Προγραμματισμός Διαδικτύου

Δρ. Μηνάς Δασυγένης
mdasygenis@uowm.gr

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.

Σκοπός ενότητας

- ✓ Εξοικείωση με τη διαχείριση πινάκων.
- ✓ Σωστή χρήση συναρτήσεων της PHP.
- ✓ Ασφάλεια και αντιμετώπιση προβλημάτων.
- ✓ Χρήση και διαχείριση συνεδριών.
- ✓ Χρήση και διαχείριση cookies.

Προγραμματισμός Διαδικτύου

Περιεχόμενα

1. Πίνακες
    - 1.1 Ταξινόμηση πινάκων
    - 1.2 Αναζήτηση πινάκων
2. Ασφάλεια
    - 2.1 Αρχεία include
    - 2.2 Αντιμετώπιση προβλημάτων ασφάλειας
3. Συνεδρίες
    - 3.1 Δημιουργίες συνεδριών/συνοδών (Sessions)
    - 3.2 Καταστροφή συνεδριών
4. Κούκις (Cookies)
    - 4.1 Χρήση
    - 4.2 Λειτουργία
    - 4.3 Μετάδοση

Διαχείριση Πινάκων

Μπορούμε να μάθουμε το μέγεθος ενός πίνακα με τη μέθοδο sizeof ή count.

```
$pin=array("v1", "v2", "v3") ;
echo sizeof($pin) ;

for ($i=0; $i<count($pin); $i++) {
      …..
}
```

Ταξινόμηση πινάκων

Υπάρχουν διάφορες μέθοδοι για την ταξινόμηση πινάκων:

sort(<πίνακας>);

- Ταξινομεί τον πίνακα με αλφαριθμητική σειρά.

rsort(<πίνακας>);

- Ταξινομεί τον πίνακα με ανάποδη αλφαριθμητική σειρά.

array_multisort(<πίνακας 1>, <πίνακας 2>, <πίνακας 3>, …);

- Ταξινομεί ταυτόχρονα πολλούς πίνακες. Ο αριθμός παραμέτρωνmείναι μεταβλητός. Για παράδειγμα, εάν είχαμε έναν πίνακα με ονόματα και έναν με βαθμούς θα τους ταξινομούσε, τον πίνακα 1 (ονόματα) σε αλφαβητική σειρά αλλά ταυτόχρονα θα άλλαζε και τις τιμές του πίνακα 2 (βαθμός) ανάλογα.

```
//Ταξινόμηση με βάση τα ονόματα
 array_multisort($names, $grades);
//Ταξινόμηση με βάση τους βαθμούς
 array_multisort($grades, $names);
```

Αναζήτηση σε πίνακα

- array_search(<τιμή>, <πίνακας>, [αυστηρότητα]);

- Ψάχνει στον πίνακα να βρει την τιμή και επιστρέφει το κλειδί.
- Η παράμετρος 'αυστηρότητα' παίρνει τιμές true ή false. Σε περίπτωση που είναι true ένας αριθμός π.χ. 3 δεν είναι το ίδιο με το "3" (δηλαδή η αλφαριθμητική τιμή θεωρείται διαφορετική). Η προκαθορισμένη (default) τιμή είναι false.

- Παράδειγμα

```
$pin1=array("a"=>"Steve","b"=>"Mary","c"=> "Nick") ;
$pin2 = array("a"=> "5", "b"=>5, "c"=>3) ;

echo array_search("Mary",$pin1) . "<br>" ;
echo array_search(5,$pin2, true) . "<br>" ;
echo array_search("5",$pin2, true) . "<br>" ;
```

Συναρτήσεις ημερομηνίας και ώρας

- **getdate()**

- Επιστρέφει έναν πίνακα ο οποίος περιέχει τα παρακάτω.

[seconds] - Δευτερόλεπτα
[minutes] - Λεπτά
[hours] - Ώρες
[mday] - Αριθμό μέρας μέσα στον μήνα
[wday] - Αριθμό μέρας μέσα στην εβδομάδα
[year] - Χρόνος
[yday] - Αριθμό μέρας μέσα στο χρόνο
[weekday] - Όνομα μέρας
[month] - Όνομα μήνα

Παράδειγμα:

```
$tmp = getdate() ;

echo "Σήμερα είναι " .$tmp["weekday"] ;
```

Συναρτήσεις ημερομηνίας και ώρας

- **time()**

- Επιστρέφει την ώρα ως ένα αριθμό. Ο αριθμός αυτός είναι το σύνολο δευτερολέπτων που έχουν περάσει από την 1η Ιανουαρίου 1970 και ώρα 00:00:00.

- Παράδειγμα:

```
$tmp = time() ;
echo $tmp ;
```

- Θα τυπώσει: 1165419267

Μαθηματικές συναρτήσεις

- **round(<πραγματικός αριθμός>);**

  - Στρογγυλοποιεί τον πραγματικό αριθμό στον πλησιέστερο ακέραιο.

- **rand([μικρότερος αριθμός], [μεγαλύτερος αριθμός]);**

  - Επιστρέφει έναν τυχαίο αριθμό μεταξύ του μικρότερου και μεγαλύτερου αριθμού που έχουν δηλωθεί. Εάν κληθεί χωρίς παραμέτρους απλά επιστρέφει έναν τυχαίο αριθμό.

    πχ. `echo rand(1,10);`

- **srand(<αριθμός>);**

  - Ορίζεται τυχαίος αριθμός ο οποίος θα χρησιμοποιηθεί από γεννήτρια τυχαίων αριθμών.

    π.χ `srand(time());`

Αλφαριθμητικές συναρτήσεις

- **trim(<αλφαριθμητικό (string)>, [ειδικοί χαρακτήρες])**
  - Αφαιρεί τους ειδικούς χαρακτήρες και από τις 2 μεριές του αλφαριθμητικού (string).
  - Ειδικοί χαρακτήρες:
    - "\0" – NULL
    - "\t" – Tab
    - "\n" – Νέα γραμμή
    - "\x0B" – Κάθετο tab
    - "\r" – Enter/Επιστροφή φορέα (carriage return)
    - " " – Κενό (White space)
  - Σε περίπτωση που δεν βάλουμε τη δεύτερη παράμετρο, τους αφαιρεί όλους τους παραπάνω χαρακτήρες.

- **rtrim(<αλφαριθμητικό (string)>, [ειδικοί χαρακτήρες])**
  - Αφαιρεί τους ειδικούς χαρακτήρες από τη δεξιά μεριά του αλφαριθμητικού (string).

- **ltrim(<αλφαριθμητικό (string)>, [ειδικοί χαρακτήρες])**
  - Αφαιρεί τους ειδικούς χαρακτήρες από τη δεξιά μεριά του αλφαριθμητικού (string).

Αποστολή e-mail

- **mail(<προς>, <θέμα>, <μήνυμα>, [έξτρα], [παράμετροι])**
  - Στέλνει email.
  - Για να χρησιμοποιηθεί, πρέπει να έχει οριστεί mail server (διακομιστής).

Παράδειγμα:

```
<?php
        $to="alex@yahoo.com";
        $subject = "Δοκιμή" ;
        $message = "Γεια σου! Αυτό είναι ένα δοκιμαστικό
         μήνυμα.";
        $from = john@hotmail.com ;
        $headers = "From: $from" ;
        mail($to, $subject, $message, $headers);

        echo "Το email στάλθηκε με επιτυχία";
?>
```

Προσοχή στα αρχεία include. Να μη μπορούν να τα διαβάσουν άλλοι

- Συχνά οι λεπτομέρειες σύνδεσης αποθηκεύονται στα αρχεία include.
- Τα διαπιστευτήρια σύνδεσης (όνομα χρήστη, κωδικός πρόσβασης, βάση δεδομένων) πρέπει να αποθηκεύονται σε μορφή αναγνώσιμη από διακομιστή http (http server).
- Αυτό γενικά σημαίνει ότι βρίσκονται σε απλό κείμενο στον διακομιστή (server) αρχείων.

Ασφάλεια στην PHP: Διασφαλίζοντας ότι ο χρήστης δίνει έγκυρο e-mail

```
<?php

$clean = array() ;

$email_pattern='/^[^@\s<&>]+@([-a-z0-9]+\.)+[a-z]{2,}$/i';

If (preg_match($email_pattern, $_POST['email']))
{

      $clean[email] = $_POST['email'];

}

?>
```

Ασφάλεια στην PHP: Διασφαλίζοντας τους ακεραίους

```
<?php

$clean = array() ;

if ($_POST['num']==strval(intval($_POST['num'])))
{

      $clean['num' = $_POST['num'];
}

?>
```

Ασφάλεια στην PHP: Διασφαλίζοντας τους πραγματικούς αριθμούς

```
<?php

$clean = array() ;

if ($_POST['num']==strval(floatval($_POST['num'])))
{

      $clean['num' = $_POST['num'];
}

?>
```

Ανοίγουμε τη βάση δεδομένων μόνο όταν χρειάζεται, αλλιώς DoS

```
<?php

$link=mysql_connect('localhost','mysql_user','mysql_password');

if (!$link) {

    die('Could not connect: ' . mysql_error());

}

echo 'Connected successfully';

[rest of the page]

mysql_close($link);

?>
```

Πρόβλημα με SQL Injection (έγχυση) (1/2)

```
<?php

$id = $_GET['id'];

$sql = 'select * from table where id=' . $id;

?>
```

Αυτό υποτίθεται ότι έχει ως αποτέλεσμα μια δήλωση SQL όπως παρακάτω:

```
select * from table where id = 5
```

Πρόβλημα με SQL Injection (έγχυση) (2/2)

Τι συμβαίνει όταν ένας εισβολέας καλεί τη διεύθυνση URL

```
http://site.tld/index.php?id=5 union select
username,password from user
```

Το προκύπτον ερώτημα γίνεται ως εξής:

```
select * from table where id=5
union select * from user
```

Ασφάλεια στα SQL Injections

Φιλτράρετε τα δεδομένα σας

Αυτό δεν μπορεί να είναι υπερβολικό. Με το καλό φιλτράρισμα των δεδομένων στη θέση τους, οι περισσότερες ανησυχίες για την ασφάλεια είναι μετριασμένες και ορισμένες εξαλείφονται πρακτικά.

Αναφέρετε τα δεδομένα σας

Εάν η βάση δεδομένων σας το επιτρέπει (η MySQL), βάλτε μονά εισαγωγικά γύρω από όλες τις τιμές στις δηλώσεις SQL, ανεξάρτητα από τον τύπο δεδομένων.

Αποφύγετε τη λειτουργία των δεδομένων σας

Μερικές φορές τα έγκυρα δεδομένα μπορούν να παρεμποδίσουν ακούσια τη μορφή της ίδιας της δήλωσης SQL. Χρησιμοποιήστε τη συνάρτηση mysql_escape_string() ή μια συνάρτηση δραπέτευσης, εγγενή στη συγκεκριμένη βάση δεδομένων σας. Αν δεν υπάρχει κάποια συγκεκριμένη, η συνάρτηση addslashes() είναι μια καλή τελευταία λύση.

Ποτέ, μα ποτέ δεν εμπιστευόμαστε τις εισόδους (inputs)

Οποιαδήποτε είσοδος χρήστη μπορεί να χρησιμοποιηθεί για να ενεργοποιήσει μια έγχυση SQL (SQL Injection).

GET, POST, HTTP_REFERER, COOKIE, είσοδος RSS, κλπ.

Ακριβώς επειδή δεν είναι εύκολο για έναν χρήστη να χειρίζεται την είσοδο που παρέχεται, **δεν σημαίνει ότι είναι ασφαλές!**

Για λόγους ασφαλείας δε χρησιμοποιούνται τα γενικά αρχεία καταγραφής (register globals)

Η ρύθμιση ελεγχόταν στο php.ini .

http://us3.php.net/manual/en/ini.core.php#ini.register-globals

Πρέπει να εξαφανιστεί πολύ σύντομα.

Επιτρέπει σε όλες τις μεταβλητές να αναφέρονται χωρίς πλαίσιο.

Κάνει `$_GET['id']` το ίδιο με το `$id`

Η προεπιλεγμένη συμπεριφορά στην PHP 3 απαιτεί πολλές εφαρμογές παλαιού τύπου.

Πρέπει να τεθεί εκτός λειτουργίας!

