

ΡΗΡ 1

Διδάσκοντες: Π. Αγγελάτος, Δ. Ζήνδρος
Επιμέλεια διαφανειών: Δ. Ζήνδρος

Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών



Στόχος της ώρας

- Εισαγωγή της γλώσσας PHP σε αρχάριο επίπεδο:
 - Βασική σύνταξη
 - Συνδυασμός PHP με άλλες γλώσσες
 - Μεταβλητές
 - Αλφαριθμητικά
 - Τελεστές
 - if, else, switch
 - for, while
 - Συναρτήσεις
 - Χειρισμός φορμών



PHP

- **PHP Hypertext Preprocessor**
- Τρέχει στον **server** και **όχι** στον client
- Συνδυάζεται με κάποιον **web server** όπως ο Apache
- Ανοιχτού κώδικα, τρέχει σε Windows, Linux, και αλλού
- **Παράγει** το HTML που θα φτάσει στο χρήστη
- Θα μελετήσουμε την έκδοση 5.2

Εναλλακτικές λύσεις

- ASP
 - Η server-side λύση της Microsoft
- JSP
 - Server-side λύση βασισμένη στη Java
- Python, Django
- Ruby, Ruby on Rails

Τι μπορεί να κάνει;

- Δυναμικές σελίδες
- Login, διαχείριση χρηστών
- Μόνιμη αποθήκευση δεδομένων (στο server)
- Διαχείριση προσωπικού περιεχομένου
 - Προσωπικά μηνύματα
 - Φωτογραφίες που μπορούν να δουν μόνο «οι φίλοι μου»
- Ανέβασμα αρχείων
- Σύνδεση με βάση δεδομένων
- Πολλά, πολλά άλλα



Δώσε μου τη σελίδα X



Φτιάξε μου τη σελίδα X



Εκτέλεση κώδικα PHP
Παραγωγή HTML

Ορίστε η σελίδα X



Ορίστε η σελίδα X



Εμφάνιση σελίδας

LAMP

- Linux
- Apache
- MySQL
- PHP



Συγγραφή PHP

- Την γράφουμε σε αρχεία .php
- Τα αρχεία .php αποθηκεύονται στον server
- Ο χρήστης **δεν** έχει άμεση πρόσβαση στα αρχεία αυτά!
- Γράφουμε σκέτη HTML
- Σε ορισμένα σημεία **παρεμβάλλεται** εκτελέσιμος κώδικας
- Για να τρέξει θα **περάσει** από το πρόγραμμα httpd
- Δεν μιλάμε πλέον για τοπικά αρχεία

Γεια σου κόσμε!

```
<html>  
  <head><title>:)</title></head>  
  <body>  
    <?php  
      echo "<p>Hello, world!</p>";  
    ?>  
  </body>  
</html>
```



```
<html>  
  <head><title>:)</title></head>  
  <body><p>Hello, world!</p></body>  
</html>
```

Βασική σύνταξη

- Παρεμβολή κώδικα
- Αρχίζει με `<?php`
 - Ή συντομογραφία `<?`
- Τελειώνει με `?>`
- Μπορούμε να παρεμβάλλουμε πολλές φορές κώδικα
- Ο κώδικας εκτελείται σειριακά
 - Η μία εντολή μετά την άλλη
- Κάθε εντολή τελειώνει σε `;`

Γεια σου κόσμε!

```
<html>
  <head><title>
    <?php
      echo “:-)” ;
    ?>
  </title></head>
  <body>
    <?php
      echo “<p>Hello, world!</p>” ;
    ?>
  </body>
</html>
```



```
<html>  
  <head><title>:)</title></head>  
  <body><p>Hello, world!</p></body>  
</html>
```

Τυπώνοντας κείμενο

- **echo**: Τυπώνει την παράμετρο στο αποτέλεσμα

Interpreter

- Δεν γίνεται compile
- Το πρόγραμμα τρέχει όπως διαβάζεται

Μεταβλητές στην PHP

- Αποθηκεύουν μία τιμή
 - Νούμερα, κείμενα, πίνακες, ...
- Αρχίζουν με \$, ακολουθεί το **όνομα**
- Το όνομα...
 - Αρχίζει με γράμμα ή _
 - Περιέχει γράμματα, αριθμούς, _
 - Έχει ευαισθησία σε πεζά-κεφαλαία
- Παρόμοιες με C, C++, Java, Pascal, ...

