

# JQUERY - AJAX

---

Διδάσκοντες: Π. Αγγελάτος, Δ. Ζήνδρος

Επιμέλεια διαφανειών: Π. Αγγελάτος, Δ. Ζήνδρος

Σχολή Ηλεκτρολόγων Μηχανικών  
και Μηχανικών Υπολογιστών



# Στόχος της ώρας

- Εξοικίωση με τη βιβλιοθήκη jQuery
  - Πρόσβαση στο έγγραφο
  - Επιλογείς
  - Αλλαγή του εγγράφου
  - Αλλαγή CSS
  - Βασικά γεγονότα
- AJAX
  - Καθαρισμός **χωρίς θαμπάδα**
  - **Απώθηση** σκόνης

# Στόχος της ώρας

- Εξοικίωση με τη βιβλιοθήκη jQuery
  - Πρόσβαση στο έγγραφο
  - Επιλογείς
  - Αλλαγή του εγγράφου
  - Αλλαγή CSS
  - Βασικά γεγονότα
- AJAX
  - **Ασύγχρονη** ανάγνωση/εγγραφή δεδομένων
  - HTTP συνδέσεις μέσω Javascript

# jQuery

- Τι είναι η jQuery;
  - «Βοηθητικές ρόδες» για Javascript
  - Βιβλιοθήκη γραμμένη η ίδια σε Javascript
  - Ό,τι κάνουμε με jQuery μπορούμε να το κάνουμε με Javascript
- Τι μπορεί να κάνει:
  - Πρόσβαση στο HTML μέσω Javascript
  - Αλλαγή ιδιοτήτων εγγράφου δυναμικά
    - Περιεχόμενο HTML
    - Μορφοποίηση CSS
  - Χειρισμός γεγονότων εύκολα

# Φόρτωση της jQuery

- Είναι ένα αρχείο js
- Βάζουμε μία ετικέτα `<script>` με src τον αρχείο της jQuery

```
<script type="text/javascript" src="jquery.js">  
</script>
```

# Φόρτωση της jQuery

- Την κατεβάζουμε από <http://jquery.com/>
- 2 εκδόσεις
  - Minified + συμπιεσμένη 26KB
  - Ασυμπιεστή 179KB
- Σελίδες με τεκμηρίωση και χρήσιμα παραδείγματα [http://docs.jquery.com/Main\\_Page](http://docs.jquery.com/Main_Page)

# Minification

- Γίνεται αυτόματα από πρόγραμμα
- Αφαιρούνται:
  - Σχόλια
  - Περιττές αλλαγές γραμμών, κενά, tab
- Ελαχιστοποιούνται τα ονόματα μεταβλητών
  
- Πλεονεκτήματα
  - Μικρό μέγεθος
  - Γρήγορη μεταφορά μέσω δικτύου και φόρτωση
- Μειονεκτήματα
  - Ελάχιστη αναγνωσιμότητα

# DOM

- Document Object Model
  - Είναι μία αναπαράσταση του εγγράφου HTML
  - Χρησιμοποιούμε τις δυνατότητές του μέσω Javascript
  - Μέσω αυτού μπορούμε να διαβάσουμε/αλλάξουμε το έγγραφο
- Λεπτομερής ανάλυση στα επόμενα μαθήματα



# Δύο είδη αναπαράστασεων

- Μπορούμε να αναπαράστούμε μία HTML ετικέτα
  - Με DOM αναπαράσταση
  - Με jQuery αναπαράσταση
- Μας προσφέρουν διαφορετικές δυνατότητες

# jQuery αναπαράσταση

- Αναπαριστά ένα **σύνολο** από ετικέτες τις οποίες μπορούμε να διαβάσουμε/αλλάξουμε

# DOM αναπαράσταση

- Αναπαριστά **μία μόνο** ετικέτα HTML

# Το αντικείμενο \$

- Global μεταβλητή με όνομα \$
  - Θα μπορούσε να είναι οτιδήποτε, δε σημαίνει κάτι το \$
- Μέσω αυτής έχουμε πρόσβαση στην jQuery
- Ορίζεται μόλις φορτώσει και τρέξει το jquery.js

# Επιλογείς jQuery

- Τρόπος που επιλέγουμε html στοιχεία
  - `$( 'επιλογέας' )`
- Ο επιλογέας είναι ένας CSS επιλογέας
- Επιστρέφει:
  - Το σύνολο των στοιχείων που θα επέλεγε ο CSS επιλογέας
- Παραδείγματα
  - Όλες οι φόρμες μιας σελίδας
    - `var forms = $( 'form' );`
  - Όλα τα `<a>` μέσα στο `<div id="foo">`
    - `var links = $( '#foo a' );`

