

# Προγραμματισμός και Συστήματα στον Παγκόσμιο Ιστό

## Εισαγωγή στην PHP

---

Δρ. Δημήτριος Κουτσομητρόπουλος

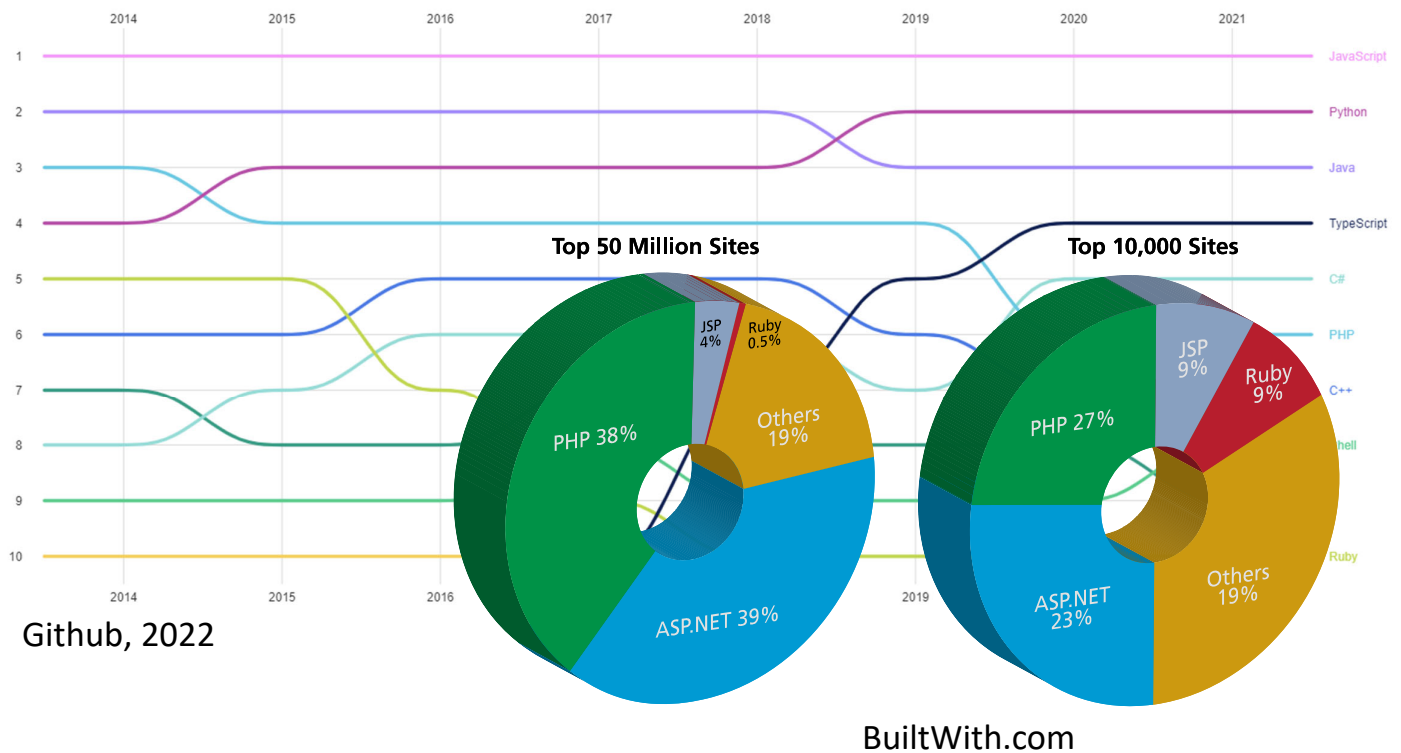


---

## Server-side προγραμματισμός

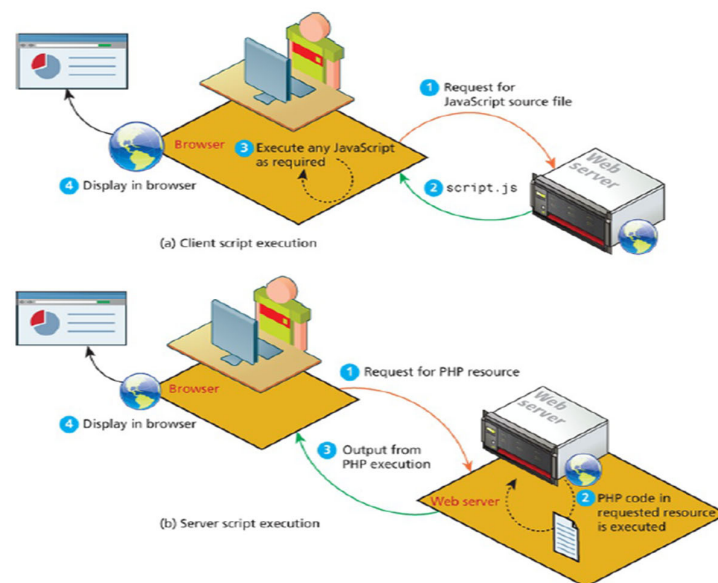
- ASP (Active Server Pages) / ASP.NET (C#)
- JSP (Java Server Pages)
- Node.js
- PHP
- Python
- Ruby on Rails
- PHP & Frameworks (π.χ. Laravel, CakePHP, Yii...)
- PHP CMS Frameworks (Wordpress, Drupal, Joomla...)

# Η PHP σήμερα



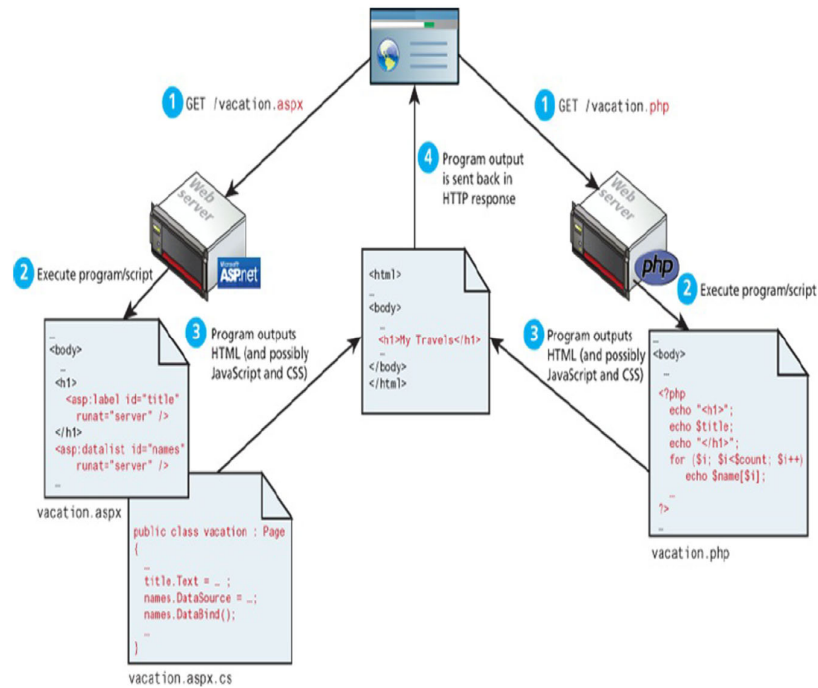
3

## Server-Side vs Client-Side



4

# Server-side προγραμματισμός



## Εισαγωγή στην PHP

**PHP : Hypertext Preprocessor**

Η PHP είναι μία scripting γλώσσα για γενική χρήση

Αντικειμενοστραφής και διαδικαστική

Τρέχει στον server και στον Web Browser  
επιστρέφει script σε μορφή HTML.

Open Source λογισμικό

Εγχειρίδιο χρήσης <http://www.php.net/docs.php>

# Εισαγωγή στην PHP

---

Ιδανική για την κατασκευή δυναμικών ιστοσελίδων  
Υποστήριξη πολλών ΣΔΒΔ όπως MySQL, Oracle, MS SQL Server, IBM DB2 κτλ.

