

## Εργαστήριο #6

### Τι πρέπει να έχετε ολοκληρώσει από το προηγούμενο εργαστήριο.

- Θα πρέπει να ξέρετε να κατασκευάσετε μια ιστοσελίδα με φόρμα HTML και στοιχεία εισόδου τύπου text, radiobutton και submit. Θα πρέπει επίσης να μπορείτε να κατασκευάσετε την αντίστοιχη σελίδα PHP για την “υποδοχή” των δεδομένων που συμπληρώνει ο χρήστης στα προαναφερθέντα στοιχεία εισόδου.

### Οδηγίες

Στο σημερινό εργαστήριο θα δουλέψετε στο ίδιο σχήμα με την προηγούμενη φορά (μία σελίδα HTML με τη φόρμα και μία δευτέρα σελίδα PHP για την υποδοχή των δεδομένων).

⇒ 1. Πριν προχωρήσετε στον χειρισμό των *checkboxes*, πρέπει να μάθετε για τους πίνακες (*arrays*) στην PHP. Διαβάστε το **Παράρτημα Α!**

⇒ 2. Διαβάστε το **Παράρτημα Β** για το στοιχείο εισόδου *checkbox* σε HTML φόρμες.

⇒ 3. Κατασκευάστε ιστοσελίδα με φόρμα HTML, η οποία θα περιέχει μια σειρά *checkboxes*, επιτρέποντας στον χρήστη να επιλέξει ένα ή περισσότερα από τα  $x^2$ ,  $x^3$ ,  $x^4$  και  $x^5$ . Με την αποστολή της φόρμας θα εμφανίζεται (μέσω PHP) ο πίνακας του προηγούμενου εργαστηρίου με την διαφορά ότι θα έχει τόσες στήλες (εκτός από εκείνη του  $x$ ), όσες είναι και οι δυνάμεις που επέλεξε ο χρήστης.

⇒ 4. Διαβάστε το **Παράρτημα Γ** για το στοιχείο εισόδου *select* σε HTML φόρμες.

⇒ 5. Προσθέστε στη φόρμα σας μια λίστα *select*, η οποία θα επιτρέπει στον χρήστη να παρουσιάζει εναλλακτικά τις δυνάμεις των αριθμών 1-100, 101-200 ή 201-300. Συμπληρώστε αντίστοιχα το πρόγραμμα PHP.

### Παράρτημα Α: Πίνακες (*arrays*) στην PHP.

Στην PHP οι πίνακες αποτελούνται από ζεύγη (κλειδί:τιμή),(κλειδί:τιμή),.... Το **κλειδί** μπορεί να είναι **ακέραιος ή string**, ενώ η **τιμή** μπορεί να είναι **οποιοδήποτε τύπου**.

### Αρχικοποίηση *arrays*

Η βασική αρχικοποίηση ενός *array* γίνεται όπως στα παρακάτω παραδείγματα:

```
$arr = array(1, 7, 6, 9);
```

Αρχικοποίηση ενός array με αρχικά στοιχεία. Το στοιχείο `$arr[0]` θα είναι ίσο με **1**, το `$arr[3]` είναι ίσο με **9** κλπ. Η λέξη `array` είναι keyword!

```
$arr = array("αβγδ", 7, 6, "abcdef");
```

Τα στοιχεία μπορούν να είναι **διαφορετικού τύπου!** Π.χ. το στοιχείο `$arr[0]` θα είναι ίσο με **"αβγδ"** ενώ το `$arr[2]` θα είναι ίσο με **6**.

```
$arr = array(1 => "αβγδ", 7 => 4, "a" => 9);
```

Τα προηγούμενα παραδείγματα **υπονοούν** το κλειδί (δείκτη) κάθε στοιχείου (θα είναι 0, 1, 2...). Μπορείτε όμως **να ορίσετε εσείς** το κλειδί για κάθε στοιχείο αρχικοποίησης.