# Επιλογείς jQuery

- **ΠΡΟΣΟΧΗ!**
- Η επιλογή γίνεται με βάση το έγγραφο όπως είναι εκείνη τη στιγμή
- Αν αλλάξει στο μέλλον το αντικείμενο **δεν** αλλάζει

# Αποτέλεσμα επιλογής jQuery

- Επιστρέφει ένα αντικείμενο
- Περιέχει:
  - Τα ίδια τα στοιχεία που επέλεξε
  - Χρήσιμες συναρτήσεις
  - Χρήσιμες ιδιότητες
- Οι συναρτήσεις δρουν μόνο στα στοιχεία που επιλέχτηκαν
- Ακόμα κι αν δεν επιλέχθηκαν στοιχεία οι συναρτήσεις υπάρχουν

```
var a = $( 'liroulirou' ); //Δεν υπάρχει ετικέτα  
liroulirou  
a.css(); //Είναι ορισμένο
```

# Αποτέλεσμα επιλογής jQuery

- Ιδιότητα length
  - Περιέχει τον αριθμό των στοιχείων που επιλέχθηκαν

```
var a = $( 'a' );  
alert( 'Υπάρχουν ' + a.length + ' σύνδεσμοι στη  
σελίδα' );
```



# Από jQuery σε DOM

- Τα στοιχεία σε αναπαράσταση DOM βρίσκονται στις θέσεις 0, 1, 2 ..., length - 1 του αποτελέσματος
- Τα παρακάτω είναι ισοδύναμα
- `var a = $( '#foo' )[ 0 ];`
- `var a = document.getElementById( 'foo' );`

# Από DOM σε jQuery

- `$( domElement )` επιστρέφει την αντίστοιχη αναπαράσταση σε jQuery
- Τα παρακάτω είναι ισοδύναμα
- `var a = $( '#foo' );`
- `var a = $( document.getElementById( 'foo' ) );`

# Συνάρτηση `css()`

- Συνάρτηση `css()`
  - Διαφορετική συμπεριφορά ανάλογα με τα ορίσματα
  - **Αλλάζει ή διαβάζει** το CSS
  - Οι κανόνες μέσω αυτής έχουν μεγαλύτερη ειδικότητα από όλους

# Συνάρτηση css()

- `css( 'ιδιότητα' )`
  - Επιστρέφει την τιμή της ιδιότητας CSS 'ιδιότητα' του πρώτου στοιχείου
- Η ιδιότητα δε μπορεί να είναι συντομευμένη ιδιότητα
  - border, padding, margin, background, font

```
<p style="color: red">Foo</p>
```

```
<p style="color: blue">Bar</p>
```

```
<script type="text/javascript">
```

```
    var foo = $( 'p' );
```

```
    var color = foo.css( 'color' ); //color = 'red'
```

```
</script>
```

# Συνάρτηση `css()`

- `css( 'ιδιότητα', 'τιμή' )`
  - Ορίζει σε κάθε στοιχείο την CSS ιδιότητα 'ιδιότητα' με τιμή 'τιμή'
- Όλες οι παράγραφοι να έχουν πράσινα γράμματα

```
var foo = $( 'p' );  
foo.css( 'color', 'green' );
```

# Συνάρτηση css()

- `css( ιδιότητες )`

- Ορίζει σε κάθε στοιχείο τις CSS ιδιότητες του *ιδιότητες*
- Το *ιδιότητες* είναι αντικείμενο της μορφής

```
{  
  'ιδιότητα': 'τιμή',  
  'ιδιότητα': 'τιμή'  
}
```

- Όλες οι παράγραφοι να έχουν άσπρο κείμενο σε μαύρο φόντο

```
var paragraphs = $( 'p' );  
paragraphs.css( {  
  'background-color': 'black',  
  'color': 'white'  
} );
```

# Συνάρτηση show()

- *Χωρίς ορίσματα*
- Εμφανίζει άμεσα τα στοιχεία στα οποία εφαρμόζεται

```
<p id="answer" style="display: none">42</p>
```

```
var a = $( '#answer' );  
a.show();
```

# Συνάρτηση show()

- show( *διάρκεια* )
  - Εμφανίζει με animation τα στοιχεία στα οποία εφαρμόζεται
  - Το animation διαρκεί *διάρκεια* milliseconds
  - Το *διάρκεια* μπορεί να είναι και 'fast' ή 'slow'
    - fast = 200ms, slow = 600ms

```
<p id="answer" style="display: none">42</p>
```

```
var a = $( '#answer' );  
a.show( 'fast' );
```