Μεταφερισιμότητα : διαθέσιμη σε διάφορα λειτουργικά συστήματα όπως Windows, Linux και Unix

Συμβατή με τους περισσότερους servers Apache, IIS

Τα PHP αρχεία μπορούν να έχουν κατάληξη “.php”, “.php3” ή “.phtml”

## Τι χρειαζόμαστε ;

---

Εγκαθιστώ τον Apache server σε σύστημα με Windows ή με Linux

Εγκαθιστώ την PHP σε σύστημα με Windows ή με Linux

Εγκαθιστώ την MySQL σε σύστημα με Windows ή με Linux

# Πλεονεκτήματα της PHP

---

Γρήγορη

Μηδενικό κόστος – Ελεύθερο Λογισμικό

Εύκολη στη χρήση

Μπορεί να τρέχει σε πολλά λειτουργικά συστήματα

Διαθέσιμη Τεχνική Υποστήριξη

Ασφαλής

Σχεδιασμένη για την υποστήριξη βάσεων δεδομένων

Εύκολα προσαρμόσιμη

9

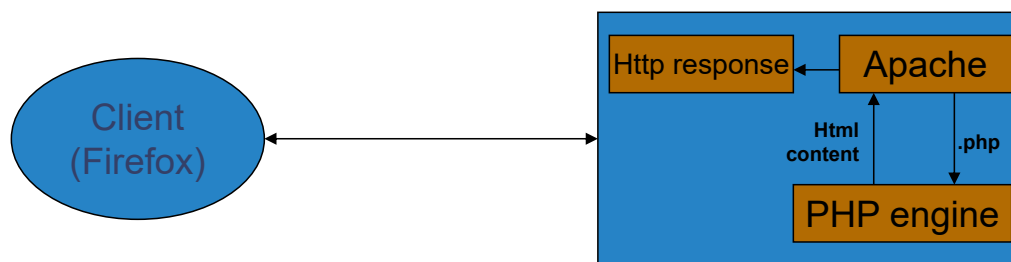
## Παράδειγμα

---

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

10

# Πως Δουλεύουν τα ServerSide Scripts



Όταν ο Browser ζητά ένα HTML αρχείο, ο server του επιστρέφει το αρχείο αυτό ως έχει, βάσει του πρωτοκόλλου

Όταν ζητηθεί ένα PHP, ο server περνά το αίτημα στη μηχανή PHP

Η μηχανή PHP διαβάζει τα αρχεία γραμμή-γραμμή και εκτελεί τον κώδικα

Τέλος, το αρχείο επιστρέφεται στον Browser ως απλή HTML

11

## Τι μπορούν να κάνουν τα Server-Side Scripts

1. Να τροποποιήσουν δυναμικά το περιεχόμενο μιας σελίδας
2. Απαντά σε ερωτήματα και δεδομένα χρηστών από φόρμες HTML.
3. Μπορεί να τρέχει σε πολλά λειτουργικά συστήματα
4. Δίνουν πρόσβαση σε δεδομένα και Βάσεις Δεδομένων και επιστρέφουν αποτελέσματα στον Browser
5. Διαμορφώνουν τη σελίδα για εξατομικευμένη χρήση

12

# Τι μπορούν να κάνουν τα ServerSide Scripts

---

Πλεονεκτήματα έναντι CGI και PERL ως προς την απλότητα και την ταχύτητα (built-in web δυνατότητες, εύκολο debug κλπ)

Παρέχει ασφάλεια, γιατί ο server-side κώδικας δεν είναι εμφανής στον πελάτη-Browser

Αφού οι επιστρεφόμενες σελίδες είναι HTML (όχι απαραίτητα valid, εξαρτάται από τον χρήστη), μπορούν να τις δουν όλοι οι Browsers

## Βασικά στοιχεία της PHP

---

## Σε αυτό το μέρος ...

---

Κανόνες Σύνταξης

Τελεστές

Μεταβλητές, Τύποι μεταβλητών, Εκχώρηση τιμών σε μεταβλητές

Ενσωμάτωση PHP στην HTML

Υποθετικές προτάσεις if και switch

Επαναληπτικές Δομές : βρόγχου while , do και for

Πίνακες

Συναρτήσεις

Χειρισμός Αρχείων

15

## Βασικοί Κανόνες Σύνταξης

---

Ο PHP κώδικας βρίσκεται ανάμεσα στα tags `<?php` και `?>`

Η σύνταξη μοιάζει με αυτή της C

Οι μεταβλητές ξεκινούν πάντα με το σύμβολο `$`

Ο τύπος της μεταβλητής καθορίζεται από την τιμή που δίνεται

Σχόλια

- `//` για μια σειρά
- `/* .. */` για πολλές σειρές

16

# Αριθμητικοί Τελεστές

Τελεστής	Περιγραφή	Παράδειγμα	Αποτέλεσμα
+	Πρόσθεση	$X=2+2$	4
-	Αφαίρεση	$X=5-2$	3
*	Πολ/σμός	$X=2*5$	10
/	Διαίρεση	$X=5/2$	2.5
%	Υπόλοιπο	$X=5\%2$	1
++	Αύξηση	$X=2$ $X++$	3
--	Μείωση	$X=2$ $X--$	1

