



# Πανεπιστήμιο Πειραιώς Τμήμα Ψηφιακών Συστημάτων

## Διαδικτυακά και Φορητά Πληροφοριακά Συστήματα (Δικτυακές Υπηρεσίες)

3<sup>η</sup> Διάλεξη

Δημοσθένης Κυριαζής

Δευτέρα 12 Μαρτίου 2018



# Σημερινή διάλεξη

---

- Σύνοψη προηγούμενης διάλεξης
- jQuery
  - Selectors, Filters, CSS
  - Attribute selector
  - Φιλτράρισμα
  - Loops
  - Προσθήκη κειμένου & HTML
  - Διαγραφή επιλογών
  - Επεξεργασία κλάσεων
  - Αλλαγή CSS
  - Γραφικά / κινήσεις
  - Εμφάνιση / απόκρυψη
  - Βαθμιαία εμφάνιση
  - Κύλιση
  - Κινήσεις



# Web storage

## □ sessionStorage / localStorage

```
<p><button onclick="butClicks()" type="button">Test</button></p>  
<div id="sesStor"></div>
```

```
function butClicks() {  
  if(typeof(Storage)!=="undefined") {  
    if (sessionStorage.clicks) {  
      sessionStorage.clicks=Number(sessionStorage.clicks)+1; }  
    else {  
      sessionStorage.clicks=1; }  
    document.getElementById("sesStor").innerHTML="Number of clicks: " +  
    sessionStorage.clicks; }  
    else { document.getElementById("sesStor").innerHTML="Not supported"; } }
```



# Web workers (1/3)

## □ Διαδικασία

- Δημιουργία αρχείου web worker (ως αρχείο Javascript)

```
var i=11;  
function countdown () {  
    i=i-1;  
    postMessage(i);  
    setTimeout("countdown()",1000); }  
countdown();
```

Η μέθοδος `postMessage()` χρησιμοποιείται για να αποστείλει ένα μήνυμα στη σελίδα

- Δημιουργία αντικειμένου web worker στη σελίδα

```
if(typeof(ww)=="undefined") {  
    ww=new Worker("xx.js"); }
```

- Προσθήκη event στη σελίδα για ανταλλαγή μηνυμάτων

```
ww.onmessage=function(event){  
    document.getElementById("result").innerHTML=event.data; };
```

Όταν ο web worker αποστέλλει ένα μήνυμα εκτελείται ο παραπάνω κώδικας και τα δεδομένα από τον worker αποθηκεύονται στο `event.data`

- Διακοπή λειτουργίας: `ww.terminate();`



# Web workers (2/3)

---

## □ Παράδειγμα

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<p>Countdown...<output id="result"></output></p>
```

```
<button onclick="startWorker()">Start</button>
```

```
<button onclick="stopWorker()">Stop</button>
```

```
<script>
```

```
var ww;
```

```
function startWorker()
```

```
{
```

```
  ww=new Worker("xx.js");
```



# Web workers (3/3)

---

```
ww.onmessage = function (event) {  
    if (event.data=="0")  
    {  
        document.getElementById("result").innerHTML="READY";  
        ww.terminate();  
    }  
    else  
    {  
        document.getElementById("result").innerHTML=event.data;  
    }  
};  
}  
function stopWorker()  
{  
    ww.terminate();  
}  
</script>  
  
</body>  
</html>
```



# Geolocation (1/2)

---

## □ Παράδειγμα

```
<p id="geoTest">Coordinates?</p>
<button onclick="getLocation()">Submit</button>
<div id="myMap"></div>
<script>
    var x=document.getElementById("geoTest");
    function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(showPosition); }
    else {
        x.innerHTML="Geolocation not supported";} }
```



# Geolocation (2/2)

---

(συνέχεια)

