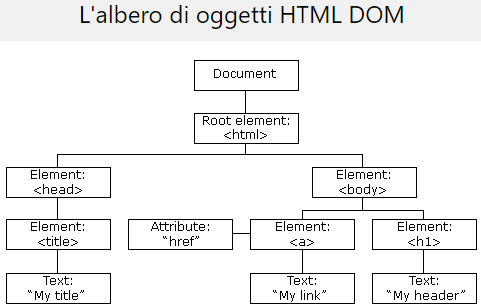
**JavaScript HTML DOM -** [**https://www.w3schools.com/js/js\_htmldom.asp**](https://www.w3schools.com/js/js_htmldom.asp)

Con il DOM HTML, JavaScript può accedere e modificare tutti gli elementi di un documento HTML.

Quando una pagina Web viene caricata, il browser crea un **D**ocument **O**bject **M**odel della pagina.

Il modello **HTML DOM** è costruito come un albero di **oggetti** :



* JavaScript può cambiare tutti gli elementi HTML nella pagina
* JavaScript può cambiare tutti gli attributi HTML nella pagina
* JavaScript può cambiare tutti gli stili CSS nella pagina
* JavaScript può rimuovere elementi e attributi HTML esistenti
* JavaScript può aggiungere nuovi elementi e attributi HTML
* JavaScript può reagire a tutti gli eventi HTML esistenti nella pagina
* JavaScript può creare nuovi eventi HTML nella pagina

HTML DOM è un modello di ***oggetto*** standard e ***un'interfaccia di programmazione*** per HTML. Definisce:

* Gli elementi HTML come ***oggetti***
* Le ***proprietà*** di tutti gli elementi HTML
* I ***metodi*** per accedere a tutti gli elementi HTML
* Gli ***eventi*** per tutti gli elementi HTML

***HTML DOM è uno standard su come ottenere, modificare, aggiungere o eliminare elementi HTML.***

***Metodi DOM HTML -*** [***https://www.w3schools.com/js/js\_htmldom\_methods.asp***](https://www.w3schools.com/js/js_htmldom_methods.asp)

I metodi DOM HTML sono **azioni** che puoi eseguire su elementi HTML. Le proprietà DOM HTML sono **valori** che è possibile impostare o modificare. È possibile accedere al ***DOM HTML*** con ***JavaScript*** (e con altri linguaggi).

Nel DOM, tutti gli elementi HTML sono definiti come **oggetti**. ***L'interfaccia di programmazione*** è proprietà e metodi di ciascun oggetto.

Una **proprietà** è un ***valore*** che puoi ottenere o impostare (ad esempio modificare il contenuto di un elemento).

Un **metodo** è ***un'azione*** che puoi fare (come aggiungere o eliminare un elemento HTML).

L'esempio seguente modifica il contenuto (***innerHTML***) <p>dell'elemento con ***id="demo":***

<p id="demo"></p>  
<script>  
document.getElementById("demo").innerHTML = "Hello World!";  
</script> <https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_method>

**Il metodo getElementById:**

Il modo più comune per accedere a un elemento HTML è utilizzare l'elemento ***id***.

Nell'esempio sopra il metodo ***getElementById*** utilizzato ***id="demo"*** per trovare l'elemento.

**La proprietà innerHTML**

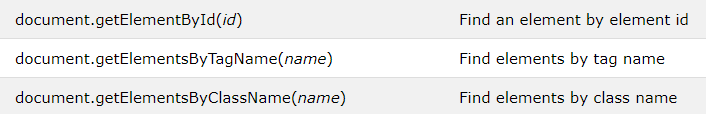
Il modo più semplice per ottenere il contenuto di un elemento è usando la proprietà ***innerHTML***, è utile per ottenere o sostituire il contenuto di elementi HTML, può essere utilizzata per ottenere o modificare qualsiasi elemento HTML, inclusi <html> e <body>.

