



# Introduzione JS

*Programmazione*

# **Introduzione a JavaScript**

**JavaScript è un linguaggio di programmazione interpretato, orientato agli oggetti e dinamico, progettato originariamente per aggiungere interattività alle pagine web.**

**Viene eseguito direttamente dal browser dell'utente, senza necessità di compilazione, rendendolo essenziale per lo sviluppo frontend.**

**Nasce nel 1995 e oggi è uno dei pilastri del web insieme a HTML e CSS.**



**Oltre a essere usato nei browser, grazie a tecnologie come Node.js, JavaScript può essere impiegato anche lato server.**

**La sua sintassi è flessibile, e il linguaggio supporta programmazione imperativa, funzionale e asincrona, il che lo rende versatile per applicazioni complesse, giochi, interfacce dinamiche e persino intelligenza artificiale.**



**JavaScript venne creato da Brendan Eich nel maggio 1995 in soli dieci giorni presso Netscape, inizialmente chiamato “Mocha”, poi “LiveScript” e infine “JavaScript” per sfruttare la popolarità di Java .**

**Nel 1997 divenne standard ECMA-262 (ECMAScript), evolvendosi con versioni chiave: ES3 nel 1999, ES5 nel 2009 (introducendo strict mode), ES6/ES2015 nel 2015 (con classi, moduli, arrow function, promise), e successive edizioni annuali fino all’ES2024 che ha aggiunto metodi come Object.groupBy e Map.groupBy .**

**Oggi JavaScript è il linguaggio più usato al mondo, supportato da un vasto ecosistema e standardato da TC39 sotto Ecma International .**



```
1. <!DOCTYPE html>
2. <html>
3. <head>
4.   <title>Esempio JS</title>
5. </head>
6. <body>
7.   <h1 id="titolo">Ciao!</h1>
8.   <button onclick="cambiaTesto()">Clicca qui</button>
9.
10.  <script>
11.    function cambiaTesto() {
12.      document.getElementById("titolo").innerText = "Hai cliccato il bottone!";
13.    }
14.  </script>
15. </body>
16. </html>
```

## Spiegazione:

- La pagina mostra un titolo (<h1>) e un pulsante (<button>).
- Quando l'utente clicca il pulsante, viene chiamata la funzione cambiaTesto().
- Questa funzione modifica il testo del titolo (innerText) accedendo all'elemento tramite il suo id.



## Dichiarazione di variabili e output

In JavaScript puoi dichiarare variabili usando `let`, `const` o `var` (più vecchio). Per stampare qualcosa sullo schermo si usa `console.log()`.

1. `let nome = "Mario";`
2. `const eta = 25;`
- 3.
4. `console.log("Ciao, " + nome + ". Hai " + eta + " anni.");`

### Spiegazione:

- `let` permette di dichiarare una variabile modificabile.
- `const` crea una variabile costante (non modificabile).
- `console.log()` stampa il messaggio nella console del browser (F12 → Console)



## Condizione if con confronto

Le condizioni if permettono di eseguire codice solo se una certa condizione è vera.

```
1. let numero = 10;  
2.  
3. if (numero > 5) {  
4.   console.log("Il numero è maggiore di 5");  
5. } else {  
6.   console.log("Il numero è 5 o minore");  
7. }
```

### Spiegazione:

- L'if controlla se `numero > 5`.
- Se la condizione è vera, esegue il primo blocco.
- Altrimenti, esegue il blocco dell'else.



## Ciclo for per scorrere un array

**Il ciclo for viene spesso usato per iterare su un array (una lista di elementi).**

```
1. let frutti = ["mela", "banana", "kiwi"];  
2.  
3. for (let i = 0; i < frutti.length; i++) {  
4.   console.log("Frutto: " + frutti[i]);  
5. }
```

### Spiegazione:

- L'array frutti contiene tre elementi.
- Il ciclo for parte da  $i = 0$  e finisce quando  $i < \text{frutti.length}$ .
- Ad ogni iterazione stampa l'elemento corrente (`frutti[i]`).





## Definizione e chiamata di una funzione

**Una funzione in JavaScript consente di racchiudere blocchi di codice riutilizzabili.**

```
1.function saluta(nome) {  
2.  console.log("Ciao, " + nome + "!");  
3.}  
4.  
5.saluta("Luca");  
6.saluta("Anna");
```

### Spiegazione:

- La funzione saluta prende un parametro nome.
- Quando viene chiamata, stampa un messaggio personalizzato.
- È un esempio base di riutilizzo del codice.



## Caratteristiche principali di JavaScript

- È interpretato e eseguito lato client (ma anche lato server con Node.js).
- Supporta la programmazione ad oggetti e funzionale.
- È event-driven, cioè reattivo agli eventi dell'utente.
- Consente manipolazione del DOM (Document Object Model).
- Ha una tipizzazione dinamica (non si specificano i tipi delle variabili).
- Supporta l'asincronia (con async/await, Promise, callback).
- È cross-platform, funziona su tutti i browser moderni.
- Può essere esteso con framework e librerie (React, Vue, Angular, jQuery, ecc.).



**Buon Davante a tutti**