```
function showPosition(position) {  
    x.innerHTML="Geografiko platos: " + position.coords.latitude +  
    "<br>Mikos: " + position.coords.longitude;  
  
    var latlon=position.coords.latitude+","+position.coords.longitude;  
    var  
    img_url="http://maps.googleapis.com/maps/api/staticmap?center=" +  
    latlon+"&zoom=14&size=500x500";  
    document.getElementById("myMap").innerHTML="<img  
    src='"+img_url+"'>";  
}  
</script>
```





# Εισαγωγή

---

- Βιβλιοθήκη Javascript
- Επιτρέπει
  - Επιλογή elements, προσθήκη περιεχομένου, απόκρυψη / εμφάνιση περιεχομένου, τροποποίηση tag attributes, καθορισμό τιμών πεδίων φόρμας, ...
- Ουσιαστικά πρόκειται για μια συλλογή συναρτήσεων JavaScript που μπορούν να προστεθούν σε δικτυακές εφαρμογές





# Προσθήκη jQuery στον κώδικα

---

## □ Στο πεδίο <head>

### ■ Τοπικά

```
<script src="jquery-1.11.2.min.js"></script>
```

### ■ Για Google CDN

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js">  
</script>
```



# document.ready

---

## □ Επεξήγηση

- Χρήση selector: `$(document)`
- Εφαρμογή μεθόδου `.ready(function() {..} )` η οποία εφαρμόζει τον κώδικα όταν έχει ολοκληρωθεί το DOM
  - Π.χ. `alert("DOM loaded");`
- Χρησιμοποιούνται παρενθέσεις γιατί μεταφέρεται ως όρισμα τη συνάρτηση για το ready event στον event handler

## □ Σύνοψη

```
$(document).ready(function() {  
    //jQuery DOM code goes here  
});
```



# Σημερινή διάλεξη

---

- Σύνοψη προηγούμενης διάλεξης
- jQuery
  - Selectors, Filters, CSS
  - Attribute selector
  - Φιλτράρισμα
  - Loops
  - Προσθήκη κειμένου & HTML
  - Διαγραφή επιλογών
  - Επεξεργασία κλάσεων
  - Αλλαγή CSS
  - Γραφικά / κινήσεις
  - Εμφάνιση / απόκρυψη
  - Βαθμιαία εμφάνιση
  - Κύλιση
  - Κινήσεις



# Selectors, Filters, CSS (1/2)

---

- Οι Selectors είναι το βασικό εργαλείο της jQuery
  - Απαιτούνται για την επιλογή του στοιχείου του DOM το οποίο θα τροποποιηθεί
  - Η σύνταξη είναι ακριβώς η ίδια με τους CSS selectors για ID, class, tag, κλπ (υπάρχουν κάποιοι επιπλέον)
  - Με χρήση selector, γίνεται έλεγχος όλου του DOM για τα elements που ορίζονται στο selector, και επιλέγονται
- Τα Filters χρησιμοποιούνται σε συνδυασμό με τους selectors για την επιλογή συγκεκριμένων στοιχείων (π.χ. βάσει θέσης, επιλογής, εμφάνισης, κλπ)



# Selectors, Filters, CSS (2/2)

- Αποτελούνται από τα εξής μέρη
  - jQuery (συνήθως χρησιμοποιείται το σύμβολο \$, αλλά το jQuery επίσης λειτουργεί)
  - Τα DOM elements που επιλέγονται – με χρήση ‘ ’ μέσα σε παρενθέσεις
  - Η μέθοδος jQuery method που θα εφαρμοστεί

jQuery	Selector	jQuery method / action
\$ ή jQuery	( ‘div’ )	.css( ‘border’, ‘1px solid black’ );



## \* Selector

---

- Επιλογή όλων των elements στο DOM



# ID Selector

---

- Ο jQuery selector για την επιλογή IDs συντάσσεται ως: ( '#' )
  - Πρέπει πάντα να περιλαμβάνεται το σύμβολο # - διαφορετικά δεν λειτουργεί σωστά!
- Χρησιμοποιεί τη μέθοδο της JavaScript getElementById()





# Element Selector

---

- Ο jQuery selector για την επιλογή elements συντάσσεται ως: (‘όνομα element’)
- Χρησιμοποιεί τη μέθοδο της JavaScript `getElementsByTagName()`
- Π.χ. Για όλα τα `<a>` tag μιας σελίδας:
  - `var linksList = $('a');`



# Class Selector

---

- Ο jQuery selector για την επιλογή classes συντάσσεται ως: ( `' .όνομα class'` )
  - Πρέπει πάντα να περιλαμβάνεται το σύμβολο `.` - διαφορετικά δεν λειτουργεί σωστά!
- Χρησιμοποιείται τη μέθοδο της JavaScript `getElementByClassName()`
- Π.χ. Για όλες τις κλάσεις `jq2014`
  - `$('jq2014')`



