**jQuery -** [**https://www.w3schools.com/jquery/jquery\_intro.asp**](https://www.w3schools.com/jquery/jquery_intro.asp)

***jQuery*** è una ***libreria*** JavaScript, che semplifica la programmazione JavaScript.

<https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_hide>\

Lo scopo di ***jQuery*** è di rendere molto più semplice l'uso di JavaScript sul tuo sito web.

***jQuery*** richiede molte attività comuni che richiedono molte linee di codice JavaScript e le racchiude in metodi che è possibile chiamare con una singola riga di codice, semplificando anche molte cose complicate da JavaScript, come le chiamate AJAX e la manipolazione DOM.

La libreria ***jQuery*** contiene le seguenti funzionalità:

* Manipolazione HTML / DOM
* Manipolazione CSS
* Metodi di eventi HTML
* Effetti e animazioni
* AJAX
* Utilità

**Sintassi jQuery -** [**https://www.w3schools.com/jquery/jquery\_syntax.asp**](https://www.w3schools.com/jquery/jquery_syntax.asp)

La sintassi jQuery è fatta su misura per selezionare elementi HTML ed eseguire un'azione sugli elementi.

La sintassi di base è: ***$ ( selettore ). azione ()*** :

* Un segno ***$*** per accedere a *jQuery*
* Un ( ***selettore*** ) per "interrogare (o trovare)" elementi HTML
* Un *jQuery* ***azione()*** per eseguire sull'elemento (s)

$(this).hide() - nasconde l'elemento corrente.

$("p").hide() - nasconde tutti gli elementi <p>.

$(".test").hide() - nasconde tutti gli elementi con class = "test".

$("#test").hide() - nasconde l'elemento con id = "test".

**L’evento ready():**

$(document).ready(function(){  
  *// jQuery methods go here...*  
});

Questo per impedire l'esecuzione di qualsiasi codice jQuery prima che il documento abbia finito di caricare.

È buona norma aspettare che il documento sia completamente caricato e pronto prima di utilizzarlo. Ciò consente inoltre di avere il codice JavaScript prima del corpo del documento, nella sezione head.

**Selettori -** [**https://www.w3schools.com/jquery/jquery\_selectors.asp**](https://www.w3schools.com/jquery/jquery_selectors.asp)

I selettori jQuery consentono di selezionare e manipolare gli elementi HTML. Sono utilizzati per "trovare" (o selezionare) elementi HTML in base al loro nome, id, classi, tipi, attributi, valori degli attributi e molto altro.

Tutti i selettori di jQuery iniziano con il simbolo del dollaro e parentesi: ***$ ().***

**Selettore di elementi:**

Il selettore di elementi jQuery seleziona gli elementi in base al nome dell'elemento: $("p")

Quando un utente fa clic su un pulsante, tutti gli elementi <p> saranno nascosti:

$(document).ready(function(){  
  $("button").click(function(){  
    $("p").hide();  
  });  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_hide_p>

**Selettore #id:**

Il selettore jQuery utilizza l'attributo id di un tag HTML per trovare l'elemento specifico: $("#test")

$(document).ready(function(){  
  $("button").click(function(){  
    $("#test").hide();  
  });  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_hide_id>

**Selettore di classi:**

Il selettore .class jQuery trova elementi con una classe specifica: $(".test")

Quando un utente fa clic su un pulsante, gli elementi con class = "test" saranno nascosti:

$(document).ready(function(){  
  $("button").click(function(){  
    $(".test").hide();  
  });  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_hide_class>



**Metodi evento -** [**https://www.w3schools.com/jquery/jquery\_events.asp**](https://www.w3schools.com/jquery/jquery_events.asp)

Un evento rappresenta il momento preciso in cui succede qualcosa.

**$(document).ready ():**

Ci consente di eseguire una funzione quando il documento è completamente caricato.

$(document).ready(function(){  
  *// jQuery methods go here...*  
});

**click():**

La funzione viene eseguita quando l'utente fa clic sull'elemento HTML.