17

## Τελεστές Ισότητας

Τελεστής	Παράδειγμα	Ισοδυναμεί με
=	$X=Y$	$X=Y$
+=	$X+=Y$	$X=X+Y$
-=	$X-=Y$	$X=X-Y$
*=	$X*=Y$	$X=X*Y$
/=	$X/=Y$	$X=X/Y$
%=	$X\%=Y$	$X=X\%Y$

18

## Τελεστές Σύγκρισης

Τελεστής	Περιγραφή	Παράδειγμα
<b>==</b>	Είναι ίσο με	4==2 (false)
<b>!=</b>	Δεν είναι ίσο με	4!=2 (true)
<b>&gt;</b>	Μεγαλύτερο από	4>2 (true)
<b>&lt;</b>	Μικρότερο από	5<2 (false)
<b>&gt;=</b>	Μεγαλύτερο ή ίσο από	5>=2 (true)
<b>&lt;=</b>	Μικρότερο ή ίσο από	

19

## Λογικοί Τελεστές

Τελεστής	Περιγραφή	Παράδειγμα
<b>&amp;&amp;</b>	And	X=6 Y=7 (X<10 && Y>3) Αποτέλεσμα true
<b>  </b>	Or	X=6 Y=7 (X<5    Y>3) Αποτέλεσμα true
<b>!</b>	not	X=6 Y=3 !(X==Y) Αποτέλεσμα true

20

## Μεταβλητές στην PHP

---

Όλες οι μεταβλητές στην PHP ξεκινούν με το σύμβολο \$. Οι μεταβλητές μπορούν να περιέχουν αλφαριθμητικά, αριθμούς, πίνακες και τιμές τύπου true-false. Π.χ.

- \$auth = true; Λογική
- \$num = 20; Ακέραιο
- \$price = 14.95; Πραγματική
- \$size = "large"; Συμβολοσειρά

21

## Μεταβλητές στην PHP

---

### Κανόνες Ονοματολογίας

Ένα όνομα μεταβλητής

- ξεκινά από γράμμα ή κάτω παύλα "\_"
- Περιέχει μόνο αλφαριθμητικούς χαρακτήρες και κάτω παύλα "\_" (a-Z, 0-9 και \_)
- Δεν περιέχει κενά π.χ. \$my\_string ή \$myString
- Όχι \$my String

22

# Παράδειγμα Χρήσης

## Παράδειγμα

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

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

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

23

## Υποθετικές Προτάσεις(if)

### Σύνταξη

```
<?php if (συνθήκη)
{
    Εντολές;
}
?>
```

### Παράδειγμα

```
<?php if ($grade<5)
{
    echo "Try next time!";
}
?>
```

24

## Υποθετικές Προτάσεις(if)

### Σύνταξη

```
<?php if (συνθήκη)
{
    Εντολές;
}
else {
    Εντολές;
}

?>
```

### Παράδειγμα

```
<?php if ($grade<5)
{
    echo "Try next time!";
}
else {
    echo "Congratulations!"
}

?>
```

25

## Υποθετικές Προτάσεις(switch)

### Σύνταξη

```
<?php switch (μεταβλητή)
{
    Περίπτωση#1:
        Εντολές
        break;
    Περίπτωση#2:
        Εντολές
        break;
    ...
    Περίπτωση#n:
        Εντολές
        break;
    default
        Εντολές;
}

?>
```

### Παράδειγμα

```
<?php if ($grade<5)
{
    echo "Try next time!";
}

?>
```

26

## Επαναληπτικές Δομές (while ή do ... while)

### Σύνταξη

```
<?php
while (συνθήκη)
{
    Εντολές;
}
?>

<?php
do{
    Εντολές;
} while (συνθήκη);
?>
```

### Παράδειγμα

```
<html><body>
<?php
    $i=1;
    while ($i<=5){
        echo "The number
is". $i." <br>";
        $i++;
    }
?>
</body></html>
```