Μεταβλητές

- Τιμή μεταβλητής ορίζεται με τον τελεστή =
- `$a = 5;`
- Δίνει στην μεταβλητή `$a` την τιμή `5`
- Οι μεταβλητές μπορούν να **αλλάξουν** τιμή
- Μπορούν να χρησιμοποιηθούν μέσα σε παραστάσεις

Μεταβλητές

```
<?php
```

```
    $a = "Hello, ";
```

```
    $b = "world!";
```

```
    echo $a;
```

```
    echo $b;
```

```
?>
```



Hello, world!

Ασθενές σύστημα τύπων

- Κάθε τιμή έχει έναν τύπο
 - int: 5, 9, 34, -18
 - float: 0.99, 3.14
 - string: “Hello world”, ‘Διονύσης’
 - boolean: true, false
- **Δεν** ορίζουμε **τύπους** μεταβλητών
- Ο τύπος μίας μεταβλητής μπορεί να αλλάξει
 - Ανάλογα με την τιμή που παίρνει

```
<?php
```

```
    $a = "Hello, ";
```

```
    echo $a; ← Τύπου string
```

```
    $a = 5;
```

```
    echo $a; ← Τύπου 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
```

```
    $a = "5";
```

```
    $b = 7;
```

```
    echo $b - $a;
```

```
?>
```

→ 2

\$a;



Μετατρέπεται σε int για να γίνει η πράξη

Αλφαριθμητικά

- Μία τιμή που είναι μία σειρά από αριθμούς, χαρακτήρες, γράμματα, ...
- Δηλαδή **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 = “Χαίρε, κόσμε!”;

Αλφαριθμητικά

- **Διπλά εισαγωγικά:** Προκαλούν αντικαταστάσεις

```
<?php
```

```
    $a = "world";
```

```
    echo "Hello, $a!";
```



Hello, world!

```
?>
```

- **Μονά εισαγωγικά:** Δεν προκαλούν αντικαταστάσεις

```
<?php
```

```
    $a = "world";
```

```
    echo 'Hello, $a!';
```



Hello, \$a!

```
?>
```

- Εξίσου γρήγορα και τα δύο

Συνένωση αλφαριθμητικών

- Με τον τελεστή `.` (τελεία)
- Παράγει ένα αλφαριθμητικό που είναι η παράθεση δύο άλλων
- `“Hello, “ . “world” → Hello, world`

Συναρτήσεις αλφαριθμητικών

- Παρέχονται έτοιμες από την PHP:
- **strlen**: Υπολογίζει μήκος αλφαριθμητικού
- **trim**: «Κόβει» τα κενά από την αρχή και το τέλος
- **strpos**: Εντοπίζει ένα αλφαριθμητικό μέσα σε άλλο
- **strtolower**: Μετατρέπει από κεφαλαία σε πεζά
- **strtoupper**: Μετατρέπει από πεζά σε κεφαλαία
- **str_repeat**: Δημιουργεί αλφαριθμητικό με επανάληψη άλλου αλφαριθμητικού

...και πολλές άλλες: <http://php.net/strings>

Ψάξτε τες όταν τις χρειαστείτε!

Ειδικοί χαρακτήρες

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

Τελεστές

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

Τελεστές

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

Τελεστές

```
$a = 5;
```

```
echo $a;            →    5
```

```
++$a;
```

```
echo $a;            →    6
```

```
--$a;
```

```
echo $a;            →    5
```

Τι θα τυπώσουν;

- `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

Σύγκριση

- Με `==`, `!=`
- Ο τύπος των τιμών μετατρέπεται ώστε να γίνει η σύγκριση
- `1 == 1` → true
- `1 == 2` → false
- `0 == "hello"` → true
- `"1" == 1` → true

