Εργαστήριο #11

Από τα προηγούμενα εργαστήρια:

Το εργαστήριο αυτό είναι συνέχεια του 10ου, το οποίο θα **πρέπει να έχετε ολοκληρώσει** (τουλάχιστον χωρίς τη μορφοποίηση!). Θα χρειαστείτε επίσης ορισμένες από τις οδηγίες μορφοποίησης CSS (ανατρέξτε στις εκφωνήσεις του 8ου και 9ου εργαστηρίου).

Οδηγίες

Ακολουθήστε τα παρακάτω βήματα. **Βεβαιωθείτε ότι το πρόγραμμά σας δουλεύει σωστά σε κάθε βήμα, πριν προχωρήσετε στο επόμενο**.

⇒ *Βήμα 1ο*.

Στο σημερινό εργαστήριο, εκτός από τον πίνακα `books` της βάσης δεδομένων, θα χρησιμοποιήσετε και τον πίνακα `courses`, ο οποίος έχει τις εξής στήλες:

1. `id`: αναγνωριστικό γραμμής.
2. `title`: ο τίτλος του μαθήματος.
3. `code`: ο κωδικός του μαθήματος στο σύστημα της γραμματείας.
4. `semester`: το εξάμηνο του μαθήματος.
5. `type`: το είδος του μαθήματος.
6. `hours`: οι διδακτικές ώρες του μαθήματος.
7. `labhours`: οι εργαστηριακές/φροντιστηριακές ώρες του μαθήματος.
8. `ects`: οι μονάδες ECTS του μαθήματος.
9. `description`: κείμενο περιγραφής του μαθήματος.

Και ο πίνακας αυτός είναι συμπληρωμένος με τα μαθήματα του Α' εξαμήνου, οπότε αρκεί να ανακτήσετε την πληροφορία του. Ο στόχος του 1ου βήματος είναι να απεικονίσετε την πληροφορία για ένα (επιλεγμένο) μάθημα.

Διαβάστε το **Παράρτημα Α**: πώς μπορείτε με μία εντολή SELECT της SQL να πάρετε ορισμένα δεδομένα με κάποιο κριτήριο; Πώς θα το κάνετε μέσω PHP; Και τι είναι η σοβαρή απειλή ασφάλειας που ονομάζεται "SQL injection";

Στη συνέχεια ξεκινώντας από το υπόδειγμα κώδικα του 10ου εργαστηρίου προσθέστε μία συνάρτηση `printcourse()`:

- Η συνάρτηση θα δέχεται ως όρισμα τον κωδικό (code) του μαθήματος και θα δημιουργεί έναν πίνακα με την πληροφορία του συγκεκριμένου μαθήματος (αν ο κωδικός υπάρχει στη βάση).
- Η συνάρτηση θα καλείται μέσα στο try block του υποδείγματος και θα πρέπει να έχει πρόσβαση στη μεταβλητή $db της σύνδεσης με τη βάση: δηλώστε το $db ως global ή περάστε το σαν πρόσθετη παράμετρο!
- Η πληροφορία που εμφανίζεται θα πρέπει να είναι μορφοποιημένη σε έναν πίνακα HTML, παρόμοιο με του site του Τμ.Πληροφορικής (π.χ. βλ.

http://di.ionio.gr/el/undergraduate-studies/modules/15/20-introduction-to-computer-science.html).

- Δοκιμάστε τη συνάρτηση δίνοντας ως όρισμα το "HY010" (ελληνικοί χαρακτήρες).

⇒ *Βήμα 2ο*.

Διαβάστε το **Παράρτημα Β** για να μάθετε για τη λειτουργία JOIN σε βάσεις δεδομένων. Στη συνέχεια επεκτείνετε την `printcourse()`, έτσι ώστε να εμφανίζονται και τα προτεινόμενα βιβλία για το επιλεγμένο μάθημα.

- Ανακτήστε την πληροφορία των βιβλίων για ένα μάθημα συνδυάζοντας (join) τους πίνακες `books` και `courses` μέσω της στήλης `course_id` του πρώτου πίνακα.

⇒ *Βήμα 3ο*.

