Javascript 1

Immagina di dover creare un linguaggio di programmazione completamente nuovo. Quanto tempo pensi ti servirebbe? Un anno? Sei mesi? Brendan Eich, nel maggio del 1995, aveva esattamente **10 giorni**.

Netscape, il browser più popolare dell'epoca, aveva bisogno di un linguaggio che rendesse le pagine web interattive.

Marc Andreessen, co-fondatore di Netscape, voleva qualcosa che fosse "semplice come Java" (il linguaggio più alla moda del momento) ma che funzionasse direttamente nel browser. Il risultato? Un linguaggio chiamato inizialmente "Mocha", poi "LiveScript" e infine "JavaScript" - un nome scelto puramente per marketing, cavalcando l'onda del successo di Java.

> Aneddoto: Il nome "JavaScript" fu una mossa di marketing geniale ma confuse generazioni di programmatori.

Cos'è JavaScript Oggi?

JavaScript è passato dall'essere il "giocattolo" del web a diventare uno dei linguaggi più utilizzati al mondo. È l'unico linguaggio che può girare nativamente in ogni browser, ed è talmente versatile che oggi lo troviamo ovunque:

Il Classico Hello World

```
console.log("Hello, World!");
```

Ma fermiamoci un attimo. Perché `console.log()` ? La console era originariamente uno strumento di debug nascosto agli utenti. Oggi è il nostro migliore amico:

Capitolo 1: Introduzione a JavaScript

JavaScript è un linguaggio di programmazione interpretato che consente di rendere le pagine web interattive e dinamiche. È uno dei tre pilastri dello sviluppo web insieme a HTML e CSS.

Caratteristiche principali:

- Linguaggio interpretato (non compilato)
- Eseguito direttamente nel browser
- Sintassi flessibile e dinamica
- Ampiamente supportato

Come inserire JavaScript

Ci sono tre modi per includere JavaScript:

1. **Inline** (sconsigliato):

```
<button onclick="alert('Ciao!')">Clicca</button>
```

1. **Interno**:

```
<!DOCTYPE html>
<html>
<head>
  <script>
    console.log("Hello, world!");
  </script>
</head>
<body>
  <h1>La mia pagina</h1>
</body>
</html>
```

1. **Esterno** (consigliato):

```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript Esterno</title>
</head>
<body>
  <h1>La mia pagina</h1>
  <script src="script.js"></script>
```

```
</body>
</html>
```

La Console del Browser

La console è uno strumento fondamentale per il debugging:

- **Chrome**: Ctrl + Shift + J (Windows) / Cmd + Option + J (Mac)
- **Firefox**: Ctrl + Shift + K (Windows) / Cmd + Option + K (Mac)

```
console.log("Questo messaggio apparirà nella console");
```

Capitolo 2: Variabili e Costanti

Le variabili sono "contenitori" che memorizzano valori. In JavaScript moderno utilizziamo `let` per le variabili e `const` per le costanti.

Dichiarazione di Variabili

```
// Dichiarazione
//var nome; vecchio metodo
let nome;

// Assegnazione
nome = "Mario";

// Dichiarazione e inizializzazione
let eta = 25;
let cognome = "Rossi";

console.log("Ciao mi chiamo "+ nome+' '+cognome+'. Ho '+ eta+ ' anni');
```

Dichiarazione di Costanti

```
const PI_GRECO = 3.14;
const NOME_SITO = "Il mio sito";

// ERRORE: non posso riassegnare una costante
```

```
// PI_GRECO = 3.15; // Questo causerebbe un errore
console.log(NOME_SITO);
```

Regole per i Nomi

- Devono iniziare con una lettera, `_` o `$`
- Possono contenere lettere, numeri, `_` e `$`
- Sono case-sensitive (`nome` è diverso da `Nome`)
- Usare camelCase per più parole (`nomeCompleto`)

```
// Esempi validi
let nome = "Mario";
let nomeCompleto = "Mario Rossi";
let eta2023 = 30;
let _privato = "segreto";
let $speciale = "valore";

// Esempi NON validi
// let 2nome = "errore"; // inizia con numero
// let nome-completo = "errore"; // contiene trattino
```

Esercizi

Esercizio 2.1: Crea una pagina HTML con JavaScript interno che mostri "Benvenuto nel mondo JavaScript!" nella console.

Esercizio 2.2: Crea un file JavaScript esterno che mostri il tuo nome e la data corrente nella console, poi collegalo a una pagina HTML.

Esercizio 2.3: Crea 5 variabili che descrivano te stesso (nome, cognome, età, città, hobby) e stampale tutte nella console.