Τα register globals οδηγούν σε προβλήματα

Μεταβλητή σύγκρουση

- Μπορεί να επιτρέψει στους κακόβουλους χρήστες να ορίσουν αυθαίρετα ή να επαναφέρουν τις κρίσιμες μεταβλητές.

```
$_SERVER['DOCUMENT_ROOT']
```

- Ενθαρρύνει τυχαία κωδικοποίηση.
- Προκαλεί σύγχυση κατά την ανάγνωση / ανασκόπηση του κώδικα.

Προβλήματα που δημιουργούνται με SQL Injection (έγχυση)

- JOIN ή UNION με άλλα δεδομένα
- Εκθέτονται ευαίσθητα δεδομένα.
- Εισάγονται κακόβουλα δεδομένα.
- Αλλάζει το περιεχόμενο της βάσης δεδομένων.
- Διαβάζονται ή γράφονται αρχεία του λειτουργικού συστήματος.
- Χειρίζονται ή εκθέτονται αρχεία του συστήματος MySQL.

Τρόποι προστασίας από την έγχυση SQL (Injection)

PHP mysql_real_escape_string()

http://us2.php.net/mysql_real_escape_string

Καθαρίζει μεταβλητές για χρήση σε ένα ερώτημα.

```
$username = mysql_real_escape_string ($_ POST['username']);
$query = mysql_query("update user set username='username');
```

Εκτέλεση τύπου μεταβλητού:

```
$query = mysql_query('select * from content where
       id ='. intval($_GET['id']);
```

Η συνάρτηση addslashes() δεν είναι αποτελεσματικός τρόπος για την προστασία από την ένεση SQL.

Παράδειγμα

```
<?php

$Conn = mysql_connect('localhost', 'user', 'pass');

mysql_select_db('db');

mysql_query("call ret_content(1,@retval)") ;

$retval = mysql_query("select @retval");

while ($row = mysql_fetch_row($retval)) echo $row[0];

mysql_close();

?>
```

Αποθηκευμένες διαδικασίες

Πρόκειται για μια από τις ασφαλέστερες προσεγγίσεις!

Τι μας επιστρέφει μια εντολή εύρεσης δεδομένων από τη βάση δεδομένων;

- Πριν ερωτήσετε τη βάση δεδομένων MySQL θα πρέπει να γνωρίζετε ότι το ΑΠΟΤΕΛΕΣΜΑ των ληφθέντων δεδομένων (αποτέλεσμα του ερωτήματος) θα μπορούσε να είναι ένας αριθμός, συμβολοσειρά, πίνακας, πίνακας δύο διαστάσεων και ούτω καθεξής.

- Για παράδειγμα, εάν το αποτέλεσμα του ερωτήματός σας μεταφέρει μόνο ένα κελί όπως ένα name('Peter') ή έναν αριθμό όπως το μέγεθος του RBC ('6.54'), τα δεδομένα που έχουν ληφθεί είναι μόνο μια συμβολοσειρά ή ένας αριθμός.

- Αλλά σε περίπτωση που επιλέξετε μια ολόκληρη σειρά δεδομένων στο MySQL το αποτέλεσμα θα είναι ένας πίνακας που περιέχει τα κελιά του πίνακα όπως το $x[0].

- Τέλος, αν το αποτέλεσμα του ερωτήματός σας είναι ένας ολόκληρος πίνακας (`SELECT * FROM pat_info`), το αποτέλεσμα θα είναι ένας πίνακας δύο διαστάσεων, όπως το $x[0][0], όπου ο πρώτος δείκτης καθορίζει τη στήλη και ο δεύτερος δείκτης ορίζει τον αριθμό σειράς.

Τοποθετώντας τα αποτελέσματα σε μεταβλητή

```
<?php
// εδώ εμφανίζονται οι μεταβλητές...
//  Δημιουργία σύνδεσης και επιλογή βάσης δεδομένων..
// ... Ερώτημα στη βάση δεδομένων...

      // Ανάκτηση αποτελεσμάτων
      $row = mysql_fetch_array($result);
      print_r($row);
      // ... Κλείσιμο της βάσης δεδομένων ... //

?>
```

Το MYSQL_FETCH_ARRAY διαβάζει την σειρά του πίνακα κατά σειρά όχι όλα.

Δίνει μόνο την πρώτη σειρά του πίνακα σας.

Τοποθετώντας τα αποτελέσματα σε μεταβλητή

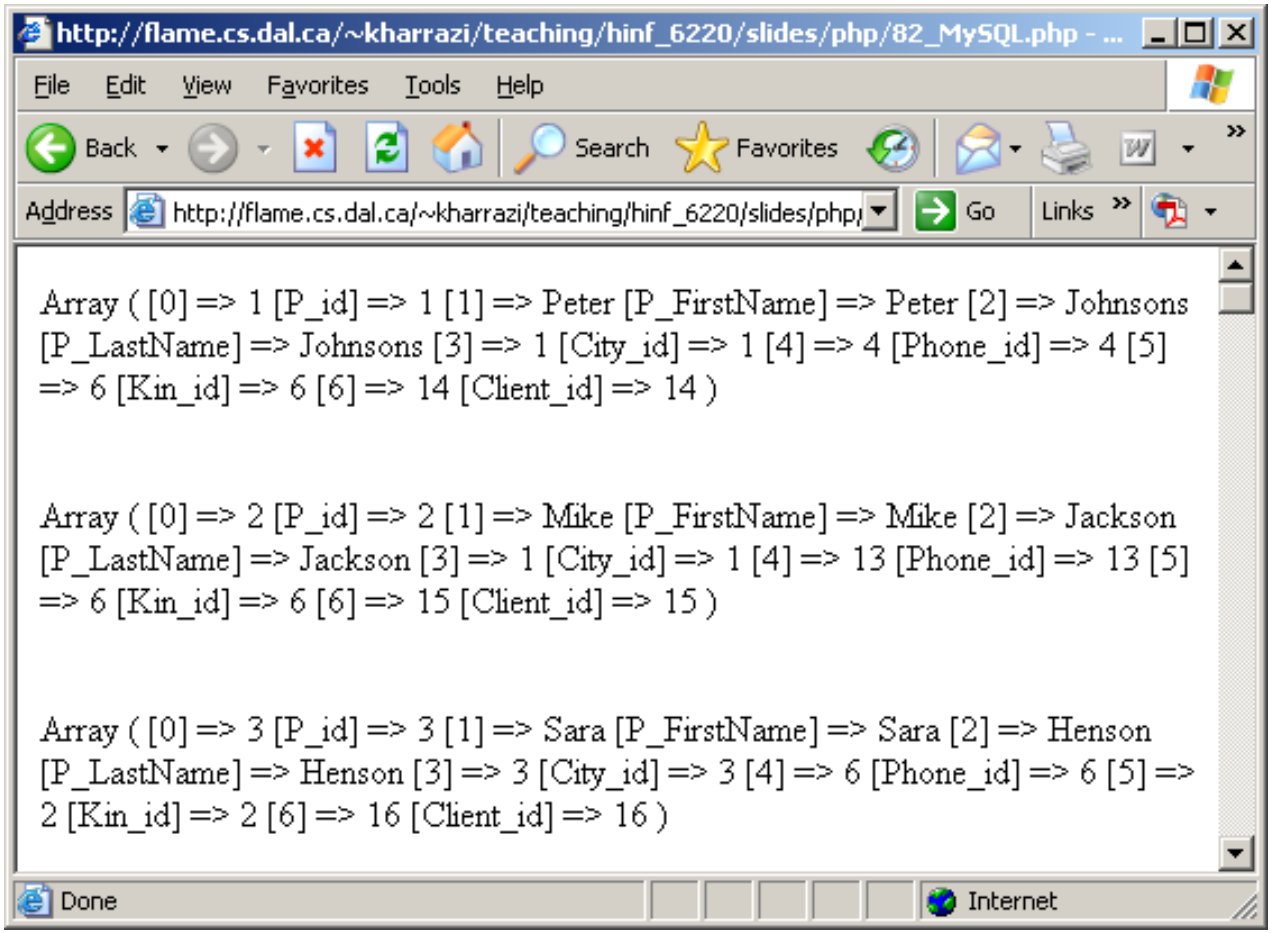
```
<?php
// εδώ εμφανίζονται οι μεταβλητές...
// Δημιουργία σύνδεσης και επιλογή βάσης δεδομένων..
// ... Ερώτημα στη βάση δεδομένων...
// Ανάκτηση αποτελεσμάτων

      while ($row = mysql_fetch_array($result)){
            print_r($row);
            echo '<br>';
      }
      // ... Κλείσιμο της βάσης δεδομένων ... //

?>
```

Λόγω του βρόχου WHILE MYSQL_FETCH_ARRAY διαβάζει όλες τις σειρές στον πίνακα.

Τα αποτελέσματα μπορεί να είναι γραμμές πίνακα

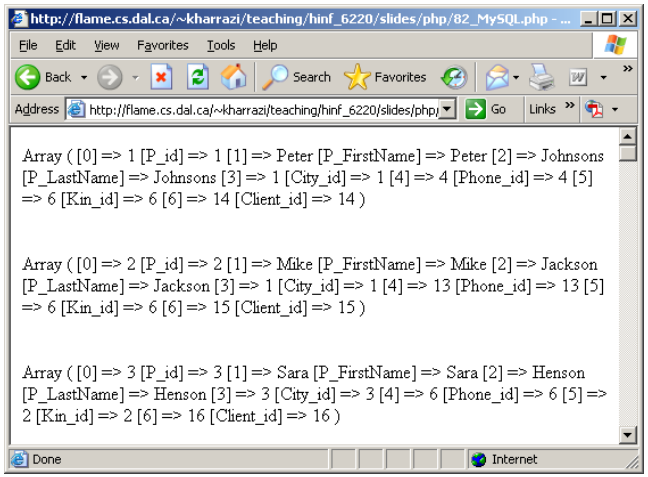


Ασθενής #1 (γραμμή #1)

Ασθενής #2 (γραμμή #2)

Ασθενής #3 (γραμμή #3)

Τα αποτελέσματα μπορεί να είναι γραμμές πίνακα



HTML/WEB

PHP

P_id	P_FirstName	P_LastName	City_id	Phone_id	Kin_id	Client_id
1	Peter	Johnsons	1	4	6	14
2	Mike	Jackson	1	13	6	15
3	Sara	Henson	3	6	2	16

Ασθενής #1 (γραμμή #1)

Ασθενής #2 (γραμμή #2)

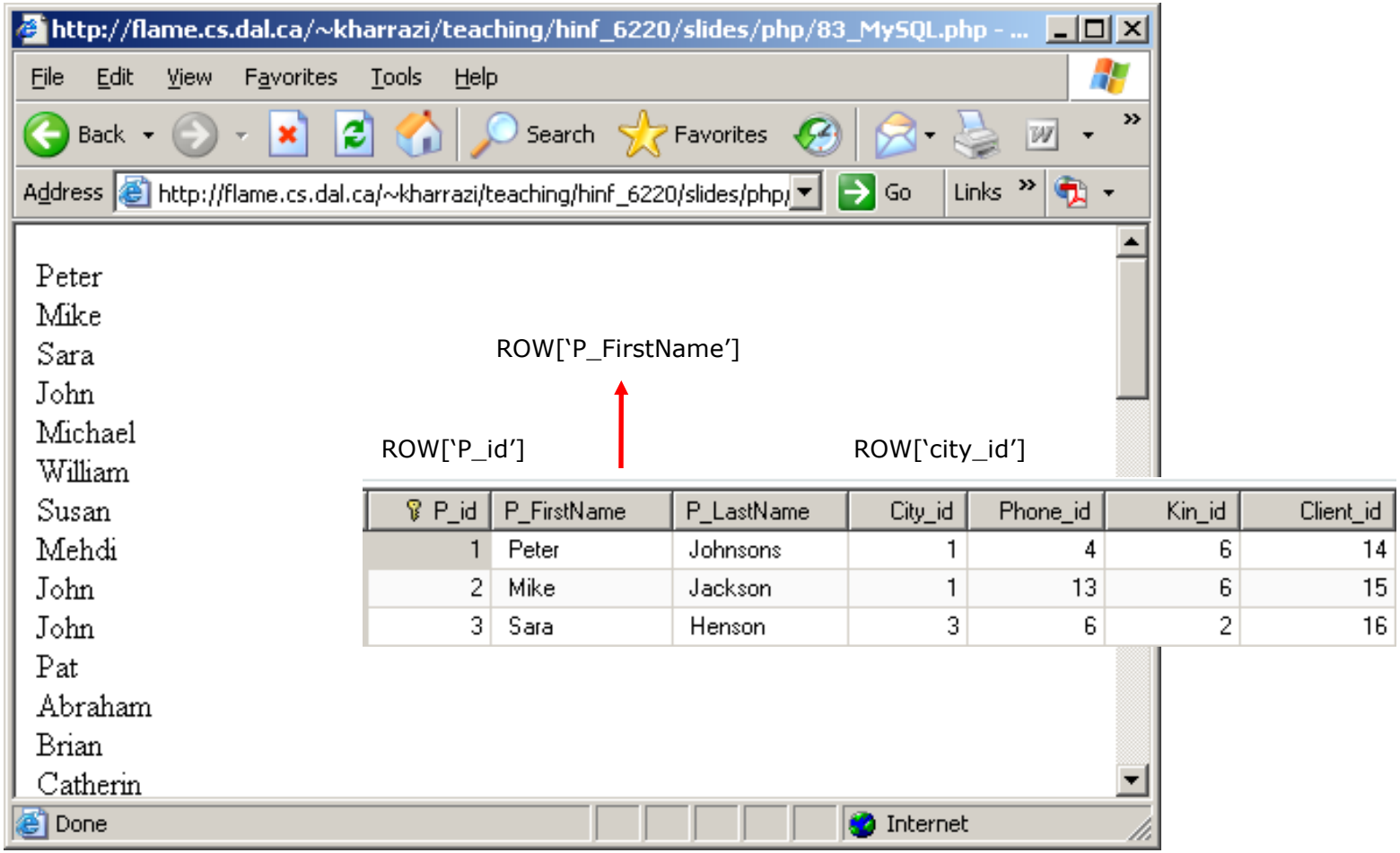
Ασθενής #3 (γραμμή #3)