# Συνάρτηση hide()

- *Χωρίς ορίσματα*
- Κρύβει άμεσα τα στοιχεία στα οποία εφαρμόζεται
- Σαν να είχαν display: none

```
<p id="answer">42</p>
```

```
var a = $( '#answer' );  
a.hide();
```

# Συνάρτηση hide()

- hide( *διάρκεια* )
  - Κρύβει με animation τα στοιχεία στα οποία εφαρμόζεται
  - Το animation διαρκεί *διάρκεια* milliseconds
  - Το *διάρκεια* μπορεί να είναι και 'fast' ή 'slow'

```
<p id="answer">42</p>
```

```
var a = $( '#answer' );  
a.hide( 'slow' );
```

# Συνάρτηση toggle()

- *Χωρίς ορίσματα*
  - Τα στοιχεία που φαίνονται τα κρύβει
  - Τα στοιχεία που είναι κρυμμένα τα εμφανίζει

```
<p style="display: none">One</p>
```

```
<p style="display: block">Two</p>
```

```
var paragraphs = $( 'p' );
```

```
p.toggle(); //κρύβεται το one εμφανίζεται το Two
```

# Συνάρτηση fadeIn()

- fadeIn( διάρκεια )
  - Εμφανίζει ένα στοιχείο με εφέ fade
  - Το εφέ διαρκεί *διάρκεια* milliseconds
  - Αν παραληφθεί η διάρκεια έχει προεπιλεγμένη τιμή 400ms



Χρόνος

# Συνάρτηση fadeOut()

- fadeOut( διάρκεια )
  - Κρύβει ένα στοιχείο με εφέ fade
  - Το εφέ διαρκεί *διάρκεια* milliseconds
  - Αν παραληφθεί η διάρκεια έχει προεπιλεγμένη τιμή 400ms
  - Μετά το τέλος του animation τα στοιχεία αποκτούν display: none



Χρόνος

# Συνάρτηση text()

- *Χωρίς ορίσματα*
- Επιστρέφει το κείμενο όλων των στοιχείων σε ένα string
- `<p>Hello</p>`
- `<p> world!</p>`
- `var text = $( 'p' ).text(); // text = 'Hello world!'`

# Συνάρτηση text()

- `text( κείμενο )`
- Ορίζει το κείμενο όλων των στοιχείων να είναι *κείμενο*
- Κάνει αυτόματα escape ειδικούς χαρακτήρες

# Συνάρτηση text()

- `<p><strong>Hello</strong></p>`
- `<p> world!</p>`
- `var text = $( 'p' ).text( '<em>jQuery rocks!</em>' );`
- `<p>&lt;em&gt;jQuery rocks!&lt;/em&gt;</p>`
- `<p>&lt;em&gt;jQuery rocks!&lt;/em&gt;</p>`



# Συνάρτηση html()

- *Χωρίς ορίσματα*
- Επιστρέφει τα html περιεχόμενα **του πρώτου** στοιχείου σε ένα string
- `<p><strong>Hello</strong></p>`
- `<p> world!</p>`
- `var html = $( 'p' ).html();`
- `// html = '<strong>Hello</strong>'`

# Συνάρτηση html()

- `html( κώδικας )`
- Ορίζει το html περιεχόμενο όλων των στοιχείων να είναι *κώδικας*
- **Δεν** γίνονται escape οι ειδικοί χαρακτήρες

# Συνάρτηση html()

- `<p><strong>Hello</strong></p>`
- `<p> world!</p>`
- `var c = $( 'p' ).html( '<em>jQuery rocks</em>' );`
- `<p><em>jQuery rocks</em></p>`
- `<p><em>jQuery rocks</em></p>`

# 6<sup>η</sup> Εργασία

- Παράδοση μέσω SSH (Secure SHell)
  - Ανεβάστε .js, .html, .css, .php εικόνες κλπ. στο home σας
  - Επιβεβαιώστε ότι έχετε πρόσβαση με **ιδιωτικό/δημόσιο κλειδί**
  - Μέχρι πριν τις διακοπές των Χριστουγέννων
- **Να βελτιώσετε τον file uploader**
  - Βελτίωση πάνω στην 4<sup>η</sup> και 5<sup>η</sup> εργασία
  - Χρήση Javascript
  - Αλλαγή της σελίδας της λίστας των αρχείων
  - Κλικ σε αρχείο στη λίστα για άμεση προβολή
  - Άμεση προβολή κάτω από την λίστα
  - Ειδικός σύνδεσμος για download
  - Άμεση προβολή: .txt, .jpg, .png, .gif