# Parent-child selectors

---

- Χρησιμοποιούνται για την επιλογή elements όταν tag, CSS, και ID elements δεν μπορούν να χρησιμοποιηθούν
- Βασική χρήση σε navigation menus
- Επιτρέπουν την επιλογή των απευθείας «παιδιών» ενός element με χρήση του >
  - Π.χ. Ο selector \$('body > p') θα επιλέξει όλα τα elements παραγράφων αρκεί να είναι παιδιά του body element



# Descendent Selectors

---

- Οι Parent-child selectors εφαρμόζονται για απευθείας παιδιά (π.χ. `<li>` tags μέσα σε `<ul>` tags). Όταν απαιτείται η επιλογή παιδιών σε μεγαλύτερο «βάθος» σε σχέση με το αρχικό element χρησιμοποιούνται descendent selectors
- Η διαφορά στη σύνταξη είναι στο `>` που σε αυτή την περίπτωση δεν χρησιμοποιείται



# Επιλογή πολλαπλών elements

---

- Για την επιλογή πολλαπλών elements (π.χ. classes, IDs, HTML tags) καθώς και σχέσεων parent-child, γίνεται χρήση jQuery selector που διαχωρίζονται με κόμμα μεταξύ τους



# Σημερινή διάλεξη

---

- Σύνοψη προηγούμενης διάλεξης
- jQuery
  - Selectors, Filters, CSS
  - Attribute selector
  - Φιλτράρισμα
  - Loops
  - Προσθήκη κειμένου & HTML
  - Διαγραφή επιλογών
  - Επεξεργασία κλάσεων
  - Αλλαγή CSS
  - Γραφικά / κινήσεις
  - Εμφάνιση / απόκρυψη
  - Βαθμιαία εμφάνιση
  - Κύλιση
  - Κινήσεις



# Attribute selector (1/2)

---

- Επιτρέπει την επιλογή elements εάν έχουν ένα συγκεκριμένο attribute
- Χρησιμοποιούνται επίσης για τον έλεγχο της τιμής του attribute
- Π.χ. Για `<img>` tags που έχουν το alt attribute
  - `$('img[alt]')`



# Attribute selector (2/2)

## □ Σύνταξη

- [attribute] – για το συγκεκριμένο attribute
- [attribute="value"] – για elements με συγκεκριμένη τιμή στο attribute, π.χ. text boxes σε μια φόρμα: `$('input[type="text"]')`
- [attribute^="value"] – για elements με την τιμή του attribute να ξεκινάει σε συγκεκριμένη τιμή - π.χ. εξωτερικά links:  
`$('a[href^="http://"]')` ή mailto links: `$('a[href^="mailto:"]')`
- [attribute\$="value"] – για elements που η τιμή του attribute τελειώνει σε συγκεκριμένη τιμή - π.χ. file extensions για PDF:  
`$('a[href$=".pdf"]')`
- [attribute\*="value"] – για elements που η τιμή του attribute περιλαμβάνει το συγκεκριμένη τιμή σε οποιοδήποτε σημείο – π.χ. link προς ένα domain: `$('a[href*="unipi.gr"]')`





# Σημερινή διάλεξη

---

- Σύνοψη προηγούμενης διάλεξης
- jQuery
  - Selectors, Filters, CSS
  - Attribute selector
  - Φιλτράρισμα
  - Loops
  - Προσθήκη κειμένου & HTML
  - Διαγραφή επιλογών
  - Επεξεργασία κλάσεων
  - Αλλαγή CSS
  - Γραφικά / κινήσεις
  - Εμφάνιση / απόκρυψη
  - Βαθμιαία εμφάνιση
  - Κύλιση
  - Κινήσεις



# Φιλτράρισμα (1/3)

---

- Χρησιμοποιούνται όταν ο στόχος είναι η επιλογή ενός ή συγκεκριμένων elements στο DOM
  - Βασικό για δυναμικές σελίδες που αλλάζουν βάσει request
- Το φίλτρο ορίζεται ως :filter



