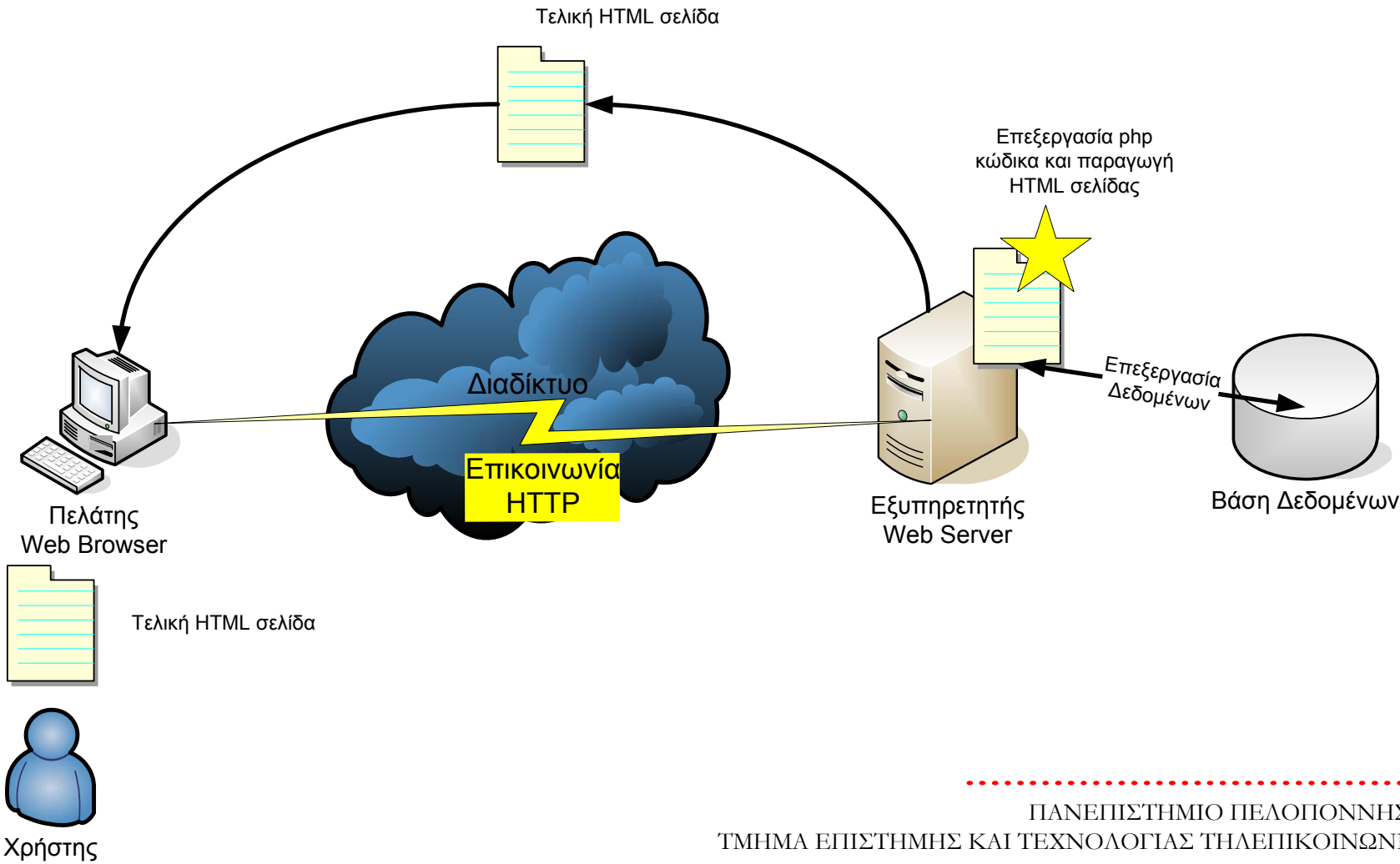
Σχεδίαση Εφαρμογών και Υπηρεσιών Διαδικτύου
4η Διάλεξη: Προγραμματισμός στην πλευρά του εξυπηρετητή: PHP