Προσθέστε στον κώδικά σας τη συνάρτηση `printform()`, η οποία:

- Θα δέχεται ως όρισμα τον κωδικό (code) ενός επιλεγμένου μαθήματος.
- Θα τυπώνει μια φόρμα HTML **με μοναδικό στοιχείο** ένα στοιχείο select (δηλαδή, χωρίς κουμπί αποστολής!), για τη δημιουργία μιας drop-down λίστας με όλα τα μαθήματα που είναι διαθέσιμα στη βάση: ανακτήστε τους κωδικούς (code) και τους τίτλους (title) μέσω ενός ερωτήματος SELECT στη βάση.
- Εάν μια επιλογή ταιριάζει με τον κωδικό που δόθηκε ως όρισμα εισόδου, το μάθημα αυτό θα πρέπει να φαίνεται επιλεγμένο. Επίσης, η συνάρτηση θα επιστρέφει true. Σε αντίθετη περίπτωση, αφήστε να φαίνεται η πρώτη επιλογή, η οποία θα έχει τίτλο "επιλέξτε μάθημα". Στην τελευταία αυτή περίπτωση, η συνάρτηση θα επιστρέφει false.
- Η συνάρτηση θα καλείται μέσα στο try block του υποδείγματος και θα πρέπει να έχει πρόσβαση στη μεταβλητή $db της σύνδεσης με τη βάση: δηλώστε το $db ως global ή περάστε το σαν πρόσθετη παράμετρο!

Καλέστε δοκιμαστικά τη συνάρτηση `printform()`, πριν την κλήση της συνάρτησης `printcourse()`. Όταν όλα είναι σωστά, προχωρήστε στον τελικό κώδικα, **πάντα μέσα στο try block**, ως εξής:

1. Αρχικά ελέγξτε αν υπάρχει (isset) παράμετρος επιλογής μαθήματος από προηγούμενη αποστολή της φόρμας.
2. Καλέστε την `printform()`, περνώντας ως όρισμα το επιλεγμένο μάθημα (αν υπάρχει, αλλιώς περάστε την τιμή null).
3. Αν επιστρέψει true, καλέστε την `printcourse()`, περνώντας επίσης ως όρισμα το επιλεγμένο μάθημα.

Θα παρατηρήσατε ότι δεν υπάρχει κουμπί αποστολής της φόρμας! Διαβάστε το **Παράρτημα Γ** για να δείτε πώς μπορείτε να στέλνετε τη φόρμα χωρίς κουμπί, κάθε φορά που ο χρήστης επιλέγει μια διαφορετική επιλογή.

Παράρτημα Α: Ερωτήματα SQL με παραμέτρους

1. SQL ερωτήματα SELECT με κριτήρια επιλογής.

Συνήθως από έναν πίνακα της βάσης δεδομένων θέλουμε να ανακτήσουμε ορισμένες μόνο σειρές, σύμφωνα με κάποιο κριτήριο. Η σύνταξη της εντολής SELECT για να το επιτύχουμε αυτό είναι όπως στο εξής παράδειγμα:

```
SELECT title,authors FROM books WHERE id<5;
```

Η προσθήκη του **WHERE**, ακολουθούμενου από κάποια συνθήκη επιλογής, μας επιτρέπει να επιλέξουμε συγκεκριμένες γραμμές από έναν πίνακα. Η συνθήκη μπορεί να περιέχει μεταξύ άλλων τα = (ίσο) και <> (διάφορο), καθώς και τα αριθμητικά μικρότερο, μεγαλύτερο κ.λ.π. Επίσης, η συνθήκη μπορεί να είναι σύνθετη, αποτελούμενη από πολλές μικρότερες, ενωμένες με τα **AND** και **OR** (υπάρχει και το **NOT**).

2. Η απειλή του SQL Injection.

Πάρα πολύ συχνά, ένα ερώτημα SQL είναι" παραμετροποιημένο": η συνθήκη που θα βάλετε στο WHERE προέρχεται από μια μεταβλητή της PHP, ίσως από είσοδο από τον χρήστη. Πώς θα φτιάξετε το τελικό ερώτημα SQL;

Η πιο κάτω τακτική είναι λάθος και εγκυμονεί μεγάλους κινδύνους:

```
// ας σχηματίσουμε το string της ερώτησης...
$code = $_GET['userinput'];
$q = 'select title from courses where code="'.$code.'"';
// ... το οποίο στέλνουμε στη βάση
$st = $db->query($q);
// ... συνέχεια με $st->fetch() ή $st->fetchAll()
```

