

# Javascript 2

## Cos'è il DOM?

Il DOM (Document Object Model) è la rappresentazione JavaScript del tuo HTML. Pensa al DOM come a un **traduttore universale**:

- **HTML** = La struttura che vedi
- **DOM** = Lo stesso contenuto ma in "linguaggio JavaScript"
- **document** = L'oggetto madre che contiene tutto

```
// Esplora il DOM
console.log(document);    // Mostra tutto l'HTML
console.dir(document);    // Mostra le proprietà JavaScript
```

💡 **Concetto chiave:** Il DOM trasforma ogni tag HTML in un **oggetto JavaScript** che puoi manipolare!

## 1. SELEZIONARE ELEMENTI

### querySelector() - Per UN elemento

```
// Seleziona per ID (# davanti)
const header = document.querySelector('#main-header');

// Seleziona per classe (. davanti)
const bottone = document.querySelector('.btn-primary');

// Seleziona per tag (nessun simbolo)
const paragrafo = document.querySelector('p');
```

💡 **Regola d'oro:** Sempre selezionare l'elemento prima di modificarlo!

## 1.2. MANIPOLARE IL CONTENUTO

### innerHTML - Cambia contenuto HTML

```
const div = document.querySelector('#myDiv');

// Sostituisce tutto il contenuto
div.innerHTML = '<h2>Nuovo titolo</h2>';

// Aggiunge al contenuto esistente
div.innerHTML += '<p>Paragrafo aggiunto</p>';

// Usa template string per variabili
const nome = 'Mario';
const eta = 30;
div.innerHTML = `
  <h3>Ciao ${nome}</h3>
  <p>Hai ${eta} anni</p>
`;
```

## innerText - Solo testo semplice

```
const p = document.querySelector('p');
p.innerText = 'Solo testo, niente HTML';

// La differenza:
element.innerHTML = '<strong>Grassetto</strong>'; // Verrà renderizzato
in grassetto
element.innerText = '<strong>Grassetto</strong>'; // Verrà mostrato come
testo normale
```

## 1.3. ESEMPIO PRATICO: STAMPARE UN ARRAY IN UNA LISTA HTML

Questo è uno dei pattern più comuni nel web development!

### HTML di partenza:

```
<ul id="lista-prodotti"></ul>
<div id="container-cards"></div>
```

## Metodo 1: Loop classico

```
const prodotti = ['iPhone', 'Samsung', 'Xiaomi', 'OnePlus'];
const lista = document.querySelector('#lista-prodotti');

// Svuota la lista prima di riempirla
lista.innerHTML = '';

// Aggiungi ogni elemento
for (let i = 0; i < prodotti.length; i++) {
  lista.innerHTML += `<li>${prodotti[i]}</li>`;
}
```

## Metodo 2: forEach (più moderno)

```
const prodotti = ['iPhone', 'Samsung', 'Xiaomi', 'OnePlus'];
const lista = document.querySelector('#lista-prodotti');

lista.innerHTML = ''; // Pulisci la lista

prodotti.forEach(prodotto => {
  lista.innerHTML += `<li>${prodotto}</li>`;
});
```

## Metodo 4: Con array di oggetti (caso reale)

```
const utenti = [
  { nome: 'Mario', email: 'mario@email.com', eta: 30 },
  { nome: 'Laura', email: 'laura@email.com', eta: 25 },
  { nome: 'Giuseppe', email: 'giuseppe@email.com', eta: 35 }
];

const container = document.querySelector('#container-cards');

const htmlCards = utenti.map(utente => `
  <div class="card">
    <h3>${utente.nome}</h3>
  `
```

```
<p>Email: ${utente.email}</p>
<p>Età: ${utente.eta} anni</p>
</div>
`).join('');
```

```
container.innerHTML = htmlCards;
```

## ESERCIZI PRATICI

### Esercizio 1: Selezione base

Crea un file HTML con questi elementi e selezionali:

```
<h1 id="titolo-principale">Il mio sito</h1>
<p class="descrizione">Questa è la descrizione</p>
<button class="btn btn-primary">Clicca qui</button>
<input type="text" id="nome-utente" placeholder="Il tuo nome">
```

**Da fare:**

```
// 1. Seleziona il titolo per ID
// 2. Seleziona la descrizione per classe
// 3. Seleziona il bottone (usa due classi)
// 4. Seleziona l'input per tipo
```