27

## Επαναληπτικές Δομές (for)

### Σύνταξη

```
<?php
for(αρχικοποίηση;συνθήκη;αύξηση)
{
    Εντολές;
}
?>
```

### Παράδειγμα

```
<html><body>
<?php
    for($i=0;$i<=5;$i++){
        echo "Hello PHP!!".
<br>";
    }
?>
</body></html>
```

28

# Πίνακες στην PHP

## Παράδειγμα

```
<?php
$pin[0] = "Γιάννης ";
$pin[1] = "Νίκος ";
$pin[2] = "Ελένη ";
?>
```

## Παράδειγμα

```
<?php
$pin = array("Γιάννης", "Νίκος",
"Ελένη");

?>
```

Δεν χρειάζεται να ορίσουμε το μέγεθος!!

# Πίνακες στην PHP (maps)

## Παράδειγμα

```
<?php

$age["Γιάννης"] = 20;
$age["Νίκος"] = 25;
$age["Ελένη"] = 37;

?>
```

## Παράδειγμα

```
<?php

$age =
    array("Γιάννης"=>20,"Νίκος"=
>25, "Ελένη"=>37);

?>
```

# Επαναληπτικές Δομές για Πίνακες

## Σύνταξη

```
<?php
foreach (πίνακας as
μεταβλήτη) {

    Εντολές;

}
?>
```

## Παράδειγμα

```
<html><body>
<?php
$spoudastes[0] = "Γιάννης";
$spoudastes[1] = "Νίκος ";
$spoudastes[2] = "Ελένη ";
foreach ($spoudastes as
$name){
    echo "Το όνομα είναι ".$name.
    "<br>";
}
?>
</body></html>
```

31

# Συναρτήσεις (functions)

## Δήλωση Συνάρτησης

```
<?php
function myFunction() {
    Εντολές;
}
?>
```

## Κλήση Συνάρτησης

```
myFunction();
```

32

# Συναρτήσεις (functions)

## Παραδείγματα

---

### Συνάρτηση με δύο παραμέτρους

```
<?php
function writeMyName($fname, $age) {
    echo $fname."είναι".$age."χρόνων";
}
?>
```

- Στην συνάρτηση δεν δηλώνουμε τον τύπο της παραμέτρου.

### Κλήση της συνάρτησης

```
<?php
writeMyName("Nick",35);
?>
```

- **Αποτέλεσμα** : Nick είναι 35 χρόνων

33

# Συναρτήσεις (functions)

## Παραδείγματα

---

### Συνάρτηση με επιστρεφόμενες τιμές

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

### Κλήση της συνάρτησης (Το \$x θα γίνει 10)

```
<?php
$x = add(3,7);
?>
```

Echo "5 + 15 ισούται με ".add(5,15) → Η συνάρτηση καλείται πριν τυπωθεί οτιδήποτε.

34

`fopen(<όνομα αρχείου>,<mode>);`

- Ανοίγει ένα αρχείο σύμφωνα με το mode
- Π.χ. `$file = fopen("a.txt","r");`

`fclose(<όνομα αρχείου>);`

- Κλείνει ένα ανοιγμένο αρχείο

`feof(<όνομα αρχείου>);`

- Επιστρέφει true εάν το αρχείο έφτασε στο τέλος του

`fgets(<όνομα αρχείου>);`

- Επιστρέφει μία γραμμή από ένα ανοικτό αρχείο

`fgetc(<όνομα αρχείου>);`

- Επιστρέφει ένα χαρακτήρα από ένα ανοικτό αρχείο

`fwrite(<όνομα αρχείου>,<string δεδομένων>);`

- Γράφει σε ένα ανοικτό αρχείο το string δεδομένων και επιστρέφει το μέγεθος των δεδομένων που γράφτηκαν
- Παράδειγμα :

```
$fl = fopen("file.txt","w");  
echo fwrite($fl,"File Testing!"); //Θα τυπώσει τον αρ. 14  
fclose($fl);
```