Έτσι δίνετε την ευκαιρία στον κακόβουλο χρήστη να επιτεθεί στο site σας μέσω της δημοφιλούς μεθόδου που ονομάζεται "SQL injection": η εισαγωγή κακόβουλων εντολών SQL προς τη βάση σας. Δείτε το παρακάτω παράδειγμα, όπου η μεταβλητή `$code` προέρχεται από δεδομένα από φόρμα HTML, τα οποία δεν έχετε ελέγξει!

```
// εάν η μεταβλητή $code περιέχει το εξής:
// 'bogus"; άλλη κακόβουλη εντολή SQL; --'
// τότε στο προηγούμενο παράδειγμα, θα στείλετε στη βάση:
select title from courses where code="bogus";
άλλη κακόβουλη εντολή SQL; -- "
```

Αυτό που θα στείλετε στη βάση είναι δύο εντολές (το ; διαχωρίζει SQL εντολές και το - - εισάγει σχόλια στην SQL), με τη δεύτερη να ανήκει στον κακόβουλο χρήστη: μπορεί να δοκιμάσει να διαγράψει δεδομένα, να υποκλέψει κωδικούς κ.ά...

3. Ερωτήματα μέσω PHP PDO με παραμέτρους.

Η PHP μας δίνει τη δυνατότητα να κατασκευάσουμε ερωτήματα SQL με παραμέτρους μέσω των λεγόμενων "προετοιμασμένων ερωτημάτων" (**prepared statements**): μπορείτε να τα δείτε ως προκατασκευασμένα ερωτήματα με θέσεις για τις παραμέτρους.

Τα "προετοιμασμένα ερωτήματα" χρησιμοποιούνται για ταχύτητα σε πολλαπλά ίδια ερωτήματα, αλλά επίσης και γιατί **φροντίζουν οι παράμετροι που εισέρχονται στα ερωτήματα να "εξουδετερώνονται" (escaped) κατάλληλα, έτσι ώστε να μην αποτελούν απειλή!**

Κατά συνέπεια, όταν βάζετε δεδομένα που δεν εμπιστεύεστε σε SQL ερωτήματα, χρησιμοποιήστε πάντα τη μέθοδο των προετοιμασμένων ερωτημάτων, όπως εδώ:

```
// δημιουργία προετοιμασμένου ερωτήματος
$st = $db->prepare("select title from courses where code=?");
// το array με τις παραμέτρους (εδώ είναι μόνο η $code)
$params = array($code);
// αντικατάσταση παραμέτρων και εκτέλεση
if ($st->execute($params)) {     // true σε επιτυχές ερώτημα
  // ... συνέχεια με $st->fetch() ή $st->fetchAll()
}
```

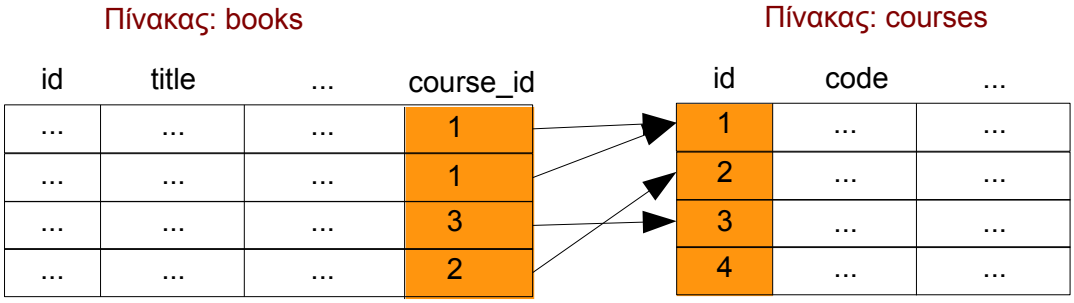
Παρατηρήστε ότι:

- Πρώτα προετοιμάζουμε το ερώτημα με την **prepare()** και μετά το εκτελούμε με την **execute()**.
- Η prepare() δέχεται ερωτήματα SQL όπου στη θέση των παραμέτρων έχει τοποθετηθεί το **?**.
- Στη συνέχεια, περνάμε στην execute() ένα array με τις παραμέτρους που θα αντικαταστήσουν τα ? (με τη σειρά εμφάνισης) πριν την εκτέλεση.

Η χρήση του ? είναι η απλούστερη περίπτωση: υπάρχουν και άλλες δυνατότητες με τα προετοιμασμένα ερωτήματα που δεν θα καλυφθούν όμως στο εργαστήριο. Για περισσότερες λεπτομέρειες δείτε στο http://www.php.net/manual/en/pdo.prepared-statements.php .