Σύγκριση

- Η σύγκριση αλφαριθμητικών γίνεται εύκολα και **σωστά**:

```
$a = "hello";  
if ( $a == "hello" ) {  
    echo '$a is hello. ' ;  
}  
else {  
    echo '$a is not hello. ' ;  
}
```

if

```
if ( συνθήκη ) {  
    σώμα 1  
}  
else if ( συνθήκη ) {  
    σώμα 2  
}  
...  
else {  
    σώμα 3  
}
```

if

- Εκτελεί κώδικα **υπό συνθήκη**
- Παρόμοιο με το if της C, C++, Java, ...
- 1^η συνθήκη **αληθής**;
 - Εκτέλεση κώδικα **σώματος if**
- Αλλιώς, 2^η συνθήκη **αληθής**;
 - Εκτέλεση κώδικα **σώματος else if**
- ...
- Όλες οι συνθήκες **ψευδείς**;
 - Εκτέλεση κώδικα **σώματος else**
- Τα **else if** και **else** είναι προαιρετικά

if

```
if ( $a == 5 ) {  
    echo "Hello!";  
}  
else {  
    echo "Goodbye!";  
}
```

switch

```
switch ( παράσταση ) {  
    case τιμή1:  
        σώμα 1;  
        break;  
    case τιμή2:  
        σώμα 1;  
        break;  
    ...  
    default:  
        εναλλακτικό σώμα  
}
```

switch

- **Επιλέγει** ένα σώμα με βάση την τιμή μίας παράστασης
- Παρόμοιο με C, C++, Java, ...
 - Τα cases **μπορούν** να είναι και **όχι σταθερές**
- Αν η παράσταση έχει τιμή1
 - Εκτέλεση σώματος 1
- Αν η παράσταση έχει τιμή2
 - Εκτέλεση σώματος2
- ...
- Αλλιώς
 - Εκτέλεση εναλλακτικού σώματος
- Το **default** είναι προαιρετικό

switch

```
switch ( $a ) {  
    case 5:  
        echo "a is 5";  
        break;  
    case 10:  
        echo "a is 10";  
        break;  
    case 15:  
        echo "a is 15";  
        break;  
    default:  
        echo "a is neither 5, nor 10, nor 15";  
}
```

switch

- Παράληψη του break οδηγεί σε **fall-through**
- Εκτελούνται τα σώματα που ακολουθούν μέχρι το επόμενο break

switch

```
$a = 10;  
switch ( $a ) {  
    case 5:  
        echo "a is 5. ";  
    case 10:  
        echo "a is 10. ";  
    case 15:  
        echo "a is 15. ";  
    default:  
        echo "a is neither 5, nor 10, nor 15";  
}
```



a is 10. a is 15. a is neither 5, nor 10, nor 15.

for

```
for ( αρχικοποίηση; συνθήκη; βήμα ) {  
    σώμα  
}
```

```
for ( αρχικοποίηση; συνθήκη; βήμα ) {  
    σώμα  
}
```

for

- Ίδιο σε C, C++, Java...
- Επαναλαμβάνει ένα σώμα σύμφωνα με κάποια συνθήκη
- Αρχικά τρέχει η **αρχικοποίηση**
- Αν η **συνθήκη** είναι **ψευδής**, τελειώσαμε
- Αν η **συνθήκη** είναι **αληθής**, τρέχει το **σώμα**
- Μετά το σώμα τρέχει το **βήμα**
- **Η συνθήκη ελέγχεται ξανά, κ.ό.κ.**

for

```
for ( $i = 0; $i < 5; ++$i ) {  
    echo "Hello, world! ";  
}
```



Hello, world! Hello, world! Hello, world!
Hello, world! Hello, world!

while

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

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

while

- Ίδιο σε C, C++, Java, ...
- Επαναλαμβάνει ένα σώμα σύμφωνα με κάποια συνθήκη
- Αρχικά ελέγχεται η συνθήκη
- Αν η **συνθήκη** είναι **ψευδής**, τελειώσαμε
- Αν η **συνθήκη** είναι **αληθής**, τρέχει το **σώμα**
- Η **συνθήκη** ελέγχεται ξανά, κ.ό.κ.

