST["grade2"];
5. $average = ($grade1 + $grade2) / 2;
6. if ($average > 89) {
7.   print ("Average=$average You got an A");
8. } elseif ($average > 79) {
9.   print ("Average=$average You got a B");
10. } elseif ($average > 69) {
11.   print ("Average=$average You got a C");
12. } elseif ($average > 59) {
13.   print ("Average=$average You got a D");
14. } elseif ($average >= 0) {
15.   print ("Grade=$grade You got an F");
16. } else {
17.   print ("Illegal average less than 0 average=$average");
18. }
19. $max=$grade1;
20. if ($grade1 < $grade2) {
21.   $max = $grade2;
22. }
23. print ("<br>Your max score was $max");
24. ?> </body></html>
```

Αποκτήστε τιμές από το \$grade1 και το \$ grade2

Υπολογίστε τον μέσο όρο του \$ grade1 και \$ grade2

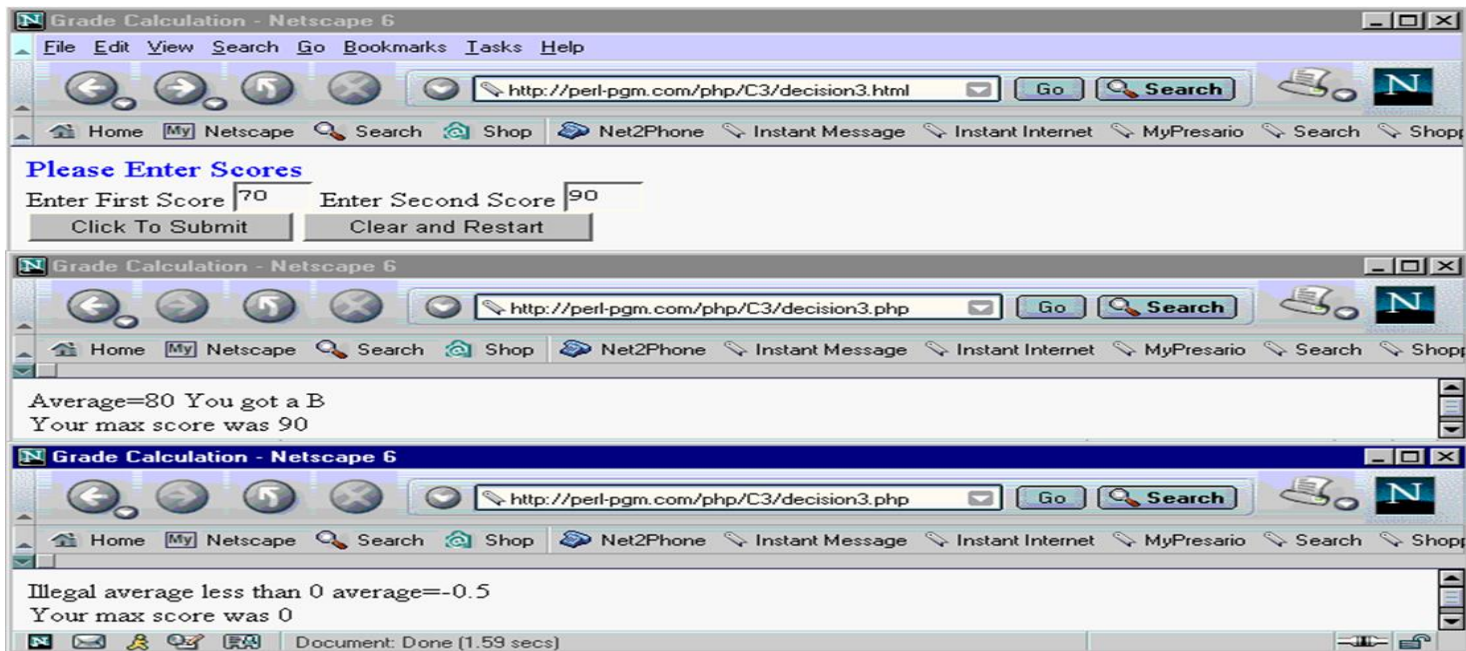
Ελέγξτε αν το \$average είναι "A", "B", "C", "D" ή "F"



# Έξοδος παραδείγματος «max score»

Ο προηγούμενος κώδικας μπορεί να εκτελεστεί σε:

<http://webwizard.aw.com/~phppgm/C3/decision.php>



# Χρησιμοποιώντας τη δήλωση *switch*

---

Χρησιμοποιήστε τη δήλωση *switch* ως μια άλλη δοκιμή υπό όρους