## Φιλτράρισμα (2/3)

Όνομα	Περιγραφή
:even and :odd	Εύρεση even και odd βάσει του index
:header	Εύρεση elements με <h1>...<h6> tags
:first (:last και :only)	Εύρεση πρώτου, τελευταίου, ή μοναδικού
:nth-child	Εύρεση n-οστού
:parent	Εύρεση parent ενός element
:visible	Εύρεση element που είναι ορατό
:hidden	Εύρεση element που είναι κρυμμένο
:contains('κείμενο')	Εύρεση element που περιλαμβάνει συγκεκριμένο κείμενο



## Φιλτράρισμα (3/3)

---

- Χρήση :not() φίλτρου για την εύρεση elements που δεν ταιριάζουν σε κάποιο τύπο selector
  - Π.χ. Επιλογή όλων των <a> tag εκτός αυτών της κλάσης  
testButton: \$('a:not(.testButton)');
- Μπορεί να χρησιμοποιηθεί με όλα τα jQuery filters και τους αντίστοιχους selectors
  - Π.χ. Επιλογή όλων των link που δεν ξεκινούν με http://:  
\$('a:not([href^="http://"])')



# Loops

---

- Loops σε σύνολο elements επιτυγχάνονται με συγκεκριμένες μεθόδους της jQuery αυτόματα
  - Π.χ. Επιλογή όλων των εικόνων σε ένα <div> tag με ID=slideshow και στη συνέχεια απόκρυψη αυτών των εικόνων: `$('#slideshow img').hide();`



# Σημερινή διάλεξη

---

- Σύνοψη προηγούμενης διάλεξης
- jQuery
  - Selectors, Filters, CSS
  - Attribute selector
  - Φιλτράρισμα
  - Loops
  - Προσθήκη κειμένου & HTML
  - Διαγραφή επιλογών
  - Επεξεργασία κλάσεων
  - Αλλαγή CSS
  - Γραφικά / κινήσεις
  - Εμφάνιση / απόκρυψη
  - Βαθμιαία εμφάνιση
  - Κύλιση
  - Κινήσεις



# Προσθήκη κειμένου & HTML (1/2)

---

- Με χρήση της `html()` είναι η αντικατάσταση του κειμένου μέσα στο συγκεκριμένο `element` με άλλο HTML
- Η χρήση της `text()` είναι ίδια με την `.html()` αλλά δεν μπορεί να περιλαμβάνει HTML tags
  - Χρήσιμη για την αλλαγή του κειμένου σε κάποιο tag
- Η `append()` προσθέτει κείμενο ως το τελευταίο `child element` για το συγκεκριμένο `element`
- Η `prepend()` λειτουργεί όπως η `append()`, αλλά προσθέτει αμέσως μετά το αρχικό tag



## Προσθήκη κειμένου & HTML (2/2)

---

- Για την προσθήκη HTML εκτός κάποια επιλογής (πριν ή μετά από το αρχικό tag ενός επιλεγμένου element)
  - before()
  - after()





# Σημερινή διάλεξη

---

- Σύνοψη προηγούμενης διάλεξης
- jQuery
  - Selectors, Filters, CSS
  - Attribute selector
  - Φιλτράρισμα
  - Loops
  - Προσθήκη κειμένου & HTML
  - Διαγραφή επιλογών
  - Επεξεργασία κλάσεων
  - Αλλαγή CSS
  - Γραφικά / κινήσεις
  - Εμφάνιση / απόκρυψη
  - Βαθμιαία εμφάνιση
  - Κύλιση
  - Κινήσεις



# Διαγραφή επιλογών

- Για την αντικατάσταση ή διαγραφή ενός επιλεγμένου element γίνεται χρήση της μεθόδου `remove()`
  - Π.χ. dialog box με ID `porup`: `$('#porup').remove();`
- Η μέθοδος `remove()` δεν περιορίζεται σε ένα μόνο επιλεγμένο element αλλά μπορεί να εφαρμοστεί σε πλήθος
  - Π.χ. διαγραφή όλως των `<span>` tags που ανήκουν στην κλάση `error`: `$('.span.error').remove();`



# Σημερινή διάλεξη

---

- Σύνοψη προηγούμενης διάλεξης
- jQuery
  - Selectors, Filters, CSS
  - Attribute selector
  - Φιλτράρισμα
  - Loops
  - Προσθήκη κειμένου & HTML
  - Διαγραφή επιλογών
  - Επεξεργασία κλάσεων
  - Αλλαγή CSS
  - Γραφικά / κινήσεις
  - Εμφάνιση / απόκρυψη
  - Βαθμιαία εμφάνιση
  - Κύλιση
  - Κινήσεις