- Στο παράδειγμα, το στοιχείο `$arr["a"]` θα είναι ίσο με **9**, και το `$arr[7]` θα είναι ίσο με **4**.
- **Ναι**, τα κλειδιά (δείκτες πίνακα) **μπορούν να είναι μη συνεχόμενα!**
- **Ναι**, τα κλειδιά **μπορούν να είναι και μη αριθμητικά!**

```
$arr1 = array(array(1, 2, 3), array(8, "abc"));
```

```
$arr2 = array("a" => array(1, 2, 3),  
             "b" => array(8, "fff"),  
             "c" => array("white", "green", "blue"));
```

Μπορείτε να χρησιμοποιήσετε `array` **μέσα σε array** για πίνακες πολλών διαστάσεων. Εδώ για παράδειγμα, το στοιχείο `$arr2["b"][1]` είναι ίσο με **"fff"**.

```
$arr = array();
```

Έτσι κατασκευάζετε **έναν κενό πίνακα**.

### Προσπέλαση μεμονωμένων στοιχείων arrays

```
echo $arr[0]; // προσπέλαση θέσης πίνακα  
  
$arr[45] = 3; // εισαγωγή νέου στοιχείου στη θέση 45  
  
$arr[] = 23; // εισαγωγή στην επόμενη θέση από τον τρέχοντα  
// μέγιστο αριθμητικό δείκτη (=παράθεση!)  
  
unset($arr[5]); // διαγραφή μεμονωμένου στοιχείου
```



Προσέξτε την περίεργη σύνταξη της PHP: `$arr[] = 23;` όπου το 23 εισάγεται ως νέο στοιχείο στο τέλος του πίνακα \$arr – θα χρειαστεί αργότερα!).

### Διάσχιση arrays με loops

Εάν ο πίνακας έχει δηλωθεί με **συνεχή κλειδιά** (0,1,2,3 κλπ) κατά τον παραδοσιακό τρόπο, μπορείτε να προσπελάσετε όλα τα στοιχεία του με ένα **for**. Χρησιμοποιήστε την **count()** για να βρείτε τον αριθμό των στοιχείων του πίνακα:

```
$arr = array(1, 7, 6, 9);  
for ($i=0;$i<count($arr);$i++) {  
    echo $arr[$i];  
}
```

Μπορείτε να χρησιμοποιήσετε επίσης τη δομή **foreach** (χρήσιμη όταν ο πίνακας δεν έχει συνεχόμενα κλειδιά):

```
$colors = array('red', 'blue', 'green', 'yellow');  
foreach ($colors as $color) {  
    echo $color;  
}
```

Εδώ, σε κάθε επανάληψη, η μεταβλητή `$color` περιέχει την επόμενη τιμή (όχι κλειδί!) του array και θα εκτελεστούν τόσες επαναλήψεις, όσες οι τιμές που περιέχονται στο array.

### Παράρτημα B: Στοιχεία checkboxes.

- a car
- a bike
- a motorcycle

Τα checkboxes επιτρέπουν την επιλογή **μίας η περισσότερων** τιμών. Για την εισαγωγή checkboxes χρησιμοποιείται πάλι το στοιχείο **input**, με την ιδιότητα **type** να έχει την τιμή “checkbox”:

```
<input type="checkbox" name="a[]" value="c">a car<br>  
<input type="checkbox" name="a[]" value="b">a bike<br>  
<input type="checkbox" name="a[]" value="m">a motorcycle<br>
```

- Εάν χρησιμοποιείτε ως “γλώσσα υποδοχής” την PHP, τότε **συνιστάται** για όλα τα checkboxes της ομάδας να χρησιμοποιήσετε το ίδιο **name**, με την προσθήκη του [ ] στο τέλος. Αυτό θα οδηγήσει την PHP να τοποθετήσει σε

ένα array τα αντίστοιχα **value** των επιλεγμένων στοιχείων (βλ. λεπτομέρειες στην παράγραφο “Υποδοχή δεδομένων στην PHP”).

