



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

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

↑
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”` $\rightarrow$ `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
<code>+, -, *, /, %</code>	Αριθμητικές πράξεις
<code> , &&, !</code>	Λογικές πράξεις: ή, και, όχι
<code>==, !=, <, >, <=, >=</code>	Σύγκριση
<code>++, --</code>	Αύξηση, μείωση
<code>.</code>	Ένωση αλφαριθμητικών
<code>=, +=, -=, *=, /=, %=, .=</code>	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