

## Εργαστήριο #13

### Από τα προηγούμενα εργαστήρια:

Θα πρέπει να έχετε ολοκληρώσει το 11<sup>ο</sup> (προσπέλαση βάσης δεδομένων μέσω PHP και 12<sup>ο</sup> εργαστήριο (βασικές τεχνικές JavaScript στον browser).

### Οδηγίες

Στο τελευταίο εργαστήριο θα υλοποιήσετε ό,τι ακριβώς και στο 11<sup>ο</sup> (σύνδεση με τη βάση δεδομένων και παρουσίαση σε πίνακα των πληροφοριών ενός μαθήματος) αλλά με τελείως διαφορετικό τρόπο: η τεχνική που θα ακολουθήσετε ονομάζεται **Ajax (Asynchronous JavaScript & XML)**. Με τον γενικό (και μάλλον marketing) όρο Ajax περιγράφονται **εφαρμογές στο web**, οι οποίες αντλούν πληροφορίες σε **μη-HTML μορφή** από τρίτες πηγές (services) και τις **απεικονίζουν δυναμικά, χωρίς να ανανεώνουν την ιστοσελίδα** τους από τον web server.

Το εργαστήριο έχει δύο μέρη:

- Στο πρώτο μέρος θα υλοποιήσετε την υπηρεσία παροχής των “ωμών” δεδομένων, με ένα **αρχείο PHP**.
- Στο δεύτερο μέρος θα υλοποιήσετε την εφαρμογή web που καταναλώνει (απεικονίζει) την προηγούμενη πληροφορία: η εφαρμογή αυτή θα αποτελείται από ένα **αρχείο HTML** και το αντίστοιχο **αρχείο JavaScript**.

### Πρώτο Μέρος: Υπηρεσία παροχής δεδομένων.

#### ⇒ Βήμα 1<sup>ο</sup>.

Διαβάστε στο **Παράρτημα A.1** πώς θα γράψετε ένα πρόγραμμα PHP για την παροχή δεδομένων σε μορφή διαφορετική από την HTML.

#### ⇒ Βήμα 2<sup>ο</sup>.

Διαβάστε στο **Παράρτημα A.2** σχετικά με το μορφότυπο JSON και στο **Παράρτημα A.3** πώς θα στείλετε δεδομένα στη μορφή αυτή μέσω της PHP.

Στη συνέχεια υλοποιήστε το εξής HTTP GET ερώτημα προς την υπηρεσία σας (απεικονίζεται μαζί με το τι θα πρέπει να επιστρέφεται σε μορφή JSON):

#### Αίτημα:

```
http://yourserver/~youruser/yourfile.php?q=listcourses
```

#### Απάντηση:

```
{"courses": [{"code": "κωδ1", "title": "τιτλ1"},  
              {"code": "κωδ2", "title": "τιτλ2"},  
              ...  
              {"code": "κωδn", "title": "τιτλn"}]}
```

Διαδικασία:

- Τα `κωδ1..κωδη` και τα `τιτλ1...τιτλη` θα αντικατασταθούν από τα στοιχεία της βάσης δεδομένων για όλα τα μαθήματα. Προσαρμόστε τη συνάρτηση `printform()` του 11<sup>ου</sup> εργαστηρίου για τον σκοπό αυτόν!

**Υπόδειξη:** αν περάσετε ως όρισμα στη `fetch()` ή στη `fetchAll()` τη σταθερά `PDO::FETCH_ASSOC`, τότε θα λάβετε τα δεδομένα από τη βάση ως array με δείκτες (κλειδιά) **μόνο τα ονόματα των στηλών**: πολύ κοντά στη μορφή που θέλετε να περάσετε στην `json_encode()`!

- Το κυρίως προγράμμα σας θα πρέπει να ελέγχει (`isset`) αν υπάρχει η μεταβλητή `$_GET['q']` και αν έχει τιμή ίση με `'listcourses'`. Αν ναι, καλέστε την αναμορφωμένη `printform()`, για να στείλετε την κατάλληλη απάντηση.
- Αν δεν υπάρχει η `$_GET['q']` ή δεν έχει τιμή `'listcourses'`, απαντήστε με ένα μήνυμα λάθους, επίσης σε μορφή JSON:

**HTTP κωδικός:**

**HTTP/1.0 400 Bad Request**

**Μήνυμα σφάλματος:**

```
{"error":{"message":"input parameters error"}}
```

Δείτε στο [Παράρτημα A.4](#) πώς θα στείλετε πρώτα τον ζητούμενο κωδικό HTTP.