Δρ. Απόστολος Γκάμας

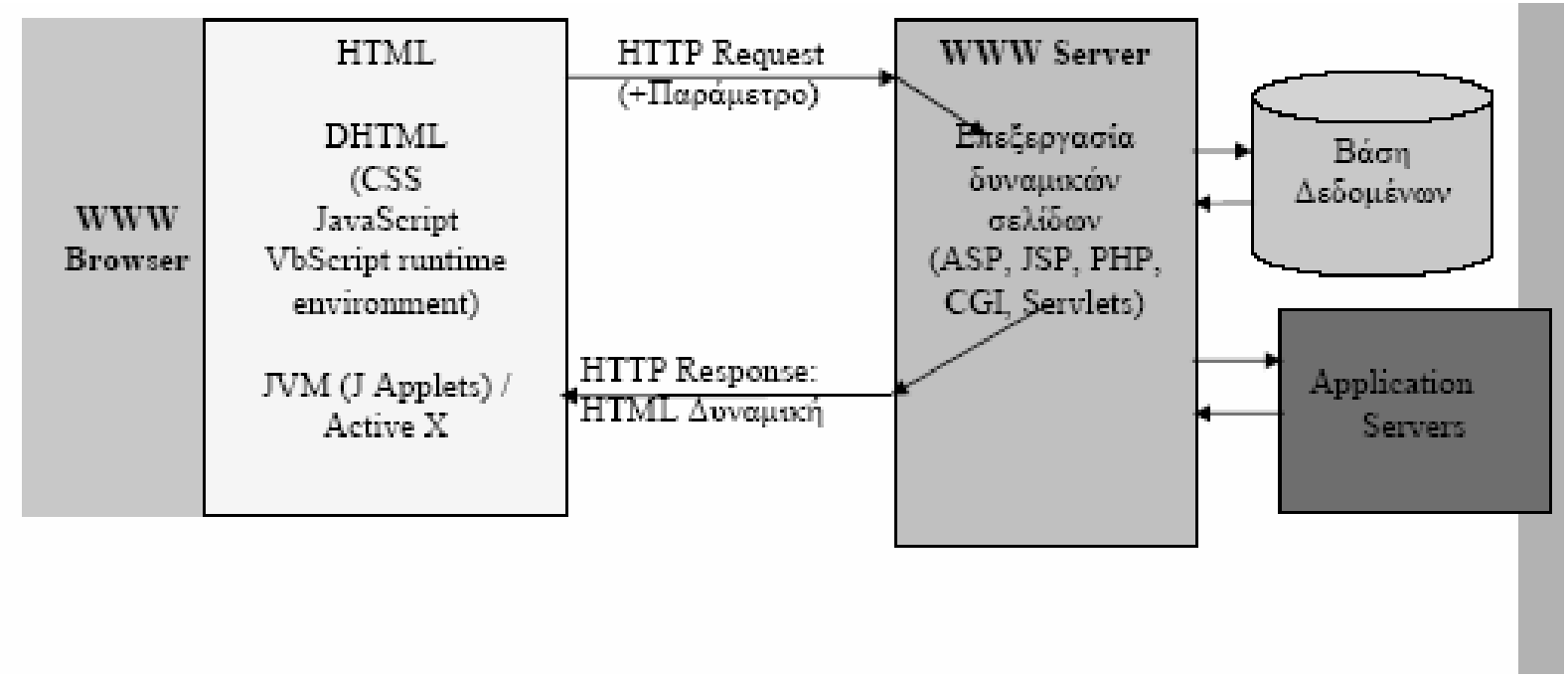
Διδάσκων (407/80)

