



Σχεδίαση Εφαρμογών και Υπηρεσιών Διαδικτύου

5^η Διάλεξη: Προγραμματισμός στην πλευρά του εξυπηρετητή: προχωρημένα θέματα PHP

Δρ. Απόστολος Γιάμας

Διδάσκων (407/80)

gkamas@uop.gr



Προκαθορισμένες μεταβλητές 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.



Πίνακας `$_SERVER`

- Παρέχει μεγάλο αριθμό πληροφοριών, πχ:
 - `'SERVER_NAME'`: Το όνομα του εξυπηρετητή
 - `'HTTP_USER_AGENT'`: Ο browser του χρήστη
 - `'REMOTE_ADDR'`: Η διεύθυνση του χρήστη
 - `'SCRIPT_FILENAME'`: Το όνομα του PHP script



Διαφορά Get – Post στις HTML φόρμες

- Get, Post: Μέθοδοι μεταφοράς δεδομένων μιας φόρμας από τον web client στον web server
 - Post: Αποστολή δεδομένων φόρμας «διαφανώς» από το χρήστη
 - Get: Τα στοιχεία της φόρμας μεταφέρονται ως τμήμα της URL, π.χ. <http://www.uop.gr/~gkamas/grade.php?grade=4>
- Ερώτηση: πως μπορώ να στείλω δεδομένα μιας φόρμας προς επεξεργασία από ένα PHP script χωρίς να χρειαστεί να ανοίξω τη φόρμα;
- Καμία διαφορά ως προς την επεξεργασία των δεδομένων από το server-side script

Φόρμα



— `$_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 αόρατη στον πελάτη. Η μέθοδος POST δεν έχει περιορισμούς ως προς το μέγεθος της πληροφορίας.



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="number_sum.php" method=post>
<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>" . $_POST[number1] . " + " . $_POST[number2] . " = " .
    ($_POST[number1] + $_POST[number2]) . "</H1>";
?>
</body>
</html>
```

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 browser ζητά μια σελίδα, στέλνει και το cookie. Τα PHP scripts μπορούν και να πάρουν και να ορίσουν τις τιμές των cookies.



Ορισμός COOKIE

- setcookie (name, value, expire, path, domain);
 - name: Το όνομα του cookie.
 - value: Η τιμή του cookie. Η τιμή αυτή αποθηκεύεται στον υπολογιστών των clients, γι'αυτό μην αποθηκεύετε ευαίσθητες πληροφορίες.
 - expire: Η ώρα λήξης του cookie.
 - path: Το path στον server στο οποίο θα είναι διαθέσιμο το cookie.
 - domain: Το domain στο οποίο το cookie είναι διαθέσιμο.
- ΠΡΟΣΟΧΗ: Η εντολή setcookie θα πρέπει να χρησιμοποιείται πριν το tag HTML



Παράδειγμα: Ορισμός Cookie