***HTML DOM Document -*** [***https://www.w3schools.com/js/js\_htmldom\_document.asp***](https://www.w3schools.com/js/js_htmldom_document.asp)

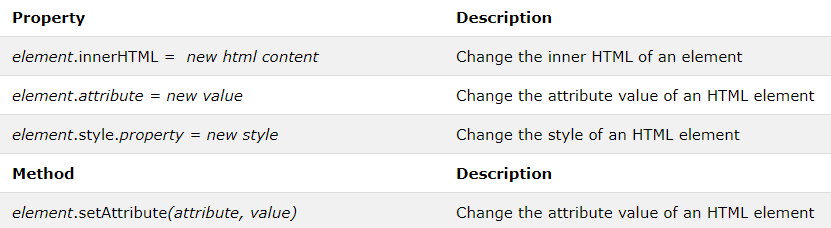
L'oggetto ***document*** è il proprietario di tutti gli altri oggetti nella tua pagina web.

Se si desidera accedere a qualsiasi elemento in una pagina HTML, si inizia con l'accesso all'oggetto ***document***.

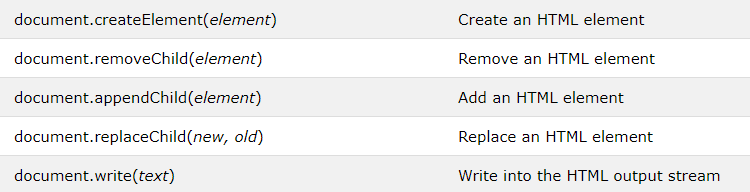
**Trovare elementi HTML:**



**Modifica degli elementi HTML:**



**Aggiunta ed eliminazione di elementi:**



**Aggiunta di gestori di eventi:**

Aggiunta del codice del gestore eventi a un evento onclick:

document.getElementById(id).onclick = function(){code}

**Elementi DOM HTML -** [**https://www.w3schools.com/js/js\_htmldom\_elements.asp**](https://www.w3schools.com/js/js_htmldom_elements.asp)

Spesso, con JavaScript, vuoi manipolare elementi HTML. Per fare ciò, devi prima trovare gli elementi. Ci sono diversi modi per farlo:

* Trovare elementi HTML per ***id***

var myElement = document.getElementById("intro");

<https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_getelementbyid>

* Ricerca di elementi HTML in base al ***nome del tag***

var x = document.getElementById("main");  
var y = x.getElementsByTagName("p");

<https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_getelementsbytagname2>

* Ricerca di elementi HTML in base al ***nome della classe***

var x = document.getElementsByClassName("intro");

<https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_getelementsbyclassname>

* Trovare elementi HTML dai ***selettori CSS***

var x = document.querySelectorAll("p.intro");

<https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_queryselectorall>

**Modifica HTML -** [**https://www.w3schools.com/js/js\_htmldom\_html.asp**](https://www.w3schools.com/js/js_htmldom_html.asp)

JavaScript può creare contenuti HTML dinamici. In JavaScript, ***document.write()*** può essere utilizzato per scrivere direttamente nel ***flusso di output*** HTML:

<script>  
document.write(Date());  
</script> <https://www.w3schools.com/js/tryit.asp?filename=tryjs_date>

Il modo più semplice per modificare il ***contenuto di un elemento*** HTML consiste nell'utilizzare la proprietà ***innerHTML***.

<p id="p1">Hello World!</p>  
<script>  
document.getElementById("p1").innerHTML = "New text!";  
</script> <https://www.w3schools.com/js/tryit.asp?filename=tryjs_change_innerhtml>

Per cambiare il ***valore di un attributo*** HTML, usa questa sintassi:

document.getElementById(*id*).*attribute = new value*

<img id="myImage" src="smiley.gif">  
<script>  
document.getElementById("myImage").src = "landscape.jpg";  
</script> <https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_image>

**Modifica CSS -** [**https://www.w3schools.com/js/js\_htmldom\_css.asp**](https://www.w3schools.com/js/js_htmldom_css.asp)

Il DOM HTML consente a JavaScript di modificare lo ***stile degli elementi*** HTML.

document.getElementById(*id*).style.*property*=*new style*

<p id="p2">Hello World!</p>  
<script>  
document.getElementById("p2").style.color = "blue";  
</script> <https://www.w3schools.com/js/tryit.asp?filename=tryjs_change_style>  
<p>The paragraph above was changed by a script.</p>

HTML DOM ti consente di eseguire codice quando si verifica un ***evento***.