Esercizio 2.4: Crea delle costanti per i giorni della settimana e usale in frasi di benvenuto (es. "Buon lunedì!").

Capitolo 3: Tipi di Dato

JavaScript ha diversi tipi di dato che si dividono in **primitivi** e **non primitivi**.

Tipi Primitivi

1. Number (Numeri)

```
let intero = 42;
let decimale = 3.14;
let negativo = -15;
console.log(typeof intero); // "number"
```

2. String (Stringhe)

```
let semplice = 'Ciao';
let doppi = "Mondo";
let template = `Ciao ${semplice}!`; // Template literal
console.log(template); // "Ciao Ciao!"
```

3. Boolean (Booleani)

```
let vero = true;
let falso = false;
let confronto = 5 > 3; // true
```

4. Undefined

```
let nonDefinita;
console.log(nonDefinita); // undefined
```

5. Null

```
let vuoto = null; // Intenzionalmente vuoto
```

Verifica del Tipo

```
let numero = 42;
let testo = "Ciao";

console.log(typeof numero); // "number"
console.log(typeof testo);  // "string"
console.log(typeof true);   // "boolean"
```

Capitolo 4: Operatori

Teoria

Gli operatori ci permettono di eseguire operazioni sui dati.

Operatori Aritmetici

```
let a = 10;
let b = 3;

console.log(a + b);  // 13 (addizione)
console.log(a - b);  // 7 (sottrazione)
console.log(a * b);  // 30 (moltiplicazione)
console.log(a / b);  // 3.333... (divisione)
console.log(a % b);  // 1 (resto/modulo)

// Incremento e decremento
let counter = 5;
counter++; // 6
counter--; // 5
```

Operatori di Assegnazione

```
let x = 10;
x += 5;  // x = x + 5 → 15
x -= 3;  // x = x - 3 → 12
x *= 2;  // x = x * 2 → 24
x /= 4;  // x = x / 4 → 6
```

Operatori di Confronto

```
let num1 = 5;
let num2 = "5";

console.log(num1 == num2);   // true (uguaglianza con conversione)
console.log(num1 === num2);  // false (uguaglianza stretta)
console.log(num1 != num2);   // false
```

```
console.log(num1 !== num2);  // true
console.log(num1 > 3);       // true
console.log(num1 <= 5);      // true
```

Operatori Logici

```
let a = true;
let b = false;

console.log(a && b);  // false (AND)
console.log(a || b);  // true (OR)
console.log(!a);      // false (NOT)

// Esempi pratici
let eta = 20;
let haPatente = true;
let puo_guidare = eta >= 18 && haPatente; // true
```

Esercizi

Esercizio 4.1: Crea 5 variabili contenenti un **numero** e scrivi un programma in modo da ottenere la somma tra questi numeri e la media.

In console poi mostra la seguente frase: *'La somma tra i numeri equivale a ... e la media equivale a...'*

```
let num1 = 1;
  let num2 = 2;
  let num3 = 3;
  let num4 = 4;
  let num5 = 5;
  let somma = num1 + num2 + num3 + num4 + num5;
  let media = somma / 5;
  console.log('La somma tra i numeri equivale a ' + somma + ' e la
  media equivale a ' + media);
```

Appunti Short Circuit (domanda Giuseppe)

- **Valori falsy** - JavaScript considera falsy: `false`, `0`, `""`, `null`, `undefined`, `NaN`

```
const count = 0;
const result = count || 10;// result sarà 10, non 0
```

- **Operatore nullish coalescing (`??`)** - Introdotto in ES2020, controlla solo `null` e `undefined`

```
const count = 0;
const result = count ?? 10;// result sarà 0
```

Capitolo 5: Condizioni

Le condizioni permettono al programma di prendere decisioni ed eseguire codice diverso in base alle circostanze.