Παράρτημα Β: Λειτουργία JOIN σε πίνακες βάσης δεδομένων

Στον πίνακα `books` του προηγουμένου εργαστηρίου, έχει προστεθεί η στήλη `course_id`, η οποία, για κάθε βιβλίο, περιέχει την τιμή της στήλης `id` του αντίστοιχου μαθήματος στον πίνακα `courses` (**σχήμα 1**).

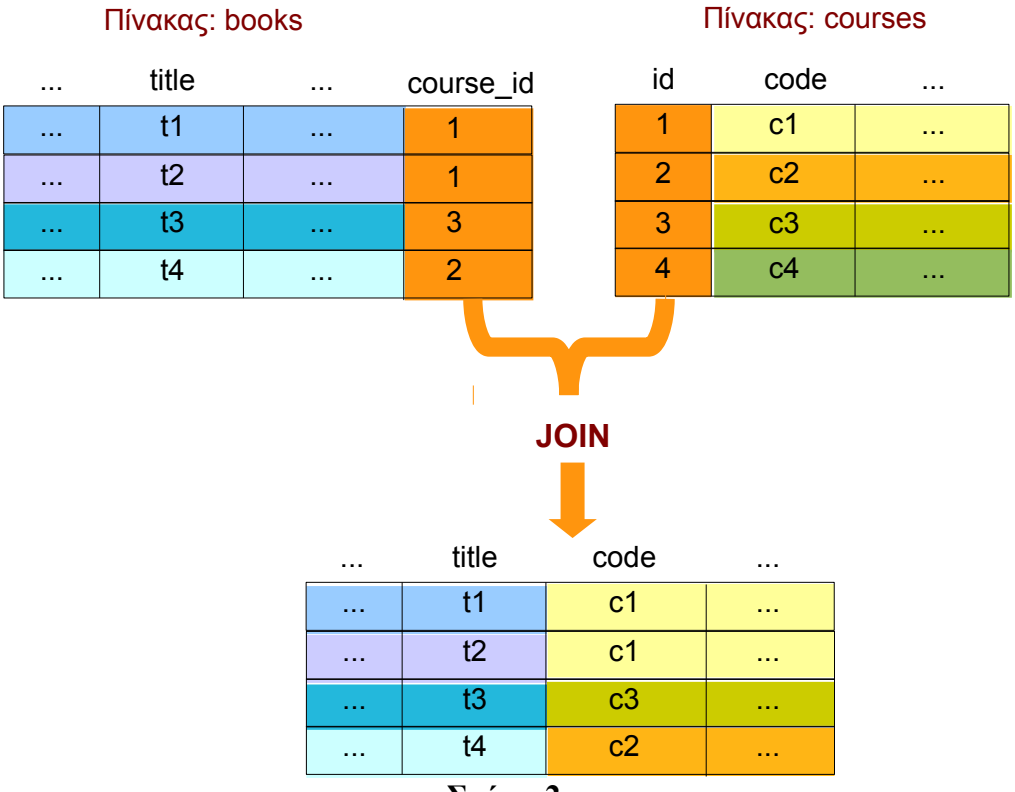


Σχήμα 1

Μέσω αυτής της **ταύτισης τιμών** του `books.course_id` και του `courses.id`, δηλώνεται η σχέση κάθε βιβλίου με το αντίστοιχο μάθημα. Αυτός είναι ο τρόπος των σχεσιακών βάσεων δεδομένων για τη δήλωση **σχέσεων** μεταξύ **οντοτήτων**.

Λειτουργία join.

Όταν υπάρχουν σχέσεις μεταξύ πινάκων, συχνά η πληροφορία αυτή πρέπει να αποτυπωθεί και στο ερώτημα SELECT. Η λειτουργία join κάνει ακριβώς αυτό (**σχήμα 2**):



Σχήμα 2

Η σύνταξη του ερωτήματος SELECT μπορεί να συμπεριλάβει τη λειτουργία join, όπως φαίνεται στο ακόλουθο παράδειγμα, το οποίο βρίσκει τους τίτλους κάθε βιβλίου και το εξάμηνο που διανέμεται:

```
SELECT books.title,courses.semester
FROM books INNER JOIN courses
ON courses.id=books.course_id;
```

Παρατηρήστε ότι όταν και οι δυο πίνακες έχουν στήλη με το ίδιο όνομα, **πρέπει να αποσαφηνίσετε ποια θέλετε**, όπως στην περίπτωση του `books.title`! Αντιθέτως, το `courses.semester` του παραδείγματος μπορεί να γραφεί και ως `semester` μόνο.