# Συνάρτηση addClass()

- `addClass( 'κλάσηΑ κλάσηΒ' )`
  - Προσθέτει στα στοιχεία τις κλάσεις που της περνάμε
  - Δεν αντικαθιστά τις ήδη υπάρχουσες κλάσεις

# Συνάρτηση `removeClass()`

- `removeClass( 'κλάσηΑ κλάσηΒ' )`
  - Αφαιρεί στα στοιχεία τις κλάσεις που της περνάμε
  - Δεν αντικαθιστά τις ήδη υπάρχουσες κλάσεις

# Συνάρτηση toggleClass()

- `toggleClass( 'κλάσηΑ κλάσηΒ' )`
  - Αν ένα στοιχείο έχει την κλάση την αφαιρεί
  - Αν ένα στοιχείο δεν έχει την κλάση την προσθέτει
  - Δεν αντικαθιστά τις ήδη υπάρχουσες κλάσεις

# Συνάρτηση hasClass()

- `hasClass( κλάση )`
- Επιστρέφει `true` αν **οποιοδήποτε** από τα στοιχεία έχει την κλάση *κλάση*

```
<p class="foo"></p>
```

```
<p class="bar"></p>
```

```
var a = $( 'p' ).hasClass( 'bar' ); //true
```



# Συνάρτηση attr()

- attr( ιδιότητα )

- Επιστρέφει την τιμή της ιδιότητας *ιδιότητα* του **πρώτου** στοιχείου

```
<img src='foo.jpg' alt='Delicious waffle' />
```

```
var description = $( 'img' ).attr( 'alt' );
```

# Συνάρτηση attr()

- attr( ιδιότητα, τιμή )
  - Ορίζει στα στοιχεία την ιδιότητα *ιδιότητα* με τιμή *τιμή*

```
<img id="photo" src='foo.jpg' alt='Delicious waffle'  
>
```

```
var photo = $( '#photo' );  
photo.attr( 'title', photo.attr( 'alt' ) );
```

# Γεγονότα

- Όταν συμβαίνει κάτι στον browser πυροδοτείται ένα γεγονός
- Πολύ χρήσιμο να εκτελούμε κώδικα κάθε φορά που συμβαίνει κάτι τέτοιο
- Διάφορα είδη γεγονότων, πληκτρολογίου, ποντικιού κ.α.
- Η jQuery μας επιτρέπει εύκολη χρήση γεγονότων

# Γεγονότα – Βασική σύνταξη

- $\$( \text{επιλογέας} ) . \text{όνομαΓεγονότος} ( \text{συνάρτηση} )$
- Κάθε φορά που συμβαίνει το γεγονός *όνομαΓεγονότος* και αφορά τα στοιχεία του επιλογέα τότε εκτελείται η *συνάρτηση* *συνάρτηση*

# Γεγονός Κλικ

- Πυροδοτείται όταν γίνεται κλικ σε κάποιο στοιχείο από το ΠΟΝΤΙΚΙ

```
<span id="button">Εμφάνιση</span>
```

```
<p id="content" style="display: none">[...]</p>
```

```
$( '#button' ).click( function() {  
    $( '#content' ).show( 'fast' );  
} );
```

# Γεγονός Υποβολή

- Πυροδοτείται όταν υποβάλλεται μία φόρμα
  - Η υποβολή σταματάει αν επιστρέψουμε false
- `$( 'form.login' ).submit( function() { ... } );`

# Διάσχιση εγγράφου

- Συναρτήσεις διάσχισης εγγράφου
  - Μας πηγαίνουν σε σημεία του εγγράφου με βάση τα επιλεγμένα στοιχεία
  - Όλες οι συναρτήσεις μπορούν να φιλτραρισθούν με επιλογή

# Διάσχιση εγγράφου

- `children( [επιλογέας] )`
  - Επιστρέφει τα άμεσα παιδιά των στοιχείων σε jQuery αναπαράσταση
  
- `siblings( [επιλογέας] )`
  - Επιστρέφει τα αδέλφια των στοιχείων σε jQuery αναπαράσταση



# Διάσχιση εγγράφου

- **next( [επιλογέας] )**
  - Επιστρέφει τους αμέσως επόμενους αδελφούς των στοιχείων σε jQuery αναπαράσταση
  
- **prev( [επιλογέας] )**
  - Επιστρέφει τον προηγούμενο αδελφό των στοιχείων σε jQuery αναπαράσταση