- Μπορείτε να χρησιμοποιήσετε την ιδιότητα **checked** (όπως στα radiobuttons) για εκείνα τα checkboxes που θέλετε να φανούν προεπιλεγμένα.
- Παρατηρήστε ότι τα στοιχεία checkboxes **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:

Ένα **επιλεγμένο** checkbox στέλνει το ζεύγος *όνομα=τιμή*, όπου το *όνομα* είναι η τιμή της ιδιότητας **name** και η *τιμή* προέρχεται από το αντίστοιχο **value**. Ένα μη επιλεγμένο checkbox δεν στέλνει τίποτα.

### Υποδοχή δεδομένων στην PHP:

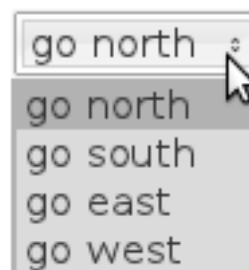
- Εάν χρησιμοποιήσετε το ίδιο name με την προσθήκη των [ ], η PHP κατά τη λήψη των δεδομένων θα δημιουργήσει ένα array με τα values των επιλεγμένων checkboxes (με τη σειρά που στέλνονται). Αυτό οφείλεται στη σύνταξη `$arr[] = value` που όπως είδαμε ισοδυναμεί με παράθεση νέων στοιχείων σε ένα array.
  - Π.χ. αν στο προηγούμενο σετ από checkboxes ο χρήστης επιλέξει τα “car” και “motorcycle” και η μέθοδος της φόρμας είναι το GET, τότε η PHP θα σας δώσει το `$_GET["a"]`, το οποίο θα είναι ένα array με δύο τιμές: το “c” και το “m”.
- 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). Για τον ίδιο λόγο, βεβαιωθείτε με τη βοήθεια της **is\_array()** ότι η μεταβλητή σας **είναι όντως array!**
- **Στη συνέχεια** χρησιμοποιήστε τη δομή **foreach** για να προσπελάσετε κάθε στοιχείο του array αυτού και, ανάλογα με την τιμή του, εκτελέστε την επιθυμητή ενέργεια :

```
if (isset($_GET['a']) && is_array($_GET['a'])) {  
    foreach ($_GET['a'] as $checkval) {  
        // do something with $checkval here...  
    }  
}
```

### Παράρτημα Γ: Στοιχεία select.

Τα στοιχεία select επιτρέπουν την επιλογή από μια προκαθορισμένη λίστα. Η λίστα αυτή είναι σταθερή: δεν μπορεί ο χρήστης να προσθέσει νέες επιλογές.

Όσον αφορά τη λειτουργικότητα, τα στοιχεία select είναι παρόμοια με τα στοιχεία radio buttons: επιτρέπουν την επιλογή μίας από πολλές τιμές.



(τα στοιχεία `select` μπορούν εναλλακτικά να χρησιμοποιηθούν για την επιλογή πολλαπλών τιμών· η περίπτωση αυτή όμως δεν θα καλυφθεί στο εργαστήριο)

Ένα στοιχείο `select` εισάγεται ως εξής:

```
<select name="goto">
  <option value="N" selected>go north</option>
  <option value="S">go south</option>
  <option value="E">go east</option>
  <option value="W">go west</option>
</select>
```

- Παρατηρήστε ότι, σε αντίθεση με τα άλλα στοιχεία `input`, τα στοιχεία `option` (παιδιά του στοιχείου `select`) περιέχουν το κείμενο που θα δει ο χρήστης! (στο παράδειγμα το ορατό κείμενο είναι σημειωμένο με κίτρινο).
- Η προαιρετική ιδιότητα `selected` επιτρέπει την εμφάνιση μιας προεπιλεγμένης τιμής της λίστας.

#### Τι στέλνεται κατά την αποστολή:

Ένα στοιχείο `select` έχει παρόμοια συμπεριφορά με τα `radio buttons`: στέλνει το ζεύγος `όνομα=τιμή`, όπου το όνομα είναι η τιμή της ιδιότητας `name` του `select` και η τιμή ισούται με την τιμή της ιδιότητας `value` για το επιλεγμένο στοιχείο της λίστας.

#### 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).
- Στη συνέχεια, χρησιμοποιήστε διακλαδώσεις `if..else` ή `switch` για να εκτελέσετε κώδικα ανάλογα με το επιλεγμένο στοιχείο της λίστας.