`rewind(<όνομα αρχείου>);`

- Μετακινεί τον δείκτη ενός ανοικτού αρχείου στην αρχή (στο 0 δηλαδή)

# Χρήσιμες Συναρτήσεις για Αρχεία

`file_exists(<όνομα αρχείου>);`

- Επιστρέφει true αν το αρχείο υπάρχει

`copy(<αρχείο1>, <αρχείο2>);`

- Αντιγράφει το αρχείο 1 στην διαδρομή/όνομα αρχείου του αρχείου 2

`file_size(<όνομα αρχείου>);`

- Επιστρέφει το μέγεθος του αρχείου

37

## Modes για τα αρχεία

Mode	Περιγραφή
<b>r</b>	Μόνο ανάγνωση. Τοποθετεί το δείκτη στην αρχή του αρχείου
<b>r+</b>	Ανάγνωση και εγγραφή. Τοποθετεί το δείκτη στην αρχή του αρχείου
<b>w</b>	Μόνο για εγγραφή. Σβήνει τα προηγούμενα δεδομένα, και εάν το αρχείο δεν υπάρχει το δημιουργεί.
<b>w+</b>	Ανάγνωση και εγγραφή. Σβήνει τα προηγούμενα δεδομένα, και εάν το αρχείο δεν υπάρχει το δημιουργεί.
<b>a</b>	Προσθήκη. Ανοίγει και προσθέτει στο τέλος του αρχείου ή δημιουργεί εάν το αρχείο δεν υπάρχει.
<b>a+</b>	Προσθήκη και ανάγνωση. Ανοίγει και προσθέτει στο τέλος του αρχείου ή δημιουργεί εάν το αρχείο δεν υπάρχει.
<b>x</b>	Μόνο εγγραφή. Επιστρέφει σφάλμα αν το αρχείο υπάρχει.
<b>x+</b>	Εγγραφή και ανάγνωση. Επιστρέφει σφάλμα αν το αρχείο υπάρχει.

38

## PHP μέρος 2ο

---

Σε αυτό το μέρος...

---

Συμπερίληψη αρχείων include

Δημιουργία και διαχείριση φορμών με τις μεθόδους GET και POST

Δημιουργία και διαχείριση συνόδων COOKIES

Δημιουργία και διαχείριση συνόδων SESSION

# Include Files

Με την εντολή `include` μπορούμε να γράψουμε ένα μεγάλο πρόγραμμα σε πολλά αρχεία και μετά να κάνουμε όλα `include` από ένα πρόγραμμα.

Με το `include` είναι σαν να κάνουμε copy-paste το αρχείο που κάνουμε `include` στο σημείο που γίνεται η κλήση του `include`

## Σύνταξη

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

41

# Μεταβλητές superglobal

Η PHP χρησιμοποιεί ειδικούς προκαθορισμένους πίνακες που ονομάζονται **superglobal μεταβλητές**

- Επιτρέπουν την προσπέλαση σε HTTP headers,
- query string parameters κ.α.
- Παράδειγμα `$ _SERVER` (`HTTP_USER_AGENT`)

- **`$GLOBALS`** Array for storing data that needs superglobal scope
- **`$_COOKIE`** Array of cookie data passed to page via HTTP request
- **`$_ENV`** Array of server environment data
- **`$_FILES`** Array of file items uploaded to the server
- **`$_GET`** Array of query string data passed to the server via the URL
- **`$_POST`** Array of query string data passed to the server via the HTTP header
- **`$_REQUEST`** Array containing the contents of `$_GET`, `$_POST`, and `$_COOKIE`
- **`$_SESSION`** Array that contains session data
- **`$_SERVER`** Array containing information about the request and the server

42

# Παραδείγματα Χρήσης

---

Σχετικά με τα προγράμματα Web Servers

Παρουσίαση των HTTP Headers

- PHP:

[Headers.php](#)

[Headers.html](#) (*HTTP\_REFERER*)

[phpinfo\(\)](#)

43

## \$\_GET

---

Η μεταβλητή \$\_GET είναι ένας πίνακας ο οποίος περιέχει τιμές που έχουν σταλεί με την μέθοδο HTTP GET.

