

Εργαστήριο #5

Τι πρέπει να έχετε ολοκληρώσει από το προηγούμενο εργαστήριο.

- Θα πρέπει να ξέρετε να εισάγετε ένα βασικό πρόγραμμα PHP μέσα σε μια ιστοσελίδα, τη χρήση της echo και τον χειρισμό απλών μεταβλητών στην PHP.

Οδηγίες

Στο προηγούμενο εργαστήριο, οι σελίδες PHP που δοκιμάσατε ήταν στατικές – θα μπορούσαν το ίδιο καλά να υλοποιηθούν με απλή HTML. Η PHP σας δίνει όμως τη δυνατότητα να γράψετε **δυναμικές ιστοσελίδες**: ιστοσελίδες που το περιεχόμενό τους αλλάζει ανάλογα με δεδομένα από τον χρήστη, από βάσεις δεδομένων κ.ο.κ.

⇒ 1. Διαβάστε το **Παράρτημα Α** (θεωρητικό υπόβαθρο: PHP, HTML φόρμες και το πρωτόκολλο HTTP). **Βεβαιωθείτε ότι έχετε καταλάβει πλήρως το κείμενο πριν προχωρήσετε στα επόμενα βήματα!**

⇒ 2. Διαβάστε το **Παράρτημα Β** για τα στοιχεία των φορμών HTML.

⇒ 3. Κατασκευάστε ιστοσελίδα με φόρμα HTML, η οποία θα περιέχει ένα πλαίσιο κειμένου και ένα κουμπί αποστολής. Ο χρήστης θα συμπληρώνει έναν αριθμό και, πιέζοντας το κουμπί αποστολής, θα ζητείται η ιστοσελίδα PHP από το προηγούμενο εργαστήριο με τις εξής τροποποιήσεις:

- Αντί του x^2 θα τυπώνεται το x^n , όπου n θα είναι ο αριθμός που συμπλήρωσε ο χρήστης στη φόρμα (χρησιμοποιήστε τη συνάρτηση **pow(x,y)** που δίνει το x^y).
- Πριν τυπώσετε, θα μετατρέπετε την είσοδο του χρήστη σε ακέραιο και εάν ο αριθμός είναι μικρότερος από 1 ή μεγαλύτερος από 10, αντί του πίνακα θα τυπώνετε ένα μήνυμα σφάλματος.

⇒ 4. Προσθέστε στη φόρμα σας μια ομάδα radio buttons, τα οποία θα επιτρέπουν στον χρήστη να παρουσιάζει εναλλακτικά τις δυνάμεις των αριθμών 1-100, 101-200 ή 201-300. Συμπληρώστε αντίστοιχα το πρόγραμμα PHP.

Παράρτημα Α: PHP, HTML φόρμες και το πρωτόκολλο HTTP.

Όταν ο web server καλείται να επιστρέψει μια ιστοσελίδα που περιέχει κώδικα PHP, τότε δημιουργείται ένα **περιβάλλον εκτέλεσης** για τον κώδικα αυτόν.

- Για κάθε αίτηση που φτάνει στον web server, δημιουργείται ένα νέο περιβάλλον εκτέλεσης.
- Το περιβάλλον εκτέλεσης χρησιμοποιείται για όλον τον κώδικα PHP μιας ιστοσελίδας.

Το περιβάλλον εκτέλεσης, μεταξύ άλλων, περιέχει μεταβλητές που περιγράφουν τις λεπτομέρειες της αίτησης καθώς και δεδομένα **από τον χρήστη προς τον web server**. Αυτές οι μεταβλητές είναι διαθέσιμες στο πρόγραμμά σας PHP και με βάση τις τιμές τους μπορείτε να δημιουργήσετε ιστοσελίδες με **δυναμικό τρόπο**.

Για να στείλει ο χρήστης δεδομένα **προς** τον web server χρησιμοποιούνται κατά κύριο λόγο οι **φόρμες HTML** (HTML forms).