### Esercizio 2: Modifica contenuti

Partendo dall'HTML dell'esercizio 1:

```
// 1. Cambia il titolo in "Benvenuto nel mio sito!"
// 2. Cambia la descrizione in "Sito rinnovato con JavaScript"
// 3. Cambia il testo del bottone in "Inizia ora"
```

### Esercizio 3: Lista della spesa

Crea una lista della spesa dinamica:

```
<h2>Lista della spesa</h2>
<ul id="spesa"></ul>
```

```
const spesa = ['Pane', 'Latte', 'Uova', 'Pasta', 'Pomodori'];

// Stampa tutti gli elementi nella lista
// Risultato atteso:
// • Pane
// • Latte
// • Uova
// • Pasta
// • Pomodori
```

## Esercizio 5: Tabella utenti

Crea una tabella dinamica:

```
<table id="tabella-utenti">
  <thead>
    <tr>
      <th>Nome</th>
      <th>Cognome</th>
      <th>Email</th>
      <th>Città</th>
    </tr>
  </thead>
  <tbody id="corpo-tabella">
    </tbody>
</table>
```

```
const utenti = [
  { nome: 'Mario', cognome: 'Rossi', email: 'mario@email.com', citta: 'Roma' },
  { nome: 'Laura', cognome: 'Bianchi', email: 'laura@email.com', citta: 'Milano' },
  { nome: 'Giuseppe', cognome: 'Verdi', email: 'giuseppe@email.com', citta: 'Napoli' }
];
```

```
a: 'Napoli' }  
];  
  
// Riempi il tbody con i dati degli utenti
```

## Soluzione Esercizio 1:

```
const titolo = document.querySelector('#titolo-principale');  
const descrizione = document.querySelector('.descrizione');  
const bottone = document.querySelector('.btn.btn-primary');  
const input = document.querySelector('input[type="text"]');  
// o document.querySelector('#nome-utente');
```

## Soluzione Esercizio 2:

```
titolo.innerText = 'Benvenuto nel mio sito!';  
descrizione.innerText = 'Sito rinnovato con JavaScript';  
bottone.innerText = 'Inizia ora';
```

## Soluzione Esercizio 3:

```
const spesa = ['Pane', 'Latte', 'Uova', 'Pasta', 'Pomodori'];  
const listaSpesa = document.querySelector('#spesa');  
  
const htmlSpesa = spesa.map(item => `<li>${item}</li>`).join('');  
listaSpesa.innerHTML = htmlSpesa;
```

## Soluzione Esercizio 5:

```
const corpoTabella = document.querySelector('#corpo-tabella');  
  
const htmlRighe = utenti.map(utente => `  
  <tr>  
    <td>${utente.nome}</td>  
    <td>${utente.cognome}</td>  
    <td>${utente.email}</td>
```

```

        <td>${utente.citta}</td>
    </tr>
`).join('');

corpoTabella.innerHTML = htmlRighe;

```

## 2. SELEZIONARE ELEMENTI

```

// Seleziona per ID (# davanti)
const header = document.getElementById('main-header');

// Seleziona per classe (. davanti)
const bottone = document.getElementsByClassName('di');

```

## 3. MANIPOLARE GLI STILI

### Metodo 1: Proprietà style (CSS inline)

```

const elemento = document.querySelector('#myElement');

// Singole proprietà (camelCase!)
elemento.style.color = 'red';
elemento.style.backgroundColor = 'yellow';
elemento.style.fontSize = '20px';

// Proprietà multiple
elemento.style.cssText = 'color: blue; background: white; padding: 10px;';

```

### Metodo 2: Classi CSS (NE)

```

const elemento = document.querySelector('#elem');

function cambiaColore() {
    if (!elemento.classList.contains('rosso')) {
        elemento.classList.add('rosso');
    }
}

```

```
} else {  
  elemento.classList.remove('rosso');  
}  
}
```

```
.square {  
  background-color: blue;  
  height: 150px;  
  width: 150px;  
}
```

```
.rosso {  
  background-color: red;  
}
```

## 4. MANIPOLARE ATTRIBUTI

```
const link = document.querySelector('a');  
  
// Leggi attributo  
console.log(link.href);  
console.log(link.getAttribute('target'));  
  
// Modifica attributo  
link.href = 'https://google.com';  
link.setAttribute('target', '_blank');  
  
// Rimuovi attributo  
link.removeAttribute('target');  
  
// Attributi speciali  
const input = document.querySelector('input');
```



```
input.value = 'Nuovo valore';  
input.disabled = true;
```

## ESERCIZI 1

### Esercizio 1.1: Cambia Colore

Crea una pagina con:

- Un titolo h1 con id="title"
- Un paragrafo con classe="text"
- Un bottone con id="colorBtn"

Al click del bottone, cambia il colore del titolo in rosso e del paragrafo in blu.