$("p").click(function(){  
  $(this).hide();  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_click>

**dblClick ():**

La funzione viene eseguita quando l'utente fa doppio clic sull'elemento HTML:

$("p").dblclick(function(){  
  $(this).hide();  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dblclick>

**mouseenter():**

La funzione viene eseguita quando il puntatore del mouse entra nell'elemento HTML:

$("#p1").mouseenter(function(){  
  alert("You entered p1!");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_mouseenter>

**mouseleave():**

La funzione viene eseguita quando il puntatore del mouse lascia l'elemento HTML:

$("#p1").mouseleave(function(){  
  alert("Bye! You now leave p1!");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_mouseleave>

**mousedown():**

La funzione viene eseguita quando il pulsante sinistro, centrale o destro del mouse viene premuto, mentre il mouse si trova sopra l'elemento HTML:

$("#p1").mousedown(function(){  
  alert("Mouse down over p1!");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_mousedown>

**mouseup():**

La funzione viene eseguita quando viene rilasciato il pulsante sinistro, centrale o destro mentre il mouse si trova sopra l'elemento HTML:

$("#p1").mouseup(function(){  
  alert("Mouse up over p1!");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_mouseup>

**hover():**

La prima funzione viene eseguita quando il mouse entra nell'elemento HTML e la seconda funzione viene eseguita quando il mouse lascia l'elemento HTML:

$("#p1").hover(function(){  
  alert("You entered p1!");  
},  
function(){  
  alert("Bye! You now leave p1!");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_hover>

**focus():**

La funzione viene eseguita quando il campo modulo diventa attivo:

$("input").focus(function(){  
  $(this).css("background-color", "#cccccc");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_focus_blur>

**blur():**

La funzione viene eseguita quando il campo modulo perde lo stato attivo:

$("input").blur(function(){  
  $(this).css("background-color", "#ffffff");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_focus_blur>

**on():**

Il metodo ***on()*** allega uno o più gestori di eventi per gli elementi selezionati. Allegare un evento click a un elemento <p>:

$("p").on("click", function(){  
  $(this).hide();  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_on_click>

**Effetti: nascondi e mostra -** [**https://www.w3schools.com/jquery/jquery\_hide\_show.asp**](https://www.w3schools.com/jquery/jquery_hide_show.asp)

Con jQuery, puoi nascondere e mostrare elementi HTML con i metodi ***hide()*** e ***show():***

$("#hide").click(function(){  
  $("p").hide();  
});  
$("#show").click(function(){  
  $("p").show();  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_hide_show>

**Sintassi:**

$(*selector*).hide(*speed,callback*);  
$(*selector*).show(*speed,callback*);

Il parametro di velocità opzionale specifica la velocità del nascondimento / visualizzazione e può assumere i seguenti valori: "low", "speed" o millisecondo.

Il parametro ***callback*** facoltativo è una funzione che deve essere eseguita dopo il completamento del metodo ***hide()*** o ***show().***

$("button").click(function(){  
  $("p").hide(1000);  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_hide_slow>

**toggle ():**

Puoi anche alternare tra nascondere e mostrare un elemento:

$("button").click(function(){  
  $("p").toggle();  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_toggle>

**Sintassi:**

$(*selector*).toggle(*speed,callback*);

**Effetti jQuery – Dissolvenza -** [**https://www.w3schools.com/jquery/jquery\_fade.asp**](https://www.w3schools.com/jquery/jquery_fade.asp)

**Effetti jQuery – Scorrevole -** [**https://www.w3schools.com/jquery/jquery\_slide.asp**](https://www.w3schools.com/jquery/jquery_slide.asp)

**Effetti jQuery – Animazione -** [**https://www.w3schools.com/jquery/jquery\_animate.asp**](https://www.w3schools.com/jquery/jquery_animate.asp)

**jQuery Stop Animations -** [**https://www.w3schools.com/jquery/jquery\_stop.asp**](https://www.w3schools.com/jquery/jquery_stop.asp)

**Funzioni di callback -** [**https://www.w3schools.com/jquery/jquery\_callback.asp**](https://www.w3schools.com/jquery/jquery_callback.asp)

Le istruzioni JavaScript vengono eseguite riga per riga. Tuttavia, con gli effetti, la prossima riga di codice può essere eseguita anche se l'effetto non è terminato. Questo può creare errori.

Per evitare ciò, è possibile creare una funzione di ***callback***. Una funzione di ***callback*** viene eseguita dopo che l'effetto corrente è finito.

