jQuery

jQuery è una libreria JavaScript che dà la possibilità di manipolare una pagina html lato client.

Con jQuery è possibile facilmente:

- Selezionare gli elementi HTML
- Eseguire operazione con gli elementi HTML
- Eseguire operazione CSS
- Eseguire funzioni a seguito di eventi HTML
- DOM HTML attraversamento modifica
- Eseguire processi AJAX (Asynchronous Javascript And Xml) che consentono di cambiare il contenuto delle pagine web senza effettuare refresh dell'intera pagina

È possibile utilizzare jQuery attraverso:

- Una installazione locale
  - Scaricando la libreria jQuery nel proprio Server (ad esempio in /jquery/jquery-2.1.3.min.js) e includendola nel codice HTML.

```
<head>
 <script src =".jquery/3.5.1/jquery.min.js">
 </script>
</head>
```

- Attraverso un Content Delivery Network (CDN)
  - Includendo la libreria jQuery direttamente da una dei CDN (ad esempio utilizzando la libreria jQuery di Google)

```
<head>
 <script> <src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
 </script>
</head>
```

jQuery viene quasi esclusivamente per leggere o manipolare un documento HTML attraverso il suo DOM (Document Object Model). Non appena il browser ha caricato la pagina e quindi creato il suo DOM è possibile utilizzare la variabile *$ (document)* del jQuery per navigare all'interno della pagina.

Se si vuole che una funzioni si attivi immediatamente dopo il caricamento della pagina, è possibile farlo richiamandola all'interno della funzione $(document).ready ().

```
$(document).ready(function() {
    // do stuff when DOM is ready
});
```

Le funzione jQuery devono essere scritte all'interno dei tag <script> </script> di HTML

```
<html><head>
<title>The jQuery Example</title>
 <script type = "text/javascript"
src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
 </script>
 <script type = "text/javascript" language = "javascript">
         $(document).ready(function() {
            $("div").click(function() {alert("Hello, world!");});
         });
 </script>
 </head>
 <body><div id = "mydiv">Click on this to see a dialogue box.</div>
 </body>
</html>
```

Questo codice produce il seguente effetto <Esempio1>

In genere il codice viene sviluppato in un file diverso, nell'esempio si utilizza il file ./jquery/custom.js, incluso nella pagina html

```
/* Filename: /jquery/custom.js */
$(document).ready(function() {
$("div").click(function() {
      alert("Hello, world!");});});});
```

```
<html><head>
 <title>The jQuery Example</title>
  <script type = "text/javascript" src =
"https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
  </script>
  <script type = "text/javascript" src = "/jquery/custom.js"></script>
  </head>
   <body>
      <div id = "mydiv">Click on this to see a dialogue box.</div>
   </body></html>
```

Per selezionare degli elementi html ed effettuare delle azioni su di essi in JQuery la sintassi base è la seguente:

```
$(selector).action()
```

- Il simbolo $ definisce l'accesso alle librerie JQuery
- *selector* rappresenta un elemento HTML o una *query* per raggiungerlo che di base usa la stessa terminologia dei CSS
- *action* rappresenta l'azione che si vuole compiere

Ad esempio:

```
//Per nascondere tutti gli elementi <hr> useremo
$("hr").hide();
//Per ascondere tutti gli elementi che hanno come class=”nomeCalsse”
$(".nomeClasse").hide();
//Nascondere tutti gli elementi con un id=”idOggetto”
$(".idOggetto").hide();
```

Selettori jQuery

Per individuare degli elementi in una pagina HTML si utilizza la seguente sintassi `:$(selector)` in cui si seleziona un elemento di una pagina HTML utilizzando la stessa tecnica dei CSS quindi:

- Basandosi sul nome del tag html

```
<script type = "text/javascript" >
$(document).ready(function(){
    $("#pulsanteNascondi").click(function(){
        $("h1").hide();});
$("#pulsanteVisualizza").click(function(){
        $("h1").show();});
}); </script>


    <body><div id = "mydiv">
<h1>Testo h1 .</h1>
<button id="pulsanteNascondi">Click to hide h1</button>
<button id="pulsanteVisualizza">Click to show h1</button>
</div></body></html>
```

Questo codice produce il seguente effetto Esempio2

- Selezione in base alla classe. `$(".classeElemento")`

Ad esempio creiamo una funzione che al click su un pulsante nasconde tutti gli elementi con classe uguale a "classeDaNascondere"

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <title>esempio3</title>
   <script type ="text/javascript"
    src ="https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
   </script>
   <script type ="text/javascript"
     $(document).ready(function(){
       $("#pulsanteNascondi").click(function(){
         $(".classeDaNascondere").hide();});
       $("#pulsanteVisualizza").click(function(){
          $(".classeDaNascondere").show();});});});
</script>
</head>
<body><div>
 <div id = "mydiv">
```

