Corso jQuery

JavaScript Query per tutti

Scopo del corso

jQuery è uno strumento largamente diffuso nello sviluppo web e anche se i moderni sviluppi tendono a non usare questa libreria, è importante conoscerne il funzionamento e le potenzialità.

All'interno di questo corso proverò a guidarvi nell'uso di queste tecnologia largamente diffusa ed utilizzata, facendo dei paralleli fra il suo utilizzo e l'utilizzo di vanilla JavaScript.

jQuery - di cosa si tratta?

jQuery è una libreria JavaScript popolare che semplifica la scrittura di codice JavaScript. Include una varietà di funzionalità che consentono di eseguire operazioni come modificare elementi HTML, gestire eventi, creare animazioni, effettuare richieste AJAX e molto altro.

Storia di jQuery

jQuery è stato originariamente sviluppato da John Resig nel 2006. Resig ha creato jQuery per aiutare a semplificare la scrittura di JavaScript. Nel corso degli anni, jQuery è diventato uno dei framework JavaScript più popolari al mondo.

Nel 2016, la fondazione jQuery ha rilasciato la versione 3.0 di jQuery, che ha introdotto una serie di miglioramenti, come una migliore performance, una maggiore velocità di esecuzione e una migliore compatibilità con i browser moderni.

jQuery continua ad essere uno dei framework JavaScript più popolari al mondo, con milioni di sviluppatori che lo usano per creare siti Web e applicazioni.

Come iniziare con jQuery

Per iniziare a usare jQuery, è necessario includere il file jQuery nella pagina HTML. È possibile scaricare il file jQuery dal sito ufficiale di jQuery o utilizzare un CDN (Content Delivery Network) per includere il file jQuery nella pagina HTML.

Esempio di utilizzo di jQuery

In questo esempio jQuery viene presa dalla CDN del sito di jQuery

```
<!DOCTYPE html>
<html lang="it">
  <head>
    <title>jQuery Example</title>
    <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  </head>
  <body>
    <h1>Hello World!</h1>
    <script>
      $(document).ready(function () {
        $("h1").css("color", "red");
      });
    </script>
  </body>
</html>
```

Concetto di ready

Perché si tende a mettere il codice jQuery all'interno di una funzione ready?

Perché il codice JavaScript viene eseguito quando viene caricata la pagina HTML. Se si tenta di selezionare un elemento HTML che non è ancora stato caricato, il codice non funzionerà.

Per risolvere questo problema, è possibile utilizzare la funzione ready() di jQuery. La funzione ready() viene eseguita quando viene caricata la pagina HTML.

Esiste anche una sintassi compatta che describe la funzione ready() di jQuery:

```
$(function () {
  // codice jQuery
});
```

ready in vanilla JavasScript

In vanilla JavaScript è possibile utilizzare la funzione addEventListener() per eseguire il codice JavaScript quando viene caricata la pagina HTML.

```
document.addEventListener("DOMContentLoaded", function () {
  // codice JavaScript
});
```

Questa sintassi è più lunga rispetto alla sintassi di jQuery, pur avendo lo stesso scopo.

Sintassi

La sintassi base di jQuery è:

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

dove **$** identifica l'utilizzo di jQuery, **selector** è il selettore CSS per selezionare gli elementi HTML e **action()** esegue un'azione sugli elementi HTML selezionati.

Selector

I selettori jQuery consentono di selezionare e manipolare gli elementi HTML.
Ecco alcuni esempi di selettori jQuery:

$(this) - seleziona l'elemento corrente

$("p") - seleziona tutti gli elementi <p>

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

$("#test") - seleziona l'elemento con id="test"

Action

Le azioni jQuery sono eseguite sugli elementi HTML selezionati.

Ecco alcuni esempi di azioni jQuery:

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

$("p").show() - mostra gli elementi <p>

$("p").toggle() - alterna tra nascondi e mostra gli elementi <p>

jQuery fadeIn() e fadeOut()

fadeIn() e fadeOut() sono due metodi di jQuery che consentono di mostrare e nascondere gli elementi HTML con un effetto di dissolvenza.

fadeIn() mostra un elemento HTML con un effetto di dissolvenza.
fadeOut() nasconde un elemento HTML con un effetto di dissolvenza.

