



Αντώνης Χρυσόπουλος
Κατερίνα Κυπριώτη
Στέλιος Μόσχογλου
Θεοδόσης Σουργκούνης

ΒΑΣΙΚΕΣ ΕΝΤΟΛΕΣ PHP

PHP 2: The Return of PHP – Η Δευτέρα Παρουσίαση

Παρακαλώ τι θα πάρετε?

1. Βασικές Οδηγίες
2. Μεταβλητές
3. Αλφαριθμητικά - Αριθμοί
4. Τελεστές
5. Συγκρίσεις
6. Βρόχοι επανάληψης
7. Συναρτήσεις
8. Χειρισμός φορμών

Τι έχετε φρέσκο?

1. Βασικές Οδηγίες

PHP Basics

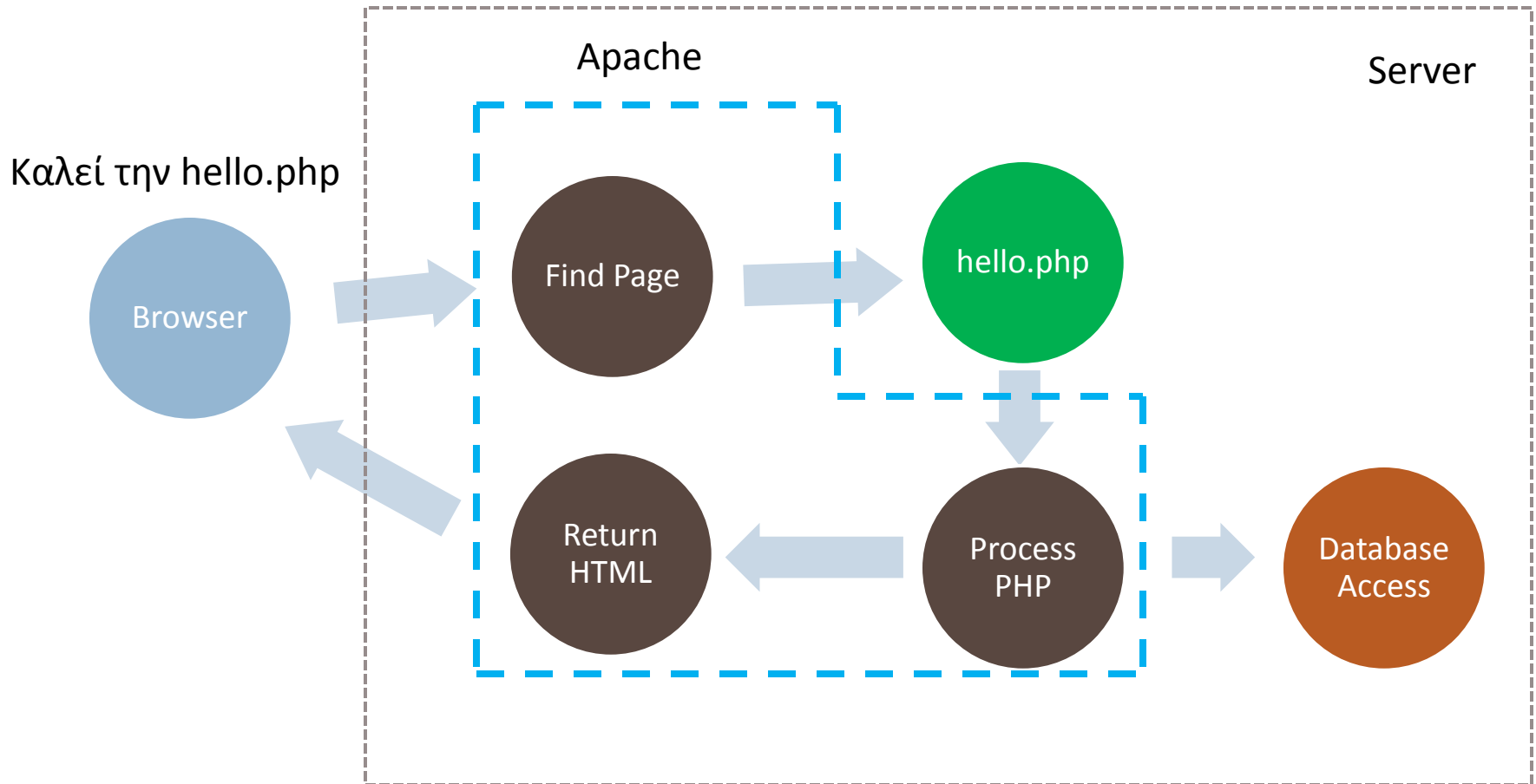
- Τα PHP αρχεία έχουν κατάληξη .php
 - ▣ Ο χρήστης **δεν** έχει άμεση πρόσβαση στα αρχεία αυτά καθώς βρίσκονται στον server!
- Ο κώδικας php παρεμβάλλεται στον κώδικα της σελίδας μας.
 - ▣ Οι εντολές της php βρίσκονται ανάμεσα σε `<?php` και `?>`
 - ▣ Οι εντολές της php τελειώνουν με semicolon (;) ΠΑΝΤΑ!!!
 - ▣ Ο κώδικας εκτελείται σειριακά, η μία εντολή μετά την άλλη
 - ▣ Space Insensitive