```
1. switch ($rating) {  
2.   case 1:  
3.     $rated = "Poor";  
4.     print "The rating was $rated";  
5.     break;  
6.   case 2:  
7.     $rated = "Fair";  
8.     print "The rating was $rated";  
9.     break;  
10.  case 3:  
11.   $rated = "Good";  
12.   print "The rating was $rated";  
13.   break;  
14.  default:  
15.   print "Error: that rating does not exist";  
16. }
```

Κλείστε τις αγκύλες

Εκτελέστε τα όταν  
έχει το *\$rating* τιμή 1

Εκτελέστε τα όταν το  
*\$rating* έχει τιμή 2

Εκτελέστε τα όταν το  
*\$rating* έχει τιμή 2

Όταν η τιμή δεν είναι  
1, 2 ή 3





# Χρησιμοποιώντας Loops για επαναλήψεις καταστάσεων

---

- Τα scripts μπορούν να χρησιμοποιήσουν τις δηλώσεις για να επαναλάβουν τμήματα του κώδικα.
- Πλεονεκτήματα των loops include.
  - Τα σενάρια μπορούν να είναι πιο συνοπτικά.
  - Μπορείτε να γράψετε πιο ευέλικτα σενάρια.
- Θα συζητήσουμε τα while loops και τα for loops τώρα.
  - Θα αναθεωρήσει τα foreach loops αργότερα.



# Χρησιμοποιώντας ένα *for* βρόχο

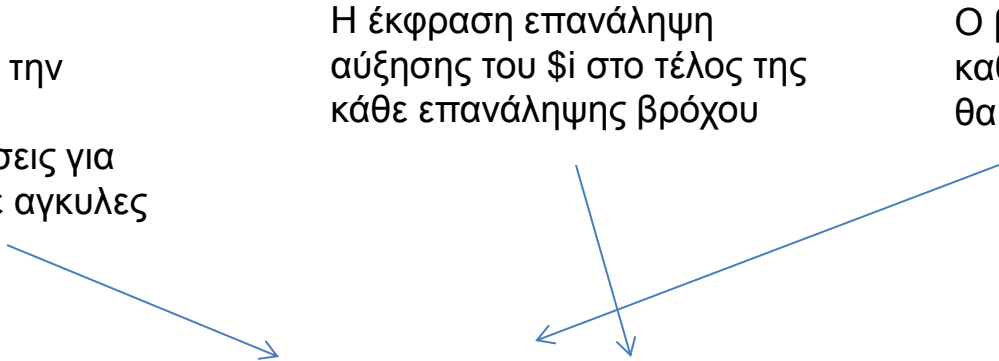
---

Χρησιμοποιήστε ένα *for* βρόχο για να επαναλάβετε της σειρές καταστάσεων, με συγκεκριμένο αριθμό φορών.

η προετοιμασία έκφραση καθορίζει την αρχική τιμή του  $i$ .  
Επισυνάψτε δηλώσεις για να επαναλάβετε σε αγκυλες

Η έκφραση επανάληψη αύξησης του  $i$  στο τέλος της κάθε επανάληψης βρόχου

Ο βρόχος-τελική συνθήκη καθορίζει πότε ο βρόχος θα τελειώσει.



```
for (  $i = 0$ ;  $i < \text{max}$ ;  $i++$  ) {  
  Set of statements to repeat  
}
```



# Πλήρης Παράδειγμα Script

---

