



Javascript **jQuery**



Impariamo JQuery

Giulio Angiani
I.I.S. "Blaise Pascal" - Reggio Emilia



JQuery javascript framework



Cos'è JQuery

JQuery è una libreria javascript per lo sviluppo cross-browser di applicazioni web based



E' uno dei framework js più utilizzati

Numerosissime funzioni e tanti plugin di facile utilizzo basati su di esso

La versione più recente è **3.5.1**

Sito di riferimento : <https://jquery.com/>

OBIETTIVO : rendere più agevole l'uso di javascript tramite **selezione** e **gestione eventi** più semplici e **cross-browser**

- SELEZIONE
- GESTIONE EVENTI
- AJAX - Eventi asincroni



JQuery - Struttura standard

Struttura base di una pagina che usa jQuery

JS

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <title>Javascript</title>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <script>
      $(document).ready(function(){
        // qui le istruzioni del mio programma
      });
    </script>
  </head>
  <body>

</body>
</html>
```

- Fondamentale è **importare** il file **jquery.min.js** dal repository cdn standard
- Utilissimo inserire le funzioni all'interno della **function(){}** che viene chiamata alla fine del caricamento della pagina
 - **\$(document).ready()** è l'evento di caricamento finito
 - **function(){}** è quanto succederà alla fine del caricamento



jQuery - Hello world!

L'hello world di jQuery è il seguente

```
<script>
  $(document).ready(function(){
    alert("Ciao mondo!");
  });
</script>
```

JS

Al caricamento della pagina appare una finestra di alert che saluta l'utente



jQuery - Event handling

In jQuery è semplice associare un evento ad un elemento della pagina anche **senza** esplicitare codice javascript all'interno dell'elemento stesso

si abbia il codice seguente

```
<body>
  <button id='btn01'>mostra</button>
</body>
```

HTML

```
$(document).ready(function(){
    $("#btn01").on("click", function() {
        alert("Hai cliccato sul bottone!");
    });
});
```

JS

Al caricamento della pagina:

- viene selezionato l'elemento con **id = btn01** (jQuery seleziona con la stessa logica dei CSS)
- a questo elemento viene agganciato l'evento onClick con **.on("click",)**
- la risposta all'evento è la funzione **function() { alert("Hai cliccato sul bottone!"); }**



jQuery - Event handling (2)

Nell'esempio seguente la risposta all'evento modifica il contenuto di un altro oggetto della pagina

si abbia il codice seguente

```
<body>
  <div id='container'></div>
  <button id='btn01'>mostra</button>
</body>
```

HTML

```
$("#btn01").on("click", function() {
  $("#container").html("Hai cliccato sul bottone!");
});
```

JS

Nella funzione di risposta all'evento infatti

- viene selezionato il div con **id=container**
- il metodo **.html()** modifica il contenuto del div stesso



JQuery - Event handling (3)

Se l'elemento sul quale fare l'operazione di risposta all'evento è lo stesso sul quale si è scatenato l'evento (nel caso specifico il button), possiamo riferirci a questo con la variabile **\$(this)**

```
$("#btn01").on("click", function() {  
    $(this).html("Hai cliccato su di me!");  
});
```

JS



JQuery - Event handling (4)

Obiettivo: eliminare ogni chiamata JS dalla parte HTML

In Javascript standard scriviamo:

```
<a href="#" onClick="confermaCancellazione(3)">cancella</a>
```

HTML

Vogliamo togliere ogni riferimento a JS

- dobbiamo aggiungere informazioni per
 - selezione
 - passaggio parametri

```
<a href="#" class='linkcancellazione' numerodiriga=3>cancella</a>
```

HTML

```
// aggancio l'evento a tutte le 'anchor' con class 'linkcancellazione'  
$("a.linkcancellazione").on("click", function() {  
    // recupero il valore del numero di riga  
    var idriga = $(this).attr("numerodiriga");  
    if (confirm(...)) {  
    }  
});
```

JS



Client-side validation form

Esercizio:

Modificare il file `ricerca_avanzata.php` per gestire il link di cancellazione tramite jQuery

10 minuti....



JQuery - Content updating

Modificare dinamicamente in contenuto di una pagina

- già visto in JS
- usiamo qui le funzioni JQ

Obiettivo:
cliccando sul bottone (o premendo invio) il messaggio inserito nel campo input viene visualizzato in coda nel box **Chat messages**

<p>Chat messages:</p> <p>>>Ciao >>come stai?</p>	<input type="text" value="digita qui il tuo messaggio"/> <input type="button" value="Invia"/>
--	--

Useremo le funzioni apposite di jQuery

- text()
- html()
- append() / prepend()



JQuery - Content updating

Abbiamo un punto in cui inserire i dati ed un bottone

```
<div class="row">  
  <input type='text' id='msg' class='w-100' placeholder='digita qui il tuo messaggio'>  
  <button value='invia' id='btnmsg'>Invia</button>  
</div>
```

HTML

ed uno di cui vogliamo modificare il contenuto

```
<div id='chatcontainer' class='mt-3'>  
  
</div>
```

HTML

Dobbiamo scegliere

- un **evento** da gestire
- una **funzione** di risposta
- un elemento da **modificare**



JQuery - Content updating

Possiamo gestire l'evento in questo modo