Όταν ολοκληρώσετε το βήμα αυτό, **ελέγξτε** εισάγοντας τη διεύθυνση URL του αιτήματος στον browser. Θα πρέπει να βλέπετε τη σωστή απάντηση JSON.

*(μην ανησυχείτε αν δεν βλέπετε τους ελληνικούς χαρακτήρες· κατά τη λήψη τους από την εφαρμογή web θα μετατραπούν σωστά)*

[Δεύτερο Μέρος: Εφαρμογή web.](#)

⇒ **Βήμα 1<sup>ο</sup>.**

Ξεκινήστε με την κατασκευή της ιστοσελίδας της εφαρμογής. Συνδέστε με το αντίστοιχο αρχείο JavaScript. Μέσα στο body της ιστοσελίδας τοποθετήστε τα εξής:

```
<form>
  <select id="courses">
    <option value="_">Επιλέξτε Μάθημα</option>
  </select>
```

```
</form>
<div id="tableplace">
</div>
```

### ⇒ Βήμα 2°.

Διαβάστε στο **Παράρτημα Β.1** για το περίφημο **XMLHttpRequest**. Στη συνέχεια, μέσα στη συνάρτηση του `window.onload` προσθέστε κώδικα για να φέρνετε τη λίστα των μαθημάτων από την υπηρεσία που φτιάξατε προηγουμένως. Η υποδοχή των δεδομένων θα γίνεται από συνάρτηση που θα γράψετε στο επόμενο βήμα.

### ⇒ Βήμα 3°.

Γράψτε τη συνάρτηση επεξεργασίας των δεδομένων που λήφθηκαν: αφού μετατρέψετε τα δεδομένα JSON σε αντικείμενο της JavaScript (βλ. **Παράρτημα Β.2**), στόχος σας είναι να τα προσθέσετε ως στοιχεία `<option>` μέσα στο `<select>` με id ίσο με 'courses'. Για να δείτε πώς μπορείτε προγραμματιστικά να τροποποιήσετε το περιεχόμενο μιας ιστοσελίδας δείτε το **Παράρτημα Γ.1**.

Αφού ελέγξετε το αποτέλεσμα (θα πρέπει να εμφανίζονται οι επιλογές των μαθημάτων στο στοιχείο `select`), στο τέλος της ίδιας συνάρτησης συνδέστε το γεγονός **onchange** του `select` με μια νέα συνάρτηση (προς το παρόν κενή).

## Πρώτο Μέρος (συνέχεια): Υπηρεσία παροχής δεδομένων.

### ⇒ Βήμα 3°.

Προσθέστε τη δυνατότητα στο πρόγραμμά σας PHP να δίνει πληροφορίες σχετικές με ένα συγκεκριμένο μάθημα. Οι λεπτομέρειες αίτησης-απάντησης είναι:

#### Αίτημα:

```
http://yourserver/~youruser/yourfile.php?q=courseinfo&code=κωδ
```

#### Απάντηση:

```
{
  "title": "τιτλ", "code": "κωδ", "semester": "εξαμ", "type": "τυπ",
  "hours": "ωρ", "labhours": "εργ", "ects": "μον", "description": "περ",
  "books": [
    {
      "title": "βτιτλ1", "authors": "συγγρ1", "publisher": "εκδ1",
      "isbn": "αρ1", "edition": "εκδ1",
    },
    {
      "title": "βτιτλ2", "authors": "συγγρ2", "publisher": "εκδ2",
      "isbn": "αρ2", "edition": "εκδ2",
    },
    ...
    {
      "title": "βτιτλn", "authors": "συγγρn", "publisher": "εκδn",
      "isbn": "αρn", "edition": "εκδn",
    }
  ]
}
```

**Σφάλμα:** αν το μάθημα με τον κωδικό αυτόν δεν υπάρχει

**HTTP κωδικός:**

**HTTP/1.0 404 Not Found**

**Μήνυμα σφάλματος:**

```
{"error":{"message":"the course requested does not exist"}}
```

- Για να απαντάτε στο ερώτημα αυτό τροποποιήστε κατάλληλα τη συνάρτηση `printcourse()` του 11<sup>ου</sup> εργαστηρίου.
- Στο κυρίως πρόγραμμα προσθέστε την αναγνώριση του `$_GET['q']` με τιμή `'courseinfo'`, ελέγχοντας ταυτόχρονα ότι υπάρχει το `$_GET['code']` και ότι έχει τιμή με μήκος (`strlen`) μεγαλύτερο του μηδενός.