Sintassi tipica: ***$ ( selettore ) .hide ( velocità, callback );***

L'esempio ha un parametro ***callback*** che è una funzione che verrà eseguita dopo che ***hide*** è stato completato:

$("button").click(function(){  
  $("p").hide("slow", function(){  
    alert("The paragraph is now hidden");  
  });  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_hide_callback>

**jQuery – Chaining -** [**https://www.w3schools.com/jquery/jquery\_chaining.asp**](https://www.w3schools.com/jquery/jquery_chaining.asp)

Con jQuery, puoi concatenare azioni / metodi. Il concatenamento ci consente di eseguire più metodi jQuery (sullo stesso elemento) all'interno di una singola istruzione.

$("#p1").css("color", "red").slideUp(2000).slideDown(2000);

<https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_chaining>

**Get su contenuti ed attributi - <https://www.w3schools.com/jquery/jquery_dom_get.asp>**

Una parte molto importante di jQuery è la possibilità di manipolare il DOM. jQuery viene fornito con una serie di metodi correlati a DOM che semplificano l'accesso e la manipolazione di elementi e attributi.

Tre semplici, *ma utili*, metodi jQuery per la manipolazione DOM sono:

* ***text()*** - Imposta o restituisce il contenuto del testo degli elementi selezionati
* ***html()*** - Imposta o restituisce il contenuto degli elementi selezionati (incluso il markup HTML)
* ***val()*** - Imposta o restituisce il valore dei campi modulo

L'esempio seguente mostra come ottenere contenuti con metodi ***text()*** e ***html():***

$("#btn1").click(function(){  
  alert("Text: " + $("#test").text());  
});  
$("#btn2").click(function(){  
  alert("HTML: " + $("#test").html());  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_html_get>

Nell'esempio seguente viene illustrato come ottenere il valore di un campo di input con il metodo ***val()*** :

$("#btn1").click(function(){  
  alert("Value: " + $("#test").val());  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_val_get>

**Ottieni attributi - attr ():**

Il metodo ***attr()*** viene utilizzato per ottenere valori di attributo.

Nell'esempio seguente viene illustrato come ottenere il valore dell'attributo ***href*** in un collegamento:

$("button").click(function(){  
  alert($("#w3s").attr("href"));  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_attr_get>

**Set su contenuti ed attributi -** [**https://www.w3schools.com/jquery/jquery\_dom\_set.asp**](https://www.w3schools.com/jquery/jquery_dom_set.asp)

L'esempio seguente mostra come impostare il contenuto con i metodi ***text(), html()*** e ***val():***

$("#btn1").click(function(){  
  $("#test1").text("Hello world!");  
});  
$("#btn2").click(function(){  
  $("#test2").html("<b>Hello world!</b>");  
});  
$("#btn3").click(function(){  
  $("#test3").val("Dolly Duck");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_html_set>

**Imposta attributi - attr ():**

Il metodo ***attr()*** viene anche utilizzato per impostare / modificare i valori degli attributi.

L'esempio seguente mostra come modificare (*impostare*) il valore dell'attributo href in un collegamento:

$("button").click(function(){  
  $("#w3s").attr("href", "https://www.w3schools.com/jquery/");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_attr_set>

**Add elementi -** [**https://www.w3schools.com/jquery/jquery\_dom\_add.asp**](https://www.w3schools.com/jquery/jquery_dom_add.asp)

Ci sono 4 metodi per aggiungere nuovi contenuti:

**append():**

inserisce il contenuto ALLA FINE del contenuto degli elementi HTML selezionati ***(Interno al tag)***:

$("p").append("<p>Some appended text.</p>");

<https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_html_append>

**prepend():**

inserisce il contenuto ALL'INIZIO del contenuto degli elementi HTML selezionati ***(Interno al tag):***

$("p").prepend("<p>Some prepended text.</p>");

<https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_html_prepend>

function appendText() {  
  var txt1 = "<p>Text.</p>";               // Create element with HTML    
  var txt2 = $("<p></p>").text("Text.");   // Create with jQuery  
  var txt3 = document.createElement("p");  // Create with DOM  
  txt3.innerHTML = "Text.";  
  $("body").append(txt1, txt2, txt3);      // Append the new elements   
} <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_html_append2>

**after ():**

inserisce il contenuto DOPO gli elementi HTML selezionati:

$("img").after("Some text after");

**before ():**

inserisce il contenuto PRIMA degli elementi HTML selezionati:

$("img").before("Some text before");

<https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_html_after>

function afterText() {  
  var txt1 = "<b>I </b>";                    // Create element with HTML    
  var txt2 = $("<i></i>").text("love ");     // Create with jQuery  
  var txt3 = document.createElement("b");    // Create with DOM  
  txt3.innerHTML = "jQuery!";  
  $("img").after(txt1, txt2, txt3);          // Insert new elements after <img>  
} <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_html_after2>

**Remove elementi -** [**https://www.w3schools.com/jquery/jquery\_dom\_remove.asp**](https://www.w3schools.com/jquery/jquery_dom_remove.asp)

**remove():**

rimuove gli elementi selezionati e i relativi elementi figli:

$("#div1").remove(); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_remove>

Il metodo remove() accetta anche un parametro, che consente di filtrare gli elementi da rimuovere.