Πρόσβαση στη στήλη των αποτελεσμάτων

```
<?php
// εδώ εμφανίζονται οι μεταβλητές...
// Δημιουργία σύνδεσης και επιλογή βάσης δεδομένων..
// ... Ερώτημα στη βάση δεδομένων...
// Ανάκτηση αποτελεσμάτων

      while ($row = mysql_fetch_array($result, ,
MYSQL_ASSOC)){
            print_r($row['P_FirstName']);
            echo '<br>';
      }
      // ... Κλείσιμο της βάσης δεδομένων ... //
?>
```

Πρόσβαση στη στήλη των αποτελεσμάτων



P_id	P_FirstName	P_LastName	City_id	Phone_id	Kin_id	Client_id
1	Peter	Johnsons	1	4	6	14
2	Mike	Jackson	1	13	6	15
3	Sara	Henson	3	6	2	16

Το πρωτόκολλο HTTP δεν διατηρεί κατάσταση

- Το πρόβλημα: Το πρωτόκολλο HTTP δεν διατηρεί κατάσταση.

  - Ο διαδικτυακός διακομιστής (web server) παίρνει μόνο διευθύνσεις URL.
  - Δεν έχει ιδέα τι συνέβη πριν.

- Πως μπορούμε να διατηρήσουμε την κατάσταση της εφαρμογής μεταξύ σελίδων;

  - Μεταξύ των σελίδων εφαρμογής.
  - Μεταξύ συνεδριών προγράμματος περιήγησης.

2 επίπεδα διατήρησης της κατάστασης

- Μακροπρόθεσμη κατάσταση εφαρμογής

  - Αποτελεσματικά διαρκούν για πάντα.
  - Λογαριασμοί χρηστών, αρχεία.
  - Χρήση βάσης δεδομένων

- Συνεδρία χρήστη

  - Μεταξύ διαδικτυακών σελίδων.
  - Ενιαία δραστηριότητα: ψωνίζοντας, συμπληρώνοντας μια φόρμα.
  - Ο χρήστης ίσως κλείσει ή επανεκκινήσει το πρόγραμμα περιήγησης (browser).
  - Μπορούν να αποθηκευτούν στο διακομιστή ή σε ένα πρόγραμμα περιήγησης

Τεχνολογίες για τη διατήρηση της κατάστασης

- Παράμετροι συμβολοσειράς URL
- Κούκις / Μπισκότα (Cookies)
- PHP διαχείριση συνεδριών

Διατήρηση της κατάστασης μέσω URL

- Σενάριο: Η χρήστες συνδέονται στη σελίδα εισόδου (login page). Πως μπορούμε να διατηρήσουμε την ταυτότητα των χρηστών σε άλλες σελίδες;

- Λύση: Περνάμε το όνομα χρήστη (username) σαν συμβολοσειρά URL ή από παράμετρο.

```
<a href="link.html?userID=<?php echo
$_REQUEST['userID']; ?>"> Link text </a>
```

- Σημειώστε ότι αυτό χρειάζεται για να προστεθεί σε όλους τους συνδέσμους (links)

Διατήρηση της κατάστασης από URL στις φόρμες

Οι τιμές μπορούν να περαστούν σαν μια κρυμμένη παράμετρος εισόδου.

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

    <input type="hidden" name="userID"
    value="<?php echo $_REQUEST["userID"];
    ?>"/>

    …

    <input type="submit" value="Next
page"/>
</form>
```

Τι στοιχεία θα πρέπει να διατηρούμε;

- Σε τι πρέπει να επιμείνουμε για να είμαστε σίγουροι ότι ένας χρήστης έχει συνδεθεί;
  - όνομα χρήστη: αλλά μπορεί να αλλοιωθεί, αντιγραφεί.
  - κωδικός πρόσβασης: δεν πρέπει να περάσει τριγύρω σε άλλους χρήστες.

- Κάθε τιμή που περνάει σε μία συμβολοσειρά URL μπορεί να έχει παραβιαστεί από έναν χρήστη.

- Η είσοδος μπορεί να δημιουργήσει ένα σημείο το οποίο είναι δύσκολο να γίνει απομίμηση.
  - Τυχαία ID παράγονται από τον διακομιστή.
    - Αποθηκεύονται στη βάση δεδομένων και επαληθεύουν κάθε σελίδα.
  - Χρονικό σημείο
    - Απορρίπτονται συνεδρίες μετά από μία χρονική περίοδο.

Πλεονεκτήματα και μειονεκτήματα της χρήσης URL για διατήρηση κατάστασης

- Πλεονεκτήματα
  - Δουλεύει παντού.
  - Τεχνικά απλή.
  - Μπορεί να είναι προσβάσιμη σε πλευρά πελάτη (client-side) [JS] και σε πλευρά διακομιστή (server-side) [PHP].

- Μειονεκτήματα
  - Μπορεί να προβληθεί ή να μεταβληθεί από χρήστη.
  - Πρέπει να ξαναγράψει όλους τους συνδέσμους ή να χρησιμοποιήσει φόρμες.
  - Χάνεται αν ο χρήστης κλείσει το παράθυρο του προγράμματος περιήγησης.

Υπάρχουν άλλοι 2 τρόποι για τη διατήρηση κατάστασης

- Sessions (Συνοδοί / Συνεδρίες)
- Cookies (Κούκις / Μπισκότα)

Διαχείριση των συνοδών (sessions)

- Η PHP παρέχει ένα στρώμα αφαίρεσης για τη διαχείριση της περιόδου λειτουργίας του προγράμματος περιήγησης (browser session).

- Μπορεί να συνδέσει τις PHP μεταβλητές (περιλαμβανομένων απλών αντικειμένων) σε μία συνεδρία χρήστη.
  - Μπορεί να ανακτήσει μεταβλητές σε κάθε σελίδα.
  - Οι μεταβλητές διαρκούν μέχρι να κλείσει το πρόγραμμα περιήγησης.

- «Κάτω από την κουκούλα» (Underneath the hood)
  - Η PHP δημιουργεί ένα τυχαίο ID.
  - Στέλνει το ID στο χρήστη σαν cookie (κούκι)
  - Η PHP αποθηκεύει το ID και τις μεταβλητές σε έγγραφο κειμένου σε πλευρά διακομιστή (in server-side).

Συνεδρίες

- Στο διαδίκτυο ο διακομιστής (web server) δεν γνωρίζει ποιος είναι ο κάθε χρήστης.
- Υπάρχουν ορισμένες περιπτώσεις που πρέπει να μεταφέρουμε δεδομένα από την μία ιστοσελίδα στην επόμενη. Π.χ όνομα χρήστη (username), τι έχει επιλέξει ότι θέλει να αγοράσει σε ένα ηλεκτρονικό κατάστημα κ.α.
- Για αυτόν τον λόγο υπάρχει ο μηχανισμός των συνεδριών.
- Οι συνεδρίες δημιουργούν ένα μοναδικό κωδικό (Unique, IDentifier) για κάθε επισκέπτη ώστε να σωθούν συγκεκριμένες μεταβλητές.

Συνεδρίες

Ο πελάτης διατηρεί μόνο τον μοναδικό αριθμό που αποθηκεύεται στη μεταβλητή PHPSESSID.
Είναι η μόνη ορατή πληροφορία από την πλευρά του πελάτη.

Ο διακομιστής αποθηκεύει όλες τις μεταβλητές σε δικό του χώρο (συνήθως σε αρχεία) που συνδέονται με αυτόν τον αριθμό.

Όποιος δηλαδή έχει πρόσβαση σε αυτόν τον αριθμό μπορεί να έχει πρόσβαση στις μεταβλητές που έχει αποθηκευμένες ο διακομιστής.

Συνεδρίες

Η πιο συνηθισμένη χρήση του ελέγχου των συνεδριώνν λειτουργίας είναι **να παρακολουθεί τους χρήστες** αφού πιστοποιηθούν μέσω ενός μηχανισμού σύνδεσης.

Δημιουργώντας τους συνοδούς (sessions)

- Χρησιμοποιήστε τη συνάρτηση session_start() για να ξεκινήσει η συνεδρία.
  - Αν δεν υπάρχουν οι αποθηκευμένες τιμές, να τις ανακτήσετε.

- Ορίστε και πάρτε τις τιμές σε $_SESSION πίνακα.

```
//φορτώστε το συνοδό
session_start() ;
//ελέγξτε να βλέπετε αν οι μεταβλητές “color” είναι ορισμένες
//αν δεν είναι, να τις ορίσετε
if(!isset($_SESSION[“color”]) {
        $_SESSION[“color”] = “vermillion” ;
}
```

Συνεδρίες / Σύνοδοι (sessions)

- Για να αρχίσουμε ένα σύνοδο πρέπει να κληθεί η εντολή session_start πριν τη δήλωση της ετικέτας <html>.

- Παράδειγμα:

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

<html>
        <body>

        </body>
</html>
```

Συνεδρίες / Σύνοδοι (sessions)

- Η υποστήριξη συνόδων στην PHP αποτελείται από έναν τρόπο διατήρησης ορισμένων δεδομένων σε επόμενες προσβάσεις. Αυτό σας δίνει τη δυνατότητα να δημιουργήσετε πιο προσαρμοσμένες εφαρμογές και να αυξήσετε την ελκυστικότητα του ιστοτόπου σας.

- Ένας επισκέπτης που έχει πρόσβαση στον ιστότοπό σας διαθέτει ένα μοναδικό αναγνωριστικό, το αποκαλούμενο αναγνωριστικό περιόδου σύνδεσης. Αυτό είτε αποθηκεύεται σε ένα cookie από την πλευρά του χρήστη ή διαδίδεται στη διεύθυνση URL.

Ορισμός μιας περιόδου λειτουργίας

```
session_start();
$_SESSION['variable_name']=value;
session_destroy();
```

Η συνάρτηση session_start() πρέπει *να* εμφανίζεται *πριν* την <html> ετικέτα.

