

Selezionare gli elementi

[Pagina principale di Javascript](#)

Per modificare la pagina web, bisogna agire sull'oggetto che lo rappresenta cioè sul document. Per scegliere gli elementi sui quali agire bisogna usare i metodi di document.

- document.**getElementById()**
- document.**getElementsByClassName()**
- document.**getElementsByTagName()**
- document.**querySelector()**
- document.**querySelectorAll()**

Analizziamoli uno alla volta

document.getElementById()

E' un metodo che viene utilizzato per ottenere un riferimento a un elemento HTML all'interno di un documento utilizzando il suo attributo id.

Questo metodo restituisce l'elemento corrispondente se presente nel documento, altrimenti restituisce **null**.

document.getElementById() seleziona un **solo elemento**, se esiste, nella pagina. Se proviamo ad agire su un elemento che non esiste, otteniamo un errore.

In Javascript si possono effettuare modifiche su un singolo oggetto alla volta, non su un raggruppamento di elementi.

```
document.getElementById()
```

```
<body>
<style>
    #titolo{color:red;}
</style>

<h1 id="titolo">Titolo della Pagina</h1>

<script>
    //Lo script deve essere dopo l'oggetto da selezionare
    // Ottenere il riferimento all'elemento con id "titolo"
    var titolo = document.getElementById('titolo');

    // Modificare l'HTML dell'elemento
    titolo.innerHTML = 'Nuovo Titolo';
</script>

</body>
```

Quando definiamo le proprietà CSS di un id, usiamo **#id**, quando usiamo `getElementById()` dobbiamo scrivere il nome dell'id **senza #**.

document.getElementsByClassName()

E' un metodo utilizzato per ottenere un elenco di tutti gli elementi HTML all'interno di un documento che hanno una determinata classe CSS.

Restituisce una collezione (non un array) di tutti gli elementi che corrispondono alla classe specificata. Nel nostro caso tratteremo array e collezioni nello stesso modo.

`getElementsByClassName()`

```
<body>
<style>
    .paragrafoRosso {
        color: red;
    }
</style>

<p class="paragrafoRosso">Questo è un paragrafo rosso.</p>
<p class="paragrafoRosso">Questo è un altro paragrafo rosso.</p>
<p>Questo è un paragrafo normale.</p><!--non verrà selezionato -->

<script>
    //Lo script deve essere dopo gli oggetti da selezionare

    // Ottenere un elenco di tutti gli elementi con classe "paragrafoRosso"
    var paragrafiRossi = document.getElementsByClassName('paragrafoRosso');
    // Questo non si può fare!!
    paragrafiRossi.innerHTML = "Ciao"
    // Bisogna iterare attraverso gli elementi per fareci qualcosa co
    for (var i = 0; i < paragrafiRossi.length; i++) {
        paragrafiRossi[i].textContent = 'Questo paragrafo è stato
selezionato e modificato.';
    }
</script>

</body>
```

Quando definiamo una classe nei CSS dobbiamo usare **.nomeClasse**, quando usiamo il metodo `getElementsByClassName()` il **punto** non va messo.

document.getElementsByTagName()

Il metodo `document.getElementsByTagName()` è un metodo utilizzato per ottenere una lista di tutti gli elementi HTML all'interno di un documento che hanno un determinato tag.

Restituisce una collezione (non un array) di tutti gli elementi che corrispondono al tag specificato.

