

## Introduzione a JavaScript

JavaScript è il linguaggio di programmazione che rende le pagine web interattive. Mentre HTML definisce la struttura e CSS lo stile, JavaScript aggiunge:

- Reazioni a click e input utente
- Animazioni e effetti dinamici
- Modifiche al contenuto senza ricaricare la pagina
- Elaborazione di dati e calcoli

## 2. Come si usa JavaScript

Puoi inserire codice JavaScript in tre modi:

1. **Internamente** nella pagina HTML, tra tag `<script>` e `</script>`
2. **Esternamente** in un file separato (estensione .js) collegato con `<script src="file.js"></script>`
3. **Inline** direttamente negli attributi HTML (sconsigliato per codice complesso)

Il posto migliore per inserire gli script è prima della chiusura del tag `</body>` per garantire che la pagina sia già caricata.

## 3. Concetti Fondamentali

### Variabili

Contenitori per memorizzare dati:

- `let` per variabili modificabili
- `const` per valori costanti
- `var` (vecchio metodo, evitare)

Esempi:

```
let nome = "Marco";  
const PI_GRECO = 3.14;
```

### Tipi di Dati

- **Stringhe:** Testo tra virgolette "ciao"
- **Numeri:** 10, 3.14
- **Booleani:** true o false
- **Array:** Liste ["pane", "latte", "uova"]
- **Oggetti:** {nome: "Luigi", età: 25}

### Operatori

- Aritmetici: +, -, \*, /
- Comparazione: >, <, ===, !==
- Logici: && (e), || (o), ! (non)

## 4. Strutture di Controllo

### Condizioni (if/else)

Copy

Download

```
Se (condizione) {  
    esegui codice  
} altrimenti se (altra condizione) {  
    esegui altro codice  
} altrimenti {  
    codice di default  
}
```

### Cicli (for/while)

- **For:** Ripete un blocco per un numero specifico di volte
- **While:** Ripete finché una condizione è vera

## 5. Funzioni

Blocchi di codice riutilizzabili:

Copy

Download

```
function nomeFunzione(parametro1, parametro2) {  
    // istruzioni  
    return risultato;  
}
```

Forma moderna (arrow function):

Copy

Download

```
const somma = (a, b) => a + b;
```

## 6. Interazione con HTML

### Selezionare Elementi

- `document.getElementById("id")` - Seleziona un elemento unico
- `document.querySelector(".classe")` - Seleziona il primo elemento che matcha
- `document.querySelectorAll("p")` - Seleziona tutti i paragrafi

### Modificare Elementi

Puoi cambiare:

- Contenuto con `.innerHTML`
- Stile con `.style`.proprietà
- Classi CSS con `.classList`

### Gestire Eventi

Reagire a azioni dell'utente:

- `click` - Clic del mouse
- `mouseover` - Passaggio del puntatore
- `keydown` - Premuto tasto
- `submit` - Invio form

Esempio di listener:

Copy

Download

```
elemento.addEventListener("tipoEvento", funzioneDaEeguire);
```

## 7. Esempio Pratico

Immagina un contatore che aumenta ogni click:

1. HTML crea un bottone e un display
2. JavaScript:
  - Memorizza il conteggio in una variabile
  - Aggiunge un listener al bottone
  - Incrementa e aggiorna il display a ogni click

## 8. Errori Comuni

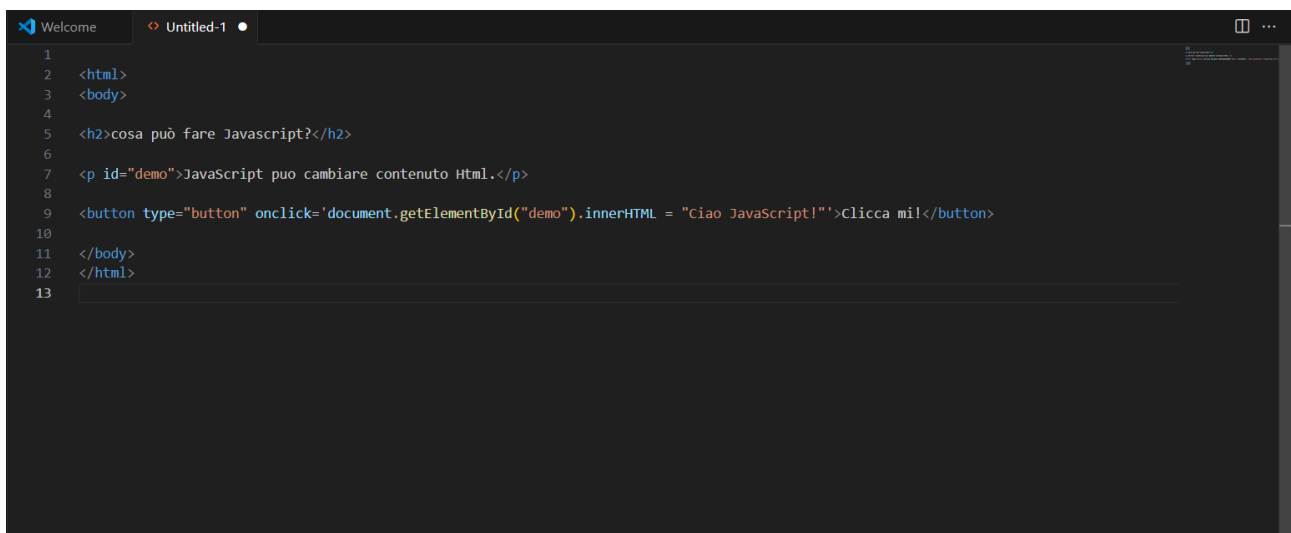
- Confondere `=` (assegnazione) con `==/===` (confronto)

- Dimenticare parentesi o punti e virgola
- Usare variabili non dichiarate
- Selezionare elementi non ancora caricati

## 9. Per Iniziare

1. Apri la console del browser (F12 > Console)
2. Prova comandi semplici come `alert("Hello")`
3. Modifica elementi HTML esistenti
4. Crea piccole interazioni (cambia colore al click)

Ecco un esempio di Javascript:



```
1
2 <html>
3 <body>
4
5 <h2>cosa può fare Javascript?</h2>
6
7 <p id="demo">JavaScript puo cambiare contenuto Html.</p>
8
9 <button type="button" onclick='document.getElementById("demo").innerHTML = "Ciao JavaScript!";'>Clicca mi!</button>
10
11 </body>
12 </html>
13
```