

# Manipolazione del DOM in JavaScript

## Sommario

- Cos'è il DOM? ..... 2
- Selezione degli Elementi del DOM ..... 3
- Manipolazione degli Attributi e del Contenuto ..... 5

# Cos'è il DOM?

Il **DOM** (Document Object Model) è una rappresentazione ad oggetti della struttura di un documento HTML o XML. Il DOM rappresenta ogni elemento di un documento come un oggetto, permettendo di modificarne dinamicamente il contenuto, gli attributi, e la struttura.

In HTML, ogni parte di una pagina è rappresentata da un nodo del DOM. Ad esempio, un elemento `<div>`, `<p>`, un attributo come `class` o `id` e persino il testo all'interno degli elementi sono tutti rappresentati come nodi nel DOM.

JavaScript permette di interagire con il DOM per:

- Accedere agli elementi della pagina.
- Modificarli, sia nel contenuto che nelle proprietà.
- Aggiungere o rimuovere nodi.
- Gestire eventi.

# Selezione degli Elementi del DOM

Per interagire con gli elementi del DOM, dobbiamo prima **selezionarli**. JavaScript fornisce vari metodi per farlo.

## 1. getElementById

Il metodo `getElementById` permette di selezionare un elemento HTML in base al suo **ID**. Poiché l'ID di un elemento deve essere unico nella pagina, questo metodo è molto veloce e diretto.

**Esempio:**

Immagina di avere un paragrafo con un ID chiamato "saluto" e vuoi modificarne il contenuto.

HTML:

```
<p id="saluto">Ciao, mondo!</p>
```

JavaScript:

```
// Seleziona l'elemento con id "saluto"
let elemento = document.getElementById("saluto");
// Modifica il contenuto
elemento.textContent = "Ciao, JavaScript!";
```

In questo esempio, il contenuto del paragrafo cambia da "Ciao, mondo!" a "Ciao, JavaScript!".

## 2. querySelector

Il metodo `querySelector` è molto più flessibile rispetto a `getElementById` perché permette di selezionare elementi non solo per ID, ma anche per **classi**, **attributi** e altre modalità di selezione CSS. Restituisce il primo elemento che corrisponde al selettore CSS dato.

**Esempio:**

Immagina di avere più paragrafi e vuoi selezionare il primo che ha una determinata classe.

HTML:

```
<p class="messaggio">Primo messaggio</p>
<p class="messaggio">Secondo messaggio</p>
```

JavaScript:

```
// Seleziona il primo elemento con la classe "messaggio"
let primoMessaggio = document.querySelector(".messaggio");
// Modifica il contenuto
primoMessaggio.textContent = "Messaggio modificato!";
```

In questo esempio, solo il primo paragrafo con la classe `messaggio` viene modificato.

## 3. querySelectorAll

A differenza di `querySelector`, che seleziona il primo elemento che corrisponde al selettore, `querySelectorAll` restituisce una **NodeList** contenente tutti gli elementi che corrispondono al selettore.

#### Esempio:

Supponiamo di avere più paragrafi con la stessa classe e desideriamo modificare il contenuto di tutti.

#### HTML:

```
<p class="messaggio">Messaggio 1</p>
<p class="messaggio">Messaggio 2</p>
<p class="messaggio">Messaggio 3</p>
```

#### JavaScript:

```
// Seleziona tutti gli elementi con la classe "messaggio"
let messaggi = document.querySelectorAll(".messaggio");

// Modifica il contenuto di ogni messaggio
messaggi.forEach(function(messaggio) {
    messaggio.textContent = "Messaggio aggiornato!";
});
```

In questo esempio, tutti i paragrafi con la classe `messaggio` avranno il contenuto modificato.

# Manipolazione degli Attributi e del Contenuto

Una volta selezionato un elemento del DOM, possiamo modificarne **gli attributi** (come `src`, `href`, `class`, etc.) o il **contenuto**.

## 1. Manipolazione del contenuto:

Per cambiare il contenuto di un elemento, possiamo utilizzare due metodi principali:

- **textContent**: Modifica solo il testo all'interno dell'elemento.
- **innerHTML**: Modifica il contenuto HTML, permettendo di aggiungere anche tag HTML all'interno dell'elemento.

**Esempio:**

Immagina di voler cambiare il testo di un paragrafo e aggiungere un altro elemento HTML (come un `<strong>`).

**HTML:**

```
<p id="descrizione">Benvenuto alla pagina!</p>
```

**JavaScript:**

```
// Modifica solo il testo del paragrafo
document.getElementById("descrizione").textContent = "Testo aggiornato!";

// Modifica il contenuto HTML, includendo un elemento <strong>
document.getElementById("descrizione").innerHTML = "Testo
<strong>importante</strong>!";
```

Il primo esempio cambia solo il testo dentro il paragrafo, mentre il secondo inserisce anche un tag HTML (`<strong>`) per rendere una parte del testo in grassetto.

## 2. Manipolazione degli attributi:

Per manipolare gli attributi di un elemento, possiamo usare i metodi `getAttribute`, `setAttribute`, e `removeAttribute`.

**Esempio:**

Immagina di avere un elemento `img` e desideri cambiare l'attributo `src`.

**HTML:**

```

```

**JavaScript:**

```
// Cambia l'attributo src
document.getElementById("immagine").setAttribute("src", "foto2.jpg");

// Ottieni il valore dell'attributo alt
```

```
let altText = document.getElementById("immagine").getAttribute("alt");
console.log(altText); // "Immagine di esempio"
```

In questo caso, l'immagine viene cambiata dinamicamente e viene ottenuto il valore dell'attributo `alt`.

### 3. Rimozione di un attributo:

Per rimuovere un attributo, possiamo utilizzare `removeAttribute`.

**Esempio:**

Immagina di voler rimuovere l'attributo `src` da un elemento `img`:

JavaScript:

```
// Rimuove l'attributo "src" dall'immagine
document.getElementById("immagine").removeAttribute("src");
```

In questo esempio, l'immagine perderà la fonte (`src`) e non verrà più mostrata.

## Riepilogo

1. **Cos'è il DOM:** È un modello ad oggetti della struttura di un documento HTML che permette l'interazione con gli elementi di una pagina web.
2. **Selezione degli elementi:**
  - o `getElementById`: Seleziona un elemento in base all'ID.
  - o `querySelector`: Seleziona il primo elemento che corrisponde a un selettore CSS.
  - o `querySelectorAll`: Seleziona tutti gli elementi che corrispondono a un selettore CSS.
3. **Manipolazione del contenuto:**
  - o `textContent`: Modifica solo il testo.
  - o `innerHTML`: Modifica il contenuto HTML.
4. **Manipolazione degli attributi:**
  - o `setAttribute`: Modifica un attributo.
  - o `getAttribute`: Ottiene il valore di un attributo.
  - o `removeAttribute`: Rimuove un attributo.