Εισαγωγή στις Φόρμες HTML

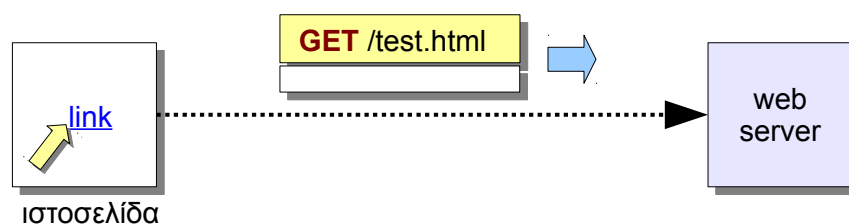
Η γλώσσα HTML μας παρέχει τα στοιχεία (elements) για να απεικονιστούν στον browser φόρμες όπου ο χρήστης μπορεί να εισάγει κείμενο (**textboxes**), να επιλέξει από μια λίστα (**selections**) ή από ένα σετ επιλογών (**radio buttons**), να “τσεκάρει” μία ή περισσότερες επιλογές (**checkboxes**) και τελικά να στείλει τα δεδομένα του στον web server, πιέζοντας ένα κουμπί αποστολής (**submit button**).



Κάθε στοιχείο εισόδου (textbox, selection, radio button, checkbox κλπ) έχει μια ιδιότητα **name**. Μπορείτε να φανταστείτε την τιμή της name ακριβώς ίδια με το όνομα μιας μεταβλητής της C: μας επιτρέπει να ανακτήσουμε το περιεχόμενο του συγκεκριμένου στοιχείου στην PHP.

Προσοχή: η τιμή της **name** δεν φαίνεται πουθενά στην απεικόνιση της ιστοσελίδας· **δεν προορίζεται για τον χρήστη, παρά μόνο για τον προγραμματιστή!** Καλό είναι να χρησιμοποιείτε ονόματα στα αγγλικά, όπως ακριβώς κάνετε με τα ονόματα των μεταβλητών σε άλλες γλώσσες.

Τι ακριβώς συμβαίνει όταν ο χρήστης πιάσει το κουμπί αποστολής μιας φόρμας; Για να φανεί καλύτερα ο μηχανισμός, ας δούμε πρώτα τι συμβαίνει όταν ο χρήστης πατά έναν απλό υπερσύνδεσμο (link) μιας ιστοσελίδας (**σχήμα 1**):



Σχήμα 1

Όπως φαίνεται στο **σχήμα 1**, η επιλογή του link οδηγεί τον browser να δημιουργήσει μια HTTP αίτηση προς τον web server, ζητώντας ένα αντικείμενο, το URL του οποίου βρίσκεται μέσα στο href του υπερσυνδέσμου. Το πρωτόκολλο HTTP ορίζει μια “**μέθοδο**” για κάθε αίτηση. Όπως φαίνεται στο σχήμα 1, η επιλογή του υπερσυνδέσμου δημιουργεί μια αίτηση με τη μέθοδο **GET**.

Η αποστολή μιας φόρμας HTML, πιέζοντας το κουμπί αποστολής (submit), δημιουργεί επίσης μια αίτηση προς τον web server:

- το URL του αντικειμένου που θα ζητηθεί περιέχεται στην ιδιότητα **action** της φόρμας (οι λεπτομέρειες θα δοθούν αργότερα).
- η “μέθοδος” της αίτησης μπορεί να είναι είτε **GET** είτε **POST** (αναλύονται στη συνέχεια) και δίνεται μέσω της ιδιότητας **method** της φόρμας.

Η διαφορά με την αίτηση λόγω επιλογής υπερσυνδέσμου είναι ότι τώρα, μαζί με την αίτηση, **μεταδίδονται επίσης προς τον server οι τιμές** (ό,τι έχει συμπληρώσει ή επιλέξει ο χρήστης πριν πατήσει το κουμπί αποστολής) **όλων των στοιχείων εισόδου της φόρμας**. Η μετάδοση αυτή γίνεται σε ζεύγη **name,value**:

```
name1 = value1  
name2 = value2  
....  
nameN = valueN
```

για κάθε στοιχείο εισόδου που έχει περιεχόμενο. Το *nameN* είναι η τιμή της ιδιότητας **name** του κάθε στοιχείου και το *valueN* ισούται με τα δεδομένα που έχει συμπληρώσει ή επιλέξει ο χρήστης για το ίδιο στοιχείο.