Ελέγξτε τη λειτουργία του προγράμματος, εισάγοντας το κατάλληλο URL στον browser.

Συμπληρωματικά, προσθέστε τον κατάλληλο κώδικα, έτσι ώστε σε περίπτωση σφάλματος επικοινωνίας με τη βάση δεδομένων (στην πρόταση `catch (PDOException)`) να στέλνετε το εξής:

**HTTP κωδικός:**

**HTTP/1.0 500 Internal Server Error**

**Μήνυμα σφάλματος:**

```
{"error":{"message":"database error"}}
```

## [Δεύτερο Μέρος \(συνέχεια\): Εφαρμογή web.](#)

### ⇒ Βήμα 4<sup>ο</sup>.

Συνεχίστε με την κατασκευή της συνάρτησης που καλείται στο γεγονός **onchange** του στοιχείου `select`:

- Ανακτήστε τον δείκτη (`selectedIndex`) της επιλεγμένης τιμής του `select`.
- Αν είναι η πρώτη επιλογή (“Επιλέξτε μάθημα”), δεν κάνετε τίποτα.
- Αλλιώς, ανακτήστε τον επιλεγμένο κωδικό του μαθήματος (συμβουλευθείτε την εκφώνηση του 12<sup>ου</sup> εργαστηρίου)
- και στη συνέχεια κάντε μια αίτηση `XMLHttpRequest` για να λάβετε τα στοιχεία του μαθήματος.

- Τέλος, προσθέστε μια συνάρτηση υποδοχής του αποτελέσματος, η οποία θα προσθέτει δυναμικά στην ιστοσελίδα, μέσα στο στοιχείο `div` με `id` ίσο με `'tableplace'`, τον πίνακα του μαθήματος, όπως ακριβώς εμφανίζεται στο 11<sup>ο</sup> εργαστήριο.

## Επίλογος

- **“Τελικά, άξιζε τον κόπο;”** Φυσικά και όχι! Για ένα τόσο μικρό εργαστηριακό παράδειγμα, η χρήση μεθόδων Ajax είναι υπερβολική. Η σημασία της γνωριμίας με τη μέθοδο αυτή όμως είναι **πολύ μεγάλη**, αν αναλογιστείτε ότι **όλες οι εμπορικές εφαρμογές web** που χρησιμοποιείτε **καθημερινά**, είναι γραμμένες στο `στυλ Ajax!` Αυτό γίνεται για να αυξηθεί η λειτουργικότητα και ευχρηστία των εφαρμογών.
- **“Δεν υπάρχουν και μειονεκτήματα;”** Φυσικά και υπάρχουν: τι θα συμβεί αν ο χρήστης έχει απενεργοποιήσει τη JavaScript ή όταν χρησιμοποιείται συσκευή ανάγνωσης για χρήστες με ειδικές ανάγκες; Για τον λόγο αυτόν, όλα τα εμπορικά sites παρέχουν μια εναλλακτική μορφή σε απλή HTML.
- **“Ποιος είναι επιτέλους ο λόγος χρήσης της επικεφαλίδας `Access-Control-Allow-Origin: *`;”** Όλοι οι browsers, για λόγους ασφαλείας απαγορεύουν τη λήψη περιεχομένου μέσω της `XMLHttpRequest` από sites διαφορετικά από εκείνο που προμήθευσε την αρχική ιστοσελίδα! Σε διαφορετική περίπτωση, η κακόβουλη ιστοσελίδα θα μπορούσε να ενσωματώσει στοιχεία από ένα άλλο site που εσείς εμπιστεύεστε: θα ήταν θέμα χρόνου να εισάγετε “ευαίσθητα” δεδομένα πιστεύοντας ότι απέναντί σας βρίσκεται μια έμπιστη ιστοσελίδα!