```
1. <html><head><title>Loops</title></head>
2. <body><font size="5" color="blue">
3. Generate Square and Cube Values </font>
4. <br>
5. <form action="http://webwizard.aw.com/~phppgm/C3/whileloop.php" method="post">
6. <?php
7. print ("Select Start Number");
8. print ("<select name=\"start\">");
9. for ($i=0; $i<10; $i++) {
10.    print ("<option>$i</option>");
11. }
12. print ("</select>");
13. print ("<br>Select End Number");
14. print ("<select name=\"end\">");
15. for ($i=10; $i<20; $i++) {
16.    print "<option>$i</option>";
17. }
18. print ("</select>");
19. ?>
20. <br><input type="submit" value="Submit">
21. <input type="reset" value="Clear and Restart"> </form></body></html>
```

Επαναλάβετε τη δήλωση εκτύπωσης 10 φορές με τιμές 0, 1, 2, ... 9 για \$ i

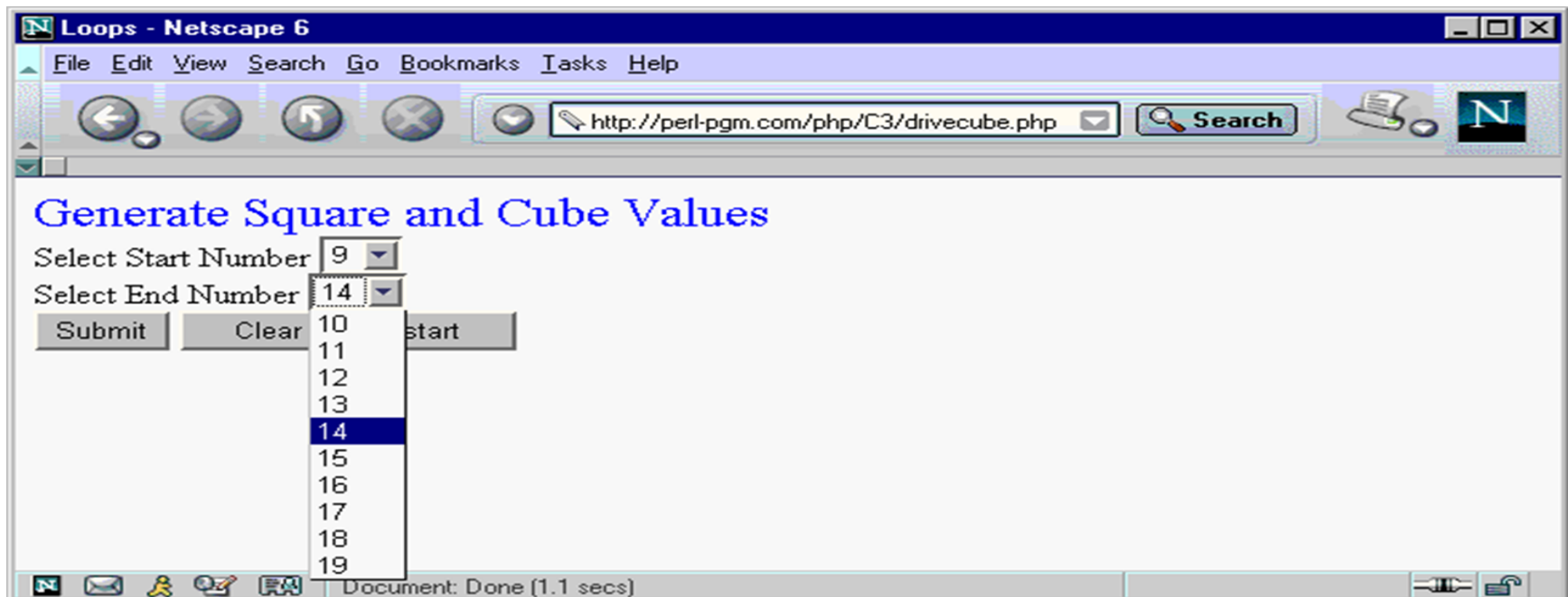
Επαναλάβετε τη δήλωση εκτύπωσης 10 φορές με τιμές 10, 11, 12, ... 19 για \$ i.



# Έξοδος παραδείγματος «δημιουργία τιμών τετραγώνου και κύβου»

Ο προηγούμενος κώδικας μπορεί να εκτελεστεί στο

<http://webwizard.aw.com/~phppgm/C3/drivecube.php>



# Ένα πλήρες παράδειγμα Script

---