gkamas@uop.gr

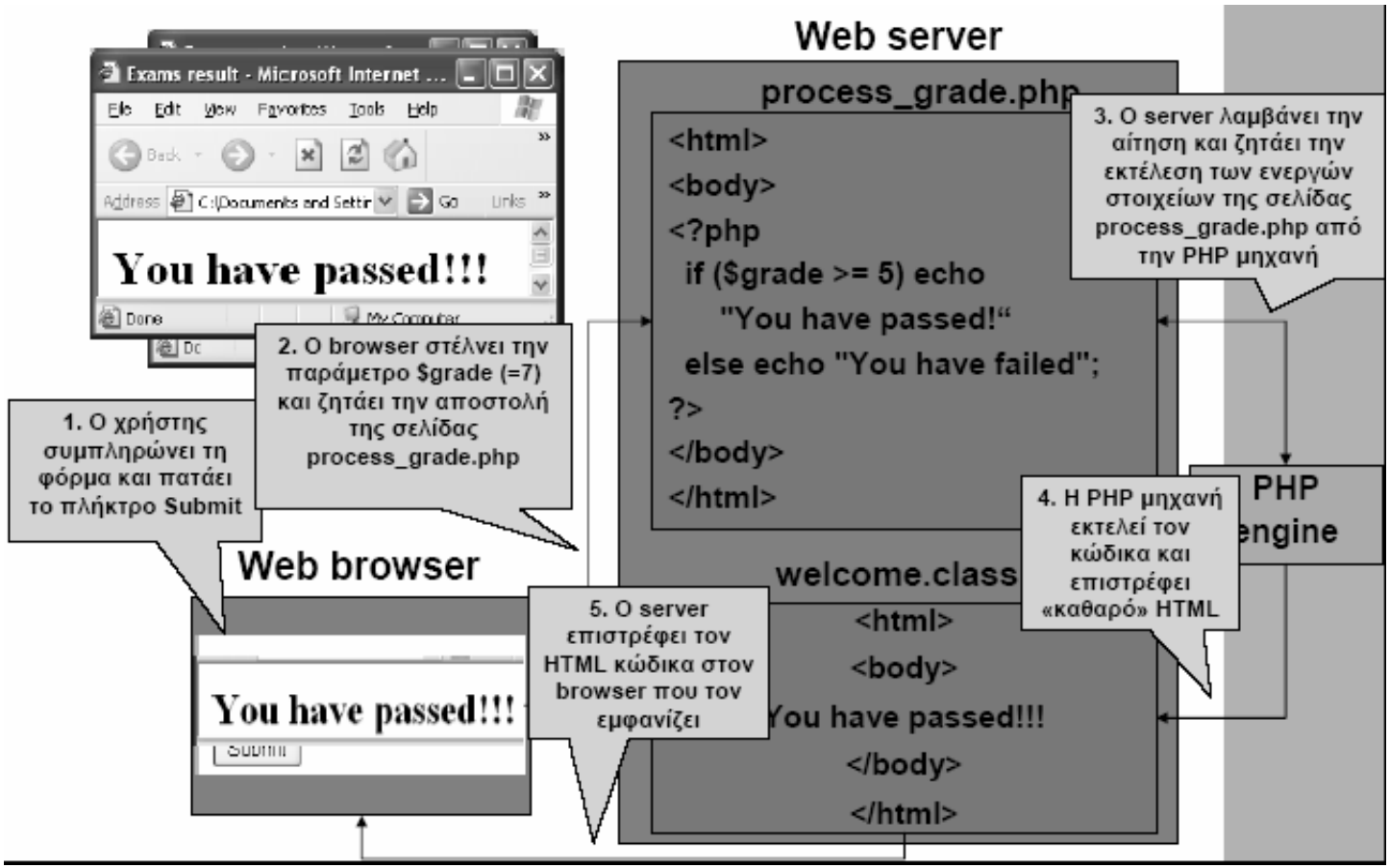
Προγραμματισμός στην πλευρά του εξυπηρετητή (server side)



Προγραμματισμός στην πλευρά του εξυπηρετητή (server side)



Αλληλεπίδραση browser-web server με χρήση τεχνολογίας server side



Server Side: Καταλληλότητα, Πλεονεκτήματα, Μειονεκτήματα

- Καταλληλότητα:
  - Δυναμική / Παραμετρική εμφάνιση περιεχομένου
  - Απαραίτητο όταν απαιτείται επικοινωνία (αλληλεπίδραση) με τον Server
  - Δυνατότητα ελέγχου των πελατών, π.χ. μετρητές επισκέψεων (hit counters), ελεγχόμενη πρόσβαση σε κάποιες σελίδες
- Πλεονεκτήματα:
  - Η επεξεργασία μεταφέρεται στο server, χρησιμοποιείται η ισχύς του server
  - Ο κώδικας είναι κρυφός
  - Η εκτέλεση του κώδικα είναι ανεξάρτητη του browser: στέλνεται «καθαρό»
  - HTML που εμφανίζεται πανομοιότυπο σε κάθε browser
  - Η μοναδική λύση για πρόσβαση στο file system του server
- Μειονεκτήματα:
  - Χρησιμοποιεί πολύτιμη επεξεργαστική ισχύ του server.
  - Κλιμάκωση (scalability);