```
<?php  
setcookie("uname", $name, time()+36000);  
?>  
  
<html>  
  
<body>  
  
<p>  
Το cookie έχει οριστεί  
  
</p>  
  
</body>  
  
</html>
```



Παράδειγμα: Ανάγνωση Cookie

```
<html>
<body>
<?php
if (isset($_COOKIE["uname"]))
echo "Καλωσήλαθε " . $_COOKIE["uname"] . "!<br />";
else
echo «Δεν έχετε κάνει Login!<br />";
?>
</body>
</html>
```



Μεταβλητές Session

- Οι μεταβλητές Session αποθηκεύουν πληροφορία σχετική με έναν χρήστη και είναι διαθέσιμες σε όλες τις σελίδες μιας εφαρμογής web στη διάρκεια του user session
- 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 ενός application

Τι είναι ο έλεγχος συνόδων λειτουργίας (Session



- Το HTTP είναι πρωτόκολλο «χωρίς κατάσταση» (stateless), δηλ. δεν έχει ένα ενσωματωμένο τρόπο να διατηρεί την κατάσταση μεταξύ δύο συναλλαγών.
- Π.χ., όταν ένας χρήστης ζητά μια σελίδα, ακολουθούμενη από μια άλλη, το HTTP δεν έχει τρόπο να μας πει ότι οι δυο αιτήσεις ήρθαν από τον ίδιο χρήστη.
- Η ιδέα των συνόδων λειτουργίας (sessions) είναι να μπορούμε να παρακολουθούμε ένα χρήστη κατά τη διάρκεια της επίσκεψής του σε ένα web site.
- Πλεονεκτήματα:
 - Εμφάνιση περιεχομένων σύμφωνα με το επίπεδο πιστοποίησης ή προσωπικών προτιμήσεων
 - Παρακολούθηση της συμπεριφοράς του χρήστη
 - Χειρισμός καλαθιών αγορών (εφαρμογές e-commerce)

Session Object



- Όταν δουλεύει κανείς με μια εφαρμογή web, την ανοίγει, κάνει ορισμένες αλλαγές και μετά την κλείνει. Αυτή η διαδικασία είναι όμοια με τη συμπεριφορά του session. Ο υπολογιστής γνωρίζει ποιος είναι ο χρήστης του. Γνωρίζει επίσης πότε ξεινιά κανείς την εφαρμογή και πότε σταματά. Στο Internet όμως υπάρχει ένα πρόβλημα ... Ο web server δεν ξέρει ποιος είναι ο χρήστης και τι ακριβώς κάνει, γιατί η ονοματολογία στο HTTP δε διατηρεί την ίδια κατάσταση κάθε φορά που συνδεόμαστε στην εφαρμογή.
- Για να λυθεί το πρόβλημα αυτό, δημιουργείται ένα μοναδικό cookie κάθε φορά για κάθε χρήστη. Το cookie στέλνεται στον πελάτη και περιλαμβάνει πληροφορία που χαρακτηρίζει τον χρήστη. Αυτό το interface ονομάζεται session object.
- Το session object χρησιμοποιείται για να αποθηκεύσει την πληροφορία για κάθε user session. Συνήθως αποθηκεύονται οι μεταβλητές που περιλαμβάνουν το όνομα, το id και οι προτιμήσεις. Ο εξυπηρετητής δημιουργεί ένα νέο session object για κάθε νέο χρήστη και καταστρέφει το session όταν έχουμε session expire.



Βασικές λειτουργίες σε session

- `session_start()`: Δημιουργία/ενεργοποίηση session
- `session_destroy()`: Τερματισμός session
- `session_register (mixed name [, mixed ...])`: Προσθήκη μεταβλητής στο session (εναλλακτικά:
`$_SESSION["test"]="Test";`)
- `session_unregister (string name)`: Διαγραφή μεταβλητής από το session
- `session_unset()`; Ακυρώνει όλες τις μεταβλητές του session



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 λειτουργίας: για να μπορεί μια μεταβλητή να παρακολουθείται από διάφορα scripts, θα πρέπει να εγγραφεί με μια κλήση της συνάρτησης `session_register()`
- Π.χ., εγγραφή της μεταβλητής `$myvar`
 - `$myvar = 5;`
 - `session_register("myvar");`
- Η μεταβλητή (η τιμή της) θα παρακολουθείται μέχρι να τερματιστεί η σύνοδος ή καταργηθεί η εγγραφή της
- Μπορούν να εγγραφούν ταυτόχρονα περισσότερες από μια μεταβλητές:
 - `session_register("myvar1", "myvar2");`



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 @$myvar;`
- Προσοχή: μπορεί μια μεταβλητή συνόδου να έχει «επικαλυφθεί» από μια μεταβλητή που έχει οριστεί από το χρήστη μέσω της αποστολής στοιχείων από κάποια HTML φόρμα (με GET ή POST)
- Για να ελέγξουμε αν μια μεταβλητή είναι όντως μεταβλητή συνόδου:
- `$result = session_is_registered("myvar"); // επιστρέφει true/false`
Χειρισμός απλών συνόδων λειτουργίας:



Βήμα 4 χειρισμού session

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



Παράδειγμα: Login

```
<?
function process_form() {
    switch ($_POST["sex"]) {
        case "yes":
            print "Καλωσήρθατε Κυριε (".$_SESSION["take_email_session"].)";
            break;
        case "no":
            print "Καλωσήρθατε Κυρια (".$_SESSION["take_email_session"].)";
            break;
        default:
            print "Καλωσήρθατε (".$_SESSION["take_email_session"].)";
            break;
    }
}
?>
```



Παράδειγμα: Login

```
<?
if(!isset($_SESSION["take_email_session"])) {
    if (((strlen($_POST["take_email"])<6) || (strlen($_POST["take_password"])<6))) {
        print "Δεν εχετε δωσει σωστά τα δεδομένα";
        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>
```



Παράδειγμα: Session

```
<?
    session_start();
?>
<html>
<?
    session_unset();
?>
```

Ευχαριστούμε που μας επισκεφθήκατε

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()` αυτομάτα 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());
?>
```