If, Else If, Else

```
let voto = 85;

if (voto >= 90) {
    console.log("Ottimo!");
} else if (voto >= 80) {
    console.log("Buono!");
} else if (voto >= 70) {
    console.log("Discreto");
} else if (voto >= 60) {
    console.log("Sufficiente");
} else {
    console.log("Insufficiente");
}
```

Operatore Ternario

```
// Sintassi: condizione ? valoreSeTruo : valoreSeFalso
let eta = 17;
let messaggio = eta >= 18 ? "Maggiorenne" : "Minorenne";
console.log(messaggio); // "Minorenne"

// Esempio pratico
let temperatura = 25;
let abbigliamento = temperatura > 20 ? "T-shirt" : "Giacca";
```

Switch Statement

```
let giorno = "lunedì";

switch (giorno) {
    case "lunedì":
        console.log("Inizio settimana!");
        break;
    case "venerdì":
        console.log("Quasi weekend!");
        break;
    case "sabato":
    case "domenica":
        console.log("Weekend!");
        break;
    default:
        console.log("Giorno feriale normale");
}
```

Valori Truthy e Falsy

```
// Valori falsy: false, 0, "", null, undefined, NaN
// Tutto il resto è truthy

if ("") {
    console.log("Non verrà mai eseguito");
}
```

```
if ("ciao") {
    console.log("Questo verrà eseguito"); // Le stringhe non vuote sono truth
y
}
```

Esercizi

Esercizio 5.1: Crea un programma che determini la stagione basandosi sul mese (1-12) usando switch statement.

Esercizio 5.2: Crea un sistema di valutazione che, data un'età, determini la categoria (bambino 0-12, adolescente 13-17, adulto 18-64, anziano 65+) e suggerisca un'attività appropriata. Gestire casistica di numeri negativi

---

Capitolo 6: Cicli

Teoria

I cicli permettono di ripetere blocchi di codice più volte, evitando duplicazioni.

Ciclo For

```
// Sintassi: for (inizializzazione; condizione; incremento)
for (let i = 0; i < 5; i++) {
    console.log("Iterazione numero: " + i);
}
```

Break e Continue

```
// Break: esce dal ciclo
for (let i = 0; i < 10; i++) {
    if (i === 5) {
        break; // Esce dal ciclo quando i è 5
    }
    console.log(i); // Stampa 0, 1, 2, 3, 4
}

// Continue: salta all'iterazione successiva
```

```
for (let i = 0; i < 10; i++) {
    if (i % 2 === 0) {
        continue; // Salta i numeri pari
    }
    console.log(i); // Stampa solo 1, 3, 5, 7, 9
}
```

Esercizi

Esercizio 6.1: Crea un programma che calcoli la somma di tutti i numeri da 1 a 100 usando un ciclo for.

Esercizio 6.2: Crea un programma che stampi la tabellina di un numero (scelto da te) da 1 a 10, saltando il numero 5 usando continue.

Esercizio 6.3: FIZZ-BUZZ: Utilizzando la logica appena appresa con l'operatore Modulo, scrivere un programma che stampi in console tutti i numeri da 1 a 30.

- Se il numero e' multiplo di 3 deve stampare "Vostro Nome";
- Se multiplo di 5 deve stampare "Cognome";
- Se multiplo di 3 e 5 (15) deve stampare "Nome e Cognome";

Capitolo 7: Funzioni

Le funzioni sono blocchi di codice riutilizzabili che eseguono compiti specifici.

Dichiarazione e Invocazione

```
// Dichiarazione
function saluta() {
    console.log("Ciao a tutti!");
}

// Invocazione
saluta(); // "Ciao a tutti!"

// Funzione con parametri
function salutaPersona(nome) {
    console.log("Ciao " + nome + "!");
}
```

```
salutaPersona("Marco"); // "Ciao Marco!"

// Funzione con return
function somma(a, b) {
    return a + b;
}

let risultato = somma(5, 3);
console.log(risultato); // 8
```

Scope delle Variabili

```
let globale = "Visibile ovunque";

function esempioDiScope() {
    let locale = "Visibile solo qui dentro";
    console.log(globale); // Funziona
    console.log(locale);  // Funziona
}

esempioDiScope();
// console.log(locale); // ERRORE: locale non è definita qui
```

Esercizi

Esercizio 7.1: Crea una funzione `calcolatrice` che prenda tre parametri (numero1, numero2, operazione) e restituisca il risultato. L'operazione può essere "+", "-", "*", "/".

Esercizio 7.2: Utilizzando un approccio funzionale (piccole funzioni che risolvono un piccolo problema), riscrivere il programma fatto in precedenza che stampi in console tutti i numeri da 1 a 30. (usare 2 funzioni massimo)

- Se il numero e' multiplo di 3 deve stampare “Vostro Nome”;
- Se multiplo di 5 deve stampare “Cognome”;
- Se multiplo di 3 e 5 (15) deve stampare “Nome e Cognome”;

Esercizio 7.3: Scrivere un programma funzionale che, dato un numero in input (max), stampi a video:

- PRIMA tutti i numeri Dispari
- DOPO tutti i numeri PARI

Capitolo 8: Array e Metodi

Gli array sono strutture dati che contengono più valori in una singola variabile.