while

```
$i = 0;  
while ( $i < 5 ) {  
    echo "Hello, world! ";  
    ++$i;  
}
```



Hello, world! Hello, world! Hello, world!
Hello, world! Hello, world!

do... while

```
do {  
    σώμα  
} while ( συνθήκη );
```

```
do {  
    σώμα  
} while ( συνθήκη );
```

do... while

- Ίδιο σε C, C++, Java, ...
- Επαναλαμβάνει ένα σώμα σύμφωνα με κάποια συνθήκη
- Αρχικά τρέχει μία φορά το σώμα
- Στη συνέχεια ελέγχεται η συνθήκη
- Αν η συνθήκη είναι **ψευδής**, τελειώσαμε
- Αν η συνθήκη είναι **αληθής**, τρέχει το **σώμα**
- Η συνθήκη ελέγχεται ξανά, κ.ό.κ.

do... while

```
$i = 0;  
do {  
    echo "Hello, world! ";  
    ++$i;  
} while ( $i < 0 );
```



Hello, world!

break

- Ίδιο σε C, C++, Java, ...
- Εμφανίζεται μέσα σε μία ροή ελέγχου
 - **for, foreach, while, do... while, switch**
- Διακόπτει την ροή και συνεχίζει αμέσως μετά
- Δεν γίνονται άλλες επαναλήψεις μετά το **break**

continue

- Ίδιο σε C, C++, Java, ...
- Εμφανίζεται μέσα σε μία ροή επανάληψης
 - **for, foreach, while, do... while**
- Διακόπτει την ροή και συνεχίζει ελέγχοντας την συνθήκη
- Μπορεί να γίνουν και άλλες επαναλήψεις μετά το **continue**

Χειρισμός φορμών

- Για να πάρουμε δεδομένα από HTTP GET:
 - Μεταβλητή `$_GET`
 - `$_GET[“όνομα_παραμέτρου”]`
- Για να πάρουμε δεδομένα από HTTP POST:
 - Μεταβλητή `$_POST`
 - `$_POST[“όνομα_παραμέτρου”]`
- Ορίζονται αυτόματα από την PHP

Χειρισμός φορμών

input.html:

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

Χειρισμός φορμών

test.php:

<p>

Πληκτρολόγησες

<?php

echo \$_POST['foo'];

?>!

</p>

Σχόλια

- `//` η υπόλοιπη γραμμή είναι σχόλιο
- Το πολύ 1 γραμμή

```
$a = 5; // assign $a to be 5
```

- `/*` τα περιεχόμενα είναι σχόλιο `*/`
- 1 ή περισσότερες γραμμές

Συναρτήσεις

```
function όνομα_συνάρτησης( ορίσματα ) {  
    σώμα;  
}
```

Συναρτήσεις

- Παρόμοιες με συναρτήσεις σε C, C++, Java, ...
- Ορίζουν υπο-ρουτίνες που κάνουν συγκεκριμένη δουλειά
- Ορίζονται με την λέξη-κλειδί function
- Ακολουθεί το **όνομα** της συνάρτησης
- Ακολουθούν τα ονόματα των **ορισμάτων** σε () χωρισμένα με κόμματα

Επιστροφή τιμής

- Οι συναρτήσεις **επιστρέφουν** τιμή με **return**
- Η τιμή επιστροφής χρησιμοποιείται όπου έγινε η κλήση
- Επιστροφή σημαίνει **τερματισμός** συνάρτησης
- Δεν ορίζουμε **τύπο** επιστροφής
- Δεν είναι υποχρεωτικό

Κλήση συναρτήσεων

- Καλούνται οπουδήποτε χρησιμοποιώντας το όνομα
- Ακολουθούν οι **τιμές** των ορισμάτων σε () χωρισμένες με κόμματα
- Σειρά ορισμάτων έχει σημασία
 - Πρώτη τιμή → Πρώτο όρισμα
 - Δεύτερη τιμή → Δεύτερο όρισμα
 - κ.ό.κ.
- Κλήση χωρίς επιστροφή:
`όνομα_συνάρτησης(τιμές_ορισμάτων);`
- Κλήση με επιστροφή:
`$a = όνομα_συνάρτησης(τιμές_ορισμάτων);`