Παράδειγμα --- Hello World

```
<html>
  |
  |   <head>
  |   |   <title>Hello World!</title>
  |   |
  |   |   </head>
  |   |   <body>
  |   |   |   <?php    echo "Hello World!";  ?>
  |   |   |
  |   |   |   </body>
  |   |
  |   </html>
```

Hello World!

Διαδρομή Λειτουργιών



Echo - Print

- Η εντολή `echo` μας δίνει την δυνατότητα να εκτυπώσουμε ένα αλφαριθμητικό στην σελίδα μας. Αυτό μπορεί να είναι από ένα απλό `String` έως μια μεταβλητή.

echo "Echo το κινητό σου, θα си κανονίσω";



Echo το κινητό σου, θα си κανονίσω!



Echo – Print (2)

Παραδείγματα:

print "Hello World!";



Hello World!

echo "Hello" . " World!";



Hello World!

echo 5+7;



12

Σχολιασμός: Μανώλης Μαυρομάτης

- Σχόλια για μόνη γραμμή
// This is a single line comment
or this...

- Σχόλια για πολλές γραμμές
/ This is a multi-line comment*
And is the second line
**/*

- Τα σχόλια χρησιμοποιούνται για να βοηθήσουν αυτούς που διαβάζουν τον κώδικά μας (και εμάς τους ίδιους).



Ναι, κάτι άλλο?

1. Βασικές Οδηγίες
2. Μεταβλητές

Μεταβλητές

- Μεταβλητή είναι η συμβολική ονομασία μιας τιμής, η οποία μπορεί να αλλάξει κατά το πέρασμα του χρόνου.
- Μεταβλητές στην PHP
 - ▣ Ξεκινούν με \$.
 - ▣ Ακολουθεί γράμμα ή underscore (_).
 - ▣ Μπορεί να περιέχει γράμματα, αριθμούς, underscores (_) ή dashes (\, /).
 - ▣ Δεν μπορεί να περιέχει κενά.
 - ▣ Είναι case sensitive (*\$AntePAOKARA* <> *\$antepaokara*).

Nasty Names

□ *\$My-name*

□ *\$___MyBooks*

□ *\$_Ahahouha*

Μεταβλητές (2)

- Η τιμή αποδίδεται σε μια μεταβλητή με χρήση του τελεστή = (ίσον).
- Παράδειγμα: `$var1 = 10;`
 - ▣ Δίνω στην μεταβλητή `$var1` την τιμή 10.
- Στις μεταβλητές δεν ορίζουμε τύπους.
 - ▣ Ο τύπος μιας μεταβλητής εξαρτάται από την τιμή της.
 - ▣ Εφόσον οι μεταβλητές μπορούν να αλλάξουν τιμή με το πέρας του χρόνου, αυτό σημαίνει ότι μπορούν να αλλάξουν δυναμικά και τύπο.

Βασικοί Τύποι Μεταβλητών

1. *int*: 11 , 256, -32, 0

2. *float*: 3.14, 5.55, 0.22

3. *String*: “Κούλα είναι κωλόπαιδο ο Κυριάκος”,
“Σκίστεεεεεε!!!”

4. *boolean*: true, false



Παραδείγματος Χάρη (Κλυνν)

```
<?php  
$var1 = "Hello ";  
$var2 = "World!";  
echo $var1;  
echo $var2;  
?>
```



Hello World!