Συνεδρίες / Σύνοδοι (Sessions)

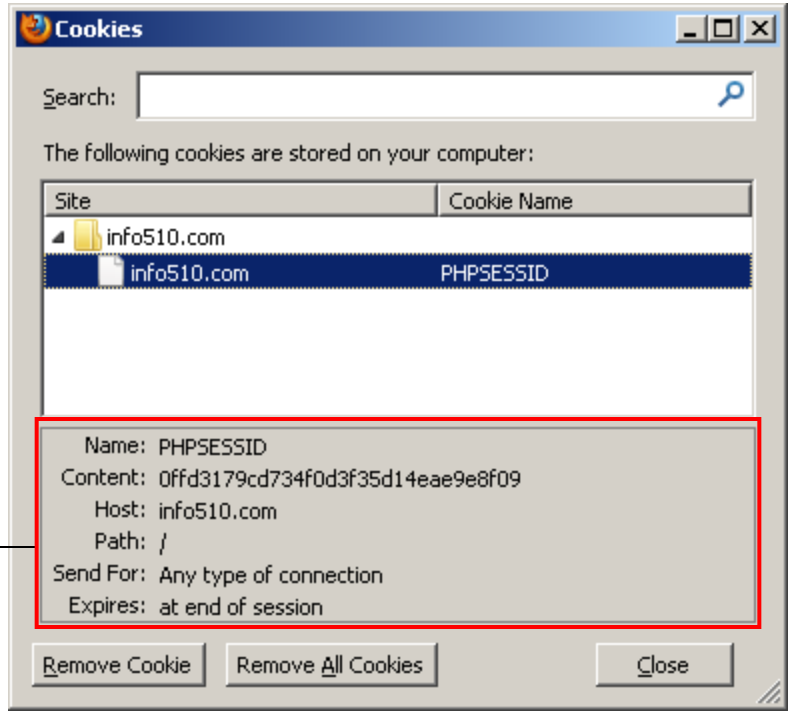
- Παράδειγμα

```
<?php
        session_start();
        //Σώζει δεδομένα
        $S_SESSION['var']=1;
?>

<html>
        <body>
                <?php
                        //Ανάκτηση δεδομένων
                        echo "Δεδομένα"=.$_SESSION['var'];
                ?>
        </body>
</html>
```

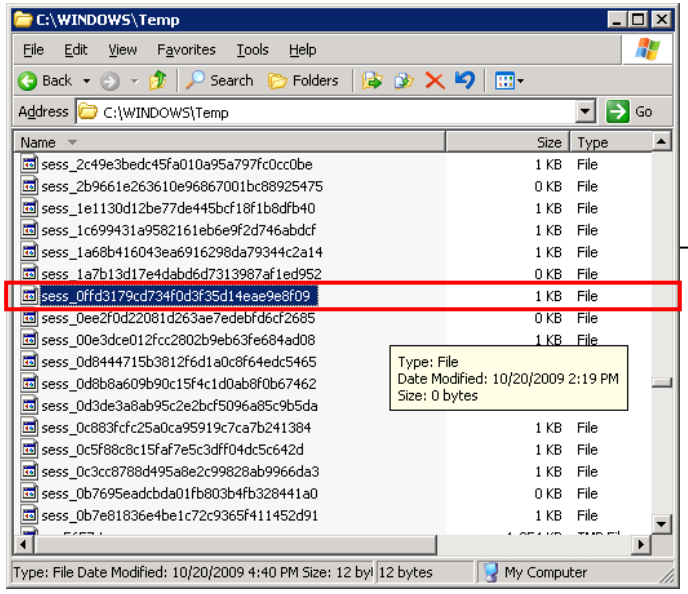
Τι είναι οι συνεδρίες (1/2) ;

```
<?php
session_start();
if (isset($_SESSION["counter"]))
{
        $_SESSION["counter"] = $_SESSION["counter"] + 1;
}else{
        $_SESSION["counter"] = 1;
}
echo "You have visited us " .
$_SESSION["counter"] . " times!";
?>
```



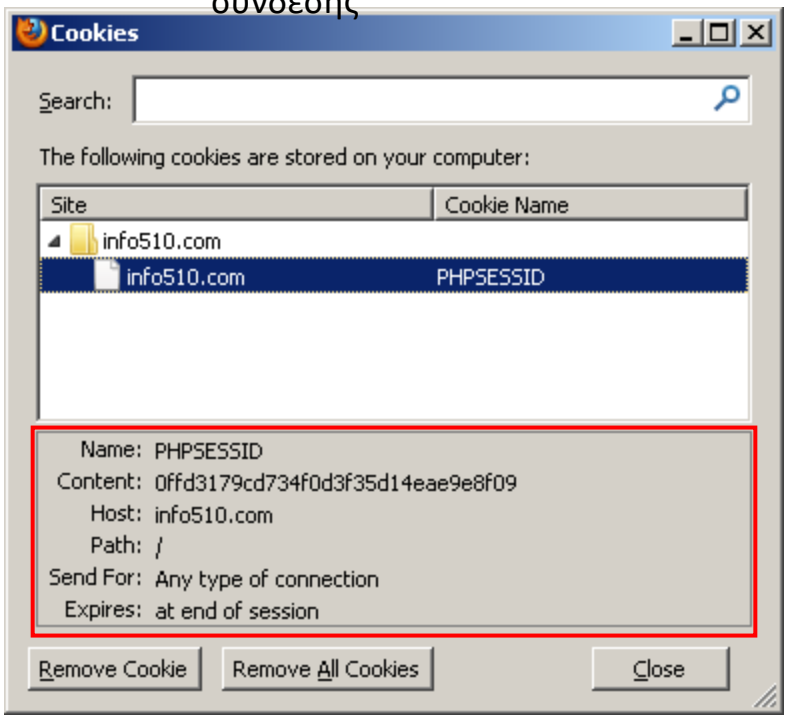
Κανένα cookie σχετικά με το 'counter'
Μόνο το cookie για την PHP συνεδρία

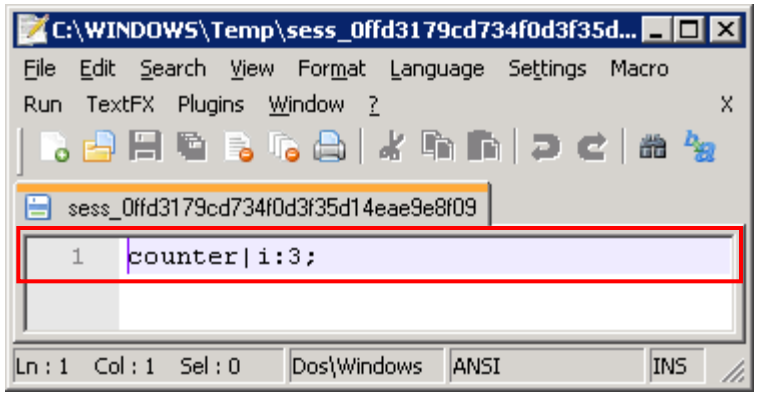
Τι είναι οι συνεδρίες (2/2) ;



Αρχεία διακομιστή / σύνδεσης

Αναγνωριστικό πελάτη / περιόδου σύνδεσης





Περιεχόμενο Συνεδρίας

Συνεδρίες / Σύνοδοι (Sessions)

- Για να κλείσουμε/καταστρέψουμε μία συνεδρία (session) χρησιμοποιούμε την εντολή session_destroy, η οποία καταστρέφει όλα τα δεδομένα τα οποία έχουν σωθεί στο session. Η εντολή αυτή χρησιμοποιείται συνήθως όταν ο χρήστης κάνει αποσύνδεση (logout).

- Μπορούμε επίσης να σβήσουμε μόνο μία μεταβλητή με την εντολή unset.

- Παράδειγμα:

```
<?php
        unset($_SESSION['var']);
?>
```

Καταστροφή των συνεδριών (session)

Χρησιμοποιείστε τη συνάρτηση session_start() για να ξεκινήσετε μία συνεδρία.

Παράδειγμα:

```
<?php
        //ξεκινήστε μια συνεδρία
        session_start() ;

        //υπόλοιπο κομμάτι κώδικα

        //καταστροφή συνεδρίας
        session_destroy() ;
?>
```

Ασφάλεια στα Session Fixation: 2 βήματα

```
<?php
        session_start();

        if(!isset($_SESSION['initiated']))
        {
                session_regenerate_id() ;
                $_SESSION['initiated'] = true ;
        }
?>
```

Α) Είναι καλύτερα να χρησιμοποιούμε μια μεταβλητή μέσα στο $_SESSION για να εμποδίζουμε τους εισβολείς (hackers) να κάνει Session Fixation (δηλαδή να δώσουν στον πελάτη ένα σταθερό αριθμό συνεδρίας (Session) και να το χρησιμοποιήσουν αυτοί στη συνέχεια).

Β) Κάθε φορά που υπάρχει αλλαγή δικαιωμάτων πρέπει να γίνεται regenerate session.

Ασφάλεια στα Session Hijacking: (1/2) Απλός τρόπος

```
<?php

   session_start();

   if(!isset($_SESSION['HTTP_USER_AGENT']))
   {
         if($_SESSION['HTTP_USER_AGENT'] !=
md5($_SERVER['HTTP_USER_AGENT']))
         {
                  /*προτροπή για κωδικό πρόσβασης
                  exit;
         }
   }
   else
{
         $_SESSION['HTTP_USER_AGENT']=md5($_SERVER['HTTP_USER_AGENT'])) ;
}

?>
```

```
GET / HTTP/1.1
Host: example.org
User-Agent: Mozilla/5.0 Gecko
Accept: text/xml, image/png, image/jpeg, image/gif, */*
Cookie: PHPSESSID=1234
```

Διατήρηση της συμβολοσειράς (string) που καθορίζει το πρόγραμμα περιήγησης (browser) στο GET ερώτημα

Ασφάλεια στα Session Hijacking: (2/2) Σύνθετος τρόπος

```
<?php

    $string=$_SERVER['HTTP_USER_AGENT'];
    $string.='SHIFLETT';

    $fingerprint = md5($string);
?>
```

Ο απλός τρόπος μπορεί να παρακαμφθεί αν ο εισβολέας (hacker) δοκιμάσει αρκετά strings (συμβολοσειρές) από User Agent Tags (ετικέτες πράκτορα χρήστη).
Προσθέτοντας μια κρυφή λέξη αυτό δε μπορεί να γίνει.

Εισαγωγή στα ‘μπισκότα’(cookies)

- Ένα ‘μπισκότο’ / κούκι (cookie) είναι ένα κομμάτι κειμένου το οποίο είναι αποθηκευμένο από το πρόγραμμα περιήγησης (browser) μεταξύ των συνεδριών.
  - Ορίστηκε στο πρόγραμμα περιήγησης από ένα ειδικό διαδικτυακό διακομιστή (web server).
  - Το πρόγραμμα περιήγησης στέλνει πληροφορίες πίσω, μόνο σε αυτόν τον διακομιστή (server).

- Κοινώς χρησιμοποιείται για να αποθηκεύσει πληροφορίες χρήστη.
  - Ή για να επισημάνει μεμονωμένα προγράμματα περιήγησης και να τα αναγνωρίσει αργότερα.

- Αποτελείται από τέσσερα στοιχεία
  - πηγαία ιστοσελίδα
  - όνομα χαρακτηριστικού
  - τιμή χαρακτηριστικού
  - ημερομηνία λήξης (από προεπιλογή, όταν κλείνει το πρόγραμμα περιήγησης (browser).