Τέλος, μπορείτε να συνδυάσετε με το join και με την επιλογή WHERE:

```
SELECT books.title,courses.semester
FROM books INNER JOIN courses
ON courses.id=books.course_id
WHERE courses.type="Κορμού";
```

Προσοχή! Όταν ζητάτε πεδία όπως το `books.title`, η PHP μέσω της `fetch()` ή της `fetchAll()` θα επιστρέψει την τιμή στο `$row['title']` κι όχι στο `$row['books.title']`! Αν αυτό σας δημιουργεί πρόβλημα, θυμηθείτε ότι μπορείτε να προσπελάσετε τα πεδία και με αριθμητικό δείκτη (με τη σειρά που τα ζητήσατε στη SELECT), π.χ. `$row[0]`.

Παράρτημα Γ: Προγραμματισμός στην πλευρά του browser

Στο παράρτημα αυτό θα διαβάσετε για τις *πολύ βασικές* έννοιες του προγραμματισμού "στην πλευρά του web client" (**client-side programming**), δηλαδή στον **browser**.

1. Τι σημαίνει "application scripting";

Πολλές σύνθετες εφαρμογές (applications), προγραμματισμένες στη γλώσσα `X`, παρέχουν "προγραμματιστική" πρόσβαση στα αντικείμενά τους μέσω μιας δεύτερης γλώσσας `Y`. Εκτός από την κλασσική επικοινωνία του με την εφαρμογή μέσω γραφικού περιβάλλοντος ή κονσόλας, ο χρήστης μπορεί να εκτελέσει λειτουργίες γράφοντας προγράμματα στη γλώσσα `Y`!

Φανταστείτε ότι ο χρήστης γράφει "σενάρια" (scripts) που περιγράφουν ακολουθίες λειτουργιών στην εφαρμογή:

- Ο **προγραμματισμός των λειτουργιών** της εφαρμογής ονομάζεται "**application scripting**"
- Ο **κώδικας** που περιγράφει τις λειτουργίες ονομάζεται "**script**".

2. Τι είναι ο προγραμματισμός "στην πλευρά του web client";

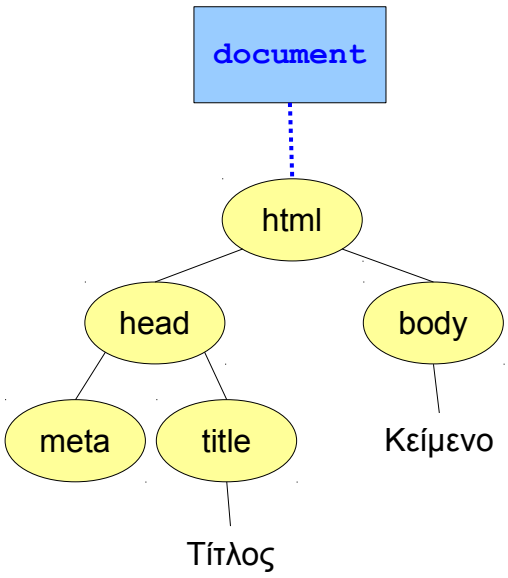
Πολύ απλά, **η δυνατότητα scripting στην εφαρμογή του browser**! Δείτε το ως το αντίθετο της PHP: η PHP εκτελείται στην πλευρά του web-server (server-side programming) και ετοιμάζει την ιστοσελίδα που θα αποσταλεί στον browser, ενώ ο προγραμματισμός στον browser επιτρέπει να αλληλεπιδρούμε δυναμικά με τα στοιχεία HTML μιας ιστοσελίδας που έχει ήδη ληφθεί από το Διαδίκτυο.

Η κύρια (και μοναδική πλέον) **γλώσσα για scripting** στον browser είναι η **JavaScript**. Η γλώσσα **δεν έχει καμία σχέση με την γλώσσα Java**! Απλά, η ονομασία αυτή αντικατοπτρίζει ένα "τέχνασμα" του marketing την εποχή που πρωτοεμφανίστηκε η JavaScript.

Η JavaScript είναι αντικειμενοστρεφής: θα προγραμματίσετε τις λειτουργίες του browser αλληλεπιδρώντας με "αντικείμενα", καλώντας τις "μεθόδους" τους και αλλάζοντας τιμές στις "ιδιότητές" τους.