L'esempio seguente rimuove tutti gli <p>elementi con **class="test":**

$("p").remove(".test"); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_remove2>

**empty():**

rimuove gli elementi secondari degli elementi selezionati:

$("#div1").empty(); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_empty>

**Classi CSS -** [**https://www.w3schools.com/jquery/jquery\_css\_classes.asp**](https://www.w3schools.com/jquery/jquery_css_classes.asp)

**addClass ():**

Aggiunge una o più classi agli elementi selezionati:

$("button").click(function(){  
  $("h1, h2, p").addClass("blue");  
  $("div").addClass("important");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_addclass>

**removeClass ():**

Rimuove una o più classi dagli elementi selezionati:

$("button").click(function(){  
  $("h1, h2, p").removeClass("blue");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_removeclass>

**toggleClass ():**

Passa dall'aggiungere / rimuovere classi dagli elementi selezionati:

$("button").click(function(){  
  $("h1, h2, p").toggleClass("blue");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dom_toggleclass>

**css() -** [**https://www.w3schools.com/jquery/jquery\_css.asp**](https://www.w3schools.com/jquery/jquery_css.asp)

Imposta o restituisce una o più proprietà di stile per gli elementi selezionati.

Per ***restituire*** il valore di una proprietà CSS specificata, utilizzare la sintassi: css("*propertyname*");

$("p").css("background-color");

<https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_css_getcolor>

Per ***impostare*** una proprietà CSS specificata, utilizzare la sintassi: css("*propertyname*","*value*");

$("p").css("background-color", "yellow");

<https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_css_setcolor>

**Dimensioni -** [**https://www.w3schools.com/jquery/jquery\_dimensions.asp**](https://www.w3schools.com/jquery/jquery_dimensions.asp)

Il metodo ***width()*** imposta o restituisce la larghezza di un elemento (esclude padding, bordo e margine).

Il metodo ***height()*** imposta o restituisce l'altezza di un elemento (esclude padding, bordo e margine).

L'esempio seguente restituisce la larghezza e l'altezza di un elemento <div> specificato:

$("button").click(function(){  
  var txt = "";  
  txt += "Width: " + $("#div1").width() + "</br>";  
  txt += "Height: " + $("#div1").height();  
  $("#div1").html(txt);  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dim_width_height>

Il metodo ***innerWidth()*** restituisce la larghezza di un elemento (include il padding).

Il metodo ***innerHeight()*** restituisce l'altezza di un elemento (include il padding).

L'esempio seguente restituisce la larghezza / altezza interna di un <div> elemento specificato :

$("button").click(function(){  
  var txt = "";  
  txt += "Inner width: " + $("#div1").innerWidth() + "</br>";  
  txt += "Inner height: " + $("#div1").innerHeight();  
  $("#div1").html(txt);  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dim_innerwidth_height>

Il metodo ***outerWidth()*** restituisce la larghezza di un elemento (include padding e border).

Il metodo ***outerHeight()*** restituisce l'altezza di un elemento (include padding e border).

L'esempio seguente restituisce la larghezza / altezza esterna di un <div> elemento specificato :

$("button").click(function(){  
  var txt = "";  
  txt += "Outer width: " + $("#div1").outerWidth() + "</br>";  
  txt += "Outer height: " + $("#div1").outerHeight();  
  $("#div1").html(txt);  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dim_outerwidth_height>

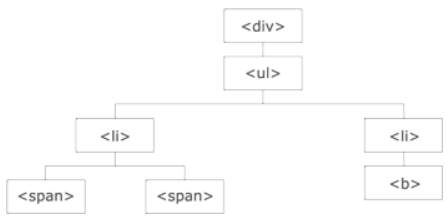
L'esempio seguente imposta la larghezza e l'altezza di un elemento <div> specificato :

$("button").click(function(){  
  $("#div1").width(500).height(500);  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_dim_width_height_set>

**Traversing -** [**https://www.w3schools.com/jquery/jquery\_traversing.asp**](https://www.w3schools.com/jquery/jquery_traversing.asp)

Significa “Muoversi attraverso”, ed è usato per *trovare o selezionare* elementi HTML in base alla loro relazione con altri elementi. Inizia con una selezione e passa attraverso, fino a raggiungere gli elementi che desideri.