# Διάσχιση εγγράφου

- **parent( [επιλογέας] )**
  - Επιστρέφει τους πατέρες των στοιχείων σε jQuery αναπαράσταση
  
- **parents( [επιλογέας] )**
  - Επιστρέφει όλους τους προγόνους των στοιχείων σε jQuery αναπαράσταση

# Callbacks

- Ασύγχρονη φύση της Javascript
- Callbacks
  - Τρέχουν κάτι όταν γίνει κάτι άλλο
- Παραδείγματα
  - Τρέξε την συνάρτηση **foo()** όταν φορτωθεί το έγγραφο
  - Τρέξε την συνάρτηση **bar()** όταν γίνει κλικ

# Callbacks

- Είναι ανώνυμες συναρτήσεις
- Σύνταξη `function() { κώδικας }`
- Περνιούνται ως παράμετροι σε άλλες συναρτήσεις
- Περισσότερα για ανώνυμες συναρτήσεις στα επόμενα μαθήματα

# Εκτέλεση κώδικα μετά τη φόρτωση

- Μπορούμε να τρέξουμε ένα τμήμα κώδικα αφού φορτώσει όλη η σελίδα:

```
$( function () {  
    // κώδικας  
} );
```

# AJAX

- Asynchronous Javascript and XML
  - Το XML δεν συνηθίζεται
  - Τα δεδομένα ανταλλάσσονται σε JSON ή άμεσα HTML/CSS
- Επιτρέπει HTTP συνδέσεις μέσω Javascript
  - POST και GET

# AJAX

- Πλεονεκτήματα:
  - Εφαρμογές που είναι ζωντανές
  - Φόρτωση νέου περιεχομένου χωρίς refresh
  - Διάβασμα των δεδομένων που θέλουμε απευθείας στην Javascript
  - Αλλαγή μόνο μέρους μιας εφαρμογής

# AJAX

- Μειονεκτήματα
  - Το κουμπί «πίσω» δε συμπεριφέρεται όπως θα περιμέναμε
  - Το κουμπί «Bookmark» δε συμπεριφέρεται όπως θα περιμέναμε
  - Οι περισσότερες αράχνες δε τρέχουν Javascript → δε καταγράφουν το περιεχόμενο που παράγεται από αυτή
  - Δε μπορούμε να φορτώσουμε οποιαδήποτε διεύθυνση



# AJAX

- Παραδείγματα χρήσης
  - Ζωντανά αποτελέσματα αναζήτησης
    - Google Instant
    - Facebook search-as-you-type
  - Chat
  - Έλεγχος διαθεσιμότητας όνομα χρήστη πριν γίνει register

# AJAX σε jQuery

- Δύο βασικές μέθοδοι:
  - `$.get()`
  - `$.post()`
- Ίδια σύνταξη

# \$.get()

- **\$.get( διεύθυνση, δεδομένα, callback )**
  - Κάνει ένα HTTP GET αίτημα στη διεύθυνση *διεύθυνση* με GET παραμέτρους τα *δεδομένα*
  - Όταν απαντήσει ο server εκτελείται το `callback`
  - Στο `callback` περνιέται ως παράμετρος **το αποτέλεσμα**

# \$.get()

```
<h2>Dynamic Paragraph</h2>
```

```
<p id="dyn"></p>
```

```
...
```

```
$.get(  
    'data.html',  
    { 'local': 'true' },  
    function( data ) {  
        $( '#dyn' ).html( data );  
    }  
);
```

# \$.post()

- **\$.post( διεύθυνση, δεδομένα, callback )**
  - Κάνει ένα HTTP POST αίτημα στη διεύθυνση *διεύθυνση* με POST παραμέτρους τα *δεδομένα*
  - Όταν απαντήσει ο server εκτελείται το `callback`
  - Στο `callback` περνιέται ως παράμετρος **το αποτέλεσμα**

# \$.post()

```
<h2>Dynamic Save</h2>
```

```
<p id="dyn"></p>
```

```
...
```

```
$.post(  
    'post.php',  
    { 'text': 'Hello' },  
    function( data ) {  
        alert( 'Post was saved!' );  
    }  
);
```

# Μάθαμε

- Εισαγωγή στην jQuery
  - Επιλογείς
  - Επεξεργασία HTML/CSS
  - Γεγονότα
- AJAX
  - GET/POST και παραδείγματα



# Την επόμενη φορά...

- Τα Javascript events σε βάθος
  - Events χωρίς jQuery
  - Ποια είναι όλα τα events που υπάρχουν;
  - Χειρισμός ποντικιού
  - Χειρισμός πληκτρολογίου
  - Διαφορετικοί τρόποι δήλωσης events