Gli eventi sono generati dal browser quando "le cose accadono" agli elementi HTML:

* + Si fa clic su un elemento
  + La pagina è stata caricata
  + I campi di input sono cambiati

<h1 id="id1">My Heading 1</h1>  
<button type="button"   
onclick="document.getElementById('id1').style.color = 'red'">  
Click Me!</button> <https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_color2>

**Eventi HTML DOM -** [**https://www.w3schools.com/js/js\_htmldom\_events.asp**](https://www.w3schools.com/js/js_htmldom_events.asp)

In questo esempio, il contenuto dell'elemento ***<h1>*** viene modificato quando un utente fa clic su di esso:

<h1 onclick="this.innerHTML = 'Ooops!'">Click on this text!</h1>

<https://www.w3schools.com/js/tryit.asp?filename=tryjs_event_onclick2>

In questo esempio, una funzione viene chiamata dal gestore eventi:

<h1 onclick="changeText(this)">Click on this text!</h1>  
<script>  
function changeText(id) {   
  id.innerHTML = "Ooops!";  
}  
</script> <https://www.w3schools.com/js/tryit.asp?filename=tryjs_event_onclick3>

***Attributi evento HTML***

Per assegnare eventi a elementi HTML è possibile utilizzare gli attributi evento.

<button onclick="displayDate()">Try it</button>

<https://www.w3schools.com/js/tryit.asp?filename=tryjs_events1>

***Assegna eventi utilizzando il DOM HTML***

Il DOM HTML ti consente di assegnare eventi ad elementi HTML usando JavaScript:

<script>  
document.getElementById("myBtn").onclick = displayDate;  
</script> <https://www.w3schools.com/js/tryit.asp?filename=tryjs_events2>

***Gli eventi***

Gli eventi ***onload e onunload*** vengono attivati ​​quando l'utente entra o esce dalla pagina.

L'evento ***onload*** può essere utilizzato per verificare il tipo di browser del visitatore e la versione del browser e caricare la versione corretta della pagina Web in base alle informazioni.

Gli eventi ***onload e onunload*** possono essere utilizzati per gestire i ***cookie***.

<body onload="checkCookies()">

<https://www.w3schools.com/js/tryit.asp?filename=tryjs_events_onload>

L' evento ***onchange*** è spesso utilizzato in combinazione con la convalida dei campi di input.

<input type="text" id="fname" onchange="upperCase()">

<https://www.w3schools.com/js/tryit.asp?filename=tryjs_onchange>

Gli eventi ***onmouseover*** e ***onmouseout*** possono essere utilizzati per attivare una funzione quando l'utente esegue il mouse sopra o fuori da un elemento HTML:

<https://www.w3schools.com/js/tryit.asp?filename=tryjs_events_mouseover>

I ***onmousedown***, ***onmouseup*** e ***onclick*** gli eventi sono tutti parti di un clic del mouse. Per prima cosa quando viene cliccato un pulsante del mouse, viene attivato l'evento ***onmousedown***, quindi, quando viene rilasciato il pulsante del mouse, viene attivato l'evento ***onmouseup***, infine, quando viene completato il clic del mouse, viene attivato l'evento ***onclick***.

**HTML EventListener -** [**https://www.w3schools.com/js/js\_htmldom\_eventlistener.asp**](https://www.w3schools.com/js/js_htmldom_eventlistener.asp)