```
  <p class="classeDaNascondere">Text uno.</p>
  <p class="classeDaNascondere">Text due.</p>
  <p class="classeDaNascondere"> Text tre.</p>
  <button id="pulsanteNascondi">Click to hide text</button>
  <button id="pulsanteVisualizza">Click to show text</button>
</div></div></body></html>
```

Questo codice produce il seguente effetto <Esempio3>

- Selezione tramite ID `$("#idElemento")`

Ad esempio creiamo una funzione che al click di bottone nasconde l'elemento con id = "elementoDaNascondere"

```
<script type = "text/javascript"
$(document).ready(function(){
    $("#pulsanteNascondi").click(function(){
        $("#elementoDaNascondere").hide();});});
</script>
```

Questi sono tutte le possibili istruzioni per i selettori JQUERY

Selettore	Esempio	Cosa seleziona
*	$("*")	Tutto
#	$("#idElemento")	Gli elementi con id=" idElemento"
.	$(".classeElemento")	Gli elementi con classe = "classeElemento"
.	$(".classe1, .classe2")	Gli elementi con classe = "classe1" oppure "classe2"
elemento	$("br")	Tutti gli elementi di tipo br
elementi	$("h1,h2,h3")	Tutti gli elementi h1, h2, h3
:first	$("h2:first")	Il primo elemento <h2>
:last	$("h2:last")	L'ultimo elemento <h2>
:even	$("li:even")	Gli elementi pari
:odd	$("li:odd")	Gli elementi dispari
:first	$("p:first-child")	Tutti gli elementi <p> che sono il primo figlio del proprio genitore
first-of-type	$("p:first-of-type")	Tutti gli elementi <p> che sono il primo figlio di tipo <p> del proprio genitore
last-child	$("p:last-child")	Tutti gli elementi <p> che sono l'ultimo figlio di tipo <p> del proprio genitore

last-of-type	**$("p:last-of-type")**	**Tutti gli elementi <p> che sono l'ultimo figlio di tipo <p> del proprio genitore**
nth-child	**$("p:nth-child(2)")**	**Tutti gli elementi <p> che sono il secondo figlio del proprio genitore**
nth-last-child	**$("p:nth-last-child(2)")**	**Tutti gli elementi <p> che sono il secondo del proprio genitore a contare dall'ultimo**
nth-of-type	**$("p:nth-of-type(2)")**	**Tutti gli elementi <p> che sono il secondo figlio di tipo <p> del proprio genitore**
nth-last-of-type	**$("p:nth-last-of-type(2)")**	**Tutti gli elementi <p> che sono il secondo figlio di tipo <p> del proprio genitore a contare dall'ultimo**
only-child	**$("p:only-child")**	**Tutti gli elementi <p> che sono l'unico figlio del proprio genitore**
only-of-type	**$("p:only-of-type")**	**Tutti gli elementi <p> che sono l'unico figlio di tipo <p> del proprio genitore**
>	**$("div > p")**	**Tutti gli elementi <p> che sono figli diretti di un elemento **
parent	**$("div p")**	**Tutti gli elementi <p> discendenti di un elemento **
+	**$("div + p")**	**Tutti gli elementi <p> vicini ad elementi **

~	$("div ~ p")	Tutti gli elementi <p> fratelli di un elemento
eq	$("ul li:eq(3)")	Il quarto elemento di una lista
gt	$("ul li:gt(3)")	Elementi di una lista con indice superiore a 3
lt	$("ul li:lt(3)")	Elementi di una lista con indice inferiore a 3
not	$("p:not(:empty)")	Tutti gli elementi <p> non vuoti
header	$(":header")	Tutti gli elementi header <h1>, <h2>, <h3> etc
animated	$(":animated")	Tutti gli elementi animati
:focus	$(":focus")	L'elemento che ha il focus
contains	$(":contains(Testo)")	Gli elementi che contengono la parola "Testo"
has	$("div:has(div)")	Gli elementi che al loro interno hanno elementi
empty	$(":empty")	Gli elementi vuoti
parent	$(":parent")	Gli elementi che sono padri di un altro elemento
hidden	$("div:hidden")	Gli elementi nascosti
visible	$("div:visible")	Gli elementi visibili

root	**$(":root")**	**L'elemento radice**
lang	**$("span:lang(it)")**	**Gli elementi che hanno "it" come attributo lang**
attribute	**$("[href]")**	**Gli elementi con un attributo di tipo href**
attribute equal	**$("[href=index.aspx]")**	**Gli elementi che hanno l'attributo href uguale a " index.aspx"**
attribute not equal	**$("[href!= index.aspx]")**	**Gli elementi che non hanno l'attributo href uguale a " index.aspx"**
attribute end value	**$("[href$='.aspx]")**	**Gli elementi che non hanno l'attributo href terminante con ".aspx"**
|=	**$("[title|='titolo']")**	**Gli elementi che il cui attributo title è uguale a 'titolo' oppure che inizia con 'titolo' seguito da un trattino**
~	**$("[title^='titolo']")**	**Gli elementi che il cui attributo title inizia con 'titolo'**
~=	**$("[title~='titolo']")**	**Gli elementi che il cui attributo title è uguale a 'titolo'**
***=**	**$("[title*='titolo']")**	**Gli elementi che il cui attributo title contiene 'titolo'**
:input	**$(":input")**	**Tutti gli elementi input**
:text	**$(":text")**	**Tutti gli elementi input con type="text"**

:password	$(":password")	Tutti gli elementi input con type="password"
:radio	$(":radio")	Tutti gli elementi input con type="radio"
:checkbox	$(":checkbox")	Tutti gli elementi input con type="checkbox"
:submit	$(":submit")	Tutti gli elementi input con type="submit"
:reset	$(":reset")	Tutti gli elementi input con type="reset"
:button	$(":button")	Tutti gli elementi input con type="button"
:image	$(":image")	Tutti gli elementi input con type="image"
:file	$(":file")	Tutti gli elementi input elements with type="file"
:enabled	$(":enabled")	Tutti gli elementi input abilitati
:disabled	$(":disabled")	Tutti gli elementi input disabilitati
:selected	$(":selected")	Tutti gli elementi input selezionati
:checked	$(":checked")	Tutti gli elementi input spuntati

Un evento in una pagina web corrisponde ad un'azione effettuata da un visitatore. Esempi sono

- Selezione di un elemento
- Movimento del mouse su un elemento
- Click su un "elemento

Quasi tutti gli eventi del DOM hanno un jQuery metodo corrispondente. Per assegnare un evento ad un elemento del DOM si usa una sitassi del tipo `$(elemento).evento()`.
Ad esempio `$("#button1").click();` corrispondente al click su di un elemento html con id=`#button1`.

Per definire cosa accade quando l'evento è intercettato si utilizza la sintassi seguente:

```
$(elemento).evento(function(){
    //Azione
});
```

Ad esempio

```
$("button1").click(function(){
    alert('hai cliccato un pulsante Button1!');
   });
```

genera una MessagBox con testo ”hai cliccato un pulsante Button1” al click dell’elemento con id=”button1”.

Le funzioni legate ai possibili eventi sono questi

Metodo	Descrizione
blur()	**il metodo blur() viene eseguito quando un elemento HTML perde il focus nella pagina.**
bind()	**Collega gli eventi al gestore**
change()	**Viene scatenato quando un elemento cambia il suo valore. si applica solo a <input>, <textarea> e <select>**
click()	**il metodo click() viene scatenato quando l'utente clicca su un elemento del DOM.**
delegate()	**Collega degli eventi a determinati figli di un elemento**
dblclick()	**il metodo dblclick() viene scatenato quando l'utente esegue un doppo click su un elemento**
event.currentTarget	**Restituisce l'elemento corrente (in genere uguale a this)**
event.data	**Contiene i dati facoltativi passati a un metodo quando viene associato il gestore di esecuzione**
event.delegateTarget	**Restituisce l'elemento a cui è stato collegato il gestore di eventi jQuery**
event.isDefaultPrevented()	**Indica se event.preventDefault( ) è stato chiamato per l' oggetto**
event.isImmediatePropagationStopped()	**Indica se event.stopImmediatePropagation() è stato chiamato per un oggetto**

event.isPropagationStopped()	**Indica se event.stopPropagation() è stato chiamato per un oggetto**
event.namespace	**Restituisce il namespace di un evento**
event.pageX	**Restituisce la posizione del mouse rispetto al bordo sinistro del documento**
event.pageY	**Restituisce la posizione del mouse rispetto al bordo superiore del documento**
event.preventDefault()	**Inibisce l'azione predefinita dell'evento**
event.relatedTarget	**Indica quale elemento è entrato o uscito dal target del mouse**
event.result	**Contiene l'ultimo valore restituito da un gestore di eventi**
event.stopImmediatePropagation()	**Impedisce ad altri gestori di eventi da essere chiamati**
event.stopPropagation()	**Ferma la propagazione degli eventi sugli elementi padre**
event.target	**Restituisce quale elemento ha attivato un determinato evento**
event.timeStamp	**Restituisce il numero di millisecondi a partire dal 1 gennaio 1970, da quando l'evento è stato attivato**
event.type	**Restituisce il tipo di evento scatenato**
event.which	**Restituisce tasto del mouse o della tastiera che ha scatenato l'evento**
focus()	**il metodo focus() viene eseguito quando un elemento HTML prende il focus nella pagina.**

focusin()	**L' evento si verifica quando un elemento o uno dei suoi contenuti ottiene il focus.**
focusout()	**L' evento si verifica quando un elemento o uno dei suoi contenuti perde il focus**
hover()	**Il metodo hover() è una combinazione dei metodi mouseenter() e mouseleave(). Accetta due funzioni; la prima viene eseguita al mouseenter(), mentre la seconda al mouseleave()**
keydown()	**Viene scatenato quando si preme un tasto della tastiera**
keypress()	**Viene scatenato quando si preme un tasto della tastiera. Simile al keydown ma non viene scatenato alla pressione di alcuni tasti come ad esempio esc, ctrl o shift**
keyup()	**si verifica quando un tasto della tastiera viene rilasciato .**
mousedown()	**Il metodo mousedown() viene eseguito quando viene premuto un tasto qualsiasi del mouse dentro un elemento**
