



# Manipolazione del DOM

*Python*

## **Cos'è il DOM**

**Il DOM (Document Object Model) è una rappresentazione strutturata di un documento HTML o XML sotto forma di un albero gerarchico di oggetti.**

**Ogni elemento, attributo o testo all'interno di una pagina web viene rappresentato come un nodo di questo albero.**

**Grazie al DOM, i linguaggi di programmazione come JavaScript possono accedere, esplorare e modificare dinamicamente la struttura, il contenuto e lo stile di una pagina web durante l'esecuzione.**



**Il DOM viene generato dal browser quando carica la pagina, ed è indipendente dal linguaggio di markup originale, consentendo una manipolazione versatile dei documenti.**

## **Cos'è la Manipolazione del DOM**

**La manipolazione del DOM consiste nell'interagire, modificare o aggiornare il DOM tramite codice, solitamente usando JavaScript.**

**Questo include operazioni come aggiungere, rimuovere o modificare elementi e attributi, cambiare testi, aggiornare stili CSS o rispondere a eventi utente (come click o input).**

**La manipolazione del DOM è fondamentale per creare pagine web dinamiche e interattive, dove il contenuto e la struttura della pagina possono cambiare in risposta alle azioni dell'utente o ad altre condizioni.**



## Esempio pratico in JavaScript

Supponiamo di voler cambiare il testo di un elemento `<p>` con `id="demo"` cliccando su un pulsante:

```
1.<!-- HTML -->
2.<p id="demo">Testo originale</p>
3.<button onclick="cambiaTesto()">Cambia Testo</button>
4.
5.<script>
6.function cambiaTesto() {
7.    // Seleziona l'elemento con id 'demo'
8.    var paragrafo = document.getElementById("demo");
9.    // Modifica il contenuto testuale dell'elemento
10.    paragrafo.textContent = "Testo modificato tramite DOM!";
11.}
12.</script>
```

- Il paragrafo `<p id="demo">` è l'elemento che vogliamo modificare.
- Il bottone richiama la funzione `cambiaTesto()` quando viene cliccato.
- All'interno della funzione, usiamo `document.getElementById("demo")` per ottenere il nodo del DOM corrispondente.
- Modifichiamo il testo dell'elemento usando la proprietà `.textContent`



## **Caratteristiche della Manipolazione del DOM**

- **Dinamica**

**Permette di aggiornare in tempo reale la struttura, il contenuto e lo stile della pagina senza ricaricare la pagina stessa.**

- **Event-driven**

**La manipolazione spesso avviene in risposta a eventi (click, input, caricamento, ecc.), rendendo l'interazione con l'utente più fluida e reattiva.**



## **Caratteristiche della Manipolazione del DOM**

- **Accesso selettivo ai nodi**

**È possibile selezionare specifici elementi tramite vari metodi (getElementById, getElementsByClassName, querySelector, ecc.), consentendo modifiche mirate.**

- **Creazione e rimozione di elementi**

**Si possono creare nuovi nodi (createElement), aggiungerli all'albero (appendChild, insertBefore) o rimuoverli (removeChild).**



## **Caratteristiche della Manipolazione del DOM**

- **Modifica di attributi e proprietà**

**Si possono modificare attributi (es. `setAttribute`, `removeAttribute`) e proprietà (es. `textContent`, `innerHTML`, `style`) degli elementi.**

- **Prestazioni e reflow/repaint**

**Le modifiche al DOM possono causare operazioni di reflow e repaint, impattando le prestazioni, soprattutto se vengono fatte molte operazioni in sequenza.**



## **Caratteristiche della Manipolazione del DOM**

- **Compatibilità cross-browser**

**Anche se gli standard sono condivisi, alcune proprietà/metodi possono comportarsi diversamente su browser diversi, richiedendo attenzione nella programmazione.**

- **Sicurezza**

**La manipolazione diretta del DOM può esporre a rischi di sicurezza, come XSS, se non si filtrano adeguatamente i dati inseriti dinamicamente (ad esempio, tramite innerHTML).**





**Buon MasterD a tutti**

