

# ΡΗΡ 2

---

Διδάσκοντες: Π. Αγγελάτος, Δ. Ζήνδρος  
Επιμέλεια διαφανειών: Π. Αγγελάτος

Σχολή Ηλεκτρολόγων Μηχανικών  
και Μηχανικών Υπολογιστών



# Στόχος της ώρας

- Εμβάθυνση στην PHP:
  - Χωρισμός κώδικα σε αρχεία
  - Εμβέλεια μεταβλητών
  - Πίνακες
  - foreach
  - Συναρτήσεις αρχείων
  - Χειρισμός ανεβασμένου αρχείου
  - Μπισκότα
  - Σύνοδοι

# Εμβέλεια μεταβλητών

- Είναι η περιοχή στην οποία ορίστηκε η μεταβλητή
- Οι μεταβλητές έξω από τις συναρτήσεις δεν είναι διαθέσιμες σε αυτές

```
$a = 5;  
function foo(){  
    echo $a; //λάθος  
}
```

# Εμβέλεια μεταβλητών

- Οι μεταβλητές είναι διαθέσιμες στα αρχεία που φορτώνονται

```
$a = 1;  
include 'b.php';  
// Το $a είναι διαθέσιμο στο b.php
```

# Εμβέλεια μεταβλητών

- Λέξη-κλειδί `global`
- Χρησιμοποιείται μέσα σε συναρτήσεις
- Δηλώνει ότι οι μεταβλητές που δηλώνονται ανήκουν έξω από τη συνάρτηση

```
$a = 5;  
function foo(){  
    global $a;  
    echo $a; //5  
}
```

# Κεφαλίδες

- `header()`
- Παίρνει παράμετρο ένα `string`
- Το στέλνει στις κεφαλίδες της απάντησης
- Πρέπει να τρέξει πριν οποιοδήποτε `Output`
  - Πριν το `<html>`
  - Πριν οποιοδήποτε `space`, `enter`, `tab`

# Κεφαλίδες

```
<html>
```

```
<?php
```

```
/* ΛΑΘΟΣ
```

```
Υπάρχει output πριν το header()*/
```

```
header( 'Location: http://www.example.com/' );
```

```
?>
```

# Κεφαλίδες

```
header( 'Location: http://www.example.com/' );
```

- Κάνει redirect τον browser στη διεύθυνση που ορίζουμε
- Η διεύθυνση πρέπει να είναι απόλυτη διεύθυνση
- Προκαλεί και την αλλαγή του κωδικού κατάστασης από 200 OK σε 302 FOUND



# Κεφαλίδες

```
header( 'Content-type: text/html; charset=utf-8' )
```

- Κάνει την ίδια δουλειά που κάνει και το

```
<meta
```

```
    http-equiv="Content-type"
```

```
    content="text/html; charset=utf-8"
```

```
/>
```

# Μπισκότα

- Χρησιμοποιούνται κυρίως για να ξεχωρίζουν χρήστες
- Είναι μία λίστα από αντιστοιχίες κλειδί → τιμή
- Έχουν ημερομηνία λήξης
- Αποθηκεύονται στον υπολογιστή του χρήστη

# Μπισκότα

- Στέλνονται σε κάθε αίτημα
- Όσο περισσότερα είναι τόσο περισσότερο overhead έχουμε
- Διαφορετικά μπισκότα για διαφορετικά domains
- Είναι πολύτιμα! Αν σου τα κλέψει κάποιος τότε γίνεται εσύ

# Παράδειγμα



Γεια!