Con ***jQuery traversing***, puoi facilmente muovere su (antenati), giù (discendenti) e lateralmente (fratelli) nell'albero, partendo dall'elemento (corrente) selezionato. Questo movimento è chiamato *attraversare o spostarsi* attraverso l'albero DOM. jQuery offre una varietà di metodi che ci permettono di attraversare il DOM.

La più grande categoria di metodi di attraversamento è il ***tree-traversal***.

**Ancestors -** [**https://www.w3schools.com/jquery/jquery\_traversing\_ancestors.asp**](https://www.w3schools.com/jquery/jquery_traversing_ancestors.asp)

**parent ():**

Restituisce l'elemento padre diretto dell'elemento selezionato. Questo metodo attraversa *solo un singolo* livello.

L'esempio seguente restituisce l'elemento padre diretto di ciascun elemento ***<span>***:

$(document).ready(function(){  
  $("span").parent();  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_parent>

**parents ():**

Restituisce tutti gli elementi degli antenati dell'elemento selezionato, fino all'elemento radice (<html>).

L'esempio seguente restituisce tutti gli antenati di tutti gli elementi ***<span>:***

$(document).ready(function(){  
  $("span").parents();  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_parents>

Puoi anche utilizzare un parametro facoltativo per filtrare la ricerca degli antenati.

L'esempio seguente restituisce tutti gli antenati di tutti gli elementi ***<span>*** che sono elementi ***<ul>:***

$(document).ready(function(){  
  $("span").parents("ul");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_parents2>

**parentsUntil ():**

Restituisce tutti gli elementi antenati tra due argomenti dati.

L'esempio seguente restituisce tutti gli elementi antenati tra a ***<span>*** e un elemento ***<div>*:**

$(document).ready(function(){  
  $("span").parentsUntil("div");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_parentsuntil>

**Descendants -** [**https://www.w3schools.com/jquery/jquery\_traversing\_descendants.asp**](https://www.w3schools.com/jquery/jquery_traversing_descendants.asp)

**children ():**

Restituisce tutti i figli diretti dell'elemento selezionato. Questo metodo attraversa solo un singolo livello lungo l'albero DOM.

L'esempio seguente restituisce tutti gli elementi che sono figli diretti di ciascun elemento ***<div>:***

$(document).ready(function(){  
  $("div").children();  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_children>

Puoi anche utilizzare un parametro facoltativo per filtrare la ricerca di bambini.

L'esempio seguente restituisce tutti gli <p>elementi con il nome della classe "***first***", che sono figli diretti di ***<div>:***

$(document).ready(function(){  
  $("div").children("p.first");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_children2>

**find ():**

Restituisce elementi discendenti dell'elemento selezionato, fino all'ultimo discendente.

L'esempio seguente restituisce tutti gli elementi ***<span>*** che sono discendenti di ***<div>:***

$(document).ready(function(){  
  $("div").find("span");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_find>

L'esempio seguente restituisce tutti i discendenti di ***<div>:***

$(document).ready(function(){  
  $("div").find("\*");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_find2>

**Siblings (fratelli) -** [**https://www.w3schools.com/jquery/jquery\_traversing\_siblings.asp**](https://www.w3schools.com/jquery/jquery_traversing_siblings.asp)

**siblings():**

Restituisce tutti gli elementi di pari livello dell'elemento selezionato.

L'esempio seguente restituisce tutti gli elementi di pari livello di ***<h2>:***

$(document).ready(function(){  
  $("h2").siblings();  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_siblings>

Puoi anche usare un parametro opzionale per filtrare la ricerca dei fratelli.

Nell'esempio seguente vengono restituiti tutti gli elementi di pari livello di ***<h2>*** che sono elementi ***<p>:***

$(document).ready(function(){  
  $("h2").siblings("p");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_siblings2>

**next ():**

Restituisce il prossimo elemento fratello dell'elemento selezionato.