Παραδείγματος Χάριν (2)

```
<?php
```

```
$var1 = "Hello ";
```

```
echo $var1;
```



```
$var1 = 5;
```

```
echo $var1;
```

int

```
?>
```

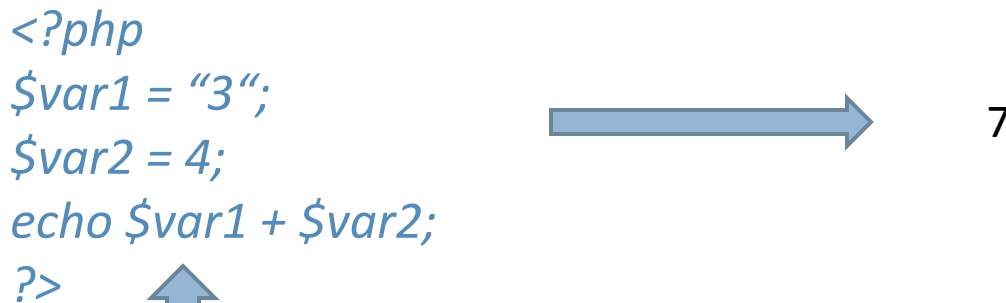


Hello 5

Δυναμικό Σύστημα Τύπων

- Μεταβλητές παίρνουν τύπο τιμής
- Διευκόλυνση στη συγγραφή κώδικα
- Σφάλματα **χρόνου εκτέλεσης** αντί συντακτικά
 - ▣ Πιο δύσκολα στον εντοπισμό
- Οι **μετατροπές** τύπων γίνονται **αυτόματα**

```
<?php
$var1 = "3";
$var2 = 4;
echo $var1 + $var2;
?>
```



Μετατροπή σε ακέραιο για να ολοκληρωθεί η πράξη

Αυτό μας τελείωσε, κάτι άλλο?

1. Βασικές Οδηγίες ✓
2. Μεταβλητές ✓
3. Αλφαριθμητικά - Αριθμοί

String Variables (Αλφαριθμητικά)

- Κάθε μεταβλητή είναι **ένα κείμενο**
- Το μήκος μπορεί να αλλάζει
- Δεν υπάρχει περιορισμός μήκους
- **Δεν** πρόκειται για πίνακες από χαρακτήρες
- **Δεν** υπάρχει διαφορετικός τύπος χαρακτήρα και αλφαριθμητικού
- Επιτρέπεται το UTF-8 άμεσα:
\$a = "Λεφτά υπάρχουν"; (Ναι στάνταρ!!!!)

Συνένωση Αλφαριθμητικών

- Αυτό γίνεται πάρα πολύ εύκολα με την τελεία (.)

```
<?php
```

```
echo "Κάποιος σ'αγαπάει," . " είμαι εγώ"
```

```
?>
```



Κάποιος σ'αγαπάει, είμαι εγώ

- Μπορεί να γίνει και με κείμενο και μεταβλητή

```
<?php
```

```
$var1 = " είμαι εγώ"
```

```
echo "Κάποιος σε ζητάει," . $var1
```

```
?>
```



Κάποιος σε ζητάει, είμαι εγώ

Συνένωση Αλφαριθμητικών (2)

- Είτε με δύο μεταβλητές

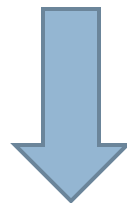
```
<?php
```

```
$var1 = “καποιος ξενυχτάει και σε περιμένει,”;
```

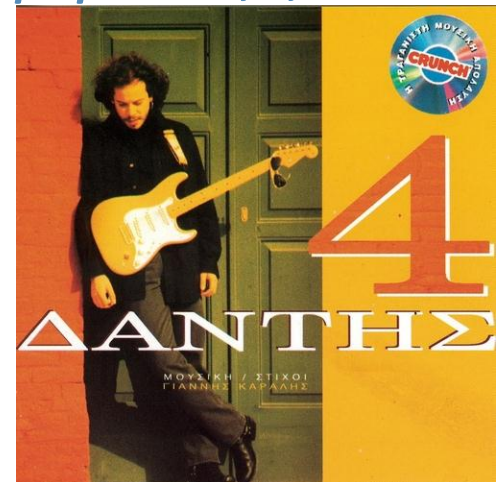
```
$var2 = “λεπτό προς λεπτό”;
```

```
echo $var1 . $var2;
```

```
?>
```



Κάποιος ξενυχτάει και σε περιμένει, λεπτό προς
λεπτό



Διπλά Αυτάκια Vs. Μονά Αυτάκια

- Διπλά εισαγωγικά: Προκαλούν αντικαταστάσεις

<?php

\$a = "Pikatsu";

echo "Καλώ εσένα, \$a!";  *Καλώ εσένα, Pikatsu*

// ακόμα καλύτερα "Καλώ εσένα, {\$a}!"

