

Introduzione a jQuery (write less do more)

Simone Kalb



Mi presento

- Mi chiamo Simone Kalb
- Lavoro come Software Engineer presso il Lab Open Media Center
- Mi occupo di sviluppo per il web, design e mobile
- Utilizzo GNU/Linux dal 1995
- E dal 2005(?) faccio parte del GulCh
- Da grande voglio fare la modella..

Di cosa parleremo oggi

- Ecco una lista abbastanza esauriente di quello che vedremo in 35-40 minuti:
 - Introduzione a jQuery
 - jQuery
 - jQuery in action
 - Come usare jQuery
 - Due parole su jQuery
 - Basta con 'sto jQuery!
 - jQueryUI
 - jQuery Mobile
 - Tutti a bere un thé caldo

jQuery

- jQuery è un framework per lo sviluppo di funzioni in JavaScript
 - Facile
 - Veloce
 - Ultraleggero (solo 31KB!)
 - Cross browser
- JQuery naque nel 2005 ad opera del suo attuale maintainer John Resig
- È rilasciato sotto licenza mista GPL/MIT
- Di esso fa parte anche l'engine del selettore *Sizzle* rilasciato sotto GPL/MIT/BSD

Perché

- jQuery è diventato uno standard
- Tutte le più grandi aziende di software lo utilizzano
- 99% dei web designers hanno questa competenza nel CV
- Già solo questo impone che lo debba conoscere anche TU!
- A meno che non ti piaccia reinventare la ruota ad ogni funzione...

Come

- Aggiungere jQuery ai nostri progetti è incredibilmente veloce:
 - `<script src="js/jquery.js"></script>`
 - Procuratevi una copia di jQuery [qui](#)
- Soluzione migliore consiste nel linkare direttamente la risorsa da un CDN
 - Content Delivery Network
 - Consente migliorare l'accesso ai dati
 - Ottimizza il caching
 - Aumenta la banda a disposizione tramite server piazzati ad hoc
 - Minimizza la latenza
- Usiamo quello ufficiale:
 - <http://code.jquery.com/jquery-1.6.4.min.js> (hosted by MT)

Dove

- Nell'head o altrove della nostra pagina HTML come di seguito:

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Demo</title>
  </head>
  <body>
    <a href="http://jquery.com/">jQuery</a>
    <script src="http://code.jquery.com/jquery-1.6.4.min.js"></script>
    <script>
    </script>
  </body>
</html>
```

- Lanciare il codice non appena la pagina è caricata

```
$(document).ready(function(){
  // Ci torneremo più avanti
});
```

Quando

- Quando si vogliono cambiare attributi di pagina al volo
- Quando si vogliono creare effetti grafici
- Quando si vogliono creare interfacce
- Quando si vogliono fare comunicazioni asincrone ma senza stress
- Quando ci si vuol concentrare sul codice e non sulla piattaforma su cui questo girerà (leggi browser)
- Quando volete scrivere un plugin :D

Il problema di onload

- Quando si vuole lanciare una funzione non appena il caricamento è terminato
 - Usiamo tradizionalmente `document.onload`
 - Con immagini grosse si crea un collo di bottiglia
 - Uno dei primi ad individuare il problema fu Michaux ^[1]
- Dean Edwards, Matthias Miller, John Resig hanno la soluzione ^[2]
 - Cross browser
 - Indipendente dal contenuto
 - Integrato in jQuery, Mootools e Low Pro (Prototype)
 - Non intrusivo
- Conseguentemente noi useremo:
 - `$(document).ready(function() {});`

La selezione dei tag/selettori

- Il principale vantaggio di jQuery è l'operatore `$()`
 - Rimpiazza l'odioso `document.getElementById`
 - È un alias di `jQuery()`
 - Serve non solo per la selezione dei selettori
 - Anche classi
 - Tag
 - Gerarchie di elementi
- `$()` ritorna un puntatore ad un oggetto che può essere manipolato a piacere
 - Tramite metodi
 - `$('#footer_image').attr('src') = 'img/logo_small.png';`
 - E concatenazione degli stessi
 - `$('#footer_image').css('color','DarkGrey').attr('alt','Alt Text')`
 - Ha un metodo `.noConflict()` per rilasciare il controllo di `$`

Flessibilità

Flessibilità

- Abbiamo visto come sia possibile fare cose complesse
 - Confrontiamo la stessa cosa con JavaScript e jQuery:
 - `var image = document.getElementById('footer_image');` [JS]
 - `image = $('#footer_image');` [jQ]
- Si possono indirizzare anche classi:
 - `$('.paragraph');`
- Selezionare tutto:
 - `$('*');`
- Gerarchie:
 - `$('div code')`, `$('li > ul')`, `$('strong + em')`, `$('strong ~ em')`

Flessibilità (continua)

- Praticamente c'è una lista infinita di selettori (CSS 1-3)^[3]
 - :animated, :eq, :file, :gt, :has, :image, :lt, :odd, etc.
 - [name |= 'value'], [name*='value'], [name!='value'], etc.
 - :form :checked :password :input
- Questi semplificano notevolmente il lavoro di selezione e ricerca del tag specifico
- È possibile passare alla funzione \$ qualsiasi cosa:
 - \$('') crea un tag img
 - \$(window.alert('Ciao')) passo un elemento del DOM
- È possibile inoltre tramite .each() si possono applicare funzioni a ogni elemento di un selettore

Filtrare/Creare e Distruggere

- Un'altro metodo molto utilizzato è `.filter()`
 - `$('div').filter('.footer')` seleziona i div con classe footer
- Inoltre il vantaggio maggiore di jQuery è quello di passeggiare avanti ed indietro nel DOM
 - tramite `.find()` `.children()` `.closest()`:
 - Salgo e scendo di uno o più livelli
- Infine si possono creare elementi e classi in maniera semplice
 - `$('img').addClass('show');`
 - Oppure `.after()` `.before()` `.insertAfter()` `.insertBefore()`
- Così come creare possiamo distruggere un elemento:
 - Con `.detach()`, `.empty()`, `.remove()`, `.unwrap()`

jQuery è event-driven

- jQuery è fortissimamente legato agli eventi classici JavaScript
 - come `.click()` `.change()` `dblclick()` `.hover()` `.keydown()` `.keypress()`
- Ad ogni evento JavaScript jQuery associa una funzione
 - `$('#img').hover(function() {`
 - `$('#img').attr('src') = '/img/logo_hover.png';`
 - `});`
- In realtà questi eventi sono solo wrapper per eventi JS già definiti
- Consentono comunque tramite la nidificazione di ottenere più controllo del flusso di eventi

Binding

- Il binding è l'atto di associazione di una funzione ad un determinato evento
- Ad ogni evento JS può essere associato un'azione
- Questa si verificherà ogni qualvolta si verifica quell'evento