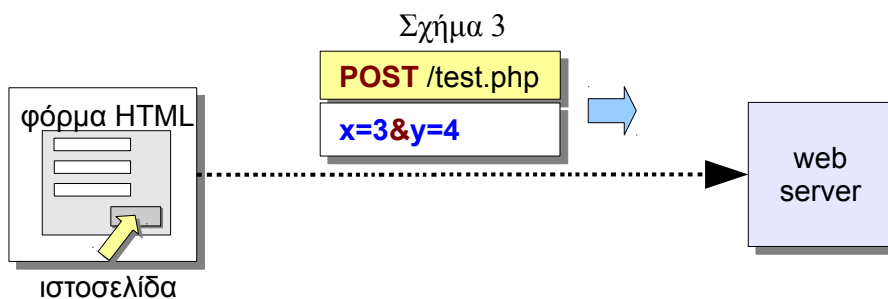
Οι μέθοδοι GET και POST

Οι δύο μέθοδοι επιτυγχάνουν ακριβώς το ίδιο, την αποστολή δηλαδή των ζευγών **name,value** προς τον web server. Η διαφορά είναι το **πώς** το κάνουν:



Σχήμα 2

Με τη μέθοδο **GET** (σχήμα 2), τα ζεύγη **name,value** **παρατίθενται στη διεύθυνση URL** της αίτησης. Αντιθέτως, η μέθοδος **POST** (σχήμα 3) προσθέτει τα ζεύγη **name,value** **στο κυρίως σώμα της αίτησης** (δεν φαίνονται στη διεύθυνση URL).



Σχήμα 3

Ποια μέθοδο να διαλέξω; Στην ερώτηση αυτή δεν υπάρχει μία καθοριστική απάντηση. Κατά καιρούς έχουν ξεσπάσει έως και “ιδεολογικοί πόλεμοι” για τη σωστή απάντηση! Επιλέξτε με βάση την πρακτικότητα:

- Με τη μέθοδο **GET** μπορείτε να φτιάξετε υπερσυνδέσμους με δεδομένα του χρήστη ή να στείλετε δεδομένα απευθείας από τη γραμμή διεύθυνσης του browser (**βολικό για το debugging στο εργαστήριο!**).
- Από την άλλη πλευρά, όταν έχετε **πάρα πολλά** δεδομένα εισόδου, δεν είναι πρακτικό να τα περάσετε σε μια **τεράστια** διεύθυνση URL· χρησιμοποιείστε τη μέθοδο **POST** για τον σκοπό αυτόν.

Ο ρόλος της PHP

Όποια μέθοδο κι αν χρησιμοποιήσετε στη φόρμα σας, η PHP σας επιτρέπει να χειριστείτε τις μεταδιδόμενες τιμές με ενιαίο τρόπο. Παρέχονται 2 **πίνακες**, οι **\$_GET** και **\$_POST**, μέσω των οποίων μπορείτε να προσπελάσετε τα δεδομένα του χρήστη από φόρμες HTML, ανάλογα με τη μέθοδο που χρησιμοποιείται.

Για παράδειγμα, εάν ένα στοιχείο εισόδου έχει ιδιότητα **name = "sometext"**, τότε ό,τι συμπλήρωσε ο χρήστης στο στοιχείο αυτό λαμβάνεται στην PHP ως

```
$_GET["sometext"] // αν χρησιμοποιείτε τη μέθοδο GET  
ή  
$_POST["sometext"] // αν χρησιμοποιείτε τη μέθοδο POST
```



Προσοχή! Όλα τα δεδομένα από HTML φόρμες περνούν πάντοτε στο πρόγραμμα PHP ως **strings**!

