JQuery

Riccardo Rosati

Linguaggi e tecnologie per il Web
Corso di laurea in Ingegneria informatica e automatica
Sapienza Università di Roma
a.a. 2018/2019

http://www.dis.uniroma1.it/~rosati/lw/

SAPIENZA
UNIVERSITÀ DI ROMA

Librerie e framework per JavaScript

Libreria:

- insieme di funzioni e strutture dati che possono essere utilizzate da una applicazione
- È il codice dell'applicazione ad accedere alla libreria (tramite collegamenti)
- **JQuery**, **React**, **JQueryUI**, **Dojo** sono esempi di librerie JS

Framework:

- architettura di supporto alla progettazione dell'applicazione
- condiziona la struttura di base e lo sviluppo del codice dell'applicazione
- il codice è inserito nelle strutture del framework (il framework "accede" al codice dell'applicazione)
- **AngularJS**, **Ember.js** sono esempi di framework per JS

JQuery

Prima versione rilasciata nel gennaio 2006

Obiettivi: creare una libreria in grado di:

- Semplificare la programmazione in JavaScript
- estendere le funzioni native di JavaScript
- garantire il funzionamento cross-browser degli script

JQuery

Il framework JQuery può essere scaricato in due formati:

- development (non compresso)
  - https://code.jquery.com/jquery-3.2.1.js
- production (compresso)
  - https://code.jquery.com/jquery-3.2.1.min.js

Esistono varie release (versioni) del framework

Includere JQuery in un documento HTML

La libreria JQuery è utilizzabile in vari modi in un documento HTML:

1. Scaricando la libreria in locale (stessa directory del documento HTML, o sua sottodirectory) e creando nel documento un collegamento con la copia locale
2. Creando nel documento un collegamento con la libreria presente nel repository ufficiale di jquery.com (JQuery CDN)
3. Creando nel documento un collegamento con la libreria presente in un altro Content Delivery Network

Includere JQuery in un documento HTML

Assumendo di avere nella cartella locale il file scaricato da jquery.com:

- Collegamento alla versione compressa di JQuery:

```
<script src="jquery-3.2.1.min.js">
  ...
</script>
```

- Collegamento alla versione non compressa di JQuery:

```
<script src="jquery-3.2.1.js">
  ...
</script>
```

Includere JQuery in un documento HTML

Collegamento al repository di jquery.com:

- Collegamento alla versione compressa di JQuery:

```
<script src="//code.jquery.com/jquery-3.2.1.min.js">
  ...
</script>
```

- Collegamento alla versione non compressa di JQuery:

```
<script src="//code.jquery.com/jquery-3.2.1.js">
  ...
</script>
```

Nota bene: in questo modo il funzionamento degli script dipende dalla raggiungibilità di queste risorse su un altro server

Uso di un CDN

- In alternativa si può utilizzare una copia del framework presente su un altro content delivery network (CDN)
- Ad esempio, dal CDN di Google:

```
<script src=
  "http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.js">
</script>
```

- Anche in questo caso il funzionamento degli script dipende dalla raggiungibilità di risorse su un altro server
- Se l'utente ha già scaricato questa risorsa nella cache del browser, si ha una diminuzione del tempo di caricamento della pagina
- Inoltre i CDN (ad es. Google) includono molte librerie/framework oltre JQuery e offrono garanzie di compatibilità se si includono più framework

Esempio

Esempio di pagina che usa la libreria JQuery:

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Demo</title>
  </head>
  <body>
    <script src="//code.jquery.com/jquery-3.2.1.min.js">
    <script> // script dell'utente... </script>
    ...
  </body>
</html>
```

JQuery: sintassi

- Sintassi di base: `$(selettore).azione()`
- Il selettore seleziona uno o più elementi del documento HTML
  - Usa la sintassi CSS
- L'azione è una funzione JQuery sugli elementi selezionati
- Esempi:
  - `$(this).hide()` nasconde l'elemento corrente
  - `$("#mio-id").fade()` fa il fading dell'elemento con id uguale a mio-id

Selettori

- Oggetto JQuery (denotato anche con il simbolo del dollaro $)
- `$("nome-elemento")` = seleziona gli elementi HTML con nome nome-elemento
- `$("#valore-id")` = seleziona l'elemento HTML con valore dell'attributo id uguale a valore-id
- `$(".nome-class")` = seleziona l'elemento HTML con valore dell'attributo class uguale a nome-class
- Esempio: `$("p.myclass")` seleziona tutti gli elementi p con valore dell'attributo class uguale a myclass
- Si possono anche usare i metodi del DOM
  - Esempio: `$(getElementById(valore-id))`

Selettori

- `$("*")` = seleziona tutti gli elementi HTML
- `$(this)` = seleziona l'elemento HTML corrente
- `$(".nome-class")` = seleziona l'elemento HTML con valore dell'attributo class uguale a nome-class
- Si possono usare filtri (preceduti da ":")
- Esempi:
  - `$("p:first")` = seleziona il primo elemento p
  - `$("p:last")` = seleziona l'ultimo elemento p
  - `$("p:even")` = seleziona gli elementi p pari
  - `$("p:odd")` = seleziona gli elementi p dispari
  - …

Selettori

- Si possono scrivere più selettori contemporaneamente:
  - `$("p:first, #mio-id")` seleziona sia il primo p che l'elemento con id mio-id
- Si possono scrivere selettori gerarchici (padre-figlio o predecessore-successore)
  - `$("#mio-id p")` seleziona gli elementi p **successori** dell'elemento con id mio-id
  - `$("#mio-id > p")` seleziona gli elementi p **figli** dell'elemento con id mio-id
  - `$("#mio-id + p")` seleziona l'elemento p che **segue** l'elemento con id mio-id
- Si possono usare condizioni sugli attributi:
  - `$("a[title='nuovo']")` seleziona gli elementi a con attributo title uguale a nuovo

Metodi per il contenuto degli elementi

Metodi per leggere/scrivere il contenuto degli elementi HTML:

- **text**: legge/scrive il contenuto testuale di un elemento
- **html**: legge/scrive il contenuto HTML di un elemento
- **val**: legge/scrive il valore dei campi delle form

Se il metodo è senza argomenti, effettua una lettura (get), se invece ha un argomento, effettua una scritture (set)

Esiste anche il metodo **attr** per leggere/scrivere i valori degli attributi

Metodi per il contenuto degli elementi

Esempi di set:

- `$("p").text("Nuovo testo dei paragrafi");`
- `$("p").html("Nuovo testo paragrafi <strong>con tag</strong>");`
- `$("#cognome").val("Rossi")`
- `$("img").attr("width","800");`
- `$("img").attr({width: "800", height: “600"});`

Esempi di get:

- `var testo = $("h1").text();`
- `var codiceHTML = $("p").html();`
- `var cognomeInseritoDaUtente = $("#cognome").val()`
- `var valoreAttributoWidth = $("img").attr("width");`

Metodi per CSS

Il metodo **css** serve per leggere/scrivere le proprietà CSS degli elementi HTML

Esempi:

- `$(#myId).css("background-color")`

  legge il valore della proprietà CSS background-color dell'elemento con id myId
- `$(#myId).css("background-color","red")`

  setta a "red" il valore della proprietà CSS background-color dell'elemento con id myId

Metodi per CSS

Altri metodi legati all'uso dei CSS:

- `addClass()`
- `removeClass()`
- `toggleClass()`
- `width()`
- `height()`
- `innerWidth()`
- `innerHeight()`
- `outerWidth()`
- `outerHeight()`

Metodi per navigare il DOM

Metodi che restituiscono il padre o i predecessori dell'elemento selezionato:

- `parent()`
- `parents()`
- `parentsUntil()`

Metodi che restituiscono i figli o i discendenti dell'elemento selezionato:

- `children()`
- `find()`

Metodi per navigare il DOM

Metodi che restituiscono i fratelli dell'elemento selezionato:

- `siblings()`
- `next()`
- `nextAll()`
- `nextUntil()`
- `prev()`
- `prevAll()`
- `prevUntil()`

Altri metodi di navigazione:

- `first()`
- `last()`
- `eq()`
- `filter()`
- `not()`

Metodi per effetti di animazione

- JQuery prevede molti metodi che realizzano effetti di animazione
- Il metodo animate permette di effettuare una animazione di alcune proprietà CSS degli elementi a cui è applicato
- Esempio:

```
$("button").click(function(){
    $(“#mioId").animate({left: ‘300px'});
});
```

- Questo metodo sposta l’elemento con id mioId verso destra, finché la proprietà left non raggiunge il valore di 300px

Passaggio di funzioni come argomenti

- Nel precedente esempio, si noti il passaggio di una funzione come argomento di un'altra funzione
- Questa è una caratteristica di JavaScript: le funzioni possono essere passate come argomento di una funzione, e possono anche essere restituite come risultato di una funzione
- Semplice sempio:

```
function square(x) {
  return x * x;
}

var square = function(x) {
  return x * x;
};
```

- JQuery usa pesantemente questa caratteristica di JavaScript

Metodi per effetti di animazione

Metodo	Descrizione
`animate()`	Esegue una animazione sugli elementi selezionati
`clearQueue()`	Cancella i metodi in coda sugli elementi selezionati
`delay()`	Setta un ritardo per le funzioni in coda sugli elementi selezionati
`dequeue()`	Toglie la prossima funzione dalla coda e la esegue
`fadeIn()`	Esegue il fade-in (assolvenza) degli elementi selezionati
`fadeOut()`	Esegue il fade-out (dissolvenza) degli elementi selezionati
`fadeTo()`	Setta l'opacità del fade-in e fade-out degli elementi selezionati
`fadeToggle()`	Alterna l'esecuzione dei metodi `fadeIn()` e `fadeOut()`

Metodi per effetti di animazione

Metodo	Descrizione
`finish()`	Interrompe, rimuove e completa tutte le animazioni in coda per gli elementi selezionati
`hide()`	Nasconde gli elementi selezionati
`queue()`	Mostra la coda dei metodi sugli elementi selezionati
`show()`	Mostra gli elementi selezionati
`slideDown()`	Mostra (scorrendo) gli elementi selezionati
`slideToggle()`	Alterna l'esecuzione dei metodi slideUp() e slideDown()
`slideUp()`	Nasconde (scorrendo) gli elementi selezionati
`stop()`	Interrompe l'animazione corrente sugli elementi selezionati
`toggle()`	Alterna l'esecuzione dei metodi hide() e show()

Gestione degli eventi

- La maggior parte degli eventi previsti nel DOM hanno un corrispondente metodo JQuery
- Si può eseguire una funzione (**event handler**) in corrispondenza dell'evento, passando tale funzione come argomento del metodo JQuery corrispondente
- Esempio:

```
$("button").click(function(){
  // codice per gestire il click sul bottone
});
```

Event handler multipli

La funzione on permette di associare più event handler allo stesso elemento

Esempio:

```
$("li").on({
        mouseenter: function(){
            $(this).css("background-color", "lightblue");
        },
        mouseleave: function(){
            $(this).css("background-color", "lightred");
        },
        click: function(){
            $(this).css("background-color", "lightgreen");
        }
  });
```

Evento (document).ready

Molto spesso tutte le funzioni JQuery dello script vengono incapsulate dentro ad un evento “document ready”:

```
$(document).ready(function(){
  …
   // qui ci sono le funzioni jQuery
  …
});
```

- L’evento (document).ready avviene al termine del caricamento del documento da parte del browser
- In questo modo si è sicuri che il codice JQuery verrà eseguito solo dopo la fine del caricamento, evitando così possibili malfunzionamenti
- Inoltre in questo modo diventa possibile mettere gli script nell’intestazione (head) del documento HTML

Evento (document).ready

Esempio:

```
$(document).ready(function(){
  $("button").click(function(){
      $(#myId).css("background-color", "red");
    });
});
```

Notare ancora una volta il passaggio di funzioni come argomento di altre funzioni (anche in modo annidato)

JQuery e AJAX

JQuery rende molto semplice l'attivazione di funzionalità AJAX

Esempio:

```
$("button").click(function(){
        $("#myDiv").load("xyz.htm");
    });
```

- la funzione load permette il caricamento (asincrono) della risorsa xyz.htm
- Al verificarsi del click su qualsiasi elemento button, il contenuto di xyz.htm viene caricato nell'elemento con id myDiv

Funzione load

La funzione load può avere, come secondo argomento, una funzione di **callback**, che viene eseguita al termine del caricamento della risorsa

Esempio:

```
<script>
$(document).ready(function(){
  $("button").click(function(){
     $("#myDiv").load(“xyx.htm", function(responseTxt, statusTxt, xhr){
       if(statusTxt == "success")  alert(“Caricamento terminato");
       if(statusTxt == "error")  alert("Errore “ + xhr.status+":" +
          xhr.statusText);
      });
   });
});
</script>
```

Metodi get e post

Metodi get e post:

- `$.get(url, callback)`
  - Invia (in modo asincrono) una richiesta http di tipo get al server
- `$.post(url, callback)`
  - Invia (in modo asincrono) una richiesta http di tipo post al server

Esempio get:

```
$("button").click(function(){
    $.get("programma-server.asp", function(data, status){
        alert("Dati ricevuti: " + data + "\nStatus: " +
status);
    });
});
```

Metodi get e post

Esempio di uso della funzione post:

```
$("button").click(function(){
    $.post("programma-server.asp",
    {
        nome: "Mario Rossi",
        matricola: "01234567"
    },
    function(data, status){
        alert("Dati ricevuti: " + data + "\nStatus: " + status);
    });
});
```

Riferimenti

- http://api.jquery.com/
- https://learn.jquery.com/
- http://www.w3schools.com/jquery/
- http://www.html.it/guide/guida-jquery/