?>

- Μονά εισαγωγικά: **ΔΕΝ** Προκαλούν αντικαταστάσεις

<?php

\$a = "Pikatsu";

echo 'Καλώ εσένα, \$a!';  *Καλώ εσένα, \$a!*

?>

Χρήσιμες Συναρτήσεις Αλφαριθμητικών

1. [strlen\(\)](#): Υπολογίζει μήκος αλφαριθμητικού
2. [trim\(\)](#): «Κόβει» τα κενά από την αρχή και το τέλος
3. [strstr\(\)](#): Εντοπίζει ένα αλφαριθμητικό μέσα σε άλλο
4. [strreplace\(\)](#): Αντικαθιστά ένα αλφαριθμητικό με ένα άλλο
5. [strtolower\(\)](#): Μετατρέπει από κεφαλαία σε πεζά
6. [strtoupper\(\)](#): Μετατρέπει από πεζά σε κεφαλαία

Μπορείτε να βρείτε και πολλά άλλα από τα docs.

Μερικές Χρήσιμες Συναρτήσεις Δεκαδικών

1. [round\(\)](#): Στρογγυλοποιεί δεκαδικά
2. [ceil\(\)](#): Βρίσκει το ceiling ενός αριθμού
3. [floor\(\)](#): Βρίσκει το floor ενός αριθμού
4. [abs\(\)](#): Βρίσκει την απόλυτη τιμή
5. [pow\(\)](#): Υψώνει σε δύναμη
6. [sqrt\(\)](#): Βρίσκει την τετραγωνική ρίζα
7. [rand\(\)](#): Δημιουργεί τυχαίους αριθμούς
8. [fmod\(\)](#): Βρίσκει το υπόλοιπο διαίρεσης

Ειδικοί Χαρακτήρες

- Δουλεύουν μόνο σε “διπλά εισαγωγικά”
- Παρόμοιοι με C, C++, Java, ...
 - `\n` → Αλλαγή γραμμής
 - `\t` → Tab
 - `\mario` (Αστεριάκι...)



Να βάλω και μια ποικιλία για τη μέση?

1. Βασικές Οδηγίες ✓
2. Μεταβλητές ✓
3. Αλφαριθμητικά - Αριθμοί ✓
4. Τελεστές

Τελεστές

Τελεστές	Λειτουργία
+, -, *, /, %	Αριθμητικές πράξεις
, &&, !	Λογικές πράξεις: ή, και, όχι
==, !=, <, >, <=, >=, ===, !==	Σύγκριση
++, --	Αύξηση, μείωση
.	Ένωση αλφαριθμητικών
=, +=, -=, *=, /=, %=, .=	Ανάθεση τιμής

Τελεστές (2)

Κώδικας	Αποτέλεσμα
$3 + 5$	8
$2 - 9$	-7
$1 / 2$	0.5
$5 * 7$	35
$102 \% 5$	2
<code>true false</code>	true
<code>!true</code>	false
<code>true && true</code>	true
$3 < 5$	true
$3 != 3$	false
<code>"Hello " . ' , world!'</code>	Hello, world!


Παραδείγματα Τελεστών

echo "1" / "2";  0.5

echo 1 . 2;  12

\$a = 5; echo "\$a";  5

\$b = 5; echo "\$b" - \$b;  0

\$c = 5; echo '\$c' . \$c;  \$c5

\$a = 3; echo ++\$a;  4

\$a = 3; echo \$a--;  3

\$b = "Hello"; \$b .= "there!!"; echo \$b;

Hello there!!