Server Side: Τεχνολογίες

Τεχνολογία	Server	Πλεονεκτήματα
Active Server Pages (.asp)	MS IIS (Internet Information Server)	Σχετικά εύκολο στη χρήση, καλή ενσωμάτωση/ ολοκλήρωση (integration), καλή υποστήριξη (support) από Microsoft
ASP.NET	IIS	Κλιμάκωση (Scalability), web services.
Java Server Pages (.jsp)		Κλιμάκωση, αξιοπιστία
PHP (.php)	Apache (open source)	Δωρεάν, εύκολη στη χρήση
Java Servlets	Any web server	Προγραμματισμός σε Java, Εύκολη πρόσβαση σε ΒΔ

Γιατί PHP;

- Υψηλή απόδοση: με ένα φτηνό server μπορούν να εξυπηρετηθούν εκατομμύρια επισκέψεων καθημερινά.
- Συνεργάζεται εύκολα με τους περισσότερους database servers (βάσεις δεδομένων)
- Σημαντικό για χτίσιμο πληροφοριακών συστημάτων (π.χ. Εφαρμογές ηλεκτρονικού εμπορίου)
- Ενσωματωμένες βιβλιοθήκες για συνήθεις web διαδικασίες: δυναμική δημιουργία εικόνων, αποστολή email, χειρισμός cookies
- Χαμηλό κόστος: δωρεάν
- Ευκολία μάθησης και χρήσης: η σύνταξή της βασίζεται σε άλλες γλώσσες (μοιάζει με Java, C)
- Υποστηρίζεται από τους περισσότερους web servers σαν module (επιπρόσθετο δομικό στοιχείο)
- Μεταφερσιμότητα (portability): ο ίδιος κώδικας δουλεύει χωρίς αλλαγές και σε άλλο λειτουργικό σύστημα
- Διαθεσιμότητα του κώδικα προέλευσης (open source): μπορούν να πραγματοποιηθούν αλλαγές στη γλώσσα

PHP - Απλό παράδειγμα

```
<html>
<head><title>PHP Example</title>
</head>
<body>
<?php (ή <?)
echo "Hi, I'm a PHP script!";
?>
</body>
</html>
```

Μεταβλητές στην PHP

- Υπάρχουν τριών τύπων μεταβλητές:
  - Βαθμωτή (scalar)
  - Πίνακας (array)
  - Συσχετιζόμενος πίνακας (associative array)
- Οι μεταβλητές είναι ο κύριος μηχανισμός για τη μεταφορά δεδομένων μεταξύ σελίδων ή τμημάτων σελίδων
- Υπάρχουν τρεις βασικές λειτουργίες που μπορούμε να κάνουμε με μία μεταβλητή: – Να την θέσουμε, να την επαναθέσουμε ή να την προσπελάσουμε

Τύποι

- Boolean, integer, floating-point number (float), string, array, object, resource, NULL
- Ο τύπος μιας μεταβλητής ΔΕΝ δηλώνεται αλλά προσδιορίζεται από την τιμή που της δίνεται
- Μετατροπή από ένα τύπο δεδομένων σε άλλο
- $mydouble = (double)$myint

Κανόνες ονομασίας μεταβλητών

- Να αρχίζει με γράμμα ή underscore(_)
- Να αποτελείται από γράμματα, αριθμούς ή underscore (_)
- Να μην είναι δεσμευμένη λέξη (όπως π.χ. print)
- Τα ονόματα των μεταβλητών είναι case-sensitive, π.χ. $baby_names και $Baby_names δεν είναι τα ίδια

Κανόνες ονομασίας μεταβλητών

- Εκτός από κείμενο, ως τιμές σε μεταβλητές μπορούμε να δώσουμε και αριθμούς καθώς και άλλα αντικείμενα (objects, booleans)
- Για να προβάλλουμε κείμενο χρησιμοποιούμε απλά ή διπλά εισαγωγικά:
  - print (“Αυτό είναι ένα παράδειγμα!”);
- Αν θέλουμε να εκτυπώσουμε το κείμενο μαζί με τα εισαγωγικά, χρησιμοποιούμε το χαρακτήρα διαφυγής \, που ορίζει στην PHP να μη θεωρήσει τον επόμενο χαρακτήρα ως μέρος του κώδικα, αλλά ως απλό κείμενο
  - print (“\”Αυτό είναι ένα παράδειγμα!\””);