### Esercizio 1.2: Lista Dinamica

Crea una lista ul vuota con id="myList". Aggiungi 5 elementi li con i nomi dei tuoi servizi usando innerHTML.

## MODULO 2: EVENTI E INTERATTIVITÀ (3 ore)

### 1. GESTIRE GLI EVENTI

#### addEventListener() - La base

```
const bottone = document.querySelector('#myBtn');  
  
// Sintassi base  
bottone.addEventListener('click', function() {  
  console.log('Bottone cliccato!');  
});  
  
// Funzione separata (più pulita)  
function handleClick() {  
  console.log('Bottone cliccato!');
```

```
}  
bottone.addEventListener('click', handleClick);
```

## Oggetto Event

```
bottone.addEventListener('click', function(event) {  
  console.log('Tipo evento:', event.type);  
  console.log('Elemento cliccato:', event.target);  
  console.log('Posizione mouse:', event.clientX, event.clientY);  
  
  // Previene comportamento predefinito  
  event.preventDefault();  
});
```

## 2. EVENTI PIÙ COMUNI

```
// CLICK - Clic del mouse  
elemento.addEventListener('click', handleClick);  
  
// INPUT - Cambio valore input  
input.addEventListener('input', function(e) {  
  console.log('Nuovo valore:', e.target.value);  
});  
  
// MOUSEOVER/MOUSEOUT - Entrata/uscita mouse  
div.addEventListener('mouseover', () => {  
  div.style.backgroundColor = 'yellow';  
});  
div.addEventListener('mouseout', () => {  
  div.style.backgroundColor = 'white';  
});  
  
// KEYDOWN/KEYUP - Pressione tasti  
document.addEventListener('keydown', function(e) {  
  console.log('Tasto premuto:', e.key);  
  if (e.key === 'Enter') {  
    console.log('Hai premuto Invio!');  
  }  
});
```

```

    }
  });

  // SCROLL - Scorrimento pagina
  window.addEventListener('scroll', function() {
    console.log('Scroll position:', window.scrollY);
  });

```

### 3. EVENTO SCROLL AVANZATO

```

// Navbar che cambia colore scrollando
const navbar = document.querySelector('#navbar');

window.addEventListener('scroll', function() {
  if (window.scrollY > 100) {
    navbar.classList.add('scrolled');
  } else {
    navbar.classList.remove('scrolled');
  }
});

```

### 4. LAVORARE CON ARRAY DI DATI

```

// Dati esempio
const products = [
  { id: 1, name: 'Laptop', price: 999, category: 'tech' },
  { id: 2, name: 'Mouse', price: 29, category: 'tech' },
  { id: 3, name: 'Libro', price: 15, category: 'books' }
];

// Renderizza tutti i prodotti
function renderProducts(productsArray) {
  const container = document.querySelector('#products');
  container.innerHTML = ''; // Pulisci container

  productsArray.forEach(product => {
    const productElement = document.createElement('div');

```

```

productElement.classList.add('product');

productElement.innerHTML = `
  <h3>${product.name}</h3>
  <p>Prezzo: €${product.price}</p>
  <span class="category">${product.category}</span>
  <button onclick="buyProduct(${product.id})">Compra</button>
`;

container.appendChild(productElement);
});
}

// Filtra prodotti
function filterProducts(category) {
  const filtered = products.filter(product =>
    category === 'all' || product.category === category
  );
  renderProducts(filtered);
}

// Inizializza
renderProducts(products);

```

## ESERCIZI 2

### Esercizio 2.1: Contatore Interattivo

Crea:

- Un div che mostra un numero (inizia da 0)
- Un bottone "+" che aumenta il numero
- Un bottone "-" che diminuisce il numero
- Un bottone "Reset" che riporta a 0

### Esercizio 2.2: Cambia Sfondo

Crea una pagina che:

- Ha 4 bottoni colorati (rosso, blu, verde, giallo)
- Al click di ogni bottone, lo sfondo della pagina diventa di quel colore
- Aggiungi un bottone "Reset" che riporta lo sfondo bianco

## **Esercizio 2.3: Todo List Interattiva**

Crea una todo list con:

- Input per scrivere una task
  - Bottone "Aggiungi"
  - Lista che mostra le task aggiunte
  - Ogni task ha un bottone "X" per eliminarla
  - Conta totale delle task
-