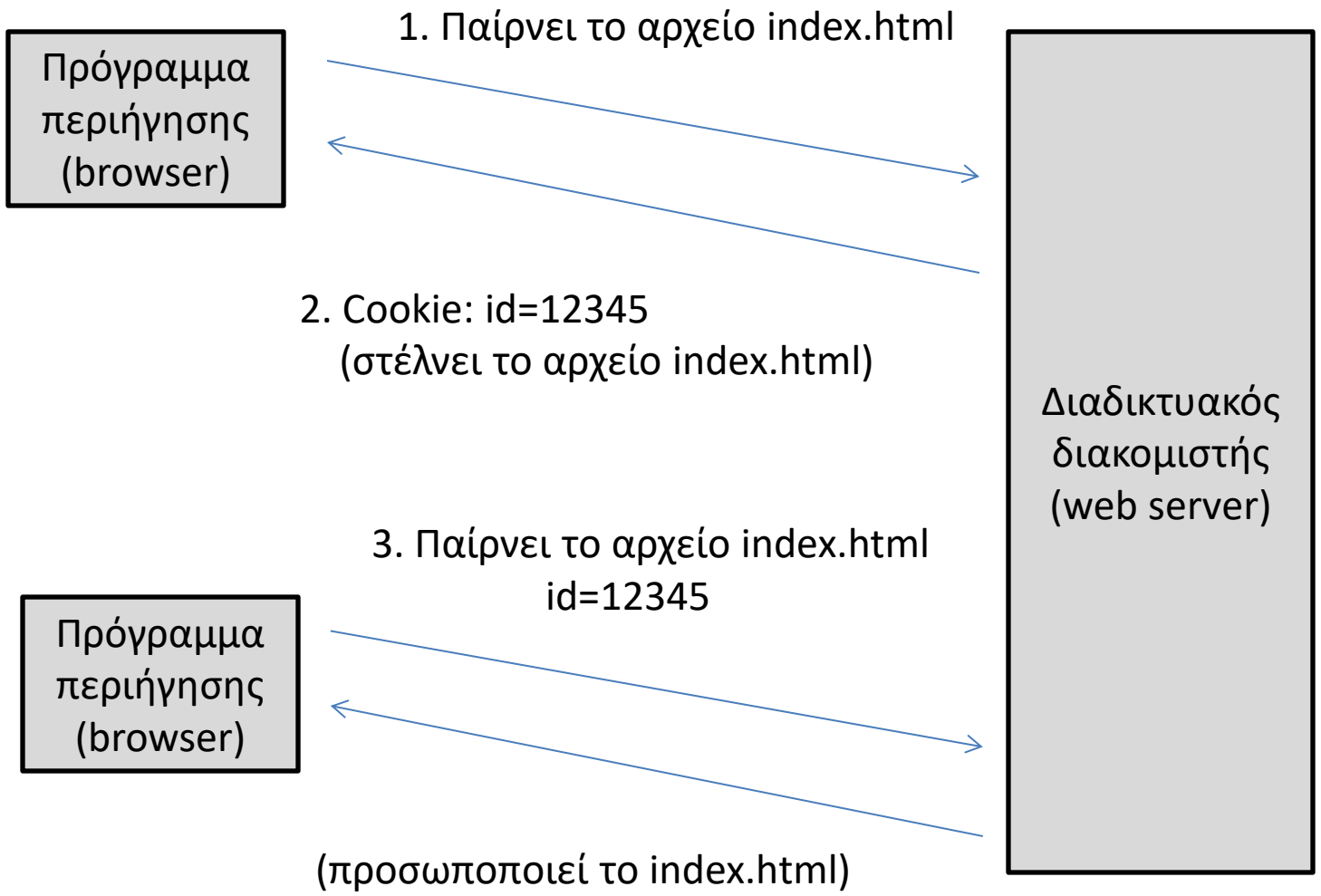
Χρήσεις των Cookies

- Αποθηκεύουν δεδομένα συνεδριών
  - Πληροφορίες σύνδεσης, σελίδες που έχουν επισκεφθεί, καλάθι αγορών.

- Παρακολουθούν τους χρήστες που επισκέπτονται την ιστοσελίδα σας
  - Εκχωρεί σε αυτούς ένα ID, το οποίο διαβάζει αργότερα.
  - Ακόμη και χωρίς να εγγραφούν.

- Επιτρέπει στους χρήστες να προσωποποιήσουν τη δική τους εμπειρία.
  - Ορίζει τις ρυθμίσεις της ιστοσελίδας, αποθηκεύει στο πρόγραμμα περιήγησης (browser).

Πως λειτουργούν τα cookies



Πως μεταδίδονται τα cookies

- Ο διακομιστής (server) στο domain.com στέλνει cookie σε HTTP απάντηση.

```
HTTP/1.0 200
Content-Length: 1276
Content-Type: text/html
Date: Tue, 06 Nov 2001 04:12:49 GMT
Expires: Tue, 06 Nov 2001 04:12:59 GMT
Set-Cookie: id=12345
<html> … </html>
```

- Το πρόγραμμα περιήγησης αποθηκεύει το cookie σε ένα τοπικό σύστημα.
- Όταν το πρόγραμμα περιήγησης επισκεφθεί ξανά το domain, στέλνει το cookie πίσω.

```
GET /index.php HTTP/1.0
Connection: Keep-Alive
Cookie: id=12345
Host: www.test.com
Refer: http://www.test.com
```

Cookies

- Τα Cookies συνήθως χρησιμοποιούνται για να γίνει η αναγνώριση ενός χρήστη από μία ιστοσελίδα.
- Τα Cookies είναι ένα μικρό αρχείο το οποίο στέλνετε από το διαδικτυακό διακομιστή (web server) στον υπολογιστή του χρήστη και περιέχει πληροφορίες σχετικά με τον επισκέπτη.
- Κάθε φορά που ο ίδιος χρήστης επισκεφτεί ξανά την ίδια ιστοσελίδα, ο υπολογιστής του χρήστη (client) θα στείλει στον διαδικτυακό διακομιστή (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.

Cookies

- Ένα cookie χρησιμοποιείται συχνά για τον εντοπισμό ενός χρήστη και είναι ένα μικρό αρχείο που ο διακομιστής ενσωματώνει στον υπολογιστή του χρήστη. Κάθε φορά που ο ίδιος υπολογιστής ζητά μια σελίδα με ένα πρόγραμμα περιήγησης, θα στείλει και το cookie. Με την PHP, μπορείτε να δημιουργήσετε και να ανακτήσετε τιμές cookie.

- **Ρύθμιση ενός cookie**

setcookie (name, value, expire, path, domain);

```
<?php
        setcookie ("user", "Peter Johnson", time()+3600);
?>
```

Cookies

- Ρυθμίστε ένα cookie ως εξής:

```
setcookie(name, value, [expiration, path, domain, secureOnly])
```

- Ανακτήστε cookies με $_COOKIES[‘cookiename’]

```
//αν ένα cookie δεν έχει οριστεί

if(!isset($_COOKIES[“color”]){
        //ρύθμισε τη διάρκεια του μέχρι 3 μέρες
        setcookie(“color”, “green”, time() +3*60*60*24);
}
```

Πλεονεκτήματα και μειονεκτήματα των Cookies

- Πλεονεκτήματα
  - Εύκολα
  - Μπορούν να χρησιμοποιηθούν και από την πλευρά του πελάτη και από του διακομιστή (client- and server- side).
  - Ένας χρήστης που γνωρίζει έχει τον έλεγχο σε ότι είναι αποθηκευμένο.
  - Επιμονή στις συνεδρίες του προγράμματος περιήγησης (browser).

- Μειονεκτήματα:
  - Μπορούν να αποθηκεύσουν μόνο κείμενο.
  - Τα cookies μπορεί να έχουν μεγαλύτερο από ένα ορισμένο μέγεθος (~4k).
  - Μπορούν μόνο να διαβαστούν από την ιστοσελίδα που τα όρισε.
  - Μερικοί χρήστες σβήνουν τα cookies στα δικά τους προγράμματα περιήγησης.
  - Οι χρήστες ίσως δεν αντιλαμβάνονται τα cookies που έχουν οριστεί.

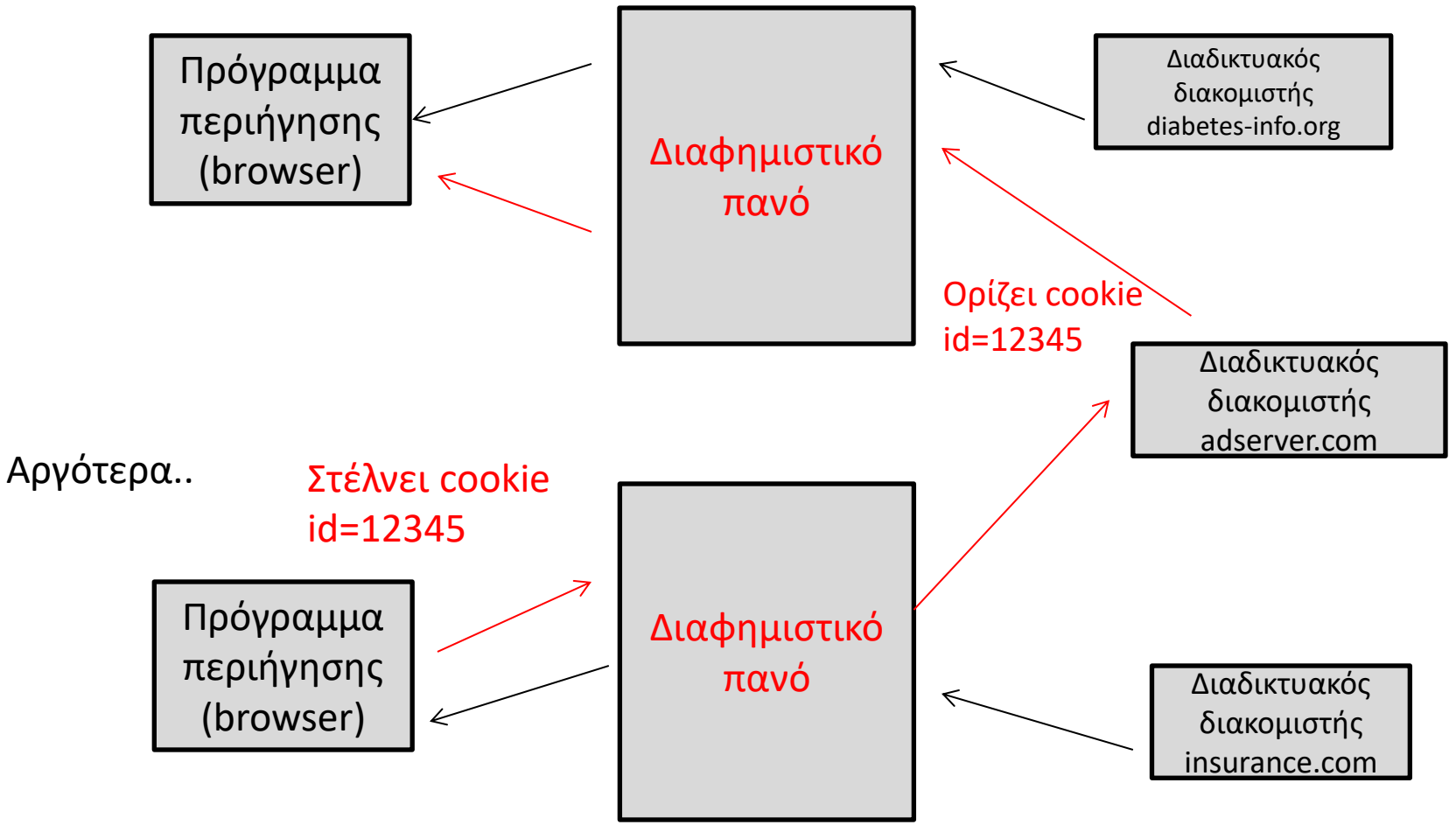
Cookies πολλαπλών ιστοσελίδων

- Γενικά, τα cookies ίσως μπορούν να διαβαστούν μόνο από την ιστοσελίδα που τα όρισε.

- Ωστόσο, μερικές φορές ένας διαδικτυακός διακομιστής (web server) παρέχει περιεχόμενο σε πολλαπλές ιστοσελίδες.
  - Διαφημιστικά πανό από διαφημιστικά πρακτορεία.

- Αυτά τα τρίτα ή “τρακαδόροι” (“trackers”) cookies μπορούν να εντοπίσουν ένα χρήστη μέσω διαφόρων ιστοσελίδων.

Cookies πολλαπλών ιστοσελίδων - Παράδειγμα



Cookies

- Τα cookies ορίζονται πριν τη δήλωση της ετικέτας <html>.

- setcookie(<όνομα>, <τιμή>, <λήξη>)
  - Ορίζει ένα cookie. Το cookie καταστρέφεται ανάλογα με τη λήξη του.
  - Για να καταστρέψουμε ένα cookie μπορούμε να το ορίσουμε χρησιμοποιώντας αρνητική λήξη.