Συναρτήσεις Μεταβλητών

- string gettype(mixed var): επιστρέφει μια συμβολοσειρά που περιέχει τον τύπο μιας μεταβλητής ή “unknown type”
- int settype(string var, string type): αλλάζει τον τύπο μιας μεταβλητής
- boolean is_array(), is_double(), is_int(), is_string(), is_object(): ελέγχουν τύπους
- int isset(mixed var): ελέγχει αν μια μεταβλητή είναι ορισμένη
- int unset(mixed var): διαγράφει μια μεταβλητή
- int empty(mixed var): ελέγχει αν μια μεταβλητή έχει μηδενική τιμή
- int intval(mixed var), double doubleval(mixed var), string strval(mixed var): μετατρέπουν τον τύπο μιας μεταβλητής

Μεταβλητές τύπου πίνακα

- Οι μεταβλητές τύπου πίνακα ξεκινούν με $, όπως και οι βαθμωτές. Η συνάρτηση array() εκχωρεί μια σειρά τιμών σε έναν πίνακα με τον ακόλουθο τρόπο:
  - $students = array("Μαρία", "Γιάννης", "Λευτέρης");
- Η παραπάνω εντολή αυτόματα εκχωρεί ένα αριθμητικό κλειδί σε κάθε στοιχείο με τη σειρά δίνοντας στο πρώτο στοιχείο το κλειδί 0. Μπορούμε τώρα να αναφερόμαστε π.χ. στο στοιχείο "Λευτέρης" ως $students[2]. Ο ακόλουθος κώδικας θα εκτυπώσει το τρίτο στοιχείο του πίνακα που είναι ο μαθητής Λευτέρης
  - <?php
  - print "$students[2]";
  - ?>

Μεταβλητές τύπου πίνακα

- Υπάρχει και άλλος τρόπος να ορίσουμε έναν πίνακα ή να προσθέσουμε στοιχεία σε έναν ήδη υπάρχοντα πίνακα:
  - $students[] = "Μαρία";
  - $students[] = "Γιάννης";
  - $students[] = "Λευτέρης";
- Για να προσθέσουμε έναν νέο μαθητή γράφουμε (ανεξάρτητα από τον τρόπο που χρησιμοποιήσαμε για τη δημιουργία του πίνακα):
  - $students[] = "Βασίλης";
- Η PHP δίνει αυτόματα στο Βασίλη ένα κλειδί, το αμέσως επόμενο κενό, που στην περίπτωση αυτή είναι το [3].

Μεταβλητές τύπου συσχετιζόμενου πίνακα

- Οι συσχετιζόμενοι πίνακες διαχωρίζουν τα περιεχόμενα στοιχεία όχι με αριθμούς, αλλά με ονόματα που εμείς καθορίζουμε. Μέσα στη συνάρτηση array() καθορίζουμε ζεύγη key=>value. Για παράδειγμα:
  - $stud = array(
  - name=>”John”,
  - haircolor=>”black”,
  - eyecolor=>”green”,
  - age=>17);
- Μπορούμε να πάμε σε οποιοδήποτε στοιχείο του πίνακα μέσω των ονομάτων των κλειδιών που ορίσαμε. Για παράδειγμα:
- print $stud[eyecolor];θα δώσει green.

Μεταβλητές τύπου συσχετιζόμενου πίνακα

- Μπορούμε επίσης να θέσουμε κάθε στοιχείο ξεχωριστά:
  - $stud[name] = ”John”;
  - $stud[haircolor] = ”black”;
  - $stud[eyecolor] = ”green”;
  - $stud[age] = 17;
- Ένας πολυδιάστατος πίνακας είναι ένας πίνακας που περιέχει άλλους πίνακες.

Πολυδιάστατος πίνακας

```
$stud = array(
array (name=>"John",haircolor=>"black",eyecolor=>"green",age=>17),
array (name=>"Mary",haircolor=>"blond",eyecolor=>"blue",age=>16),
array (name=>"Kenny",haircolor=>"brown",eyecolor=>"
      brown",age=>17),
array (name=>"Bill",haircolor=>"blond",eyecolor=>"green",age=>16)
);
```

- Για να προσπελάσουμε ένα στοιχείο:

```
print $stud[2][age];
```

Συναρτήσεις Πινάκων

- Ταξινόμηση
  - Αριθμητικών
    - $products=array("Tires","Oil"); sort($products);
    - $products=array(13,3,7); sort($products);
  - Συσχετιζόμενων
    - $prices = array("tires"=>100,"Oil"=>10, "Spark Plugs"=>4);
    - asort($prices) // ("Spark Plugs"=>4,"Oil"=>10, "Tires"=>100)
    - ksort($prices) // ("Oil"=>10, "Spark Plugs"=>4 "Tires"=>100)
  - Αντίστροφες Ταξινομήσεις
    - array_reverse();
- Τυχαία σειρά: shuffle();
- Πλήθος στοιχείων πίνακα: $num = count($products)