Sintassi

```
$(selector).fadeIn(speed,callback);
$(selector).fadeOut(speed,callback);
```

Esempio

```
$("#div1").fadeIn();
$("#div2").fadeIn("slow");
$("#div3").fadeIn(10000);
```

jQuery slideDown() e slideUp()

slideDown() e slideUp() sono due metodi di jQuery che consentono di mostrare e nascondere gli elementi HTML con un effetto di scorrimento.

slideDown() mostra un elemento HTML con un effetto di scorrimento.
slideUp() nasconde un elemento HTML con un effetto di scorrimento.

Sintassi

```
$(selector).slideDown(speed,callback);
$(selector).slideUp(speed,callback);
```

Esempio

```
$("#div1").slideDown();
$("#div2").slideDown("slow");
$("#div3").slideDown(10000);
```

jQuery animate()

Il metodo animate() di jQuery consente di creare animazioni personalizzate sugli elementi HTML.

Sintassi:

```
$(selector).animate({params},speed,callback);
```

Esempio:

```
$("#div1").animate({
    opacity: '0.5',
    height: '150px',
    width: '150px'
}   , 3000);
```

jQuery stop()

Il metodo stop() di jQuery interrompe le animazioni o le code in esecuzione per gli elementi selezionati.

Sintassi

```
$(selector).stop(stopAll,goToEnd);
```

Esempio

```
$("#div1").stop();
```

jQuery Callback

Una callback è una funzione che viene passata come argomento ad un'altra funzione e viene eseguita dopo che la funzione è stata completata.

jQuery callback sono state progettate per essere utilizzate con le animazioni, ma possono essere utilizzate anche con altri metodi.

Sintassi

```
$(selector).hide(speed,callback);
```

Esempio

```
$("button").click(function(){
  $("p").hide("slow", function(){
    alert("Il paragrafo è stato nascosto");
  });
});
```

jQuery Chaining

La concatenazione è una tecnica per collegare più metodi di jQuery insieme.

La concatenazione consente di eseguire più azioni sugli stessi elementi HTML selezionati con una sola riga di codice.

Sintassi

```
$(selector).hide().show().slideUp().slideDown();
```

Esempio

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

jQuery text / html / val - get/set

Il metodo text() imposta o restituisce il testo (contenuto) di elementi selezionati.
Il metodo html() imposta o restituisce il contenuto (HTML) degli elementi selezionati.
Il metodo val() imposta o restituisce il valore dell'attributo value degli elementi selezionati.

Sintassi

```
$(selector).text();
$(selector).text(text);
$(selector).html();
$(selector).html(htmlString);
$(selector).val();
$(selector).val(value);
```

jQuery text / html / val - utilizzo

Esempio di GET

```
$("#test").text();
$("#test2").html();
$("#test3").val();
```

Esempio di SET

```
$("#test").text("Nuovo testo");
$("#test2").html("Nuovo <b>testo</b>");
$("#test3").val("Nuovo testo");
```

jQuery - append() / prepend()

Il metodo append() inserisce il contenuto all'interno dell'elemento selezionato, dopo il contenuto esistente.
Il metodo prepend() inserisce il contenuto all'interno dell'elemento selezionato, prima del contenuto esistente.

Sintassi

```
$(selector).append(content);
$(selector).prepend(content);
```

Esempio

```
$("p").append("-testo dopo-");
$("p").prepend("-testo prima-");
```

jQuery - after() / before()

Il metodo after() inserisce il contenuto dopo l'elemento selezionati.

Il metodo before() inserisce il contenuto prima dell'elemento selezionati.

Sintassi

```
$(selector).after(content);
$(selector).before(content);
```

Esempio

```
$("p").after("Testo dopo");
$("p").before("Testo prima");
```

jQuery - remove() / empty()

Il metodo remove() rimuove gli elementi selezionati e i loro figli.

Il metodo empty() rimuove il contenuto degli elementi selezionati e i loro figli.

Sintassi

```
$(selector).remove([selector]);
$(selector).empty();
```

Esempio

```
$("p").remove(".test");
$("p").empty();
```

jQuery - addClass() / removeClass() / toggleClass()

Il metodo addClass() aggiunge una o più classi agli elementi selezionati.

Il metodo removeClass() rimuove una o più classi dagli elementi selezionati.

Il metodo toggleClass() alterna tra aggiungere / rimuovere le classi dagli elementi selezionati.