Μπορείτε να δείτε **αν υπάρχει** μια μεταβλητή (παράμετρος εισόδου) με τη συνάρτηση **isset()**. Προσοχή: η **isset()** επιστρέφει TRUE ακόμα κι όταν η μεταβλητή υπάρχει αλλά είναι κενή!

```
if (isset($_POST["age"])) {  
    ...  
}
```

Η συνάρτηση **empty()** επιστρέφει TRUE αν η μεταβλητή είναι κενή, 0, "0" ή δεν έχει τεθεί.

```
if (empty($_GET["name"])) {  
    ...  
}
```



Προσοχή! Όταν στο πρόγραμμά σας PHP δέξετε είσοδο από φόρμες HTML **δεν πρέπει να κάνετε καμία υπόθεση για την τιμή που θα λάβετε από τα διάφορα στοιχεία εισόδου!** Οι κακόβουλοι χρήστες караδοκούν και μπορούν να σας στείλουν δεδομένα **χωρίς καν να χρησιμοποιήσουν τη φόρμα σας!** Έτσι, θα πρέπει **πάντοτε να ελέγχετε** αν οι τιμές που πήρατε είναι μέσα στα πλαίσια της εφαρμογής σας!

(περισσότερα για το θέμα αυτό όταν αναφερθούμε στα διάφορα στοιχεία εισόδου)

Παράρτημα Β: Στοιχεία φορμών HTML.

Το στοιχείο form

Το στοιχείο **form** περικλείει όλα τα στοιχεία εισόδου της φόρμας (είναι ο πατέρας τους). Περιέχει τις ιδιότητες **method** και **action**, όπως αυτές περιγράφηκαν στο Παράρτημα Α:

```
<form method="get(ή post)" action="...URL...">
```

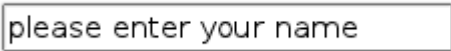

...όλα τα στοιχεία εισόδου της φόρμας ορίζονται εδώ...

```
</form>
```

Το στοιχείο **form** δεν έχει κανένα ορατό μέρος (δεν εμφανίζει κείμενο στην ιστοσελίδα).

Το στοιχείο input

Το στοιχείο **input** χρησιμοποιείται για τη δημιουργία διαφόρων στοιχείων εισόδου, ανάλογα με την ιδιότητα **type**:

type	στοιχείο εισόδου
text	πλαίσιο για εισαγωγή κειμένου, μία γραμμή 
radio	επιλογή μίας από σελ τιμών <input type="radio"/> ..-20 <input checked="" type="radio"/> 20-40 <input type="radio"/> 40-60 <input type="radio"/> 60-..
checkbox	επιλογή μίας η περισσότερων τιμών <input type="checkbox"/> a car <input type="checkbox"/> a bike <input type="checkbox"/> a motorcycle
submit	κουμπί αποστολής 

α) input, type=text

Ένα πλαίσιο κειμένου δημιουργείται όπως στο παράδειγμα:

```
<input type="text" name="username" value="enter your name">
```

Η ιδιότητα **value** είναι **προαιρετική** και, αν υπάρχει, ορίζει το αρχικό κείμενο που θα φανεί προ-συμπληρωμένο στο πλαίσιο κειμένου.

Τι στέλνεται κατά την αποστολή:

Ένα πλαίσιο κειμένου στέλνει πάντα ένα ζεύγος *όνομα=τιμή*, όπου το *όνομα* είναι η τιμή της ιδιότητας **name** και η *τιμή* είναι το περιεχόμενο κείμενο (ακόμα κι όταν είναι άδειο!).