Όταν χρησιμοποιείται η μέθοδος GET οι τιμές οι οποίες στέλνονται ΦΑΙΝΟΝΤΑΙ στην μπάρα διεύθυνσης (URL).

- Συνεπώς, η μέθοδος αυτή δεν είναι καλή για την αποστολή κωδικών χρήσης.
- Είναι χρήσιμη όμως στην περίπτωση που ο χρήστης επιθυμεί να βάλει την σελίδα στα «Αγαπημένα» (Bookmarks).

Η μέθοδος μπορεί να αποστείλει μεταβλητές που το περιεχόμενό τους δεν ξεπερνάει τους 100 χαρακτήρες.

44

# \$\_GET - Παράδειγμα

## Φόρμα Συμπλήρωσης Στοιχείων

- **Αρχείο HTML**

```
<HTML><BODY>
<form name="login_form"
method="get"
action="get_login.php">
Όνομα : <input type="text"
name="fname"><br><br>
Επώνυμο : <input type="text"
name="lname"><br><br>
Email : <input type="text"
name="email"><br><br>
<input type="submit"
value="Login">
</form> </BODY></HTML>
```

- **Αρχείο PHP**

```
<?php
echo "Τα στοιχεία που
εισάγατε είναι τα εξής
:<br>";
echo "1.Όνομα :
".$_GET["fname"]."<br>";
echo "2.Επώνυμο :
".$_GET["lname"]."<br>";
echo "3.Mail :
".$_GET["email"]."<br>";
?>
```

45

# \$\_POST

Η μεταβλητή \$\_POST είναι ένας πίνακας ο οποίος περιέχει τιμές που έχουν σταλεί με την μέθοδο HTTP POST.

Όταν χρησιμοποιείται η μέθοδος POST οι τιμές οι οποίες στέλνονται ΔΕΝ ΦΑΙΝΟΝΤΑΙ στην μπάρα διεύθυνσης (URL).

- Συνεπώς, η μέθοδος αυτή είναι καλή για την αποστολή κωδικών χρήσης.

Η μέθοδος δεν έχει περιορισμό στο μέγεθος των δεδομένων που πρέπει να στείλει.

46

# \$\_POST- Παράδειγμα

## Φόρμα Συμπλήρωσης Στοιχείων

- **Αρχείο HTML**

```
<HTML><BODY>
<form name="login_form" method="post"
action="post_login.php">
Όνομα : <input type="text" name="fname">
<br><br>
Επώνυμο : <input type="text" name="lname">
<br><br>
Email : <input type="text" name="email">
<br><br>
<input type="submit" value="Login"></form>
</BODY></HTML>
```

- **Αρχείο PHP**

```
<?php
echo "Τα στοιχεία
που εισάγατε είναι
τα εξής :<br>";
echo "1.Όνομα :
".$_POST["fname"]." "<br>";
echo "2.Επώνυμο :
".$_POST["lname"]." "<br>";
echo "3.Mail :
".$_POST["email"]." "<br>";
?>
```

47

## Διαχείριση Κατάστασης

48

# Cookies

## Τι είναι το COOKIE ;

Ένα cookie είναι ένα μικρό αρχείο που ο εξυπηρετητής ενσωματώνει στον browser του πελάτη.

- Το cookie χρησιμοποιείται για την αναγνώριση του χρήστη.
- Κάθε φορά που ο ίδιος χρήστης επισκεφτεί ξανά την ίδια ιστοσελίδα, ο Η/Υ του χρήστη θα στείλει στον web server και το cookie αυτό.
- Με αυτόν τον τρόπο η ιστοσελίδα μπορεί να γνωρίζει σχετικά με τις προτιμήσεις του επισκέπτη και να προσαρμοστεί ανάλογα.

Μπορούμε να έχουμε έως 300 cookies ή 4kb/cookie ή 20 cookie ανά server ή domain.

- Σημείωση ότι εντελώς καθορισμένοι servers και domain θεωρούνται ως ξεχωριστές οντότητες και έχουν το 20 cookie για το καθένα ξεχωριστά) - RFC 2109

`bool setcookie ( string name [, string value [, int expire [, string path [, string domain [, bool secure [, bool httponly]]]]]) )`

49

## Cookies - Παράδειγμα

### Cookie.php

Τα cookies ορίζονται πριν τη δήλωση του <html> tag.