`getElementsByTagName()`

```
<body>
<style>
  .paragrafoRosso {
    color: red;
  }
</style>

<p class="paragrafoRosso">Questo è un paragrafo rosso.</p>
<p class="paragrafoRosso">Questo è un altro paragrafo rosso.</p>
<p>Questo è un paragrafo normale.</p>

<script>
  //Lo script deve essere dopo gli oggetti da selezionare
  // Ottenere un elenco di tutti gli elementi con classe "paragrafoRosso"
  var paragrafi = document.getElementsByTagName('p');
  // Questo non si può fare!!
  paragrafi.innerHTML = "Ciao"
  // Bisogna iterare attraverso gli elementi per fareci qualcosa co
  for (var i = 0; i < paragrafi.length; i++) {
    paragrafi[i].textContent = 'Questo paragrafo è stato selezionato e
modificato.';
  }
</script>

</body>
```

document.querySelector()

Il metodo `document.querySelector()` è un'importante metodo che permette di selezionare e recuperare un elemento HTML all'interno del documento utilizzando un selettore CSS. Questo metodo restituisce solo il primo elemento corrispondente al selettore specificato.

`querySelector()`

```
<body>
  <p>Primo paragrafo</p> <!--seleziona questo-->
  <p>Secondo paragrafo</p>
  <p>Terzo paragrafo</p>

  <div id="mioId">
    <p>paragrafo dentro div#mioId</p> <!--seleziona questo-->
  </div>
```

```
<script>
  let primoP = document.querySelector('p');
  primoP.innerHTML = "<strong>Primo paragrafo</strong>";

  let pInDiv = document.querySelector('#mioId p');
  primoP.innerHTML = "<strong>paragrafo dentro div#mioId</strong>";
</script>
</body>
```

document.querySelectorAll()

Il metodo `document.querySelectorAll()` è un metodo molto potente che consente di selezionare uno o più elementi all'interno di un documento HTML utilizzando un selettore CSS. Restituisce una `NodeList` (non un array) di tutti gli elementi che corrispondono al selettore specificato. Per i nostri scopi tratteremo una `NodeList` come un array.

`querySelectorAll()`

```
<body>

<p>Questo è un paragrafo.</p>
<p class="highlight">Questo è un paragrafo con la classe "highlight".</p>
<ul>
  <li>Elemento 1</li>
  <li>Elemento 2</li>
  <li>Elemento 3</li>
</ul>

<script>
  // Ottenere un elenco di tutti gli elementi <p> e <li> con la classe
  "highlight"
  var elementiConClasseHighlight = document.querySelectorAll('p.highlight,
li.highlight');

  // Iterare attraverso gli elementi e fare qualcosa con essi
  elementiConClasseHighlight.forEach(function(elemento) {
    elemento.textContent = 'Questo elemento è stato selezionato e
modificato.';
  });
</script>

</body>
```

Esercizi

1. Vedere cosa succede se si usa `document.getElementById()` con due elementi con lo stesso id.
2. Vedere cosa succede se si usa `document.getElementsByClassName()` se non ci sono elementi

- con quella classe
3. Vedere cosa succede se si usa `document.getElementsByTagName()` se non ci sono con il tag selezionato
 4. Vedere cosa succede se si usa `document.querySelector()` con un selettore CSS che non seleziona niente
 5. Vedere cosa succede se si usa `document.querySelectorAll()` con un selettore CSS che non seleziona niente
 6. Selezionare tutti i paragrafi presenti in una pagina e sostituire il contenuto attuale di ciascun paragrafo con lo stesso testo in maiuscolo
 7. Selezionare tutti i paragrafi con classe `miaClasse` presenti in una pagina e sostituire il contenuto attuale di ciascun paragrafo con lo stesso testo in minuscolo
 8. Creare una funzione `contaTag` che prende in input un selettore CSS e ci dice quanti sono gli elementi selezionati dal selettore
 9. Creare una funzione `contaTag2` che prende in input un tag HTML e ci dice quanti sono gli elementi selezionati dal selettore
 10. Creare una funzione `listaCont` che prende in input una classe e restituisce un array con il contenuto di tutti i tag che hanno quella classe

From:

<https://www.wiki.webtocom.com/> - **WTC wiki**

Permanent link:

<https://www.wiki.webtocom.com/doku.php?id=page:selettori>

Last update: **2024/04/09 10:40**