L'esempio seguente restituisce il fratello successivo di <h2>:

$(document).ready(function(){  
  $("h2").next();  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_next>

**nextAll():**

Restituisce tutti i successivi elementi di pari livello dell'elemento selezionato.

L'esempio seguente restituisce tutti i successivi elementi di pari livello di ***<h2>:***

$(document).ready(function(){  
  $("h2").nextAll();  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_nextall>

**nextUntil ():**

Restituisce tutti i successivi elementi di pari livello tra due argomenti dati.

L'esempio seguente restituisce tutti gli elementi di pari livello tra a ***<h2>*** e un elemento ***<h6>:***

$(document).ready(function(){  
  $("h2").nextUntil("h6");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_nextuntil>

I metodi ***prev(), prevAll()*** e ***prevUntil()*** funzionano come i metodi di cui sopra, ma con funzionalità inverso: tornano elementi di pari livello precedenti (attraversano indietro lungo elementi di pari livello nell'albero DOM, invece che in avanti).

**Filtering -** [**https://www.w3schools.com/jquery/jquery\_traversing\_filtering.asp**](https://www.w3schools.com/jquery/jquery_traversing_filtering.asp)

**first ():**

Restituisce il primo elemento degli elementi specificati.

L'esempio seguente seleziona il primo elemento ***<div>:***

$(document).ready(function(){  
  $("div").first();  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_first>

**last ():**

Restituisce l'ultimo elemento degli elementi specificati.

L'esempio seguente seleziona l'ultimo elemento ***<div>:***

$(document).ready(function(){  
  $("div").last();  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_last>

**eq ():**

Restituisce un elemento con un numero indice specifico degli elementi selezionati.

I numeri indice iniziano da 0, quindi il primo elemento avrà il numero indice 0 e non 1. L'esempio seguente seleziona il secondo elemento ***<p>*** (numero indice 1):

$(document).ready(function(){  
  $("p").eq(1);  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_eq>

**filter ():**

Consente di specificare un criterio. Gli elementi che non corrispondono ai criteri vengono rimossi dalla selezione e quelli che corrispondono saranno restituiti.

L'esempio seguente restituisce tutti gli elementi ***<p>*** con nome di classe "***intro***":

$(document).ready(function(){  
  $("p").filter(".intro");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_filter>

**not ():**

Restituisce tutti gli elementi che non corrispondono ai criteri.

***Suggerimento***: il metodo ***not()*** è l'opposto di ***filter()***.

L'esempio seguente restituisce tutti gli elementi ***<p>*** che non hanno il nome di classe "***intro***":

$(document).ready(function(){  
  $("p").not(".intro");  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_not>

**jQuery - Introduzione AJAX -** [**https://www.w3schools.com/jquery/jquery\_ajax\_intro.asp**](https://www.w3schools.com/jquery/jquery_ajax_intro.asp)

***AJAX*** è l'arte di scambiare dati con un server e di aggiornare parti di una pagina web senza ricaricare l'intera pagina.

***AJAX = JavaScript asincrono e XML.*** AJAX riguarda il caricamento dei dati in background e la loro visualizzazione.

Con i metodi ***jQuery AJAX***, puoi richiedere ***testo***, ***HTML***, ***XML*** o ***JSON*** da un server remoto utilizzando ***HTTP*** ***Get*** e ***HTTP*** ***Post***. E puoi caricare i dati esterni direttamente negli elementi HTML selezionati della tua pagina web.

**jQuery - Metodo AJAX load() -** [**https://www.w3schools.com/jquery/jquery\_ajax\_load.asp**](https://www.w3schools.com/jquery/jquery_ajax_load.asp)

Il metodo ***load()*** carica i dati da un server e inserisce i dati restituiti nell'elemento selezionato.

*Sintassi:* ***$(selector).load(URL,data,callback);***

* Il parametro ***URL*** richiesto specifica l'URL che desideri caricare.
* Il parametro *facoltativo* ***data*** specifica un insieme di coppie chiave / valore ***querystring*** da inviare insieme alla richiesta.
* Il parametro *facoltativo* ***callback*** è il nome di una funzione da eseguire dopo il completamento del metodo ***load()***.

***Contenuto del file .txt :***

<h2>jQuery and AJAX is FUN!!!</h2>  
<p id="p1">This is some text in a paragraph.</p>

L'esempio seguente carica il contenuto del file "***demo\_test.txt***" in un elemento ***<div>*** specifico :

$("#div1").load("demo\_test.txt"); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_ajax_load>

È anche possibile aggiungere un ***selettore jQuery*** al parametro ***URL***.