`setcookie(<όνομα>,<τιμή>,<λήξη>)`

- Ορίζει ένα cookie. Το cookie καταστρέφεται ανάλογα με την λήξη του.
- Για να καταστρέψουμε ένα cookie μπορούμε να το ορίσουμε χρησιμοποιώντας αρνητική λήξη.
- Τιμή 0 σημαίνει ότι το cookie λήγει όταν κλείσει ο browser

```
<?php
//setcookie() defines a cookie to be
sent along with the rest of the HTTP
headers.
//Once the cookies have been set,
they can be accessed on the next page
load
$cookie_name = "ceiduser";
$cookie_value = "koutsoni";
setcookie($cookie_name,
$cookie_value, time() + (86400 *
30)); // 86400 = 1 day

if(isset($_COOKIE['ceiduser']))
{
    echo "Cookie OK! :- ) -
Cookie Value: " . $_COOKIE['ceiduser'];
}
else
    echo "Cookie burned! :- (" ;
?>
```

50

# Sessions

## Μεταβλητές Session

Οι μεταβλητές Session χρησιμοποιούνται για τη διατήρηση πληροφορίας σε περίπτωση διαδοχικών προσβάσεων.

- Με αυτόν τον τρόπο είναι δυνατή η εξατομίκευση του περιεχομένου.

Συνήθως η πληροφορία που αποθηκεύεται στις μεταβλητές αυτές είναι τα username & userid.

Για να δημιουργηθεί μια session μεταβλητή, την αποθηκεύουμε σε ένα Session Object.

- Το session object χρησιμοποιείται για να αποθηκευτεί πληροφορία ή και να αλλάξουν οι ρυθμίσεις ενός user session.
- Οι μεταβλητές που αποθηκεύονται στο session object κρατούν την πληροφορία για έναν χρήστη κάθε φορά και είναι διαθέσιμες σε όλες τις σελίδες μιας εφαρμογής.

Τα sessions σε αντίθεση με τα cookies αποθηκεύονται στον server και όχι στον client

Τα sessions λήγουν (invalidate) συνήθως μετά από 30' (default) **ή όταν ο χρήστης κλείσει το φυλλομετρητή του (default στην php)**

## Φόρμα - Login

### login\_form.html

```
<form name="login_form" method="post" action="login.php">
Username: <input type="text" name="username">
<br>Password: <input type="password" name="password">
<br><input type="submit" value="Login">
</form>
```

# Επεξεργασία φόρμας

## Login.php

```
1  <?php
2  session_start();
3  ?>
4  <html>
5  <head>
6  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-7"/>
7  </head>
8  <?php
9
10 if (isset($_SESSION['session_username']))
11 {
12     echo "Έχεις κάνει ήδη login <b>".$_SESSION['session_username']. "</b>! Μια φορά αρκεί.";
13     echo "<br><a href='logout.php'>Log off</a>";
14 }
15 else
16 {
17     if (strlen($_POST['username']) < 6 // strlen($_POST['password']) < 6)
18     {
19         echo "<h2>Δεν έχετε δώσει τα σωστά δεδομένα!</h2>";
20         echo "<br /><a href='login_form.php'><img src='meme.png' /></a>";
21     }
22     else
23     {
24         $_SESSION['session_username'] = $_POST['username'];
25         echo "Welcome <b>".$_SESSION['session_username'];
26         echo "<br><a href='logout.php'>Log off</a>";
27     }
28 }
29
30 ?>
```

53

# Φόρμα - Logout

## Logout.php

```
<?
    session_start();
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8"/>
</head>
<?
    session_unset();
?>
<p>Ευχαριστούμε που μας επισκεφθήκατε</p>
</html>
```

54

# Sessions vs Cookies

## Cookies

- Τα cookies έχουν μεγάλο χρόνο ζωής
- Δεν υπάρχει πρόβλημα με web clusters

## Sessions

- Οι χρήστες δεν θέλουν πληροφορία να αποθηκεύεται στο pc τους
- **Ο χρήστης δεν γνωρίζει, και ούτε μπορεί να δει, τι αποθηκεύεται στον server**
- Δεν χρειάζεται να μεταφέρεται η πληροφορία στον client
- Μεγαλύτερο μέγεθος από τα cookies -128M