Τελεστές Ισότητας

- Χρησιμοποιούνται τα `==`, `!=`, `===`, `!==`
- Οι τύποι των μεταβλητών μετατρέπονται για να γίνει σωστά η σύγκριση στην περίπτωση των δύο πρώτων.
 - `1 == 1` True
 - `1 == 2` False
 - `0 != "hello"` True
 - `"1" == 1` True
 - `"1" === 1` False

Να φέρω μια ή δύο να φτάσουν για όλους?

1. Βασικές Οδηγίες ✓
2. Μεταβλητές ✓
3. Αλφαριθμητικά - Αριθμοί ✓
4. Τελεστές ✓
5. Συγκρίσεις

Εντολή if

```
if ( συνθήκη 1) {  
    // εντολές  
}
```

Αν ισχύει η συνθήκη 1
εκτελείται αυτό το σώμα

```
else if (συνθήκη 2) {  
    // εντολές  
}
```

Αν ισχύει η συνθήκη 2
εκτελείται αυτό το σώμα

...

...

```
else {  
    // εντολές  
}
```

Αν δεν ισχύει καμία συνθήκη
εκτελείται το τελευταίο σώμα

Παραδείγματα

```
<?php
if ( $a == 5) {
    echo "Five of a kind!";
}
else {
    echo "Can say if stupid or just trolling!";
}
?>
```

Παραδείγματα (2)

```
<?php
$a = "Justin Bieber"
if ( $a == "Justin Bieber") {
    echo "That's my girl!";
}
else {
    echo "Bitch please!";
}
?>
```

Εντολή switch

switch (παράσταση) {

case τιμή1:

σώμα 1;

break;

Αν ισχύει η συνθήκη 1

εκτελείται το σώμα1



case τιμή2:

σώμα 2;

break;

Αν ισχύει η συνθήκη 2

εκτελείται το σώμα2



...

default:

default σώμα

Αν δεν ισχύει καμία συνθήκη

εκτελείται το τελευταίο σώμα



}

Παράδειγμα

```
<?php
switch ( $day ){
    case “Δευτέρα”:
        echo “Κάτι έχω”;
        break;
    case “Τρίτη”:
        echo “Δεν αντέχω”;
        break;
    case “Τετάρτη”:
        echo “Πως βαριέμαι”;
        break;
    ...
}
```

Παράδειγμα (2)

```
<?php
$day = "Δευτέρα";
switch ( $day){
    case "Δευτέρα":
        echo "Ξέχασες το Break!";
    case "Τρίτη":
        echo "Σε σένα μιλάω!";
    case "Τετάρτη":
        echo "Είσαι βλακάκος?";
    default:
        echo "Ό,τι να 'ναι είσαι...";
}
```



Ξέχασες το Break! Σε σένα μιλάω! Είσαι βλακάκος? Ό,τι να 'ναι είσαι...

ΠΡΟΣΟΧΗ!!!

Αν δεν βάλουμε το break μετά από κάθε σώμα θα συνεχιστεί η εκτέλεση των εντολών μέχρι το επόμενο break.

Από βρόχους τι θα πάρετε?

1. Βασικές Οδηγίες ✓
2. Μεταβλητές ✓
3. Αλφαριθμητικά - Αριθμοί ✓
4. Τελεστές ✓
5. Συγκρίσεις ✓
6. Βρόχοι επανάληψης

Εντολή for

```
for ( αρχική συνθήκη; συνθήκη τερματισμού; Βήμα)  
{  
    // σώμα εντολών  
}
```

1. Αρχικοποιούμε μια μεταβλητή.
2. Εκτελείται το σώμα εντολών του βρόχου.
3. Αλλάζουμε την τιμή της μεταβλητής σύμφωνα με το βήμα που έχουμε.
4. Ελέγχουμε αν ισχύει η συνθήκη τερματισμού
 - ▣ Αν ισχύει σταματάμε τις επαναλήψεις.
 - ▣ Αν όχι τότε συνεχίζουμε τις επαναλήψεις.

Παράδειγμα

```
<?php  
for ($i = 1; $i < 6; $i++){  
    echo "High-$i!";  
}
```



High-1!High-2!High-3!High-4!High-5!



Εντολή while

```
while (συνθήκη) {  
    // σώμα εντολών  
}
```

1. Αρχικά ελέγχεται η συνθήκη.
 - ▣ Αν ισχύει τότε εκτελείται το σώμα εντολών του βρόχου.
 - ▣ Σε αντίθετη περίπτωση σταματάει η εκτέλεση του βρόχου
 2. Ελέγχεται εκ νέου η συνθήκη τερματισμού κτλ
- ❖ Υπάρχει περίπτωση το σώμα του βρόχου να μην εκτελεστεί ούτε μια φορά.