Γεια, να θυμάσαι ότι είσαι admin



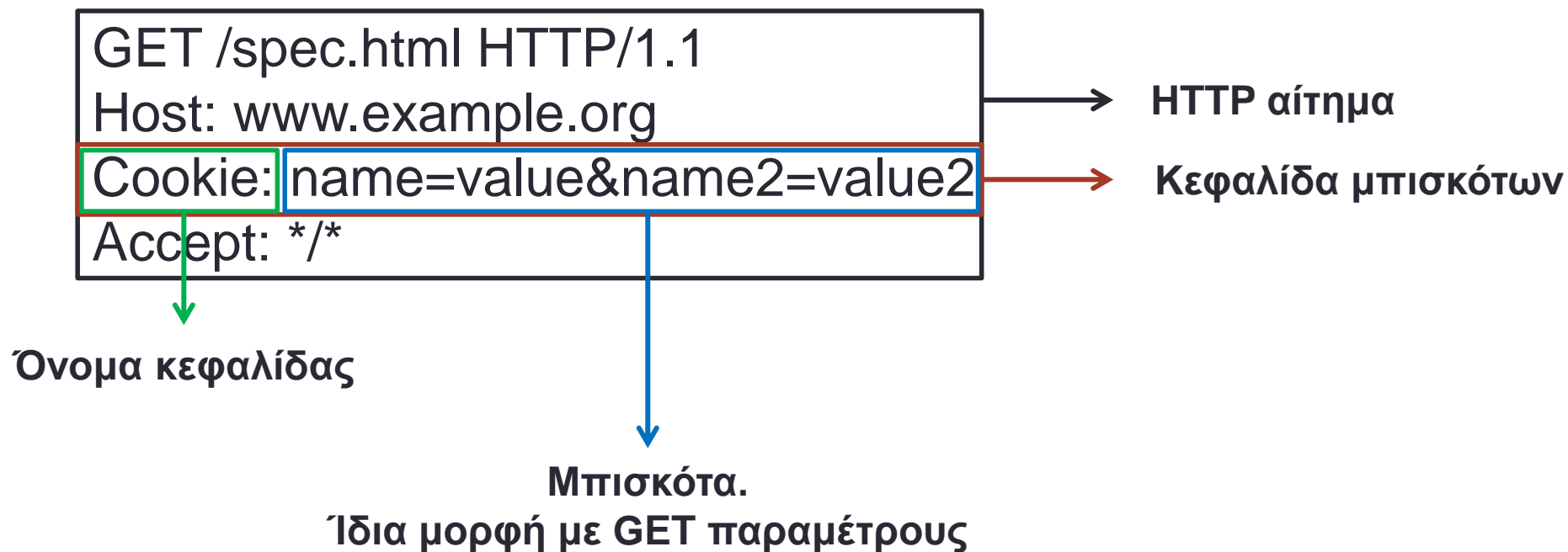
Είμαι είμαι admin



ΟΚ, κάνε ό,τι θες



# Μπισκότα – Πως μοιάζουν;



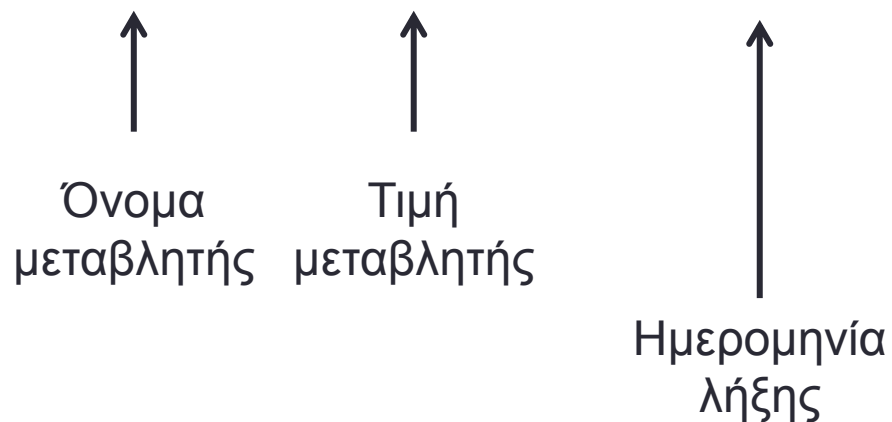
- Εδώ 2 μπισκότα
  - Μπισκότο **name** με τιμή **value**
  - Μπισκότο **name2** με τιμή **value2**

# Μπισκότα

- Μεταφέρονται στις κεφαλίδες των αιτημάτων
- Για να ορίσουμε ένα cookie σε ένα χρήστη κάνουμε:  
`setcookie( $name, $value, $expire )`
- Πρέπει να τρέξει πριν οποιοδήποτε output συμπεριλαμβανομένου και του `<html>`

# Μπισκότα

- `setcookie( $name, $value, $expire )`



- Αν το `expire` είναι 0 τότε το μπισκότο σβήνεται όταν κλείσει ο browser

# Μπισκότα

- `setcookie( 'name', 'petros', time() + 3600)`



Θα λήξει σε  
μία ώρα

- `time()` → επιστρέφει την ώρα σε μορφή Unix timestamp
- Λήγει σε 3600 δευτερόλεπτα από τώρα



# Μπισκότα

- Βρίσκονται στη μεταβλητή `$_COOKIE`
- Έχει κλειδιά τα ονόματα των μπισκότων
- Έχει τιμές τις τιμές των μπισκότων