Η **“πολιτική ασφάλειας της ίδιας πηγής”** προστατεύει μεν τον χρήστη, αλλά είναι πολύ περιοριστική για τις Ajax εφαρμογές: πρακτικά σημαίνει ότι **δεν μπορείτε να ανακτήσετε δεδομένα παρά μόνο από το δικό σας site** (που προμήθευσε και την αρχική ιστοσελίδα). Πώς θα προσπελάσετε λοιπόν (νόμιμα) δεδομένα από τρίτους;

Υπάρχουν διάφορες τεχνικές: η τεχνική **CORS** χρησιμοποιεί επικεφαλίδες, ενώ η τεχνική **JSONP** χρησιμοποιεί ένα νόμιμο “`τρικ`” της JavaScript. Η επικεφαλίδα `Access-Control-Allow-Origin: *` ανήκει στην πρώτη τεχνική, και λέει στην ουσία “**επιτρέπω τη χρήση των δεδομένων μου από όλους!**”

*(οι τεχνικές λεπτομέρειες των πιο πάνω μεθόδων ξεφεύγουν από τα πλαίσια του εργαστηρίου· αν θέλετε να μάθετε περισσότερα, αναζητήστε τις στο Διαδίκτυο)*

## Παράρτημα Α: Υπηρεσίες δεδομένων και το μορφότυπο JSON

### Α.1. Σελίδες PHP που δεν επιστρέφουν HTML.

Μέσω μιας σελίδας PHP μπορείτε να επιστρέψετε οποιοδήποτε είδος δεδομένων και σε οποιαδήποτε μορφή. Μέχρι τώρα επιστρέφετε αποκλειστικά και μόνον HTML. Είναι πολύ εύκολο όμως να στείλετε κάτι διαφορετικό, αρκεί να προσέξετε τα παρακάτω:

**1. Πολύ σημαντικό:** το `<?php` πρέπει να είναι **το πρώτο πράγμα** μέσα στο αρχείο σας! Με άλλα λόγια, **το πρώτο byte του αρχείου** πρέπει να είναι το '<', το δεύτερο το '?', κλπ. Σε διαφορετική περίπτωση, η PHP δεν θα σας αφήσει να ρυθμίσετε το είδος των δεδομένων που στέλνετε (θα είναι εξ'ορισμού τύπου HTML)!

*(αν χρησιμοποιείτε Windows, προσοχή σε κειμενογράφους που εισάγουν “αόρατους” χαρακτήρες ελέγχου στην αρχή του κειμένου!)*

**2. Πριν στείλετε οτιδήποτε** (π.χ. με την echo), πρέπει να θέσετε την **επικεφαλίδα (header)** με το είδος των δεδομένων σας, καθώς και όποιες άλλες επικεφαλίδες!

*(οι επικεφαλίδες δεν ανήκουν στο περιεχόμενο που στέλνετε· ανήκουν στο πρωτόκολλο HTTP και, ενώ δεν εμφανίζονται στον χρήστη, είναι πολύ σημαντικές: καθορίζουν πώς θα συμπεριφερθεί ο browser!)*

Η PHP έχει εξειδικευμένη συνάρτηση για την εισαγωγή επικεφαλίδων, την **header()**. Χρησιμοποιήστε την ως εξής **στην αρχή του κώδικα PHP**:

```
// τα δεδομένα που στέλνετε είναι σε μορφή JSON
header("Content-type: application/json");
// το επόμενο επιτρέπει τη χρήση των δεδομένων από τρίτους
header("Access-Control-Allow-Origin: *");
```

Το μορφότυπο JSON θα εξηγηθεί στη συνέχεια, ενώ η σημασία της δεύτερης επικεφαλίδας θα δοθεί συνοπτικά στον επίλογο του εργαστηρίου.

### Α.2. Το μορφότυπο JSON (JavaScript Object Notation).

Το μορφότυπο JSON επιτρέπει την ανταλλαγή δεδομένων σε μια απλή μορφή, εύκολα μετατρέψιμη στον αποστολέα και τον παραλήπτη. Βασίζεται στον τρόπο που η JavaScript περιγράφει την αρχικοποίηση αντικειμένων (objects).

*(τα αντικείμενα της JavaScript υποστηρίζουν όλα τα γνωστά αντικειμενοστρεφή χαρακτηριστικά: μεθόδους, μεταβλητές-μέλη, constructors, κλπ. Εδώ όμως βλέπετε τα αντικείμενα αποκλειστικά ως **συλλογές δεδομένων – data containers!**)*

Το επόμενο παράδειγμα περιγράφει στο μορφότυπο JSON ένα αντικείμενο με 3 ιδιότητες (properties): η πρώτη έχει τιμή (value) string, η δεύτερη int και η τρίτη αποτελείται από μια λίστα που περιέχει με τη σειρά της άλλα 2 αντικείμενα!