```
1. <html>
2. <head><title>While Loop</title></head>
3. <body>
4. <font size="4" color="blue"> Table of Square and Cube Values
   </font>
5. <table border=1>
6. <th> Numb </th> <th> Sqr </th> <th> Cubed </th>
7. <?php
8.   $i = $start;
9.   while ($i <= $end) {
10.    $sqr=$i*$i;
11.    $cubed=$i*$i*$i;
12.    print ("<tr><td>$i</td><td>$sqr</td><td>$cubed</td></tr>");
13.    $i = $i + 1;
14.  }
15. ?></table></body></html>
```



# Ένα Πλήρες Παράδειγμα Script με τα REGISTER\_GLOBALS απενεργοποιημένα

---

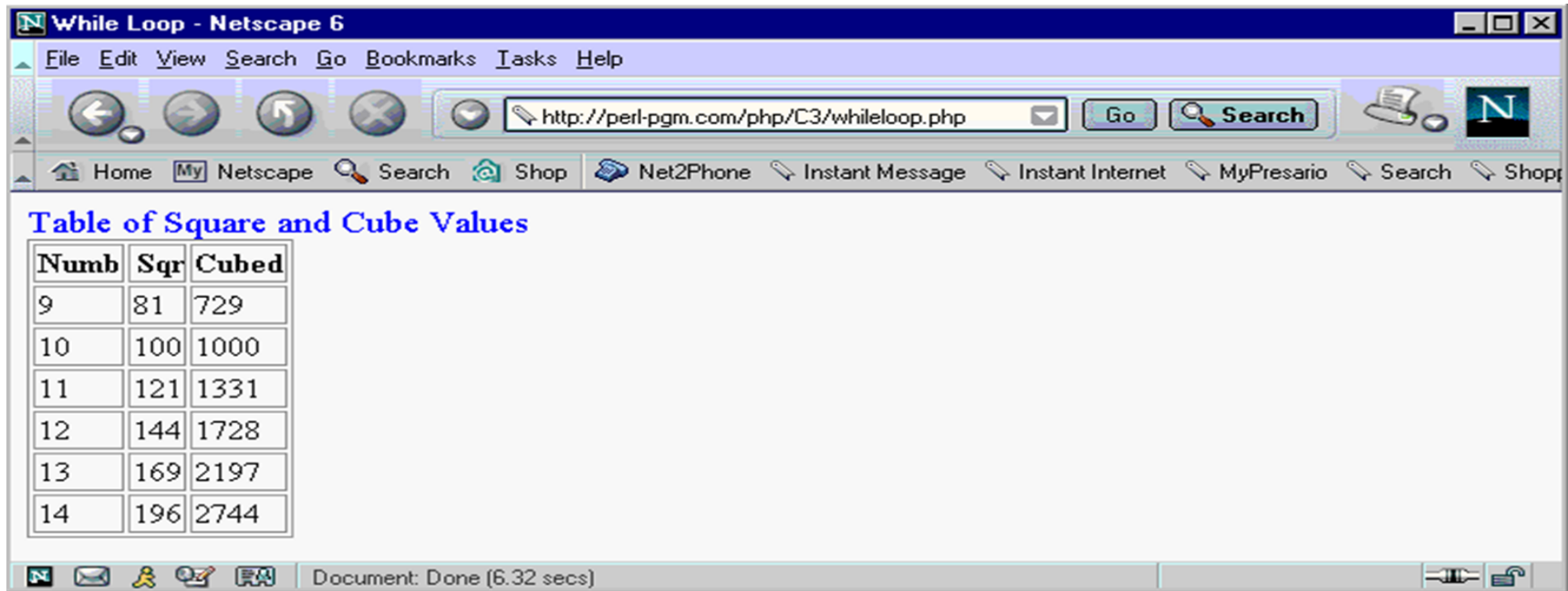
```
1. <html>
2. <head><title>While Loop</title></head>
3. <body>
4. <font size="4" color="blue"> Table of Square and Cube Values </font>
5. <table border=1>
6. <th> Numb </th> <th> Sqr </th> <th> Cubed </th>
7. <?php
8.   $start = $_POST["start"]; $end = $_POST["end"];
9.   $i = $start;
10.  while ($i <= $end) {
11.    $sqr=$i*$i;
12.    $cubed=$i*$i*$i;
13.    print ("<tr><td>$i</td><td>$sqr</td><td>$cubed</td></tr>");
14.    $i = $i + 1;
15.  }
16. ?></table></body></html>
```



# Έξοδος προηγούμενου παραδείγματος σε πίνακα

Η προηγούμενη κώδικας μπορεί να εκτελεστεί σε

<http://webwizard.aw.com/~phppgm/C3/whileloop.php>



The screenshot shows a Netscape 6 browser window titled "While Loop - Netscape 6". The address bar contains the URL "http://perl-pgm.com/php/C3/whileloop.php". The main content area displays a table titled "Table of Square and Cube Values". The table has three columns: "Numb", "Sqr", and "Cubed". The data rows are as follows:

Numb	Sqr	Cubed
9	81	729
10	100	1000
11	121	1331
12	144	1728
13	169	2197
14	196	2744

The browser status bar at the bottom indicates "Document: Done (6.32 secs)".



# Χρησιμοποιώντας τον βρόχο *while*

---

Χρησιμοποιήστε το βρόχο *while* για να επαναλάβετε μια σειρά δηλώσεων εφ' όσον η δοκιμή υπό όρους είναι *true*.

Οι συνθήκες δοκιμής μπαίνουν σε παρένθεση

Επαναλάβετε για όσο διάστημα η δοκιμη υπό όρους είναι *true*