```
$("#btnmsg").on("click", function(){  
    // recupero messaggio  
    var msg = $("#msg").val();  
    // appendo all'elemento con id=chatcontainer  
    $("#chatcontainer").append(msg);  
});
```

JS

Dove:

- **`$("#btnmsg")`** è l'elemento sul quale ascolto l'evento
- **`click`** è l'evento atteso
- **`function(){...}`** è la funzione di risposta



JQuery - Content updating

Problemi e miglioramenti

```
$("#btnmsg").on("click", function(event){  
    event.preventDefault();  
  
    // recupero messaggio  
    var msg = $("#msg").val();  
  
    // appendo all'elemento con id=chatcontainer  
    // creo un elemento DIV in modo da poterci applicare funzioni e classi CSS  
    $("#chatcontainer").append("<div> >>" + msg + "</div>");  
  
    // svuoto il campo input  
    $("#msg").val("");  
});
```

JS

Dove:

- **function(event){...}** è la funzione di risposta che gestisce l'evento
- **event.preventDefault()** evita il submit della form se premo INVIO
- **append("<div> >>" + msg + "</div>");** appende elementi DIV e non solo testo
- **\$("#msg").val("");** cancella il contenuto del campo dopo averlo utilizzato



JQuery - Content updating

Esercizio:

Modificare il file precedente per simulare una chat con due utenti

- predisporre due campi input, uno per ogni utente della chat
- visualizzare i messaggi di colore diverso, uno per ogni utente
- visualizzare i messaggi allineando quelli di un utente a destra e quelli dell'altro a sinistra
- (opzionale) visualizzare in carattere più piccolo l'ora dell'invio del messaggi di un utente

Chat messages:	Utente 1: digita qui il tuo messaggio
ciao	
che fai?	
	dormo
	e tu?
studio jQuery!	
	bravo!!!

Chat messages:	Utente 1: digita qui il tuo messaggio
Ciao!	
alle 11:21 del 29/05/2020	
che fai?	
alle 11:21 del 29/05/2020	
	dormo... e tu?
	alle 11:21 del 29/05/2020

30 minuti...



Timing in Javascript

Finora abbiamo visto come gestire le risposte ad eventi generati da **utente** - click del mouse - invio dei dati - pressione di un tasto

Le funzioni di **timing** di Javascript servono per **temporizzare** eventi di gestione della pagina che non devono rispondere necessariamente ad azioni dell'utente

Due funzioni predefiniti di Javascript:

- **setTimeout**

- permette di eseguire **una sola volta** una funzione js **dopo** un certo intervallo di tempo definito dal programmatore
- sintassi: **setTimeout(<funzione da eseguire>, <delay in milliseconds>);**
- esempio: `setTimeout(function(){ alert("Hello"); }, 3000);`

- **setInterval**

- permette di eseguire **ripetutamente** una funzione js **ogni** certo intervallo di tempo definito dal programmatore
- sintassi: **setInterval(<funzione da eseguire>, <delay in milliseconds>);**
- esempio: `setInterval(function(){ alert("Hello"); }, 3000);`



Timing in Javascript

Esempio:

```
function saluta() {  
    alert('ciao ciao');  
}  
  
$(document).ready(function(){  
    $('#saluta').on('click', function(){  
        // saluta dopo 2 secondis  
        setTimeout(function() {alert('ciao');}, 2000);  
    });  
  
    $('#salutasempre').on('click', function(){  
        // saluta dopo 2 secondis  
        setInterval(saluta, 2000);  
    });  
});
```

JS + JQ

Bello!!

ma come fermo il `setInterval`?

ref: javascript/time.html



Timing in Javascript

Terza funzione di timing è **clearInterval** che serve proprio a questo
sintassi: **clearInterval(identificativo);**

Dove:

- **identificativo** è un numero che viene restituito da **setInterval** quando invocato

```
var ferma = null; // una variabile di gestione scelta per il nostro esempio

$('#salutasempre').on('click', function(){
    // saluta dopo 2 secondi
    ferma = setInterval(saluta, 2000);
});

$('#fermasaluto').on('click', function(){
    console.log("fermo saluta...");
    clearInterval(ferma);
});
```

JS + JQ



JQuery - Content updating

Esercizio: Scrivere un programma javascript che ogni 5 secondi ricarica la pagina **chat.php**

La pagina **chat.php** contiene le istruzioni che mostrano lo scambio di messaggi inviati da vari utenti

I dati sono presenti sul database nella tabella **messaggi** che ha i seguenti campi

- id
- nome utente
- messaggio
- data ora del messaggio

Se tutto è scritto correttamente, inserendo una nuova riga nella tabella, si dovrebbe visualizzare automaticamente un nuovo messaggio nella pagina

suggerimenti

- simulare un po' di ritardo nella pagina php con l'istruzione **sleep()**
- usare **location.reload()** per ricaricare la pagina via javascript

30 minuti...

Chat		loading....
giulio		
Ciao! che fai?		
2020-06-20 10:00:00		
francesca		
Sto studiando javascript... e tu?		
2020-06-20 10:02:00		
giulio		
Ho appena finito! Molto bello!		
2020-06-20 10:03:00		





Giulio Angiani
I.I.S. "Blaise Pascal" - Reggio Emilia