Παράδειγμα

```
<?php  
$i = 1;  
while ($i < 6) {  
    echo "High-$i!";  
    $i++;  
}
```



High-1!High-2!High-3!High-4!High-5!

Εντολή do...while

```
do {  
    // σώμα εντολών  
} while (συνθήκη)
```

1. Ακριβώς ίδια με την while.
2. Το σώμα του βρόχου θα εκτελεστεί οπωσδήποτε μια φορά, είτε είναι λανθασμένη η συνθήκη είτε όχι, γιατί ο έλεγχος γίνεται κατόπιν (εορτής).

Παράδειγμα

```
<?php  
$i = 5;  
do {  
    echo "Hi- $\$i$ !";  
} while ( $\$i > 6$ )
```



Hi-5!



Εντολές `break` - `continue`

□ *break*

- Εμφανίζεται μέσα σε βρόχους επανάληψης (`for`, `while` etc) ή σώμα συγκρίσεων (`switch`).
- Διακόπτει την ροή που βρισκόμαστε και συνεχίζει με την επόμενη αμέσως εντολή.
- Δεν γίνονται άλλες επαναλήψεις.

□ *continue*

- Εμφανίζεται μέσα σε ροή επανάληψης (`do...while` etc).
- Διακόπτει την ροή των εντολών πηγαίνοντας στον έλεγχο της συνθήκης τερματισμού.
- Μπορεί να γίνουν και άλλες επαναλήψεις μετά από το `continue`.

Αυτά, κάτι άλλο?

1. Βασικές Οδηγίες ✓
2. Μεταβλητές ✓
3. Αλφριθμητικά - Αριθμοί ✓
4. Τελεστές ✓
5. Συγκρίσεις ✓
6. Βρόχοι επανάληψης ✓
7. Συναρτήσεις

Συναρτήσεις

- Οι συναρτήσεις είναι υπό-ρουτίνες που φέρουν σε πέρας μια συγκεκριμένη εργασία.
- Ορίζονται με την λέξη-κλειδί `function` στην αρχή της γραμμής.
- Στην συνέχεια ακολουθεί το όνομα της συνάρτησης και μέσα σε παρένθεση τα ορίσματα που παίρνει αυτή, χωρισμένα με κόμμα (,).

```
function όνομα-συνάρτησης (ορίσματα) {  
    // σώμα εντολών  
}
```

Ορίσματα Συναρτήσεων

- **Δίνουν** πληροφορίες σε μία συνάρτηση
- Ακολουθούν ίδια ονοματολογία με μεταβλητές
- Αρχίζουν με **\$**, ακολουθεί το **όνομα**
- Το όνομα...
 - ▣ Αρχίζει με γράμμα ή `_`
 - ▣ Περιέχει γράμματα, αριθμούς, `_`
 - ▣ Έχει ευαισθησία σε πεζά-κεφαλαία

Εντολή επιστροφής (return)

- Η εντολή *return* επιστρέφει πίσω στο κυρίως πρόγραμμα μια τιμή που έχει προκύψει κατά την ροή της συνάρτησης.
- Η τιμή αυτή λαμβάνεται στο σημείο που έγινε η κλήση της συνάρτησης.
- Η *return* πρέπει να είναι η τελευταία εντολή στο σώμα εντολών γιατί μετά σταματάει η εκτέλεση της συνάρτησης.
- Δεν ορίζεται τύπος επιστροφής, όπως στις άλλες γλώσσες προγραμματισμού.
- Δεν είναι υποχρεωτική.

Κλήση Συνάρτησης

- Για να καλέσουμε μια συνάρτηση χρησιμοποιούμε το όνομα της.
- Στην συνέχεια, ορίζουμε μέσα σε παρένθεση τις τιμές που θα πάρουν τα ορίσματα της.
- Έχει σημασία η σειρά των ορισμάτων
 - ▣ Πρώτο όρισμα → πρώτη τιμή
 - ▣ Δεύτερο όρισμα → δεύτερη τιμή...