*(η στοίχιση και τα κενά δεν παίζουν κανέναν ρόλο)*

```
{ "name": "john", "age": 3, "jobs": [ { "name": "acme", "salary": 50 },  
                                     { "name": "ajax", "salary": 100 } ] }
```

Το πιο πάνω είναι μια έγκυρη τιμή αρχικοποίησης (literal) στην JavaScript: θα μπορούσατε να την αναθέσετε σε μια μεταβλητή και να δημιουργήσετε με αυτόν τον τρόπο ένα αντικείμενο με τις ιδιότητες αυτές!

Κατά το μορφότυπο JSON, μπορείτε να συνδυάσετε αντικείμενα και λίστες με οποιονδήποτε τρόπο και σε οποιοδήποτε βάθος το ένα μέσα στο άλλο. Έτσι μπορείτε να περιγράψετε και να στείλετε πολύ σύνθετες δομές δεδομένων στο web.

### A.3. JSON και PHP.

Η μορφή JSON είναι τόσο απλή, ώστε μπορείτε να στείλετε τα δεδομένα σας με την echo, π.χ. μπορείτε να πείτε: `echo ' {"name": "john", "age": 3, ... } ';`

Για να αποφύγετε όμως τη διαχείριση λεπτομερειών, όπως η εξουδετέρωση (escaping) των " κλπ, η PHP παρέχει ειδική συνάρτηση για τον σκοπό αυτόν, την `json_encode()`. Ως όρισμα της συνάρτησης συνήθως δίνετε ένα array με κλειδιά, τα οποία αντιστοιχούν στα ονόματα των ιδιοτήτων του αντικειμένου JSON.

Στο παράδειγμα που ακολουθεί, δείτε τι πρέπει να περάσετε στη συνάρτηση `json_encode()`, έτσι ώστε να πάρετε το αντικείμενο JSON του προηγούμενου παραδείγματος:

```
// κώδικας PHP  
$jsonarr = array('name'=>'john', 'age'=>3,  
                 'jobs'=>array(array('name'=>'acme', 'salary'=>50),  
                               array('name'=>'acme', 'salary'=>100)));  
// η κλήση αυτή θα δώσει το προηγούμενο παράδειγμα  
echo json_encode($jsonarr);
```

### A.4. Αλλάζοντας τον κωδικό HTTP της απάντησης.

Αν δεν ορίσετε διαφορετικά, η PHP θα στείλει στις επικεφαλίδες του πρωτοκόλλου HTTP της απάντησης τον κωδικό επιτυχίας (**200 OK**). Όταν η απάντησή σας υποδεικνύει σφάλμα, καλό είναι να αλλάζετε κωδικό, χρησιμοποιώντας κάποιον από τους κωδικούς σφάλματος του πρωτοκόλλου HTTP.

Αυτό μπορεί να γίνει μέσω της `header()` όπως και πριν, με τη διαφορά ότι η επικεφαλίδα πρέπει να έχει μια ειδική μορφή (εξαρτάται από τον web server):

```
// παράδειγμα για τον κωδικό σφάλματος 400  
header("HTTP/1.0 400 Bad Request");
```

Αφού στείλετε τον κωδικό HTTP που θέλετε, καθώς και το μήνυμα λάθους σε μορφή JSON, μπορείτε στη συνέχεια να καλέσετε τη συνάρτηση `die()`: είναι αντίστοιχη της `exit()` της C και θα προκαλέσει τον άμεσο τερματισμό της PHP.



## Παράρτημα Β: JavaScript και XMLHttpRequest.

Το όνομα προδίδει ότι το XMLHttpRequest σχεδιάστηκε αρχικά για να μεταφέρει δεδομένα σε μορφή XML. Αυτό δεν είναι πλέον αναγκαίο· μπορείτε να μεταφέρετε οποιοδήποτε είδος δεδομένων!

### B.1. Χρήση της XMLHttpRequest.

Για να δημιουργήσετε μια αίτηση XMLHttpRequest για δεδομένα, ακολουθήστε τα εξής βήματα:

```
// η διεύθυνση URL της αίτησης
var jsonurl = "http://...";
// δημιουργία νέου αντικειμένου XMLHttpRequest
var request = new XMLHttpRequest();
// ορίστε τη μέθοδο και τη διεύθυνση
request.open("GET", jsonurl);
// ορίστε τι θα γίνει κατά την παραλαβή
request.onload = function() {
    // αν η λήψη είναι επιτυχής
    if (request.status==200) {
        // καλέστε π.χ. μια συνάρτηση επεξεργασίας
        // (το περιεχόμενο είναι στο request.responseText)
        yourFunction(request.responseText);
    }
};
// τέλος, στείλτε την αίτηση
request.send(null);
```