```
$("#text").bind("focus blur", {"background-color" : "#FFFCC"}, function (event) {  
  if (event.type == "focus") {  
    $(this).css(event.data);  
  } else {  
    $(this).css("background-color", "");  
  }  
});
```

Unbinding

- L'operazione inversa al binding
- Elimina tutte o alcune associazioni a determinati oggetti
- Quegli oggetti non risponderanno più a quell'evento
- Almeno non come erano stati definiti precedentemente

```
$(".text").unbind(); // rimuovi tutti gli eventi dai campi di testo  
$(".text").unbind("blur focus"); //rimuovi gli handlers per gli eventi focus e blur  
$(".text").unbind("blur",blurFunction); // rimuovi uno specifico handler per l'evento blur
```

Ricapitoliamo

- Si definisce un ambito di azione
 - un selettore
 - una classe CSS
 - un elemento del DOM
- Si raggiunge l'elemento con `$()`
- Si applica un metodo tipo crea o copia ecc.
 - O un'azione come click, doppio click etc.
- A questo punto impartire un comando finale
 - come `.css('color','blue')`
- Logicamente tutto questo può essere annidato a piacere

Animazioni

Animazioni con jQuery

- jQuery ha un potente set di animazioni
- Consentono di gestire scambi di dati con AJAX
 - Acronimo per Asynchronous Javascript and XML
 - Combinazione di HTML/CSS per la parte di frontend
 - DOM manipolato tramite Javascript
 - l'oggetto XMLHttpRequest (in alcuni casi iframe)
 - XML viene usato per lo scambio dei dati col server
 - Può essere usato anche JSON, EBML o anche HTML preformattato
- Le animazioni vengono eseguite una in coda all'altra
 - Appena termina una inizia un'altra
 - Questo per evitare sovrapposizioni e sfarfallii

I metodi queue e dequeue

- Nonostante questa limitazione è possibile creare code di effetti
- Tramite .queue() e .dequeue() si altera la coda degli eventi
- Passando un nome alla coda con .queue('fx'):
 - si determina quali funzioni sono presenti nella coda
- Invece passando una funzione si accoda agli effetti
 - viene eseguita al termine della coda degli effetti
 - `$('#div').queue('fx',function() { $(this).fadeOut('slow'); $(this).dequeue();`
- La funzione dequeue ritorna il controllo all'esecuzione del codice

Funzioni di animazione

- Vediamo brevemente alcune funzioni di animazione
 - .show() e hide() si commentano da sé
 - Equivalgono alla proprietà CSS visibility: {none, visible}
 - .fadeOut() e .fadeIn() modificano l'opacità
 - .fadeTo() invece setta l'opacità ad un valore [0,1]
 - .slideUp() .slideDown() e slideToggle() agiscono sull'altezza
 - Questi ultimi sono molto utilizzati
 - Il più complesso è .animate() che può impostare diversi valori