mouseenter()	**Il metodo mouseenter() viene eseguito quando il puntatore del mouse entra all'interno di un elemento.**
mouseleave()	**Il metodo mouseleave() viene eseguito quando il puntatore del mouse esce da elemento.**
mousemove()	**Viene scatenato ogni volta che il puntatore del mouse si muove all'interno dell'elemento selezionato.**

mouseout()	**si verifica quando il puntatore del mouse lascia l'elemento selezionato.**
mouseover()	**si verifica quando il puntatore del mouse si trova sopra un elemento.**
mouseup()	**Il metodo mouseup() viene eseguito quando viene rilasciato un tasto qualsiasi del mouse dentro un elemento**
off()	**Viene utilizzato per rimuovere i gestori di eventi collegati con il metodo on()**
on()	**Il metodo on(), permette di assegnare uno o più eventi ad un oggetto del DOM in una volta sola method.**
one()	**Aggiunge uno o più gestori di eventi che però vengono lanciati una volta sola**
$.proxy	**Prende una funzione esistente e ne restituisce una nuova con un particolare contesto.**
ready()	**$(document).ready() consente di capire quando il documento è stato completamente caricato.**
resize()	**L' evento si verifica quando la finestra del browser cambia dimensione**
scroll()	**Viene attivato quando l'utente scorre un elemento specificato. Funziona sull'oggetto finestra e su tutti gli gli elementi scorrevoli**

select()	**L' evento select si verifica quando è selezionato un testo in una textarea o in una textbox**
submit()	**Submit() si verifica quando viene inviato un modulo.**
trigger()	**il metodo trigger() attiva un determinato evento per gli elementi selezionati.**
triggerHandler()	**il metodo triggerHandler() attiva un determinato evento per gli elementi selezionati, ma rispetto al metodo trigger() si sostituisce all'evento e non lo scatena.**
unbind()	**rimuove i gestori di eventi da elementi selezionati**
undelegate()	**Permette di rimuovere uno o più gestori di eventi**

Un altro dei più comuni utilizzi di jQuery è implementare *le chiamate asincrone* mediante la tecnologia Ajax.

Ajax(Asynchronous JavaScript and XML, è una tecnica usata per la realizzazione di applicazioni web interattive attraverso chiamate asincrone ossia chiamate ad una risorsa esterna che non interferisce con l'esecuzione della risorsa chiamante.

I risultati della risorsa esterna saranno utilizzabili solo quando disponibili senza "tempi morti" per l'utilizzatore poiché il caricamento della risorsa esterna avviene in background.

Attraverso Ajax è possibile cambiare dinamicamente il contenuto di una pagina o di una sua porzione senza effettuare il suo ricaricamento.

Le informazioni necessarie per il cambiamento vengono prelevate da una risorsa web statica (un file HTML o un XML, ad esempio) o dinamica (uno script PHP, JSP o altro), che viene contattata attraverso una chiamata HTTP lanciata tramite Javascript.

In base alle restrizioni del Same Origin Policy, per questioni di sicurezza, non è possibile effettuare chiamate Ajax a risorse presenti all'esterno del proprio dominio, quindi la risorsa contattata tramite Ajax deve essere una risorsa locale.

JQuery consente di effettuare una chiamata asincrona attraverso il metodo jQuery.ajax() metodo, che viene applicato direttamente all'oggetto jQuery, e permette di effettuare una chiamata Ajax e personalizzarla attraverso i molti parametri disponibili.

La chiamata del metodo può essere effettuata attraverso due differenti sintassi:

```
jQuery.ajax(...)// estesa
$.ajax(...);// abbreviata
```

I parametri principali di una chiamata ajax sono sicuramente l'URL della risorsa che effettua il lavoro e la funzione da eseguire al termine della chiamata.

I principali parametri ammessi dal metodo .ajax():

async (default: true)	Determina se la chiamata deve essere asincrona (true) o no (false).
complete	Consente di specificare la funzione da eseguire al termine della chiamata sia che essa abbia dato successo o errore; è eseguita dopo success o error.

Data	Contiene i dati che devono essere inviati alla risorsa che elabora la richiesta; il formato può essere sotto forma di oggetto (contenente delle coppie chiave/valore) oppure sotto forma di semplice stringa &key1=val1&key2=val2.
dataType	Tipo di dato che si riceve di ritorno, anche se jQuery tenta di capirlo da solo, è sempre meglio specificarlo. I tipi possibili sono: "xml", "html", "script", "json", "jsonp" e "text".
Error	Consente di specificare una funzione che è eseguita in caso di errore.
success	Consente di specificare una funzione che verrà eseguita al successo.
timeout	Durata massima (in millisecondi) della chiamata;
type (default: GET)	Tipo di richiesta da effettuare, principalmente POST o GET; ma anche altri metodi HTTP (PUT, DELETE, ...) ma non tutti i browser li supportano.
url (default:pagina corrente)	URL della risorsa alla quale viene inviata la richiesta; se omessa verrà contattata la pagina corrente.

Esempio di chiamata AJAX

```
function Submit() {
    var x = document.getElementById("id_input");
        var value=x.value;
        x.readOnly = true;
        x.style.cursor = "wait";
        var res = encodeURIComponent(value);
     $.ajax({
      type: "POST",
      url: "http://www.math.unipa.it/simplehealth/simple2/service/eng/",
      data: "textaraea=" + res+"&debug=1",
      dataType: "html",
      success: function(msg)
      {$(".div_input").hide();
            $("#div_output").show();
            $("#p_output").html(msg);
      },
      error: function()
      {alert("Chiamata fallita, si prega di riprovare...");}
    });
```

Esercitazione:

Nella scorsa lezione abbiamo realizzato uno script in php che visualizza i dati di una tabella Mysql.

Lo script prevedeva la chiamata allo stesso script per l'ordinamento dei dati secondo i tre campi della tabella.

Vediamo ore come realizzarlo in jQuery con una chiamata del metodo Ajax.

```
<!DOCTYPE html>
<head>
 <title>Esempio Jquery AJAX</title>
<script type ="text/javascript" src  =
"https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
 </script>
<script type ="text/javascript">
function Go(value)
{var Url= "http://147.163.15.90/xxx/GetDatiFromDB.php?ord="+value;
$.ajax({url:Url,
      success: function(msg)
      {$("#divPrincipale").html(msg)},
      error: function()
      { text="Chiamata fallita URL "+Url+" non risponde , si prega di
riprovare...";
              alert(text);}});}
$(document).ready(function() {
 Go(0);});
<?php
```

GetDatiFromDB.php

```php
$conn=mysqli_connect("localhost","studente","infopa","studente") or
die("Errore connessione DB ".mysqli_connect_error());
$ord=0;
if(isset($_GET['ord'])) $ord=$_GET['ord'];
$sql="SELECT * FROM `nome.cognome`";
switch($ord)
{case 0:$sql.=" ORDER BY ID";break;
 case 1:$sql.=" ORDER BY nome";break;
 case 2:$sql.=" ORDER BY cognome";break;
 case 3:$sql.=" ORDER BY datadinascita";break;
}
$ris=mysqli_query($conn,$sql) or die("Errore nella chiamta SQL
".$sql);
printf("<table>\n");
printf("<tr>
<td OnClick=\"Go(0)\">ID</td>
<td OnClick=\"Go(1)\">Nome</td>
<td OnClick=\"Go(2)\">Cognome</td>
<td OnClick=\"Go(3)\">DN</td></tr>");
```

```php
while($riga=mysqli_fetch_array($ris))
{printf("<tr><td><b>%s</b></td>",$riga['ID']);
 printf("<td><b>%s</b></td>",$riga['nome']);
 printf("<td><b>%s</b></td>",$riga[2]);
 printf("<td><b>%s</b></td><tr>\n",$riga[3]);
}
echo "</table>";
?>
```