# Μπισκότα – Διαγραφή

- Αρκεί ο ορισμός του μπισκότου με `expire` στο παρελθόν
- `setcookie( 'name', 'petros', time() - 3600 )`



Έληξε πριν μία ώρα

# Μπισκότα – παράδειγμα

```
<?php
```

```
    $expire = time() + 60 * 60 * 24 * 7;  
    setcookie( 'visited', 'true', $expire );
```

```
    if ( isset( $_COOKIE[ 'visited' ] ) ) {  
        echo 'welcome back!';
```

```
    }
```

```
    else {  
        echo 'Hello, stranger';
```

```
    }
```

```
?>
```

# Σύνοδοι

- Βολική λειτουργία της PHP
- Αντί να έχουμε πολλά μπισκότα έχουμε ένα
- Το μπισκότο λέγεται Session ID
  - Είναι ένας τυχαίος αριθμός
  - Ορίζεται την πρώτη φορά που ανοίγει κάποιος τη σελίδα
  - Διαφορετικό για κάθε χρήστη
- Με βάση αυτό αποθηκεύονται στον **server** μεταβλητές
- Τις διαχειρίζεται η PHP για μας

# Σύνοδοι

- `session_start()`
- Αρχίζει μία σύνοδο ή επαναφέρει την προηγούμενη αν ο χρήστης έστειλε έγκυρο Session ID
- Επειδή ορίζει μπισκότο την καλούμε πριν οποιοδήποτε output

# Σύνοδοι

- Μεταβλητή `$_SESSION`
- Τη χρησιμοποιούμε αφού καλέσουμε το `session_start()`
- Οι μεταβλητές αποθηκεύονται και θα είναι διαθέσιμες σε επόμενα αιτήματα με το ίδιο Session ID

# Παράδειγμα



Γεια!

Γεια, όταν μου μιλάς θα μου λες ότι είσαι ο 532

943 – Γιώργος  
453 – Νίκος



SID – 532



Είμαι ο 532, με λένε Πέτρο

OK

532 – Πέτρος  
943 – Γιώργος  
453 – Νίκος



# Παράδειγμα

SID – 532



Είμαι ο 532, γεια!



Γεια σου Πέτρο



532 – Πέτρος  
943 – Γιώργος  
453 – Νίκος



SID – 324



Είμαι ο 324



Δε σε ξέρω



532 – Πέτρος  
943 – Γιώργος  
453 – Νίκος





# Παράδειγμα κώδικα

```
<?php
    session_start();
    if ( isset( $_SESSION[ 'pageviews' ] ) ) {
        ?><p>Έχεις δει τη σελίδα <?php
        echo $_SESSION[ 'pageviews' ];
        ?> φορές</p><?php
        ++$_SESSION[ 'pageviews' ];
    }
    else {
        $_SESSION[ 'pageviews' ] = 1;
    }
?>
```

# 4<sup>η</sup> Εργασία

- Παράδοση μέσω SSH (Secure SHell)
  - Ανεβάστε .php, .html, .css, εικόνες κλπ. στο home σας
  - Μέχρι **22/11/2010**
- **Να δημιουργήσετε ένα file uploader**
  - Μία σελίδα
    - Λίστα όλων των αρχείων που έχουν ανέβει
    - Κλικ για κατέβασμα αρχείου
    - Πλαίσιο για ανέβασμα αρχείου
  - Ένα επιπλέον αρχείο .php για διαχείριση ανεβάσματος

# 4<sup>η</sup> Εργασία

- Απλά πράγματα!
- Όχι βάση δεδομένων
- Όχι σύστημα χρηστών
- Όποιος θέλει ανεβάζει ό,τι θέλει
- Αποθηκεύετε τα αρχεία σε έναν φάκελο στον server
  - π.χ. C:/uploads σε Windows
  - π.χ. /home/uploads σε Linux
- Η λίστα εμφανίζεται με απλό opendir/readdir
- Μην σας νοιάζει αντικατάσταση υπάρχοντων αρχείων

# Συναρτήσεις χειρισμού μεταβλητών

- **unset**
- Παίρνει παράμετρο μία μεταβλητή
- Αν είναι ορισμένη την διαγράφει

# Συναρτήσεις χειρισμού μεταβλητών

```
<?php
```

```
// Διαγραφή μιας μεταβλητής
```

```
unset( $foo );
```

```
// Διαγραφή περισσότερων μεταβλητών
```

```
unset($foo1, $foo2, $foo3);
```

```
?>
```