# Επεξεργασία κλάσεων

- Με χρήση μεθόδων για την προσθήκη / διαγραφή / αντικατάστασή
  - `addClass()`: Προσθήκη κλάσης (δεν περιλαμβάνεται η `.`)
    - ▣ Π.χ. `$('a[href^="http://"]').addClass("intro");`
  - `removeClass()`: Διαγραφή κλάσης
  - `toggleClass()`: Αντικατάσταση κλάσης (χρησιμοποιείται για δυναμικές αλλαγές όπως προκύπτουν συνήθως από radio buttons)



# Σημερινή διάλεξη

---

- Σύνοψη προηγούμενης διάλεξης
- jQuery
  - Selectors, Filters, CSS
  - Attribute selector
  - Φιλτράρισμα
  - Loops
  - Προσθήκη κειμένου & HTML
  - Διαγραφή επιλογών
  - Επεξεργασία κλάσεων
  - Αλλαγή CSS
  - Γραφικά / κινήσεις
  - Εμφάνιση / απόκρυψη
  - Βαθμιαία εμφάνιση
  - Κύλιση
  - Κινήσεις



# Αλλαγή css (1/2)

---

- Γίνεται με χρήση της μεθόδου `css()` με 3 τρόπους
  - Εύρεση τιμής CSS για κάποιο `element` με χρήση του ονόματος της ιδιότητας
    - ▣ Π.χ. `var fontSize= $('body').css('font-size');`
  - Ορισμός μιας ιδιότητας CSS για κάποιο `element` με χρήση 2 ορισμάτων (ιδιότητα, τιμή)
    - ▣ Π.χ. `$('body').css('font-size', baseFont*2);`
    - ▣ Η τιμή μπορεί να είναι `string` ή `αριθμός`



# Αλλαγή css (2/2)

---

- Γίνεται με χρήση της μεθόδου `css()` με 3 τρόπους (συν)
  - Ορισμός πολλαπλών ιδιοτήτων CSS
    - ▣ Χρήση μεθοδολογίας για 1 ιδιότητα πολλές φορές
      - `$('#test').css('background-color','green');`
      - `$('#test').css('border','2px solid black');`
    - ▣ Χρήση object literal στη μέθοδο `css()` με σύνταξη: ιδιότητα:τιμή, ιδιότητα:τιμή
      - Π.χ. `{ 'background-color' : 'green', 'border' : '2px solid black' }`



# Σημερινή διάλεξη

---

- Σύνοψη προηγούμενης διάλεξης
- jQuery
  - Selectors, Filters, CSS
  - Attribute selector
  - Φιλτράρισμα
  - Loops
  - Προσθήκη κειμένου & HTML
  - Διαγραφή επιλογών
  - Επεξεργασία κλάσεων
  - Αλλαγή CSS
  - Γραφικά / κινήσεις
  - Εμφάνιση / απόκρυψη
  - Βαθμιαία εμφάνιση
  - Κύλιση
  - Κινήσεις





# Γραφικά / κινήσεις

---

- Κάθε μέθοδος γραφικών μπορεί να λαμβάνει ως πρόσθετα ορίσματα
  - Ταχύτητα, ως χρόνος για την ολοκλήρωση του γραφικού
  - Ενέργειες μετά την ολοκλήρωση των γραφικών (με χρήση της callback function)
- Ο χρόνος μπορεί να οριστεί με τιμές
  - String: 'fast' (200), 'normal' (400), 'slow' (600)
  - Αριθμό που εκφράζεται σε milliseconds
- Π.χ. `$('p').fadeOut('slow');`
- Π.χ. `$('p').fadeOut(500);`



# Εμφάνιση / απόκρυψη

---

- `show()`: Εμφανίζει ένα κρυμμένο (hidden) element.
  - Χωρίς ορισμό χρόνου εμφανίζεται άμεσα.
- `hide()`: Αποκρύπτει ένα element που εμφανίζεται στη σελίδα.
  - Δεν έχει κάποιο αποτέλεσμα σε ήδη κρυμμένα element.
  - Χωρίς ορισμό χρόνου αποκρύπτεται άμεσα.
- `toggle()`: Αλλάζει την τιμή της εμφάνισης / απόκρυψης του αντικειμένου βάσει της τρέχουσας τιμής
  - Είναι ιδανική μέθοδος για την αλλαγή με χρήση ενός μόνο σημείου ελέγχου (π.χ. κουμπί)