Εισαγωγή στις Βάσεις Δεδομένων / SQL



- Σχεσιακές Βάσεις Δεδομένων ή Αρχεία?
- Πλεονεκτήματα αποθήκευσης δεδομένων σε σχεσιακές βάσεις δεδομένων αντί σε αρχεία κειμένου:
 - Γρηγορότερη πρόσβαση στα δεδομένα
 - Δημιουργία ερωτημάτων/συνθηκών για την «εξαγωγή» δεδομένων
 - Ενσωματωμένος μηχανισμός για το χειρισμό ταυτόχρονης πρόσβασης
 - Ενσωματωμένα συστήματα δικαιωμάτων

Πίνακες



— Πίνακας

CustomerID	Name	Address	City
1	Κώστας Νικολάου	Παλαμά 15	Πάτρα
2	Πέτρος Πετρίδης	Κορνάρου 32	Τρίπολη
3	Παύλος Πέτρου	Παλαμά 35	Αθήνα

— Στήλες ή πεδία ή ιδιότητες

— Γραμμές ή εγγραφές

— Τιμές

Κλειδιά



CUSTOMERS

Πρωτεύοντα κλειδιά

CustomerID	Name	Address	City
1	Παπαδόπουλος Στέλιος	Δημοκρατίας 4	Αθήνα
2	Μιρέλλη Μαρία	Βικέλα 37	Ερμούπολη
3	Σώκος Πέτρος	Τηλίου 10	Βόλος

ORDERS

Ξένο κλειδί

OrderID	CustomerID	Ammount	Date
1	3	32.90	2/2/2003
2	1	25.00	12/3/2003
3	2	20.10	1/4/2001
4	1	75.80	3/4/2003

Σχέσεις



- Ένα – προς – Ένα (π.χ. Αν είχαμε βάλει τους πελάτες και τις διευθύνσεις τους σε ξεχωριστούς πίνακες)
- Ένα προς πολλά (π.χ. Customers – Orders)
- Πολλά – προς – Πολλά (π.χ. πίνακες Books – Authors)



SQL - SELECT

- Σύνταξη:
 - SELECT column_name(s) FROM table_name
- Παραδείγματα:
 - SELECT Name, Comment FROM `students`
 - SELECT * FROM `students`



SQL - WHERE

- Σύνταξη:
 - SELECT column FROM table WHERE column operator value
- Όπου operator: =, <>, >, <, >=, <=, BETWEEN, LIKE
- Παραδείγματα:
 - SELECT * FROM `students` WHERE id=1



SQL - INSERT INTO

- Σύνταξη :
 - INSERT INTO table_name VALUES (value1, value2,...)
 - INSERT INTO table_name (column1, column2,...) VALUES (value1, value2,...)
- Παραδείγματα:
 - INSERT INTO students VALUES ('Apostolis',45,'Test entry')
 - INSERT INTO students (Name,ID) VALUES ('Apostolis2',48)



SQL - UPDATE

- Σύνταξη :
 - UPDATE table_name SET column_name = new_value WHERE column_name = some_value
- Παραδείγματα:
 - UPDATE students SET Comment ='New Comment' WHERE id = 48



SQL - DELETE

- Σύνταξη :
 - DELETE FROM table_name WHERE column_name = some_value
- Παραδείγματα:
 - DELETE FROM students WHERE id = 48
 - DELETE FROM students WHERE id > 25



SQL – Άλλες Επιλογές

- ORDER BY: `SELECT * FROM students ORDER BY Name`
- AND/OR: `SELECT * FROM students WHERE NAME='test' OR NAME='test2'`
- IN: `SELECT column_name FROM table_name WHERE column_name IN (value1,value2,..)`
- BETWEEN / AND: `SELECT column_name FROM table_name WHERE column_name BETWEEN value1 AND value2`



SQL - CREATE

- Σύνταξη :
 - CREATE DATABASE database_name
 - CREATE TABLE table_name (column_name1 data_type, column_name2 data_type,)
- Παραδείγματα:
 - CREATE TABLE address (id int(11), adr varchar(100), city varchar(50), PRIMARYKEY(id))



SQL – ALTER TABLE

- Σύνταξη:
 - ALTER TABLE table_name ADD column_name datatype
 - ALTER TABLE table_name DROP COLUMN column_name
- Παραδείγματα:
 - ALTER TABLE address ADD Person varchar(30)
 - ALTER TABLE address DROP COLUMN Person



SQL - DROP TABLE

- Σύνταξη:
 - DROP TABLE table_name
- Παραδείγματα:
 - DROP TABLE address