# Συναρτήσεις χειρισμού μεταβλητών

- **isset**

- Παίρνει παράμετρο μία μεταβλητή
- Αν είναι ορισμένη και δεν είναι **NULL** επιστρέφει **true**
- Αν είναι ορισμένη και είναι **NULL** επιστρέφει **false**
- Αν δεν είναι ορισμένη επιστρέφει **false**

# Συναρτήσεις χειρισμού μεταβλητών

- **empty**
- Παίρνει παράμετρο μία μεταβλητή
- Αν δεν είναι ορισμένη επιστρέφει **true**
- Αν είναι ορισμένη και δεν έχει τιμή επιστρέφει **true**
- Αν είναι ορισμένη και έχει τιμή επιστρέφει **false**

# Συναρτήσεις χειρισμού μεταβλητών

- Τα παρακάτω θεωρούνται άδειες μεταβλητές
  - "" (άδειο string)
  - 0 (0 ως ακέραιος)
  - "0 " (το string με τον χαρακτήρα 0)
  - **NULL**
  - **FALSE**
  - **array()** (ένας άδειος πίνακας)
  - **var \$var;** (μία μεταβλητή που ορίστηκε χωρίς τιμή)



# Χάρτες

- Είναι τύπος δεδομένων
- Περιέχει αντιστοιχίες
- Αντιστοιχίζει **κλειδί** → **τιμή**
- Μπορεί να χρησιμοποιηθεί ως πίνακας, λίστα, ουρά, στοίβα κλπ

# Πίνακες

- Στην PHP είναι ταξινομημένοι χάρτες με βάση τα κλειδιά
- Τα κλειδιά μπορεί να είναι ακέραιος ή string
- Οι τιμές μπορεί να είναι οτιδήποτε

# Πίνακες

- Δημιουργία πίνακα με την εντολή `array()`

```
//πίνακας με 2 στοιχεία
```

```
//κλειδί 0 έχει τιμή 1
```

```
//κλειδί 1 έχει τιμή 'bar'
```

```
$foo = array( 0 => 1, 1 => 'bar' );
```

```
echo $foo[ 0 ]; // 1
```

```
echo $foo[ 1 ]; // bar
```

# Πίνακες

- Παράλειψη κλειδιών
- Το στοιχείο έχει ως κλειδί το μέγιστο ακέραιο κλειδί +1

```
//πίνακας με 2 στοιχεία
```

```
//κλειδί 0 έχει τιμή 1
```

```
//Τιμή 'bar'
```

```
$foo = array( 0 => 1, 'bar' );
```

```
echo $foo[ 0 ]; // 1
```

```
echo $foo[ 1 ]; // bar
```

# Πολυδιάστατοι Πίνακες

- Τόσο απλό όσο ένα array() σε ένα array()

```
//πίνακας 3x3
```

```
$foo = array(  
    array( 1, 2, 3 ),  
    array( 4, 5, 6 ),  
    array( 7, 8, 9 )  
);
```

```
echo $foo[ 1 ][ 2 ]; // 6
```

```
echo $foo[ 2 ][ 0 ]; // 7
```

# Πίνακες – Σύνταξη αγκύλων

- `$foo[ 'κλειδί' ] = τιμή`
- Αν το `$foo` δεν έχει οριστεί δημιουργείται πίνακας
- Αν το `$foo[ 'κλειδί' ]` υπάρχει η τιμή αντικαθίσταται
  
- `$foo[] = τιμή`
- Προσθέτει την τιμή με κλειδί το μέγιστο ακέραιο κλειδί +1
- Αν το `$foo` δεν υπάρχει δημιουργείται πίνακας

# Πίνακες – Διαγραφή στοιχείου

- Γίνεται με την συνάρτηση **unset**
- Ο πίνακας **δε** γίνεται reindex
- Αν επιθυμούμε reindex το κάνουμε με την συνάρτηση **array\_values**

# Πίνακες διαγραφή στοιχείου

```
$arr = array(  
    1 => 1,  
    2 => 2,  
    3 => 3  
);
```

```
unset( $arr[ 2 ] );
```

```
/*$arr = [  
    1 => 1,    ΚΑΙ ΟΧΙ  
    3 => 3,  
]
```

```
$arr= [  
    1 => 1,  
    2 => 3  
] */
```