```
$("#box").animate({  
  "borderWidth" : "4px", //bordo a 4 pixel  
  "width" : "+=20px" //aumenta la larghezza di 20 pixel  
},  
"slow", //animazione lenta  
"linear", //andamento lineare  
});
```

Eventi AJAX

Eventi AJAX

- Per gestire lo scambio di dati è possibile usare dei metodi:
 - Locali
 - possono essere impostati all'interno di `$()`
 - Globali
 - includono tutto il DOM, possono essere impostati con `.bind()`
- La catena di eventi AJAX coinvolge i seguenti metodi:
 - `ajaxStart` e `ajaxStop`
 - `beforeSend`
 - `success` e `AjaxSuccess`
 - `error` e `ajaxError`
 - `complete` e `ajaxComplete`

Esempio

- In questo modo si possono visualizzare informazioni:

```
$(document)
  .ajaxStart(
    function () {
      $("#messaggio").text("caricamento...");
    }
  )
  .ajaxStop(
    function () {
      $("#messaggio").text("");
    }
  );
```

Browser Detection

- jQuery ha dei metodi per il riconoscimento del browser in uso
- Sono immagazzinati nell'oggetto \$.browser
- Ad esempio un \$.browser.msie == true individua IE
- Ancora \$.browser.version ne individua la versione
- Inoltre è pieno di altri slots come
 - boxModel, htmlSerialize, objectAll
 - cssFloat, leadingWhitespace, opacity
 - hrefNormalized, noCloneEvent, scriptEval
 - tbody

Due parole sui plugins

- Oltre a poter definire nuovi selettori e nuovi metodi
 - È possibile definire anche nuovi plugins
 - Possono implementare nuove funzioni
 - La sintassi non è difficile e possiamo rendere due servizi
 - riusare il codice scritto da noi
 - fornire un servizio alla comunità rilasciandolo

- La sintassi:

```
$.fn.extend({  
  
    nomeplugin : function () {  
  
        return this.each(function () {  
            //this è l'elemento  
        });  
  
    });  
});
```

jQuery UI

jQueryUI: cosa è?

- È una libreria accessoria a jQuery
- Fornisce effetti molto simili a quelli di Scriptaculous
- Richiede la versione di jQuery almeno 1.3.2
- Importatela in questa maniera:

```
<link type="text/css" href="css/smoothness/jquery-ui-1.7.2.custom.css" rel="Stylesheet" />  
<script src="js/jquery-1.3.2.min.js"></script><!-- Opzionale -->  
<script src="js/jquery-ui-1.7.2.custom.min.js"></script>
```

Cosa fare con le animazioni

- Fondamentalmente i metodi sono gli stessi ma
 - Si può impostare un *easing* più avanzato
 - Sono disponibili più curve di transizione
- Il metodo `.effect()` contiene molti nuovi effetti
 - come `.explode`,
 - `.blind`,
 - `.clip`,
 - `.drop`,
 - `.highlight`,
 - `.puff`

Oggetti draggables

- Essendo una libreria per le UI non possono mancare
- Draggables
 - `$("#box").draggable();`
 - `$("#box").draggable("disable");`
- `draggable()` ha un sacco di opzioni ed eventi associati
 - `appendTo`
 - Contenitore di elementi draggabili
 - `axis`
 - Consente di limitare il drag ad un solo asse
 - `containment`
 - Definisce il box in cui è consentito il drag
 - `snap`
 - Consente di far incollare l'elemento ad un altro

Droppables e resizable

- I primi sono uguali ai precedenti
- Definiscono degli oggetti che possono essere rilasciati in determinate aree
- I resizable sono delle classi che consentono ad un oggetto di essere ridimensionabile a piacere
- Nello stesso modo di una finestra
- `$("#box").resizable();`
 - maxHeight / maxWidth / minHeight / minWidth
 - containment
 - aspectRatio
 - handles

Altri elementi

- Sortable e selectable
- Widget
 - Accordion
 - Datepicker
 - Dialog
 - Progressbar
 - Slider
 - Tabs

Sviluppi futuri

- jQuery v1.7 Beta
 - Nuovi metodi
 - .on() .off() andranno a sostituire bind()/unbind() live()/die() etc.
 - AMD per il caricamento sicuro della libreria
 - Migliore supporto per HTML5
 - html5shiv

Ultime due notizie

- Slides scaricabili gratuitamente dal sito del **Gulch**
- Potete contattarmi su [@simonekalb](https://twitter.com/simonekalb)
- Andate e commentate il mio blog su <http://simone.kalb.it> oppure www.kalb.it/simone
- Mandatemi una mail all'indirizzo [simone at nodelay dot org](mailto:simone@nodelay.org)
- Venite a trovarmi al **Crs4**
- Fatami le domande adesso!

GRAZIE PER L'ATTENZIONE