Τελεστές σύγκρισης

Τελεστής	Περιγραφή
==	Επιστρέφει true εάν η τιμή της μεταβλητής αριστερά είναι ίση με αυτή δεξιά. Αλλιώς, επιστρέφει false.
===	Παρόμοιος με ==, αλλά επιστρέφει true μόνο αν οι μεταβλητές είναι του ίδιου τύπου. (μόνο στην PHP4)
!=	Επιστρέφει true εάν η τιμή της μεταβλητής αριστερά δεν είναι ίση με αυτή δεξιά. Αλλιώς, επιστρέφει false.
<>	Επιστρέφει true εάν η τιμή της μεταβλητής αριστερά δεν είναι ίση με αυτή δεξιά. Αλλιώς, επιστρέφει false.
!==	Παρόμοιος με !=, αλλά επιστρέφει true και αν οι μεταβλητές δεν είναι του ίδιου τύπου. (μόνο στην PHP4)
<	Επιστρέφει true εάν η τιμή της μεταβλητής αριστερά είναι μικρότερη από αυτή δεξιά. Αλλιώς, επιστρέφει false.
>	Επιστρέφει true εάν η τιμή της μεταβλητής αριστερά είναι μεγαλύτερη από αυτή δεξιά. Αλλιώς, επιστρέφει false.
<=	Επιστρέφει true εάν η τιμή της μεταβλητής αριστερά είναι μικρότερη ή ίση από αυτή δεξιά. Αλλιώς, επιστρέφει false.
>=	Επιστρέφει true εάν η τιμή της μεταβλητής αριστερά είναι μεγαλύτερη ή ίση από αυτή δεξιά. Αλλιώς, επιστρέφει false.

Λογικοί και Αριθμητικοί τελεστές

- Λογικοί
  - and ή &&, or ή||, xor, !
- Αριθμητικοί
  - +, -, *, /, %
- Υπάρχουν ακόμα και οι τελεστές αύξησης/μείωσης και οι σύνθετοι τελεστές (π.χ. $a+=5;)

if-else-elseif

```
if(cond)
        …; // μια εντολή
 if(cond)
 {
     …;
     …;
 }
```

```
if(cond)
 {
     …;
 }
 else
 {
     …;
 }
```

```
if(cond)
 {
     …;
 }
 elseif
 {
     …;
 }
 elseif
 {
     …;
 }
```

Switch

```
switch($var)
{
    case "a":
        …;
        break;
    case "a":
        …;
        break;
    default :
        …;
        break;
}
```

while-for-do...while

```
while(cond)
{
      …;
}
```

```
for(exp;cond;exp)
 {
    …;
 }
```

```
do
{
      …;
}
while(cond)
```

Σπάζοντας δομές ελέγχου,επαναλήψεις και Script

- Έξοδος από έλεγχο
  - break
- Μεταπήδηση επόμενη επανάληψη βρόγχου
  - continue
- Σταμάτημα εκτέλεσης PHP Script
  - Exit

Σχόλια

```
<?php echo “Test"; // one-line comment
/* This is a
multi line comment */
echo “Test"; # This is shell-style style comment
?>
```

Επαναχρησιμοποίηση Κώδικα

- Δυνατότητα για επαναχρησιμοποίηση κώδικα από άλλα αρχεία (php, html, οποιοδήποτε άλλο)
- Συνήθως αρχεία .inc αλλά προσοχή! Ο πηγαίος κώδικας ενός .inc μπορεί να φανεί αν φορτωθεί απευθείας από browser -> καλύτερα να χρησιμοποιούμε .php ή «κρύψιμο» των .inc (αποθήκευση σε κατάλογο που δεν είναι 'δημοσιευμένος'
- require() & include()
- include()-> δεν υπολογίζεται αν η εντολή δεν εκτελεστεί
- Πλεονεκτήματα επαναχρησιμοποίησης κώδικα:
- Μικρότερο «κόστος» (όχι περιττή επανεγγραφή κώδικα)
- Αυξημένη αξιοπιστία (αν ο κώδικας δουλεύει κάπου, δουλεύει με τον ίδιο τρόπο παντού)