```
while ($ctr < $max) {
```

```
    Set of statements to repeat
```

```
}
```

Κλείστε τις αγκύλες





# Περισσότερα για το βρόχο *while*

---

- Ένας βρόχος *while* θα επαναληφτεί εφ' όσον η δοκιμή υπό όρους του βρόχου είναι *true*.
- Αν αρχικά είναι *false*, τότε οι δηλώσεις στο σώμα βρόχου ποτέ δεν θα τρέξουν.
- Μια κακή ιδέα να δημιουργήσετε έναν άπειρο βρόχο.
- Εάν η υπό όρους δοκιμή του βρόχου είναι πάντα *true*, τότε ο βρόχος δεν θα τελειώσει ποτέ (Άπειρος βρόχος).
- Θα καταναλώνουν πόρους του διακομιστή Web και ενδεχομένως να επιβραδύνει άλλη δραστηριότητα του server. (Ενδέχεται να χρειαστεί να βγείτε από το παράθυρο που τρέχει το script)



# ***TIP χρησιμοποιώντας το βρόχο `while` ή το βρόχο `for` για κάποια προβλήματα***

---

Για κάποιους βρόχους μπορείτε να χρησιμοποιήσετε είτε το `while` βρόχο είτε το `for` βρόχο. Για παράδειγμα, η ακόλουθη έξοδος και για τους δυο βρόχους είναι :  
“i=0 i=1 i=2 i=3 i=4”.

- `for ( $i=0; $i<5; $i++ ) { print “i=$i “; }`
- `$i = 0; while ( $i < 5 ) { print “i=$i “; $i=$i + 1; }`



# Χρησιμοποιώντας λογικούς τελεστές δοκιμής (Test)

---

Η PHP υποστηρίζει μια σειρά από λογικούς τελεστές που μπορούν να χρησιμοποιηθούν για την δημιουργία εκφράσεων ένωσης δοκιμής.

- Μπορούν να χρησιμοποιηθούν μέσα σε μια εντολή if ή while για να καθορίσουν περισσότερες από μία συνθήκες δοκιμής.
- Για παράδειγμα, εξετάστε την ακόλουθη γραμμή

```
while ($x > $max && $found != 1) {
```



# Λογικοί τελεστές δοκιμής (Test)

---

Η PHP υποστηρίζει τρεις λογικούς τελεστές test.

**1.** `&&` - τον τελεστή *AND*. Χρησιμοποιείτε σε περιπτώσεις καταστάσεων και `while` βρόχων.

Παράδειγμα

```
while ($ctr < $max && $flag == 0) {
```

Κάθε φορά που κάποια από  
Αυτές τις εκφράσεις είναι *false*,  
ο βρόχος θα τερματίζεται



# Ο τελεστής OR

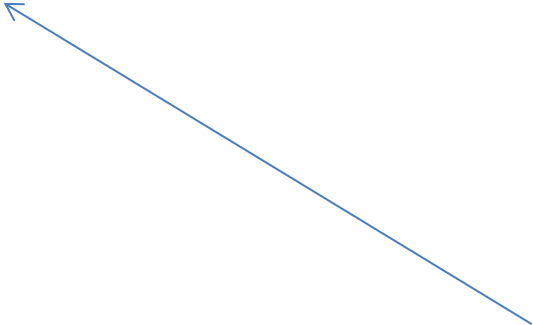
---

**2.** `||` - ο τελεστής OR. Χρησιμοποιείται πολύ όπως και ο AND τελεστής σε περιπτώσεις καταστάσεων και while βρόχων.

Παράδειγμα