Υποδοχή δεδομένων στην PHP:

- Αρχικά, χρησιμοποιήστε **πάντα** την **isset()** για να βεβαιωθείτε ότι τα δεδομένα υπάρχουν (μην ξεχνάτε ότι η αίτηση μπορεί να προέρχεται από οπουδήποτε αλλού εκτός από τη φόρμα σας).
- Αν περιμένετε κείμενο, ελέγξτε ότι το μήκος του string δεν είναι κενό με τη βοήθεια της **strlen()**:

```
if (strlen($_GET["name"])>0) {  
    ...  
}
```

- Μην ξεχνάτε ότι πάντα λαμβάνετε τα δεδομένα ως strings. Αν περιμένετε **αριθμητικά δεδομένα**, μετατρέψτε πριν χρησιμοποιήσετε με τη βοήθεια των συναρτήσεων **intval()** και **floatval()** (για ακεραίους και αριθμούς κινητής υποδιαστολής αντίστοιχα) όπως στο παράδειγμα:

```
$i=intval($_POST["age"]);
```

Προφυλάξεις:

- **ΠΟΤΕ!** δεν τυπώνετε απευθείας στην ιστοσελίδα σας ό,τι σας στέλνουν σε ένα πλαίσιο κειμένου! Πιθανόν να έχει οδηγίες HTML ή άλλα scripts που εκτελούν κακόβουλο κώδικα! Χρησιμοποιήστε τη συνάρτηση **htmlspecialchars()** για να αποφύγετε το ενδεχόμενο αυτό, όταν θέλετε να τυπώσετε ό,τι σας στέλνει ο χρήστης:

```
$userinput=htmlspecialchars($_GET["sometext"]);
```

β) input, type=radio

Κουμπιά για μοναδική επιλογή (πατώντας το ένα, σβήνουν τα άλλα), συναντώνται σε ομάδες, όπως στο παράδειγμα:

```
<input type="radio" name="age" value="1"> ..-20<br>  
<input type="radio" name="age" value="2" checked>20-40<br>  
<input type="radio" name="age" value="3">40-60<br>  
<input type="radio" name="age" value="4">60-..<br>
```

Παρατηρήστε ότι η ομαδοποίηση γίνεται μέσω της ιδιότητας **name**: όσα στοιχεία έχουν **το ίδιο όνομα**, ανήκουν **στην ίδια ομάδα** και ο χρήστης μπορεί να επιλέξει μόνο ένα από αυτά.

Χρησιμοποιήστε την ιδιότητα **checked** για να εμφανίσετε ένα εξ'ορισμού επιλεγμένο στοιχείο. Καλό είναι να υπάρχει πάντα ένα στοιχείο στην ομάδα ως αρχική επιλογή!

Τέλος, παρατηρήστε ότι τα στοιχεία radio buttons **δεν ορίζουν το κείμενο που θα δει ο χρήστης!** Το ορατό κείμενο βρίσκεται **έξω από τα στοιχεία** (στο παράδειγμα το ορατό κείμενο είναι σημειωμένο με κίτρινο).

Τι στέλνεται κατά την αποστολή:

Μία ομάδα radio buttons στέλνει **μόνο ένα** ζεύγος *όνομα=τιμή*, εκείνο του **επιλεγμένου στοιχείου**. Το *όνομα* είναι η τιμή της ιδιότητας **name** της ομάδας και η *τιμή* ισούται με την τιμή της ιδιότητας **value** για το επιλεγμένο στοιχείο. Αν δεν υπάρχει κανένα επιλεγμένο στοιχείο στην ομάδα, **τότε δεν στέλνεται απολύτως τίποτα!**

Υποδοχή δεδομένων στην PHP:

- **Αρχικά**, χρησιμοποιήστε **πάντα** την **isset()** για να βεβαιωθείτε ότι τα δεδομένα υπάρχουν (μη ξεχνάτε ότι η αίτηση μπορεί να προέρχεται από οπουδήποτε αλλού εκτός από τη φόρμα σας).
- **Στη συνέχεια**, χρησιμοποιήστε διακλαδώσεις if..else ή switch για να εκτελέσετε κώδικα ανάλογα με το επιλεγμένο radio button της ομάδας.

β) input, type=submit

Κουμπί αποστολής. Παράδειγμα:

```
<input type="submit" value="Send">
```

Η τιμή της ιδιότητας **value** χρησιμοποιείται ως **λεζάντα** του κουμπιού.

(το στοιχείο input, type=checkbox θα παρουσιαστεί στο επόμενο εργαστήριο)