3. Με ποια αντικείμενα μπορούμε να αλληλεπιδράσουμε προγραμματιστικά στον browser;

Με όλα τα στοιχεία HTML μιας ιστοσελίδας που βρίσκεται στον browser μας (και όχι μόνο)! Θυμηθείτε από το 2ο εργαστήριο ότι τα στοιχεία της ιστοσελίδας σχηματίζουν μια δενδρική δομή αντικειμένων που ονομάζεται **DOM (Document Object Model)**. Το δέντρο αυτό μπορείτε να το προσπελάσετε προγραμματιστικά μέσα από ένα κεντρικό αντικείμενο που αντιπροσωπεύει την τρέχουσα ιστοσελίδα και ονομάζεται **document** (**σχήμα 3**):



Σχήμα 3

Στο σημερινό εργαστήριο θα χρειαστείτε μόνο τα εξής:

```
Το κεντρικό αντικείμενο της ιστοσελίδας:
document


Μια συλλογή με όλα τα στοιχεία form της ιστοσελίδας:
document.forms


Το αντικείμενο που αντιπροσωπεύει ένα form με την ιδιότητα
name ή την ιδιότητα id ίση με 'formname':
document.forms['formname']


Εάν έχετε ένα αντικείμενο που αντιπροσωπεύει ένα form, τότε
μπορείτε να προκαλέσετε προγραμματιστικά την αποστολή του μέσω
της μεθόδου του submit(), π.χ.:
document.forms['formname'].submit()
```

4. Πότε εκτελείται ο κώδικας JavaScript; Και πού τον προσθέτω;

Θα θυμάστε ότι η PHP εκτελείται στον web server μόλις ζητήσετε μια ιστοσελίδα αρχίζοντας από την αρχή της ιστοσελίδας με τα δεδομένα εισόδου του χρήστη, έως το τέλος του κώδικα. Από αυτή την άποψη, ένα πρόγραμμα PHP μοιάζει με ένα κλασσικό πρόγραμμα σειριακής εκτέλεσης τύπου "κονσόλας", π.χ. σε C.

Η JavaScript εκτελείται στον browser, ο οποίος είναι μια σύνθετη εφαρμογή σε γραφικό περιβάλλον. Συνεπώς, δεν έχει νόημα το μοντέλο της σειριακής εκτέλεσης!

Αντιθέτως, η JavaScript εκτελεί κομμάτια κώδικα αντιδρώντας σε **γεγονότα (events)** που προκαλεί η αλληλεπίδραση του χρήστη με το γραφικό περιβάλλον (επιλογή στοιχείων, τσεκάρισμα επιλογών, μετακίνηση ποντικιού...), η έλευση νέων δεδομένων από το Διαδίκτυο, η ολοκλήρωση μέτρησης ενός timer κ.ά.

Το μοντέλο προγραμματισμού μέσω της αντίδρασης σε γεγονότα ονομάζεται "οδηγούμενο από γεγονότα" (**event-driven programming**). Σε κάθε γεγονός που εμφανίζεται καλείται η αντίστοιχη συνάρτηση χειρισμού του (**event handler**). Κάθε στοιχείο HTML του browser προκαλεί την εμφάνιση διαφορετικών γεγονότων, ανάλογα με το είδος του.

Στο σημερινό εργαστήριο θα χρειαστείτε το γεγονός **onchange** που εκπέμπει ένα στοιχείο select, όταν ο χρήστης επιλέγει μια νέα επιλογή. Ο κώδικας χειρισμού του μπορεί να τοποθετηθεί μέσα στην HTML ως εξής:

```
<select name="test" onchange="κώδικας χειρισμού γεγονότος">
```

Θα πρέπει εδώ να σημειωθεί ότι η ανάμιξη του κώδικα JavaScript μέσα στην HTML δεν αποτελεί την πλέον πρότυπη λύση· από την άλλη πλευρά, αποτελεί τον ευκολότερο τρόπο για πολύ μικρά κομμάτια κώδικα!

Σύμφωνα με τα παραπάνω, για να επιτύχετε την αποστολή της φόρμας του σημερινού εργαστηρίου θα πρέπει:

1. Να δώσετε ένα όνομα μέσω της ιδιότητας **name** στο στοιχείο **form** που περιέχει τη λίστα select για την επιλογή μαθημάτων.
2. Να προσθέσετε την ιδιότητα **onchange** στο στοιχείο select.
3. Ως κώδικα χειρισμού του onchange να προσθέσετε τον κατάλληλο κώδικα JavaScript για την αποστολή του form – βλ. το παράδειγμα με την **submit()**!