```
if ($ctr != $max || $flag == 0) {
```

Διεξάγει τις δηλώσεις  
στο πλαίσιο της εντολής if αν το  
\$ctr δεν είναι ίσο με το \$max  
ή το \$flag ίσο με 0



# Ο τελεστής NOT

---

**3.** ! - ο χειριστής NOT . Χρησιμοποιείται για να ελεγχθεί εάν μια έκφραση είναι *false* (χρησιμοποιείται στους while βρόχους και στις περιπτώσεις καταστάσεων).

Παράδειγμα

```
if (!$flag == 0) {
```

Αυτή η δήλωση είναι *true* ,  
όταν το *\$flag* είναι οτιδήποτε  
άλλο εκτός από το 0



# Παράδειγμα

---

Παράδειγμα, ζητήστε από το χρήστη να μαντέψει ένα «μυστικό» διψήφιο συνδυασμό, με τη χρήση λογικών τελεστών δοκιμής.

Η μορφή HTML εισόδου χρησιμοποιεί τα παρακάτω για να ρυθμίσει το pick1. Μια παρόμοια σύνολα ρυθμίζει και τη μεταβλητή pick2.

```
<font size=4 > Pick a number from 1 to 9 <br>
<input type="radio" name="pick1" value="1">1
<input type="radio" name="pick1" value="2">2
<input type="radio" name="pick1" value="3">3
<input type="radio" name="pick1" value="4">4
<input type="radio" name="pick1" value="5">5
<input type="radio" name="pick1" value="6">6
<input type="radio" name="pick1" value="7">7
<input type="radio" name="pick1" value="8">8
<input type="radio" name="pick1" value="9">9
```



# Ένα πλήρες παράδειγμα Script

---