Παράδειγμα:

```
<?php
        setcookie("username", "Ελένη", time()*3600);
?>
<html>
        <body>
        </body>
</html>
```

Ανάγνωση / Διαγραφή cookies

- Ανάκτηση ενός cookie

Η μεταβλητή PHP $_COOKIE χρησιμοποιείται για να ανακτήσει ένα cookie.

```
<?php
        echo $_COOKIE[“user”]; //Εκτυπώνει ένα cookie
        print_r ($_COOKIE); // Ένας τρόπος να δούμε όλα τα cookies
?>
```

- Διαγραφή ενός cookie

```
<?php
        //ορίζει την ημερομηνία λήξης για μία ώρα μετά
        setcookie(“user”, “”, time()*3600);
?>
```

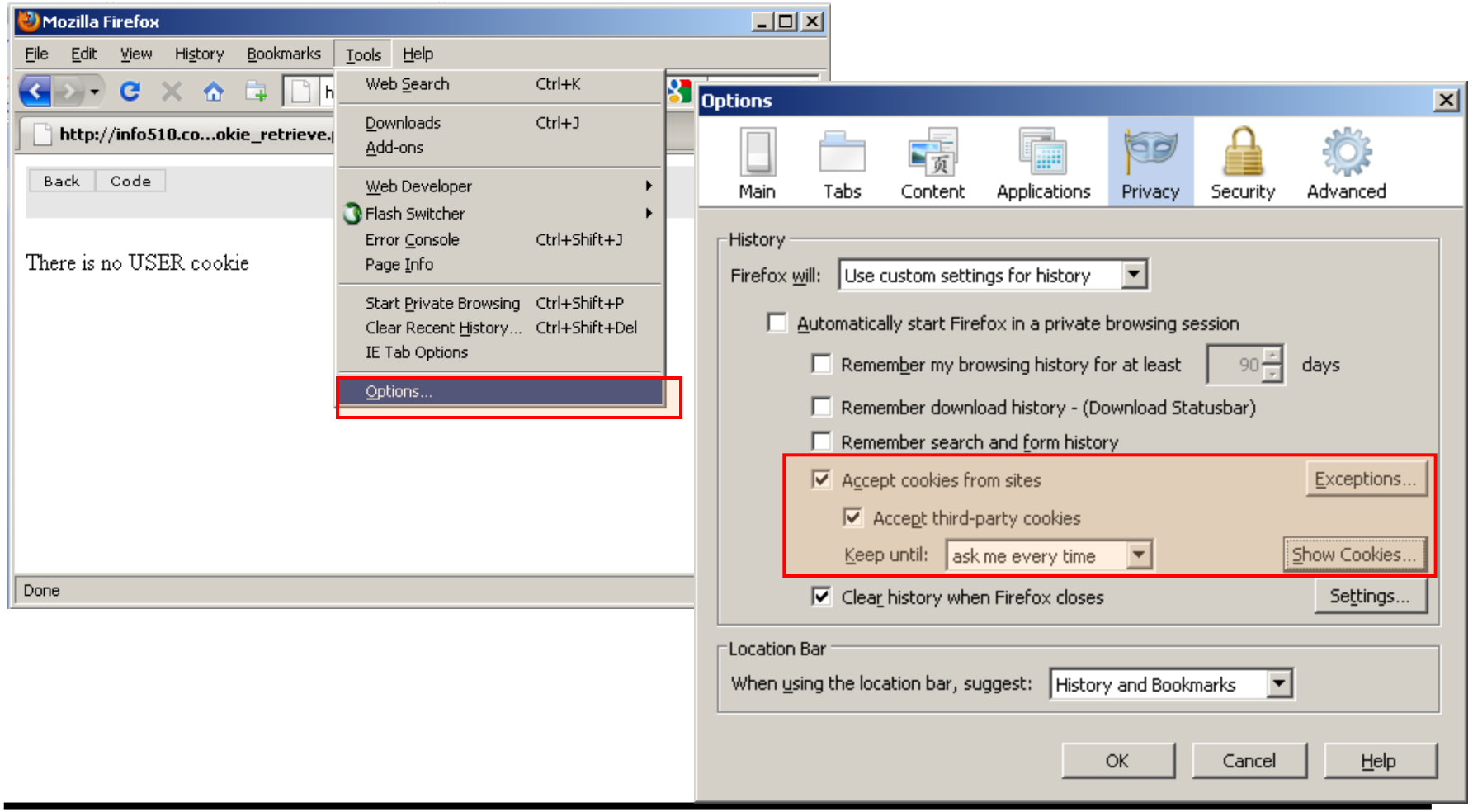
Cookies

Τα δεδομένα των cookies τα διαβάζουμε με την μεταβλητή $_COOKIE.

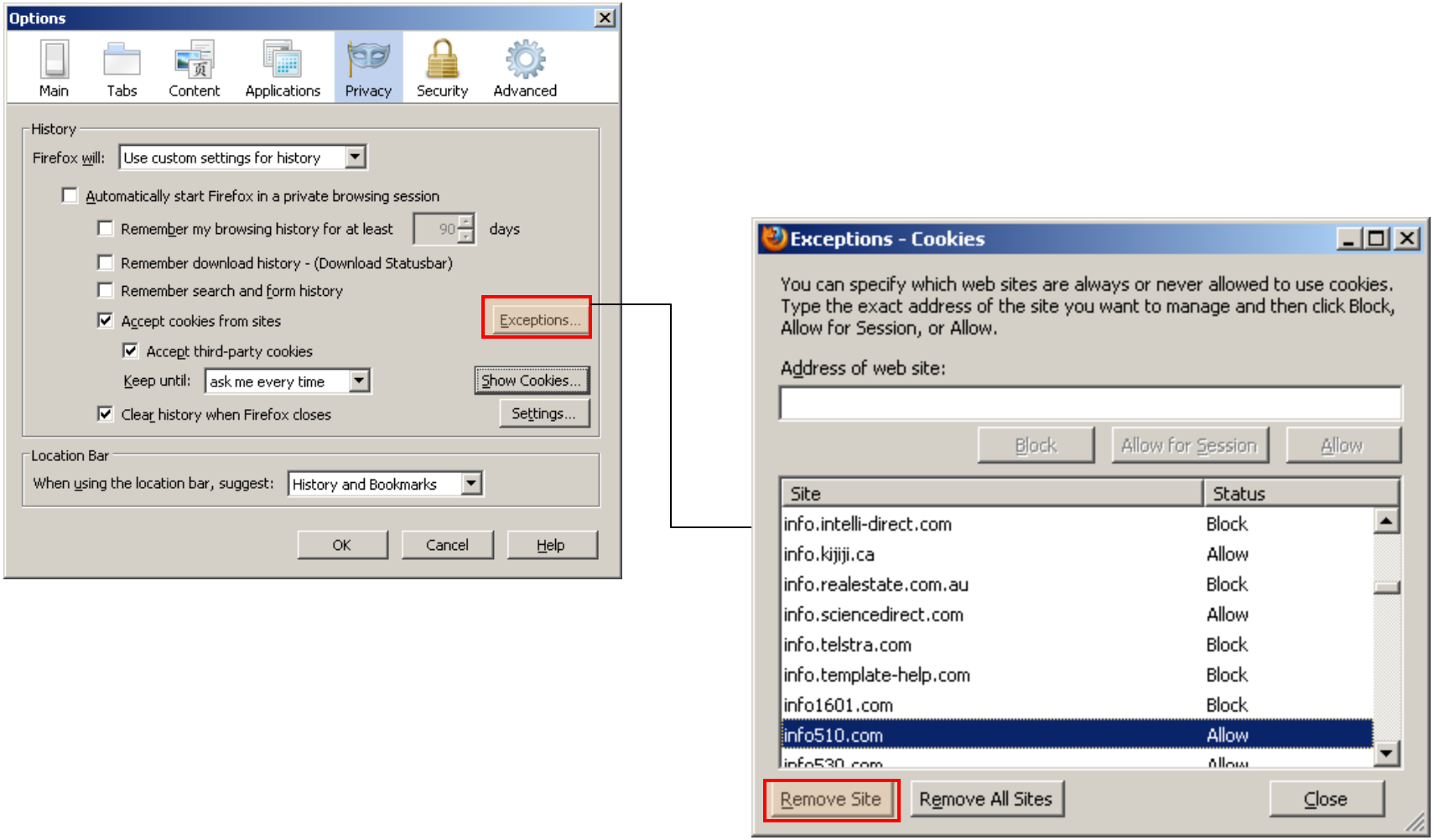
Παράδειγμα:

```
<html>
<body>
<?php
        if(isset($_SESSION[“username”]))
           echo “Γεια σου ”, $_COOKIE[“username”]. “!<br/>”;
        else
           echo “Γεια σου άγνωστε χρήστη!<br/>”;
?>
</body>
</html>
```

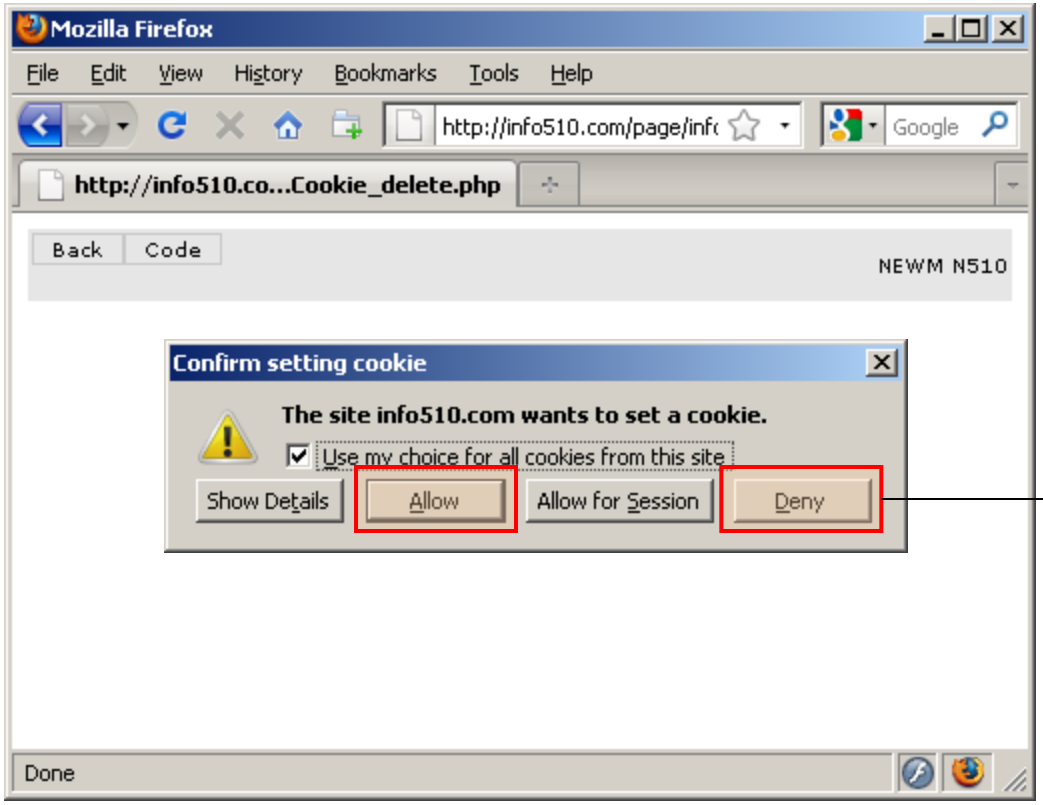
Το πρόγραμμα περιήγησης μπορεί να ρυθμιστεί να αποδέχεται ή όχι τα cookies



Μπορούμε να τοποθετήσουμε εξαιρέσεις για cookies



Ο χρήστης μπορεί να επιτρέψει συγκεκριμένα cookies

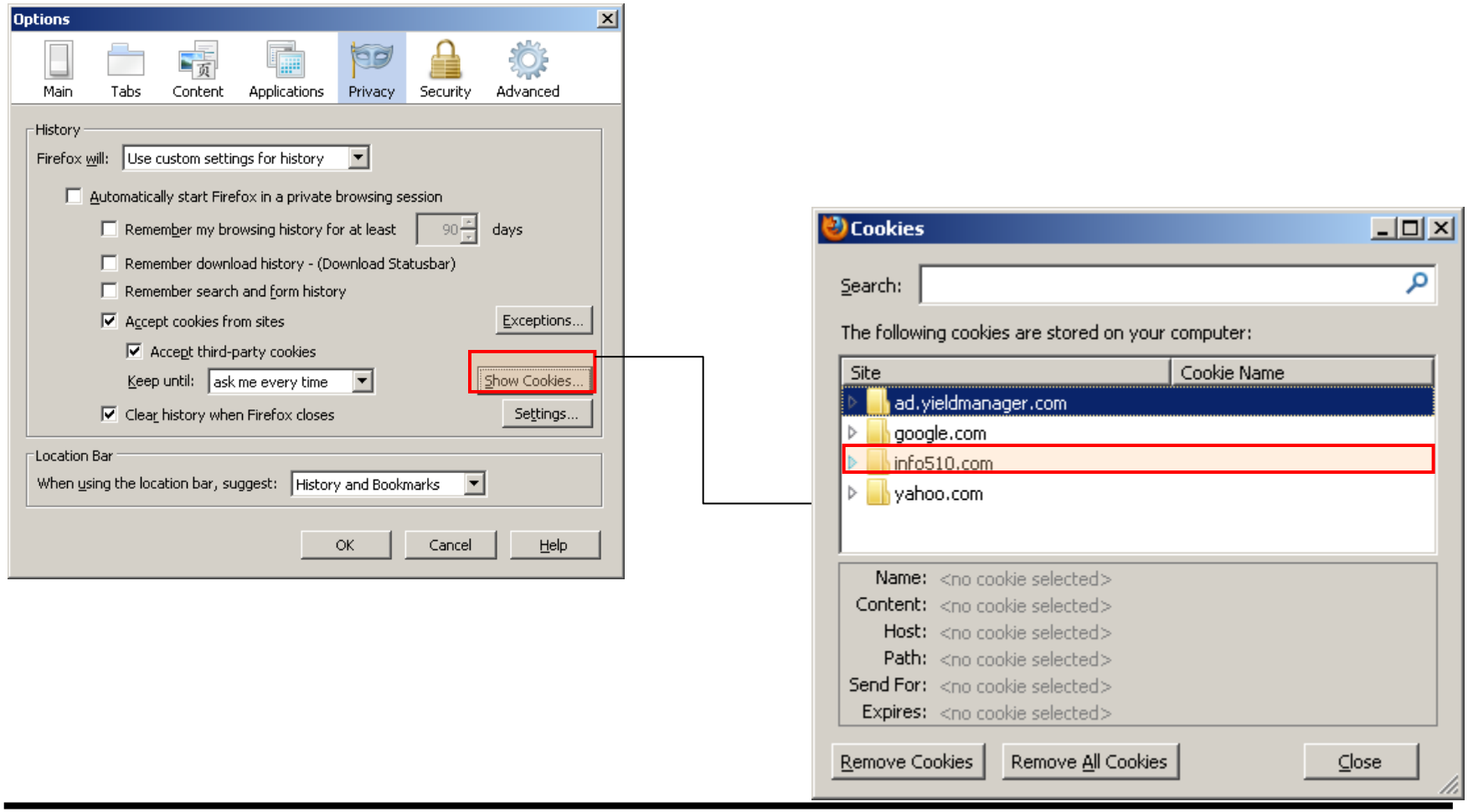


Αν απορριφθεί, τότε το cookie δεν θα λειτουργεί πια.