Ορίσματα

- **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

Συναρτήσεις

```
function add( $a, $b ) {  
    $c = $a + $b;  
    return $c;  
}
```

Όνομα συνάρτησης

Ορισμός συνάρτησης

```
echo "The sum of 3 and 5: " . add( 3, 5 );
```

Κλήση συνάρτησης

Συναρτήσεις Πρώτο όρισμα

```
function add( $a, $b ) {  
    $c = $a + $b; Ορίσματα  
    return $c;  
}
```

Δεύτερο όρισμα

```
echo "The sum of 3 and 5: " . add( 3, 5 );
```

Τιμές ορισμάτων

Τιμή πρώτου ορίσματος

Τιμή δεύτερου ορίσματος

Συναρτήσεις

```
function add( $a, $b ) {  
    $c = $a + $b;  
    return $c;  
}
```

Τιμή επιστροφής



```
echo "The sum of 3 and 5: " . add( 3, 5 );
```

Παίρνει αυτή τη θέση



Συναρτήσεις

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

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

Συναρτήσεις

```
function is_prime( $a ) {  
    for ( $i = 2; $i < $a; ++$i ) {  
        if ( $a % $i == 0 ) {  
            return false;  
        }  
    }  
    return true;  
}
```

```
if ( is_prime( 5 ) ) {  
    echo "5 is a prime number."  
}
```


Προαιρετικά ορίσματα

- Μπορούν να είναι:
- Τα **τελευταία** μίας συνάρτησης
- Όσα θέλουμε
- Ορίζουμε μία **προεπιλεγμένη** τιμή με το = μετά το όνομα του ορίσματος

Προαιρετικά ορίσματα

Όλα τα ορίσματα προαιρετικά

```
function makeCoffee  
  ( $type = "frappe", $milk = true ) {  
  
    $str = "Making a cup of $type";  
    if ( $milk ) {  
        $str .= ' with milk';  
    }  
    $str .= ".\n";  
    return $str;  
}  
  
echo makeCoffee( "espresso" );
```



Προαιρετικά ορίσματα

2° όρισμα προαιρετικό

```
function makeCoffee
  ( $type, $milk = true ) {

  $str = "Making a cup of $type";
  if ( $milk ) {
    $str .= ' with milk';
  }
  $str .= ".\n";
  return $str;
}

echo makeCoffee( "espresso" );
```

Προαιρετικά ορίσματα

```
function makeCoffee
    ( $type = 'frappe', $milk ) {

    $str = "Making a cup of $type";
    if ( $milk ) {
        $str .= ' with milk';
    }
    $str .= ".\n";
    return $str;
}

echo makeCoffee( "espresso" );
```

Εγκαταστήστε Apache + PHP!

- Ήρθε η ώρα να στήσετε έναν server στον υπολογιστή σας
- Εγκαταστήστε τον Apache από το <http://apache.org/>
- Εγκαταστήστε την PHP από το <http://php.net/>
- **Πειραματιστείτε!** 😊
- Θα τα χρειαστείτε για την 4^η εργασία!

Μάθαμε

- Εισαγωγή της γλώσσας PHP σε αρχάριο επίπεδο:
 - Βασική σύνταξη
 - Συνδυασμός PHP με άλλες γλώσσες
 - Μεταβλητές
 - Αλφαριθμητικά
 - Τελεστές
 - if, else, switch
 - for, while
 - Συναρτήσεις
 - Χειρισμός φορμών

Συγχαρητήρια!

- Μάθατε PHP.
- Μπορείτε να **κάνετε δυναμική** την πρώτη σας σελίδα!



Την επόμενη φορά...

- Εμβάθυνση στην PHP
- Πίνακες και λεξικά
- foreach
- Διαχείριση αρχείων
- Ημερομηνίες
- Διαχωρισμός κώδικα σε πολλά αρχεία
- Ανέβασμα αρχείων από τον χρήστη
- Μπισκότα και σύνοδοι
- Αποστολή e-mail