```
1. <html><head><title>Number Guess Results </title></head>
2. <body>
3. <?php
4. $combo1=5;
5. $combo2=6;
6. if (($pick1 == $combo1) && ($pick2 == $combo2)) {
7.     print ("Congratulations you got both secret numbers
8.     $combo1 $combo2!");
9. } elseif (($pick1 == $combo1) || ($pick2 == $combo2)){
10.     print ("You got one number right.");
11. } else {
12.     print ("Sorry, you are totally wrong!");
13. }
14. ?></body></html>
```





# Ένα πλήρες παράδειγμα Script με τα REGISTER\_GLOBALS απενεργοποιημένα

---

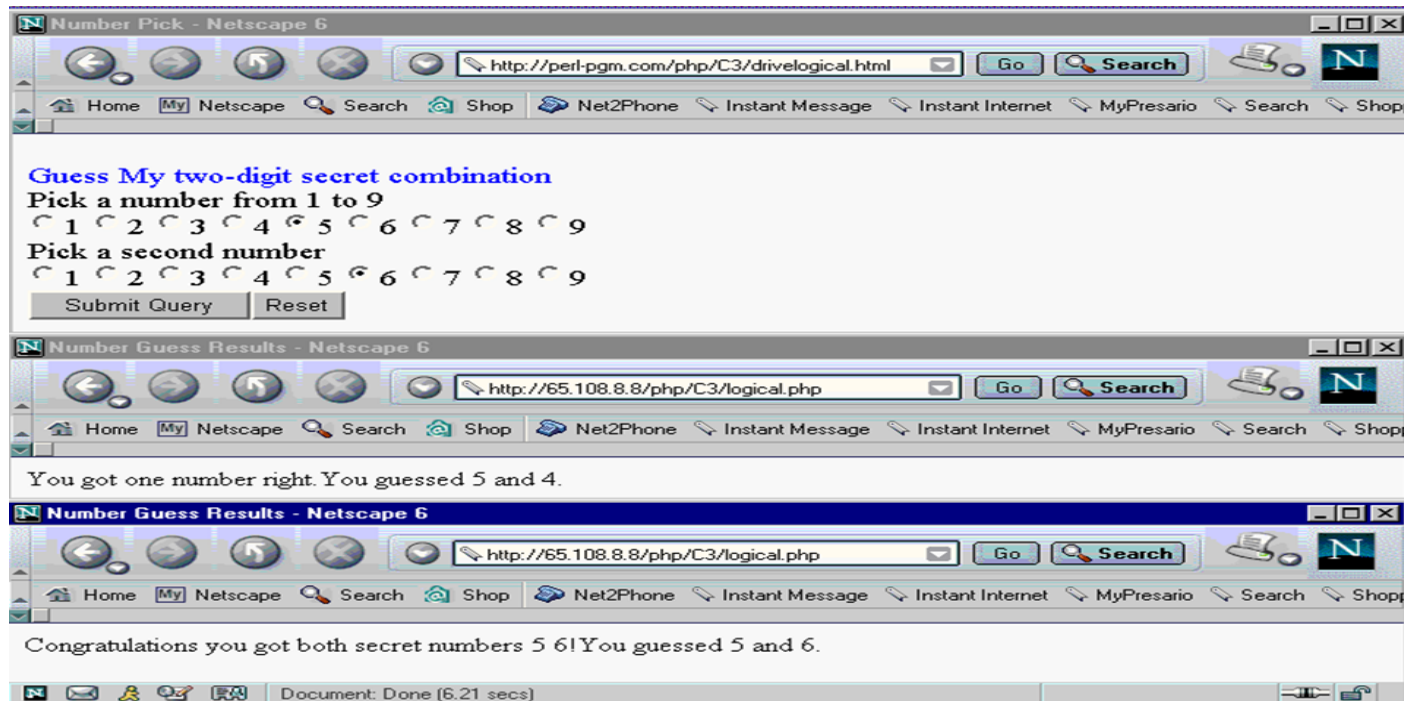
```
1. <html><head><title>Number Guess Results </title></head>
2. <body>
3. <?php
4. $pick1=$_POST["pick1"]; $pick2=$_POST["pick2"];
5. $combo1=5;
6. $combo2=6;
7. if (($pick1 == $combo1) && ($pick2 == $combo2)) {
8.     print ("Congratulations you got both secret numbers
9.     $combo1 $combo2!");
10. } elseif (($pick1 == $combo1) || ($pick2 == $combo2)){
11.     print ("You got one number right.");
12. } else {
13.     print ("Sorry, you are totally wrong!");
14. }
15. ?></body></html>
```



# Έξοδος παραδείγματος «guess the combination»

Ο προηγούμενος κώδικας μπορεί να εκτελεστεί σε :

<http://webwizard.aw.com/~phppgm/C3/drivelogical.html>



# Περίληψη

---

- Χρησιμοποιήστε τις δηλώσεις υπό όρους για την εξέταση ορισμένων συνθηκών και με βάση τα αποτελέσματα της δοκιμής να εκτελέσετε συγκεκριμένες δηλώσεις στο script.
- Τα Loops επεκτείνουν τους τύπους προγραμματισμού με προβλήματα που μπορείτε να λύσετε και σας επιτρέπει να λύσετε κάποια προβλήματα προγραμματισμού με πολλά άλλα συνοπτικά.
- Χρησιμοποιήστε λογικό ΚΑΙ (&&), ΟΡ (||) και ΝΟΤ (!) για τη διεξαγωγή δοκιμών ένωσης.



# Περίληψη

---

- Οι μεταβλητές χρησιμοποιούνται για την αποθήκευση και την πρόσβαση δεδομένων στη μνήμη του υπολογιστή.

Μπορείτε να συνδέσετε μια τιμή με μια μεταβλητή, αλλάξετε την τιμή αυτή, εκτυπώσετε την, και εκτελεστέ πολλές διαφορετικές λειτουργίες σε αυτό.

- Η PHP υποστηρίζει τόσο αριθμητικές όσο και string μεταβλητές που χρησιμοποιούν διαφορετικές μεθόδους για την αξία της χειραγώγησης (για παράδειγμα, συνένωση) από αριθμητικές μεταβλητές.