Sintassi

```
$(selector).addClass(className);
$(selector).removeClass(className);
$(selector).toggleClass(className);
```

Esempio

```
$("p").addClass("test");
$("p").removeClass("test");
$("p").toggleClass("test");
```

jQuery - css()

Il metodo css() imposta o restituisce una o più proprietà di stile per gli elementi selezionati.

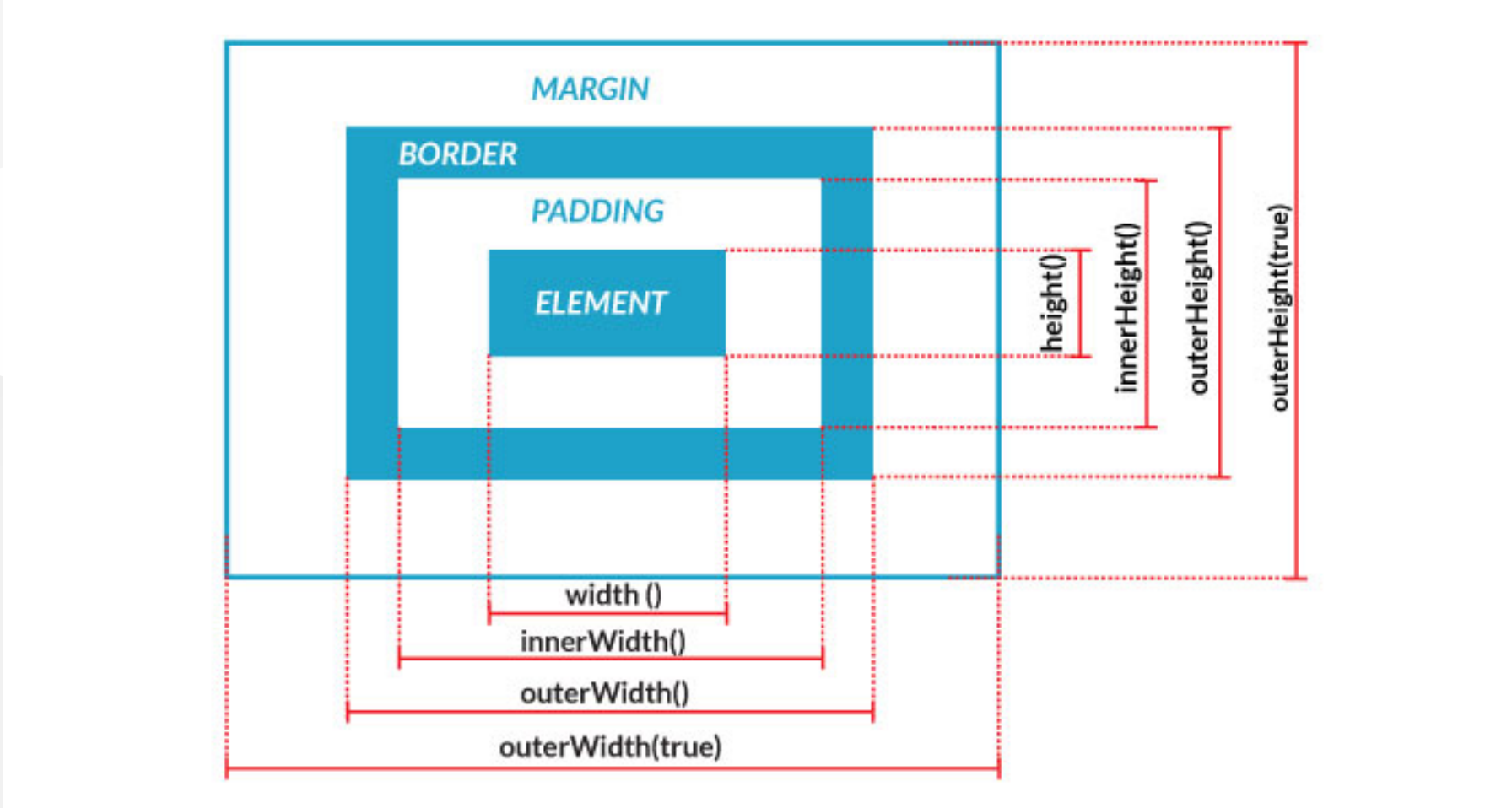
Sintassi

```
$(selector).css(propertyName);
$(selector).css(propertyName, value);
$(selector).css({propertyName:value, propertyName:value, ...});
```

Esempio

```
$("p").css("background-color");
$("p").css("background-color", "yellow");
$("p").css({"background-color": "red", "font-size": "200%"});
```

jQuery - dimensioni - riassunto



https://www.tutorialspoint.com/jquery/jquery-dimensions.htm

jQuery - dimensioni - width() / height()

Il metodo width() imposta o restituisce la larghezza degli elementi selezionati.