Εάν επιτρέπεται αλλά δεν υπάρχει σημάδι ελέγχου τότε θα το ζητάει κάθε φορά.

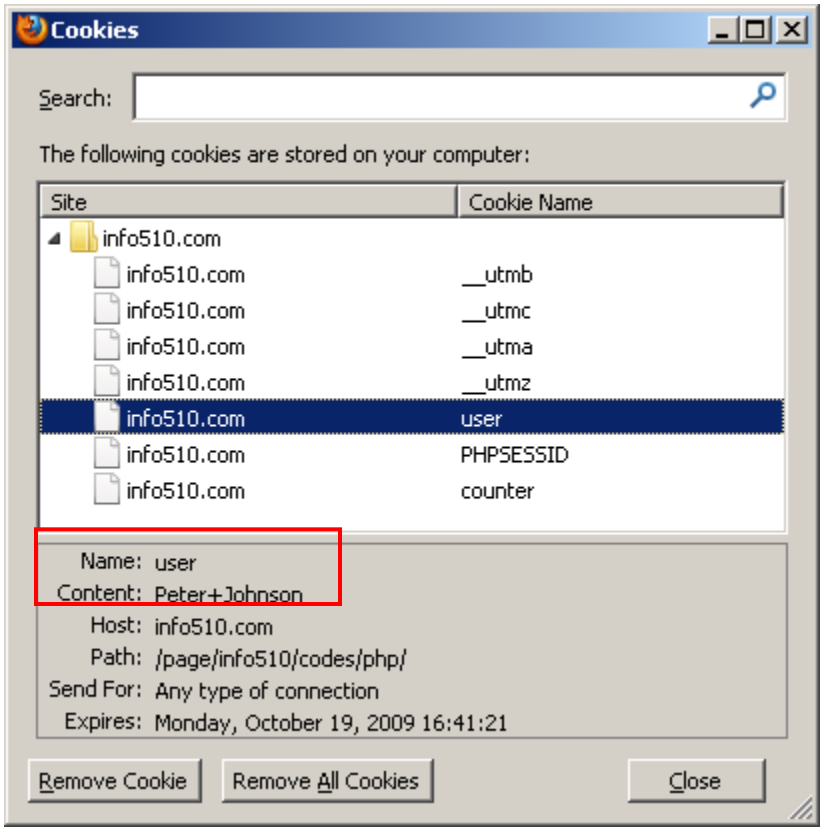
```
<?php
      setcookie ("user", "Peter Johnson", time()+3600);
?>
```

Μπορούμε να δούμε τα cookies (1/2)

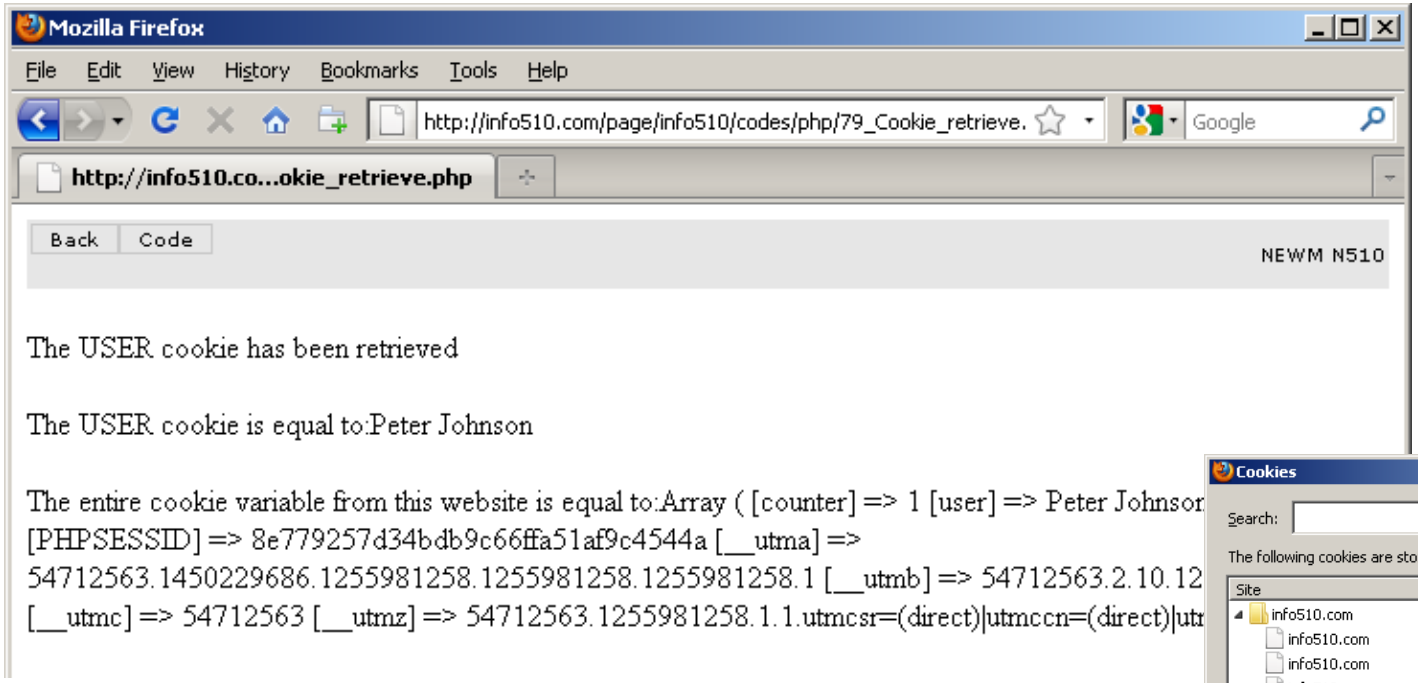


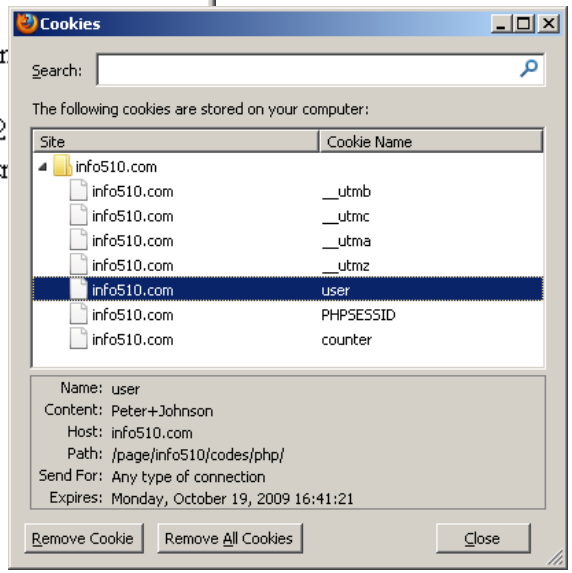
Μπορούμε να δούμε τα cookies (2/2)

```
<?php
        setcookie ("user", "Peter Johnson", time()+3600);
?>
```