# Χρήσιμες συναρτήσεις πίνακα

Όνομα συνάρτησης	Περιγραφή
<code>count</code>	Επιστρέφει τον αριθμό των στοιχείων σε ένα πίνακα
<code>array_flip</code>	Κάνει τις τιμές κλειδιά και τα κλειδιά τιμές
<code>array_pop</code>	Αφαιρεί και επιστρέφει το τελευταίο στοιχείο
<code>array_push</code>	Βάζει στοιχεία στο τέλος του πίνακα
<code>array_splice</code>	Κόβει ένα κομμάτι και το αντικαθιστά με κάτι άλλο
<code>array_search</code>	Ψάχνει στον πίνακα για την τιμή που δίνουμε

- Και πολλές άλλες!
- Ψάξτε τες στο <http://php.net/> και το function name
- πχ [http://php.net/array\\_flip](http://php.net/array_flip)

# foreach

- Συνδυάζεται με κάποιο πίνακα
- Τρέχει ένα σώμα κώδικα για κάθε στοιχείο του πίνακα
- Σύνταξη:

```
foreach ( $array as $value ) {  
    σώμα  
}
```

Κάθε φορά το **\$value** περιέχει την τιμή του τρέχοντος στοιχείου

# foreach

- Αν αλλάξει το \$value ΔΕΝ αλλάζει ο πίνακας

```
$array = array ( 1, 2, 3 );  
foreach ( $array as $value ) {  
    $value = 0;  
}  
echo $array[ 0 ]; // 1
```

# foreach

- Εναλλακτική μορφή

```
foreach ( $array as $key => $value ) {  
    σώμα  
}
```

Το **\$key** περιέχει το κλειδί του τρέχοντος στοιχείου

Το **\$value** περιέχει την τιμή του τρέχοντος στοιχείου

# foreach

```
$fruits = array(
    'banana' => 'yellow',
    'apple' => 'red',
);
foreach ( $fruits as $fruit => $color ) {
    echo $fruit . "'s color is $color\n";
}
/* banana's color is yellow
   apples's color is red
*/
```

# foreach – πολυδιάστατοι πίνακες

```
$matrix = array(  
    array( 1, 2, 3 ),  
    array( 4, 5, 6 ),  
    array( 7, 8, 9 ),  
);  
foreach ( $matrix as $row ) {  
    foreach ( $row as $element ) {  
        echo "$element ";  
    }  
    echo "\n";  
}
```

# Χωρισμός αρχείων

- Μπορούμε να φορτώσουμε τον κώδικα ενός αρχείου σε ένα άλλο αρχείο
- Βοηθάει στην οργάνωση κώδικα
- Επαναχρησιμοποιούμενος κώδικας

# Χωρισμός αρχείων

- `include`
- Σύνταξη
- `include 'library.php' ;`
- Τρέχει τον κώδικα του αρχείου `library.php`
- Αν δε βρει το αρχείο παράγει προειδοποίηση και συνεχίζει
- Είναι το ίδιο με το να κάναμε `c/r` τον κώδικα του αρχείου στη θέση του `include`



# Χωρισμός αρχείων

- `include_once`
- Σύνταξη
- `include_once 'library.php';`
- Παρόμοιο με το `include`
- Τρέχει τον κώδικα μόνο την πρώτη φορά που φορτώνει ένα αρχείο

# Χωρισμός αρχείων

- `require`
- Σύνταξη
- `require 'library.php';`
- Παρόμοιο με το `include`
- Αν δε βρει το αρχείο δε συνεχίζει η εκτέλεση, παράγεται `error`

# Χωρισμός αρχείων

- `require_once`
- Σύνταξη
- `require_once 'library.php';`
- Παρόμοιο με το `require`
- Τρέχει τον κώδικα μόνο την πρώτη φορά που φορτώνει ένα αρχείο

# Χωρισμός αρχείων

- Παραδείγματα
  - Φόρτωση βιβλιοθήκης συναρτήσεων
  - Φόρτωση Header και Footer σελίδας

# Τελεστές απόλυτης σύγκρισης

- `$a === $b`
- Είναι **true** μόνο αν το **\$a** είναι ίσο με το **\$b** και είναι του ίδιου τύπου
- `$a = 1;`
- `$b = '1';`
- `$a == $b //true`
- `$a === $b //false`

# Τελεστές απόλυτης σύγκρισης

- `$a !== $b`
- Είναι **true** μόνο αν το **\$a** δεν είναι ίσο με το **\$b** ή είναι διαφορετικού τύπου
- `$a = 1;`
- `$b = '1';`
- `$a != $b //false`
- `$a !== $b //true`