L'esempio seguente carica il contenuto dell'elemento con ***id = "p1"***, all'interno del file ***"demo\_test.txt"***, in un elemento ***<div>*** specifico :

$("#div1").load("demo\_test.txt #p1"); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_ajax_load2>

La funzione di ***callback*** può avere parametri diversi:

* ***responseTxt*** - contiene il ***contenuto*** ***risultante*** se la chiamata ha esito positivo
* ***statusTxt*** - contiene lo ***stato*** della chiamata
* ***xhr*** - contiene ***l'oggetto*** ***XMLHttpRequest***

L'esempio seguente mostra una finestra di avviso dopo il completamento del metodo load (), se ha avuto successo, visualizza "Contenuto esterno caricato con successo!", E se fallisce visualizza un messaggio di errore:

$("button").click(function(){  
  $("#div1").load("demo\_test.txt", function(responseTxt, statusTxt, xhr){  
    if(statusTxt == "success")  
      alert("External content loaded successfully!");  
    if(statusTxt == "error")  
      alert("Error: " + xhr.status + ": " + xhr.statusText);  
  });  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_ajax_load_callback>

**jQuery metodi AJAX get() post() -** [**https://www.w3schools.com/jquery/jquery\_ajax\_get\_post.asp**](https://www.w3schools.com/jquery/jquery_ajax_get_post.asp)

I metodi ***jQuery*** ***get()*** e ***post()*** vengono utilizzati per richiedere dati dal server con una richiesta ***HTTP GET*** o ***POST***.

***GET*** è fondamentalmente utilizzato per ottenere (recuperare) alcuni dati dal server. ***Nota***: il metodo ***GET*** può restituire dati memorizzati nella cache.

Il ***POST*** può anche essere usato per ottenere alcuni dati dal server. Tuttavia, il metodo ***POST*** non memorizza ***MAI*** dati e viene spesso utilizzato per inviare dati insieme alla richiesta.

**Metodo jQuery $.get():**

*Sintassi*: ***$.get(URL,callback);***

* Il parametro ***URL*** richiesto specifica l'URL che desideri richiedere.
* Il parametro *facoltativo* ***callback*** è il nome di una funzione da eseguire se la richiesta ha esito positivo.

Nell'esempio seguente viene utilizzato il metodo ***$.get()*** per recuperare i dati da un file sul server:

$("button").click(function(){  
  $.get("demo\_test.asp", function(data, status){  
    alert("Data: " + data + "\nStatus: " + status);  
  });  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_ajax_get>

Il primo parametro di ***callback*** contiene il contenuto della pagina richiesta e il secondo parametro di ***callback*** mantiene lo stato della richiesta.

***demo\_test.asp:***

<%  
response.write("This is some text from an external ASP file.")  
%>

**Metodo jQuery $.post():**

*Sintassi:* ***$.post(URL,data,callback);***

* Il parametro ***URL*** richiesto specifica l'URL che desideri richiedere.
* Il parametro *facoltativo* ***dati*** specifica alcuni dati da inviare insieme alla richiesta.
* Il parametro *facoltativo* ***callback*** è il nome di una funzione da eseguire se la richiesta ha esito positivo.

Nell'esempio seguente viene utilizzato il metodo ***$.post()*** per inviare alcuni dati insieme alla richiesta:

$("button").click(function(){  
  $.post("demo\_test\_post.asp",  
  {  
    name: "Donald Duck",  
    city: "Duckburg"  
  },  
  function(data, status){  
    alert("Data: " + data + "\nStatus: " + status);  
  });  
}); <https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_ajax_post>

Lo script ASP in "***demo\_test\_post.asp***" legge i parametri, li elabora e restituisce un risultato.

Il primo parametro di ***callback*** contiene il contenuto della pagina richiesta e il secondo parametro di ***callback*** mantiene lo stato della richiesta.

***demo\_test\_post.asp:***

<%  
dim fname,city  
fname=Request.Form("name")  
city=Request.Form("city")  
Response.Write("Dear " & fname & ". ")  
Response.Write("Hope you live well in " & city & ".")  
%>

**jQuery – Filters per ricerche -** [**https://www.w3schools.com/jquery/jquery\_filters.asp**](https://www.w3schools.com/jquery/jquery_filters.asp)