Il metodo height() imposta o restituisce l'altezza degli elementi selezionati.

Sintassi

```
$(selector).width();
$(selector).width(value);
$(selector).height();
$(selector).height(value);
```

Esempio

```
$("p").width();
$("p").width("500px");
$("p").height();
$("p").height("500px");
```

jQuery - dimensioni - innerWidth() / innerHeight()

Il metodo innerWidth() restituisce la larghezza di un elemento, inclusi il padding.

Il metodo innerHeight() restituisce l'altezza di un elemento, inclusi il padding.

Sintassi

```
$(selector).innerWidth();
$(selector).innerHeight();
```

Esempio

```
$("p").innerWidth();
$("p").innerHeight();
```

jQuery - dimensioni - outerWidth() / outerHeight()

Il metodo outerWidth() restituisce la larghezza di un elemento, inclusi il padding e il bordo.
Il metodo outerHeight() restituisce l'altezza di un elemento, inclusi il padding e il bordo.
In entrambi i casi, passando true come argomento, è possibile includere anche il margine.

Sintassi

```
$(selector).outerWidth([true]);
$(selector).outerHeight([true]);
```

Esempio

```
$("p").outerWidth(); $("p").outerHeight();
$("p").outerWidth(true); $("p").outerHeight(true);
```

jQuery - traversing

Il traversing è il processo di scorrere gli elementi HTML per trovare (o selezionare) gli elementi HTML specificati.

jQuery dispone di una serie di metodi per poter effettuare correttamente il traversing.

Lo scopo è quello di navigare all'interno del DOM per trovare gli elementi che si desidera manipolare.

Gli elementi vengono catalogati in genitori (ancestors) e figli (descendants) e fratelli (siblings).

jQuery - traversing - ancestors

Gli elementi HTML possono essere visti come un albero gerarchico.
L'elemento superiore è l'elemento radice, e gli elementi che sono al di sotto di esso sono i suoi figli.
Gli elementi che sono al di sotto dei figli di un elemento sono i suoi nipoti.

Sintassi

```
$(selector).parent();
$(selector).parents();
$(selector).parentsUntil();
```

Esempio

```
$("span").parent();
$("span").parents();
$("span").parentsUntil("div");
```

jQuery - traversing - descendants

Gli elementi HTML possono essere visti come un albero gerarchico.
L'elemento superiore è l'elemento radice, e gli elementi che sono al di sotto di esso sono i suoi figli.
Gli elementi che sono al di sotto dei figli di un elemento sono i suoi nipoti.

Sintassi

```
$(selector).children();
$(selector).find();
```

Esempio

```
$("div").children();
$("div").find("span");
```

jQuery - traversing - siblings

Gli elementi HTML possono essere visti come un albero gerarchico.
L'elemento superiore è l'elemento radice, e gli elementi che sono al di sotto di esso sono i suoi figli.
Gli elementi che sono al di sotto dei figli di un elemento sono i suoi nipoti.

Sintassi

```
$(selector).siblings();
$(selector).next();
$(selector).nextAll();
$(selector).nextUntil();
$(selector).prev();
$(selector).prevAll();
$(selector).prevUntil();
```

jQuery - traversing - siblings - esempio

Esempio

```
$("div").siblings();
$("div").next();
$("div").nextAll();
```

jQuery - traversing - filter

Il metodo filter() consente di specificare un criterio.

Gli elementi che soddisfano il criterio vengono restituiti.

Sintassi

```
$(selector).filter(filter);
```

Esempio

```
$("div").filter(".test");
```

Fonti usate per la creazione di queste slide

https://it.wikipedia.org : definizioni e argomenti

https://www.w3schools.com/Jquery/ : jQuery tutorial

https://chat.openai.com : ChatGPT

Ogni immagine inserita riporta la fonte

Disclaimer

L'autore ha generato questo testo in parte con GPT-3, il modello di generazione del linguaggio su larga scala di OpenAI. Dopo aver generato la bozza della lingua, l'autore ha rivisto, modificato e rivisto la lingua a proprio piacimento e si assume la responsabilità ultima del contenuto di questa pubblicazione.