# Συναρτήσεις αρχείων

- `opendir()`
  - Ανοίγει φακέλους
  - Μία παράμετρο, τη θέση του φακέλου
  - Επιστρέφει τιμή που χρησιμοποιείται από
    - `readdir()`
    - `closedir()`
    - `rewinddir()`
- Αν αποτύχει επιστρέφει `false`

# Συναρτήσεις αρχείων

- `readdir()`
- Μία παράμετρο, την τιμή της `opendir()`
- Επιστρέφει το όνομα του επόμενου αρχείου στον φάκελο
- Αν αποτύχει επιστρέφει `false`



# Συναρτήσεις αρχείων

- `closedir()`
- Μία παράμετρο, την τιμή της `opendir()`
- Κλείνει τον φάκελο
- Την καλούμε όταν τελειώσουμε με κάποιο
- Αν αποτύχει επιστρέφει `false`

# Συναρτήσεις αρχείων

- Εκτύπωση περιεχομένων φακέλου

```
$handle = opendir( '/path/to/files' ) );
```

```
while ( false !== ( $file = readdir(  
$handle ) ) ) {  
    echo "$file\n";  
}
```

# Συναρτήσεις αρχείων

- `file_get_contents()`
- Μία παράμετρο, το αρχείο
- Διαβάζει το αρχείο και επιστρέφει τα περιεχόμενά του
- Αν αποτύχει επιστρέφει `false`

# Συναρτήσεις αρχείων

- `file_put_contents()`
- Δύο παράμετροι
  - Το αρχείο
  - Τα περιεχόμενα
- Αποθηκεύει στο αρχείο τα περιεχόμενα.
- Αντικαθιστά τα προηγούμενα

# Συναρτήσεις αρχείων

- Υπάρχουν και οι συναρτήσεις
- `fopen`
- `fwrite`
- `fclose`
- `fgets`
- κλπ
  
- Ψάξτε τες όταν τις χρειαστείτε!
- <http://gr.php.net/manual/en/ref.filesystem.php>

# Ανέβασμα αρχείων

- Στέλνονται στον server με αίτημα POST
- Αποθηκεύονται σε προσωρινή θέση όσο τρέχει το πρόγραμμά μας
- Αν δεν τα αποθηκεύσουμε χάνονται μετά το τέλος της εκτέλεσης

# Ανέβασμα αρχείων

```
<form enctype="multipart/form-data" method="post">  
  Send this file:  
  <input name="userfile" type="file" />  
  <input type="submit" value="Send File" />  
</form>
```



**Με βάση αυτό βρίσκουμε τα  
αρχεία στην PHP**

# Ανέβασμα αρχείων

- Μεταβλητή `$_FILES`
- Περιέχει όλα τα αρχεία που ανέβασε ο χρήστης
- `$_FILES[ 'userfile' ] [ 'name' ]`
- Περιέχει το πραγματικό όνομα του αρχείου
- `$_FILES[ 'userfile' ] [ 'tmp_name' ]`
- Περιέχει τη θέση του προσωρινού αρχείου
- `$_FILES[ 'userfile' ] [ 'size' ]`
- Περιέχει το μέγεθος του αρχείου σε bytes



# Ανέβασμα αρχείων

- `move_uploaded_file( $filename, $destination )`
- Ελέγχει αν το `$filename` έχει όντως προέλθει από ανέβασμα
- Αν είναι το αποθηκεύει στο `$destination`

# Ανέβασμα αρχείων

```
<?php
    $destination = 'C:\Uploads\'
    if( !empty( $_FILES ) ) {
        $destination .= $_FILES[ 'foo' ][ 'name' ];
        $filename = $_FILES[ 'foo' ][ 'tmp_name' ];
        move_uploaded_file( $filename, $destination );
    }
?>
```

# Μάθαμε

- Εμβάθυνση στην PHP:
  - Χωρισμός κώδικα σε αρχεία
  - Εμβέλεια μεταβλητών
  - Πίνακες
  - foreach
  - Συναρτήσεις αρχείων
  - Χειρισμός ανεβασμένου αρχείου
  - Μπισκότα
  - Σύνοδοι

# Την επόμενη φορά...

- Bonus θέμα!
- Γραφικά στην PHP με τη βιβλιοθήκη GD
  - Πώς να φτιάξετε ένα πρόγραμμα γραφικών παραστάσεων