Κλήση χωρίς επιστροφή

όνομα-συνάρτησης (ορίσματα)

Κλήση με επιστροφή

ξα = όνομα-συνάρτησης (ορίσματα)

Ορισμός Συναρτήσεων

```
function multiply($a, $b){  
    $c = $a * $b;  
    return $c;  
}
```

Όνομα συνάρτησης

ορισμός συνάρτησης

```
echo "Multiplying 1 and 2 gives" . multiply(1, 2);
```

Κλήση συνάρτησης

Ορισμός Συναρτήσεων (2)

```
function multiply($a, $b) {  
    $c = $a * $b;  
    return $c;  
}
```

Ορίσματα

Πρώτο όρισμα

Δεύτερο όρισμα

```
echo "Multiplying 1 and 2 gives" . multiply(1, 2);
```

τιμή πρώτου ορίσματος

Τιμή δεύτερου ορίσματος

Ορισμός Συναρτήσεων (3)

```
function multiply( $a , $b ){  
    $c = $a * $b;  
    return $c;  
}
```

Επιστροφή

```
echo "Multiplying 1 and 2 gives" . multiply( 1 , 2 );
```

μπαίνει στην θέση του



Multiplying 1 and 2 gives 2

Παράδειγμα

```
function average( $a, $b ) {  
    $c = $a + $b;  
    return $c / 2;  
}
```

```
echo 'The average of 3, 5: ' . average( 3, 5 );  
echo "\n";  
echo 'The average of 1, 9: ' . average( 1, 9 );
```

Παραδείγμα (2)

```
function choose($a){  
    if ($a == "1"){  
        return "Pikachu";  
    }  
    else {  
        return "Καμπαμαρού";  
    }  
}
```

```
echo choose(3);
```



Προαιρετικά Ορίσματα

- Μπορούμε αν θέλουμε κάποια από τα ορίσματα μιας συνάρτησης να τα θέσουμε κατά τον ορισμό της.
 - Τα προαιρετικά ορίσματα πρέπει να είναι τα τελευταία σε σειρά.
 - Μπορούν να είναι όσα θέλουμε σε αριθμό.
 - Πρέπει να ορίσουμε με ίσον (=) την τιμή που θα έχουν ως προεπιλεγμένη.

Παράδειγμα

```
function play( $title, $band = "Mazoo and the Zoo") {  
    $str = "I am playing "  
    $str .= $title . " by " . $band;  
    return $str;  
}
```

Προαιρετικό όρισμα

```
echo play("Η Αρκούδα");
```

Αν είχαμε όμως

```
function play( $title = "Η Αρκούδα" , $band )
```

στον ορισμό θα ήταν λάθος.

Παράδειγμα

function play

```
($title = "Careless Whisper", $band = "George Michael") {
```

```
$str = "I am playing ";
```

```
$str .= $title . " by " . $band;
```

```
return $str;
```

```
}
```

Όλα είναι προαιρετικά ορίσματα

```
echo play();
```

Θα πιείτε κάτι?

1. Βασικές Οδηγίες ✓
2. Μεταβλητές ✓
3. Αλφριθμητικά - Αριθμοί ✓
4. Τελεστές ✓
5. Συγκρίσεις ✓
6. Βρόχοι επανάληψης ✓
7. Συναρτήσεις ✓
8. Χειρισμός Φορμών

Χειρισμός Δεδομένων Φόρμας

- Για να πάρουμε δεδομένα χρησιμοποιούμε δύο μεθόδους:
 - HTTP GET
 - Μεταβλητή `$_GET`
 - `$_GET[όνομα παραμέτρου]`
 - HTTP POST
 - Μεταβλητή `$_POST`
 - `$_POST[όνομα παραμέτρου]`
- ❖ Οι μεταβλητές αυτές ορίζονται αυτόματα από την PHP

Παράδειγμα

test.html

```
<form action="test.php" method="post">  
  <input type="text" name="foo" />  
  <input type="submit" value="Πάρε" />  
</form>
```

Παράδειγμα

test.php

```
<p>
```

```
    Δώσε
```

```
    <?php
```

```
        echo $_POST[ 'foo' ];
```

```
    ?>!
```

```
</p>
```

ΣΥΓΧΑΡΗΤΗΡΙΑ

- Μάθατε τις πρώτες σας εντολές σε ΡΗΡ!!!



THE END!!!

- Ευχαριστούμε πάρα πολύ για την προσοχή σας