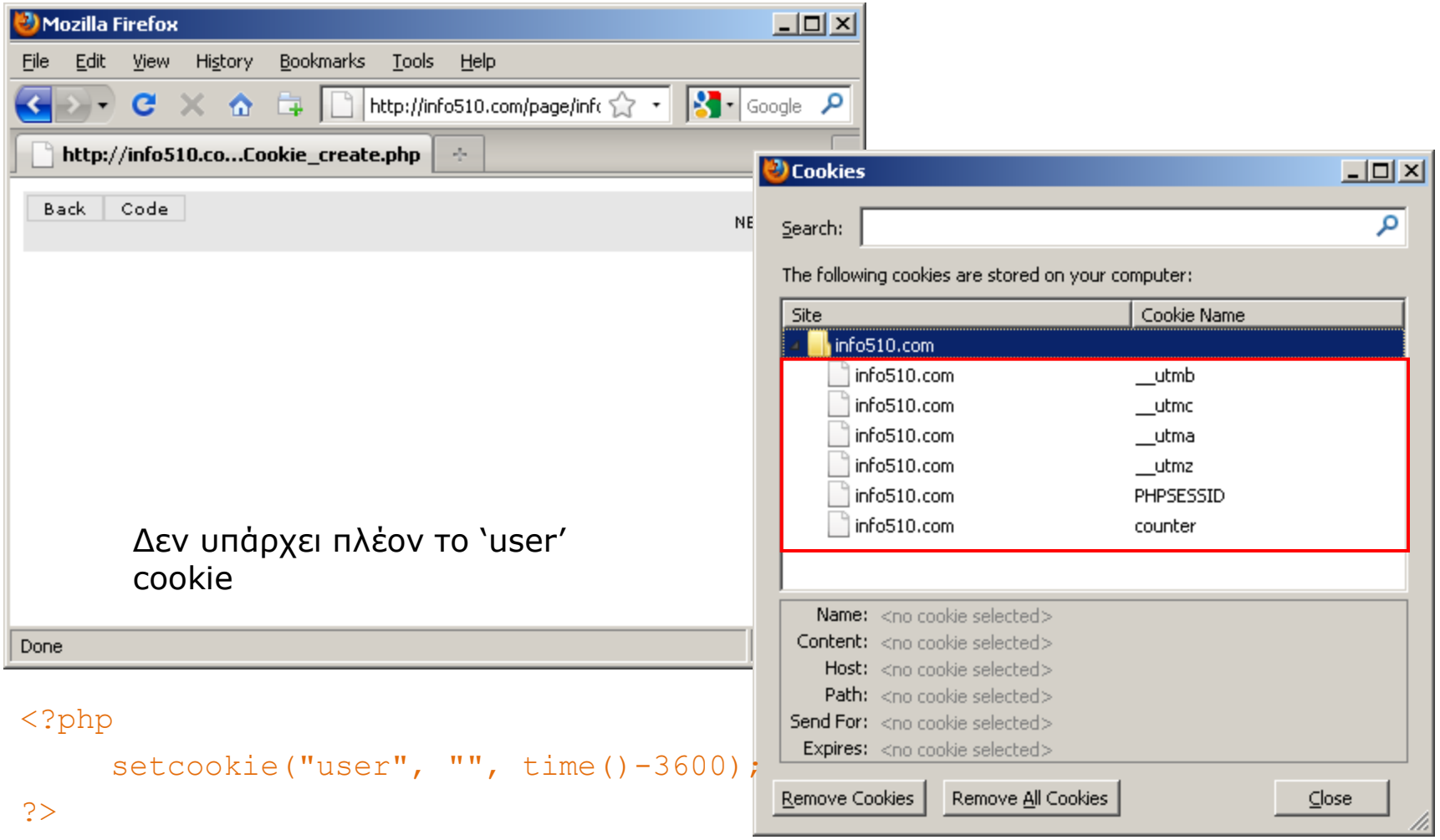
Εμφάνιση των cookies στην PHP





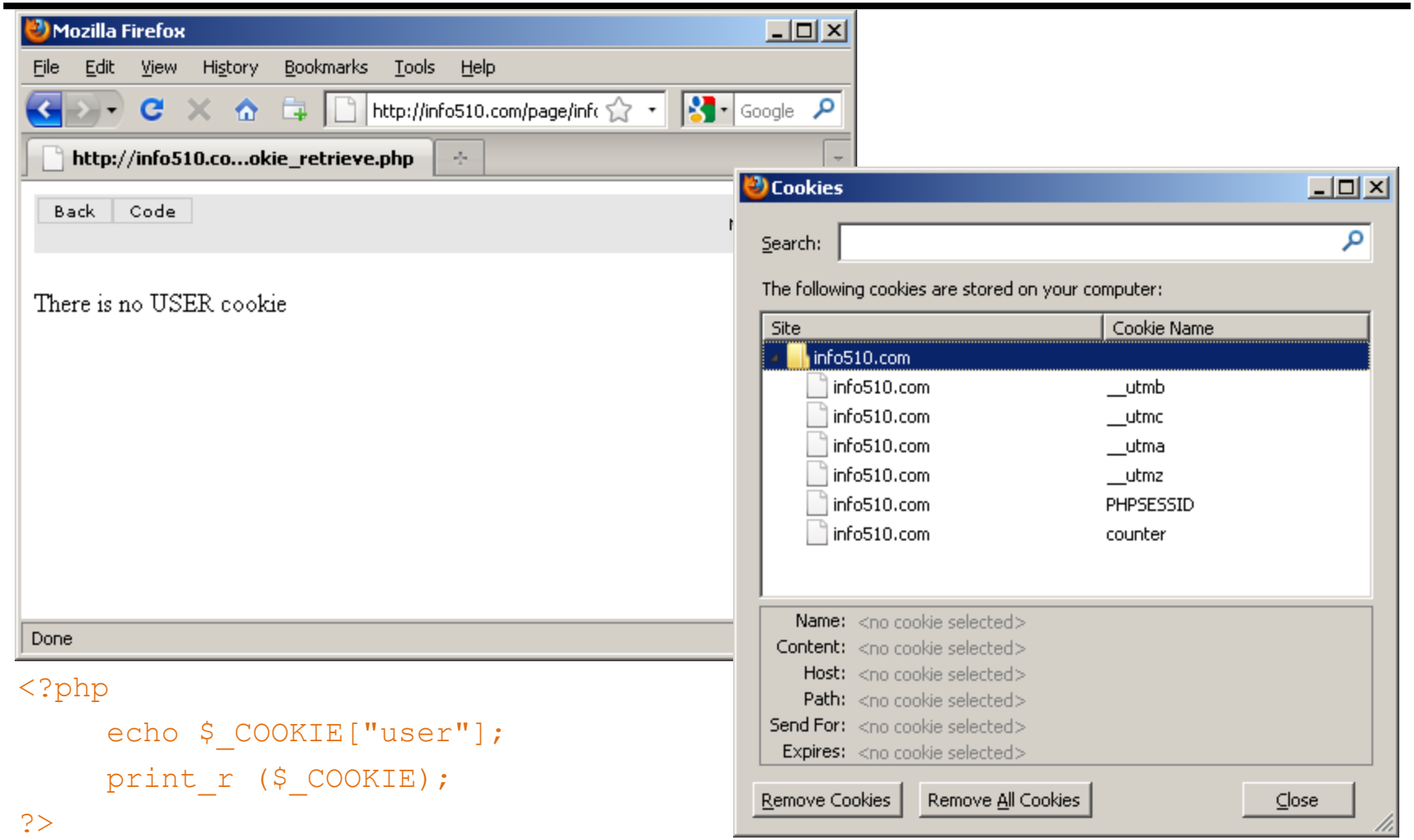

```
<?php
     echo $_COOKIE["user"];
     print_r ($_COOKIE);
?>
```

Διαγραφή των cookies




```
<?php
     setcookie("user", "", time()-3600);
?>
```

Διαγραφή των cookies και εμφάνιση




```
<?php
     echo $_COOKIE["user"];
     print_r ($_COOKIE);
?>
```

Αυθεντικότητα χρήστη (1/2)

```
<?php

   session_start();
   if (isset($_POST['username']) &&
   isset($_POST['password])) {

   //..σύνδεση στη βάση δεδομένων ..

   $sql="select * from users where name='$username' and
       password=sha1('$password')";

   //..εκτέλεση ερωτήματος.

   If ($result->num_rows>0)
{$_SESSION['valid_user']=$username;}

   ?>
```

Αυθεντικότητα χρήστη (2/2)

```
if (isset($_SESSION['valid_user']))

{
   echo 'You are logged in as:
   '.$_SESSION['valid_user'].'<br />';

   echo '<a href=”logout.php”>Log out </a><br />';

}
```

Σε κάθε σελίδα που θέλουμε να έχουν πρόσβαση μόνο εγκεκριμένοι χρήστες πρέπει να ελέγχουμε τη μεταβλητή $_SESSION['valid_user'].

Συναρτήσεις hash

Συνάρτηση μίας διαδρομής

md5
“MINAS” => f19f787f2759b3acb24c3b1fa48eaa06
“MINaS”=>033d5c4cf354556e63bc13d6e28f7ce9
sha
“MINAS”=>80c0c3586c8fa79fd508a5007951866d89c80332cfd140945bb69d6156b74a84
“MINaS”=>ee6b3434d3f509407e6e77352b8085355b3d2e69010c21e76c03f978c106111a

Χρήσιμες συναρτήσεις για ασφάλεια στην PHP

string **urlencode (string $str)**

Μετατρέπει όλους τους μη αλφαριθμητικούς χαρακτήρες, ώστε να μπορούν να αναπαρασταθούν σωστά στη διεύθυνση URL.

string **htmlspecialchars (string $str)**

Μετατρέπει <, >, ", ', και & σε HTML οντότητες.

string **htmlspecialchars_decode (string $str)**

Αντίθετο της συνάρτησης **htmlspecialchars() .**

string **htmlentities (string $str)**

Παρόμοια με τη συνάρτηση **htmlspecialchars()** εκτός από το ότι αυτή η λειτουργία μετατρέπει επίσης όλους τους χαρακτήρες που έχουν ισοδύναμες οντότητες HTML σε οντότητες HTML

Χρήσιμες συναρτήσεις για ασφάλεια στην PHP

string **html_entity_decode (string $str)**

Αντίθετο της συνάρτησης **htmlentities().**

string **strip_tags (string $str, [string $allowable_tags])**

Αυτή η λειτουργία προσπαθεί να επιστρέψει μια συμβολοσειρά με όλες τις ετικέτες HTML και PHP, απογυμνώνεται από μια δεδομένη σελίδα.

Π.χ.

```
// Επίτρεψε <p> και <a>
$stripped_text = strip_tags($text, '<p><a>');
```

Σημείωση: Οι περισσότερες από αυτές τις λειτουργίες έχουν πρόσθετες παραμέτρους. Ανατρέξτε στο ηλεκτρονικό εγχειρίδιο PHP για περισσότερες πληροφορίες σχετικά με κάθε λειτουργία.

Οι κανονικές εκφράσεις θωρακίζουν την ιστοσελίδα

Χρήσιμο για την επικύρωση της εισαγωγής και την αφαίρεση ανεπιθύμητων χαρακτήρων από μια συμβολοσειρά.

π.χ. Αφαιρέστε όλους τους μη αλφαριθμητικούς χαρακτήρες στην $str .

```
$new_str =
        ereg_replace ("[A-Za-z0-9]", "", $ str) ;
```

Προστασία ευαίσθητων δεδομένων

Π.χ των ευαίσθητων δεδομένων: κωδικοί πρόσβασης, δεδομένα πιστωτικών καρτών.

Χρησιμοποιήστε τη μέθοδο POST (αντί για τη μέθοδο GET) για τη μετάδοση δεδομένων.

Μην κρατάτε ευαίσθητα δεδομένα σε cookies .

Αποθηκεύστε τους κωδικούς πρόσβασης σε hash ή κρυπτογραφημένα έντυπα
π.χ. χρησιμοποιήστε τις λειτουργίες SHA1 () ή MD5 () .

Αποφύγετε την αποθήκευση δεδομένων πιστωτικής κάρτας.

Χρησιμοποιήστε HTTPS για να μεταφέρετε ευαίσθητα δεδομένα.

Προστασία ευαίσθητων δεδομένων

HTTP μέσω Secure Socket Layer (SSL) .

Χρησιμοποιήστε το "https:" στη διεύθυνση URL για να υποδείξετε ότι πρόκειται να χρησιμοποιηθεί το HTTPS.
Ο διακομιστής ιστού πρέπει να είναι ρυθμισμένος ώστε να δέχεται το HTTPS.

Το HTTPS κρυπτογραφεί τα δεδομένα στην κεφαλίδα και το σώμα μιας αίτησης HTTP. Τα δεδομένα που κωδικοποιούνται στη διεύθυνση URL δεν είναι κρυπτογραφημένα δηλαδή, το HTTPS δεν κρυπτογραφεί τα δεδομένα που αποστέλλονται μέσω της μεθόδου GET .
Μπορεί να εξασφαλίσει λογική προστασία από τους καταπατητές και τις επιθέσεις από τον άνθρωπο στη μέση.

Προστασία της εφαρμογής - Ταυτοποίηση

Επιβάλλετε ισχυρό κωδικό πρόσβασης.

Για να αποφευχθεί η υπόθεση του κωδικού πρόσβασης .
Μετά από ορισμένους αριθμούς ανεπιτυχών προσπαθειών σύνδεσης, καθυστερούν ή αποκλείουν τις μελλοντικές προσπάθειες σύνδεσης από τον ίδιο χρήστη.

Χρησιμοποιήστε περισσότερο από ένα αναγνωριστικό περιόδου σύνδεσης για να ελέγξετε αν έχει ήδη συνδεθεί κάποιος χρήστης
π.χ. ελέγξτε επίσης την τιμή του πεδίου "User-Agent" στην κεφαλίδα HTTP.

Όταν ένας χρήστης θέλει να αλλάξει τον κωδικό πρόσβασης, ρωτήστε τον χρήστη για νέους και παλιούς κωδικούς πρόσβασης.

Έλεγχοι προσωρινής μνήμης.
Ζητήστε από τον πελάτη (web client) να μην αποθηκεύει προσωρινά δεδομένα φόρμας, έτσι ώστε κανείς να μην μπορεί να χρησιμοποιήσει το κουμπί "Πίσω" του προγράμματος περιήγησης για να υποβάλει ξανά τα δεδομένα σύνδεσης

Προστασία των συνεδριών (Sessions)

Μην κρατάτε αναγνωριστικό περιόδου σύνδεσης (Session ID) στη διεύθυνση URL.

Περιορίστε τη διάρκεια ζωής του cookie που περιέχει το αναγνωριστικό περιόδου σύνδεσης (Session ID)

Εάν ο κεντρικός υπολογιστής εξυπηρετεί πολλούς χρήστες βεβαιωθείτε ότι η "διαδρομή" του cookie έχει οριστεί κατά τέτοιο τρόπο ώστε ο πελάτης (web client) να επιστρέψει μόνο το cookie στα σενάρια στον φάκελο της εφαρμογής σας.

Βεβαιωθείτε ότι τα αρχεία που διατηρούν δεδομένα περιόδου σύνδεσης δεν είναι προσβάσιμα από άλλους χρήστες στον ίδιο κεντρικό υπολογιστή.
Όταν ένας χρήστης των αποσυνδεθεί από την εφαρμογή ιστού σας, βεβαιωθείτε ότι έχουν διαγραφεί όλα τα δεδομένα που σχετίζονται με την περίοδο σύνδεσης.

Προστασία των μηνυμάτων λάθους

Μην εμφανίζετε λεπτομερή μηνύματα σφάλματος στους χρήστες.

Όσα λιγότερα γνωρίζει ένας εισβολέας (hacker) σχετικά με το πως λειτουργεί η εφαρμογή σας τόσο καλύτερα.

Καταγράψτε όλα τα σφάλματα και τις λεπτομέρειες τους.

Τα TOP10 σφάλματα στις εφαρμογές διαδικτύου

A1 - Cross Site Scripting (XSS)

A2 - Injection Flaws

A3 - Malicious File Execution

A4 - Insecure Direct Object Reference

A5 - Cross Site Request Forgery (CSRF)

A6 - Information Leakage and Improper Error Handling

A7 - Broken Authentication and Session Management

A8 - Insecure Cryptographic Storage

A9 - Insecure Communications

A10 - Failure to Restrict URL Access