# Βαθμιαία εμφάνιση (fading) (1/2)

- `fadeIn()`: Εμφανίζει ένα κρυμμένο αντικείμενο σταδιακά
  - Αρχικά εμφανίζεται ο χώρος που θα καλυφθεί από το αντικείμενο (μπορεί να αλλάξουν θέση άλλα αντικείμενα) και στη συνέχεια βαθμιαία εμφανίζεται το αντικείμενο.
  - Χωρίς ορισμό χρόνου, το αντικείμενο εμφανίζεται βαθμιαία ('normal': 400 milliseconds)
- `fadeOut()`: Βαθμιαία απόκρυψη αντικειμένου
- `fadeToggle()`: Αλλαγή μεταξύ βαθμιαίας απόκρυψης / εμφάνισης
  - Π.χ. `$('#button').click(function()  
{  
  $('#p').fadeToggle(500);  
})`



## Βαθμιαία εμφάνιση (fading) (2/2)

---

- `fadeTo()`: Αλλάζει την εμφάνιση του αντικειμένου σε συγκεκριμένη «διαφάνεια» (π.χ. εικόνα `semitransparent`)
  - Αντίθετα με άλλα γραφικά, θα πρέπει να ορισθεί η ταχύτητα / χρόνος
  - Επίσης ορίζεται μια δεύτερη τιμή, από το 0 ως το 1 που αντιστοιχεί στη διαφάνεια του αντικειμένου
- Π.χ. `$('p').fadeTo('normal', 0.75);`



# Κύλιση (sliding)

- `slideDown()`: Εμφανίζεται ένα κρυμμένο αντικείμενο με χρήση κύλισης.
  - Αρχικά εμφανίζεται το επάνω μέρος του αντικειμένου και στη συνέχεια το υπόλοιπο.
  - Δεν υπάρχει κάποια αλλαγή αν το element εμφανίζεται ήδη.
  - Χωρίς ορισμό ταχύτητας / χρόνου γίνεται χρήση της τιμής 'normal' (400 milliseconds)
- `slideUp()`: Απόκρυψη element με χρήση κύλισης (από κάτω προς τα επάνω)
- `slideToggle()`: Χρησιμοποιεί τη συνάρτηση `slideDown()` αν το element είναι κρυμμένο και τη `slideUp()` αν εμφανίζεται



# Σημερινή διάλεξη

---

- Σύνοψη προηγούμενης διάλεξης
- jQuery
  - Selectors, Filters, CSS
  - Attribute selector
  - Φιλτράρισμα
  - Loops
  - Προσθήκη κειμένου & HTML
  - Διαγραφή επιλογών
  - Επεξεργασία κλάσεων
  - Αλλαγή CSS
  - Γραφικά / κινήσεις
  - Εμφάνιση / απόκρυψη
  - Βαθμιαία εμφάνιση
  - Κύλιση
  - Κινήσεις



# Κινήσεις (1/2)

- Επιτυγχάνονται με χρήση της μεθόδου `animate()`
  - Επιτρέπουν μορφοποίηση όλων των CSS property που δέχονται αριθμητικές / ποσοστιαίες τιμές (π.χ. pixel, em, %)
- Η συνάρτηση μπορεί να λάβει πολλά ορίσματα
  - Το πρώτο είναι ένα object literal που περιλαμβάνει όλες τις αλλαγές στα CSS properties
  - Το δεύτερο είναι ο χρόνος / ταχύτητα (σε milliseconds)
  - Π.χ. `$('#message').animate(  
 {  
 opacity: 0.5,  
 fontSize: '24px'  
 },  
 1500  
);`



## Κινήσεις (2/2)

---

- Όταν χρησιμοποιείται μέσα σε κάποιο event, γίνεται χρήση του selector `$(this)` για την αναφορά στο element που αντιστοιχεί στο συγκεκριμένο event
- Π.χ. `$('#moveMe').click(function() {  
 $(this).animate(  
 {  
 fontSize: '50px'  
 },  
 100);  
});`