Require

```
// reusable.php
<?
echo "Here is a very simple PHP statement.<BR>";
?>


<?
echo "This is the main file.<BR>";
require( "reusable.php" );
echo "The script will end now.<BR>";
<?
```

Include

If ($var==true)

require("file1.inc"); //αυτή η εντολή θα υπολογιστεί κατά την
//ανάλυση του script

Else require ("file2.inc"); // κι αυτή ομοίως

If ($var==true)

include ("file1.inc"); // θα υπολογιστεί όταν εκτελεστεί η εντολή

// αν ικανοποιηθεί η συνθήκη

else

include ("file2.inc");

Χρησιμοποιώντας συναρτήσεις στην PHP

- Κλήση συνάρτησης ΧΩΡΙΣ πέρασμα παραμέτρων
  - function_name();
- Κλήση συνάρτησης ΜΕ πέρασμα παραμέτρων
  - function_name(“parameter”);
- Παραδείγματα:
  - function_name(2);
  - function_name(“string”);
  - function_name($variable);

Κλήση συναρτήσεων στην PHP

- Η κλήση εξαρτάται από το πρωτότυπο της συνάρτησης
  - π.χ. array explode (string separator, string str [int limit])
- Κλήση της συνάρτησης explode:
  - $str = “abc def ghi”;
  - $str_array = explode(“ ”, str); // επιστρέφει array 3 στοιχείων
  - $str_array = explode(“ ”, str, 2); // επιστρέφει array 2 στοιχείων
  - Οι κλήσεις σε συναρτήσεις ΔΕΝ είναι ευαίσθητες σε κεφαλαία-πεζά: function_name() = FUNCTION_NAME() = Function_Name()
- ΠΡΟΣΟΧΗ! Τα ονόματα των μεταβλητών ΕΙΝΑΙ ευαίσθητα σε πεζά-κεφαλαία: $name ≠ $Name

Ορίζοντας τις δικές μας συναρτήσεις

- Δήλωση μιας απλής συνάρτησης:
  - function my_function() {
  - echo “My function was called”;
  - }
- Κλήση της συνάρτησης:
  - my_function();
- Περιορισμοί στην ονομασία συναρτήσεων:
  - Όχι ίδια ονόματα με υπάρχουσες συναρτήσεις
  - Ονόματα μονάχα από γράμματα, ψηφία και χαρακτήρες υπογράμμισης
  - Τα ονόματα δεν μπορούν να ξεκινούν με ψηφίο
  - Έγκυρα ονόματα: name(), name2(), name_three()
  - Άκυρα ονόματα: 5name(), name-six(), explode()

Παραδείγματα Χρήσης

```
<html>
<body>
<b>
<?
echo "Hello, World!";
?>
</b>
<?
echo " <b> Hello, World! </b>"
?>
</body>
</html>
```

Παραδείγματα Χρήσης

```
<html><body>
<?
$greeting="Hello ";
$num=3+2;
$num=$num+1;
print $greeting.$num." people!";
?>
</body></html>
```

Παραδείγματα Χρήσης

```
<html><body>
<?
$strUser=$_SERVER["HTTP_USER_AGENT"];
print $strUser;
?>
</body></html>
```

Παραδείγματα Χρήσης

```
<?
$h=strftime("%H");
print "<p>".strftime("%m/%d/%Y %H:%M:%S %p")."</p>";
if ($h<12)
        print "Kalhmera";
  else{
  if ($h==12)
        print "Kalo mesimeri";
    else
        print "Kalo apogeyma";
  }
?>
```

Παραδείγματα Χρήσης

```
<?
for ($i=1; $i<=6; $i=$i+1){
  print "<h".$i.">This is header ".$i."</h".$i.">";
 }
?>
```

Παραδείγματα Χρήσης

```
<?
$students[0]="Nikos";
$students[1]="Maria";
for ($i=0; $i<=1; $i=$i+1){
  print $students[$i]."<br>";
}
?>
```

Παραδείγματα Χρήσης

```
<?
$students[0]="Nikos";
$students[1]="Maria";
$students[2]="Mpampis";
foreach ($students as $name){
  print $name."<br>";
}
?>
```

Παραδείγματα Χρήσης

```
<h1> Footer Test Page</h1>

<p>

This is the first paragraph.

</p>

<p>2nd p<p>3rd p<p>4th p

<? include ("../footer.htm") ?>
```