Παρατηρήστε ότι:

- Η αίτηση δεν στέλνεται παρά μόνο στο τέλος (με τη send()), αφού έχετε ορίσει σε ποια διεύθυνση και με ποια HTTP μέθοδο θα γίνει, καθώς και τι θα συμβεί όταν τα δεδομένα φτάσουν.
- Για το τι θα συμβεί στην υποδοχή της απάντησης, φροντίζει η (ανώνυμη) συνάρτηση που συνδέεται με την ιδιότητα onload της αίτησης. Η συνάρτηση αυτή θα κληθεί όταν τα δεδομένα θα είναι διαθέσιμα.

*(ο κώδικας που δόθηκε λειτουργεί με το XMLHttpRequest Level 2 που υποστηρίζουν όλοι οι μοντέρνοι browsers – αναζητήστε πληροφορία στο Διαδίκτυο, αν θέλετε να υποστηρίξετε και παλαιότερους browsers)*

## B.2. JavaScript και λήψη δεδομένων σε μορφή JSON.

Όταν έχετε λάβει τα δεδομένα που ζητήσατε σε μορφή JSON (σε μια μεταβλητή string), είναι πολύ εύκολο να τα μετατρέψετε σε αντικείμενο(α) της JavaScript:

```
// μετατρέψτε την string μεταβλητή responseText
// σε αντικείμενα της JavaScript
var obj = JSON.parse(responseText);
```

Υποθέτοντας ότι το responseText περιείχε το εξής:

```
{ "name": "john", "age": 3, "jobs": [ { "name": "acme", "salary": 50 },
                                     { "name": "ajax", "salary": 100 } ] }
```

τότε μπορείτε να χειριστείτε τη μεταβλητή obj όπως στα πιο κάτω παραδείγματα:

```
// Παράδειγμα 1: προσπέλαση της ιδιότητας "name"
var n = obj.name;    // θα έχει τιμή "john"

// Παράδειγμα 2: έλεγχος ύπαρξης ιδιότητας "age"
if ('age' in obj) { // θα είναι true
...
}

// Παράδειγμα 3: διάσχιση array
var joblist = obj.jobs;
for (var i=0;i<joblist.length;i++) {
    // κάνετε κάτι με τα joblist[i].name
    // και joblist[i].salary
}
```

## Παράρτημα Γ: JavaScript και δυναμικός χειρισμός στοιχείων HTML

Μέσω της JavaScript μπορείτε να εισάγετε δυναμικά, να αλλάξετε ή να διαγράψετε στοιχεία HTML από την ιστοσελίδα που είναι φορτωμένη στον browser. Μπορείτε επίσης να θέσετε τις ιδιότητες και το περιεχόμενο των στοιχείων HTML.

### Γ.1. Εισαγωγή νέων στοιχείων HTML.

Θα πρέπει πάντα να θυμάστε ότι η ιστοσελίδα παράγει ένα δέντρο DOM με την ιεραρχία των κόμβων να αντιστοιχεί στα στοιχεία HTML. Κάθε στοιχείο HTML έχει έναν κόμβο “πατέρα” και ενδεχομένως και κόμβους “παιδιά”.

Ξεκινήστε, δημιουργώντας ένα **νέο στοιχείο**, προσδιορίζοντας το είδος του, π.χ.:

```
var option = document.createElement("option");
```

Στη συνέχεια, μπορείτε να θέσετε διάφορες **ιδιότητες** (attributes) του στοιχείου HTML:

```
// το παρακάτω θα έχει το ίδιο αποτέλεσμα με το να  
// γράφατε στην HTML: <option value="test">  
option.setAttribute("value", "test");
```

ή να θέσετε το **περιεχόμενο** του στοιχείου HTML:

```
// το παρακάτω θα έχει το ίδιο αποτέλεσμα με το να  
// γράφατε στην HTML: <option value="test">δοκιμή!</option>  
option.innerHTML = "δοκιμή!";
```

Αφού ολοκληρώσετε την αρχικοποίηση του στοιχείου, μπορείτε να το **προσθέσετε** στον κόμβο “πατέρα”:

```
// υποθέτοντας ότι το select αντιπροσωπεύει ένα στοιχείο  
// <select>, το πιο κάτω προσθέτει το νέο στοιχείο <option>  
// στη λίστα επιλογών του <select>  
select.appendChild(option);
```