Creazione e Accesso

```
// Creazione
let frutti = ["mela", "banana", "arancia"];
let numeri = [1, 2, 3, 4, 5];//omogenei
let misto = ["testo", 42, true, null];//Disomogenei

// Accesso agli elementi (indice inizia da 0)
console.log(frutti[0]);    // "mela"
console.log(frutti[1]);    // "banana"
console.log(frutti.length); // 3

// Modifica elementi
frutti[0] = "pera";
console.log(frutti); // ["pera", "banana", "arancia"]
```

Metodi degli Array

```
let animali = ["gatto", "cane"];

// Aggiungere elementi
animali.push("pesce");      // Aggiunge alla fine
animali.unshift("uccello"); // Aggiunge all'inizio
console.log(animali); // ["uccello", "gatto", "cane", "pesce"]

// Rimuovere elementi
animali.pop();    // Rimuove l'ultimo
```

```
animali.shift();  // Rimuove il primo
console.log(animali); // ["gatto", "cane"]

// Altri metodi utili
console.log(animali.indexOf("gatto")); // 0
console.log(animali.includes("cane")); // true
console.log(animali.sort())// ordinamento
console.log(animali.reverse())// ordinamento
```

Iterazione sugli Array

```
let colori = ["rosso", "verde", "blu"];

// Con for classico
for (let i = 0; i < colori.length; i++) {
    console.log(colori[i]);
}

// Con forEach
colori.forEach(function(colore, indice) {
    console.log(`${indice}: ${colore}`);
});
```

Esercizi

Esercizio 8.1: Dato il seguente array mischiato e confuso:

Un Anello per domarli, un Anello per trovarli, un Anello per ghermirli e nel buio incatenarli.

```
let array_1 = [
  ['un', 'per', 'incatenarli.'],
  ['Anello', 'trovarli,'],
  ['ghermirli', 'e'],
  ['gondor', 'mark'],
];
let array_2 = [
  [['trovarli,']],
  ['tu,', 'sciocchi'],
```

```
  ['tu,', 'sciocchi', ['padron', 'Sauron']],
  ['nel', ['fuggite', 'gandalf']],
  [['domarli,', 'passare'], 'buio']
];
console.log(
    array_1[0][0],
  array_1[1][0],
  array_1[0][1],
  array_2[3][0][0],
  array_1[0][0],
  array_1[1][0],
  array_1[0][1],
  array_2[0][0][0],
  array_1[0][0],
  array_1[1][0],
  array_1[0][1],
  array_1[2][0],
  array_1[2][1],
  array_2[2][0],
  array_2[3][1],
  array_1[0][2],
)
```

Esercizio 8.2: Crea un array con i nomi di 5 tuoi amici. Scrivi funzioni per: aggiungere un amico, rimuovere un amico dalla testa, cercare se un amico è nella lista, mostrare tutti gli amici in ordine alfabetico.

Esercizio 8.3: Dato un array di numeri [1, 15, 3, 8, 12, 9, 7, 20], usa i metodi degli array per: trovare e contaretutti i numeri maggiori di 10, calcolare la somma di tutti i numeri, creare un nuovo array con tutti i numeri moltiplicati per 2.

Esercizio 8.4: Scrivere un programma che, data una stringa in input, conti quante vocali sono state inserite

```
//Esempio

Aiuole = Nella parola "Aiuole" ci sono 5 vocali
Alba = Nella parola "Alba" ci sono 2 vocali
Fgrty = Nella parola "Fgrty" ci sono 0 vocali
```

Capitolo 9: Oggetti

Gli oggetti sono strutture dati che raggruppano proprietà e metodi correlati.

Creazione e Accesso

```
// Creazione di un oggetto
let persona = {
    nome: "Mario",
    cognome: "Rossi",
    eta: 30,
    citta: "Roma",
    sposato: false
};

// Accesso alle proprietà
console.log(persona.nome);          // "Mario" (dot notation)
console.log(persona["cognome"]);    // "Rossi" (bracket notation)

// Modifica proprietà
persona.eta = 31;
persona["citta"] = "Milano";

// Aggiunta nuove proprietà
persona.professione = "Programmatore";
```

Metodi negli Oggetti

```
let automobile = {
    marca: "Toyota",
    modello: "Corolla",
    anno: 2020,
    accesa: false,

    // Metodi
    accendi: function() {
        this.accesa = true;
        console.log("Auto accesa!");
    },
```

```
    spegni: function() {
        this.accesa = false;
        console.log("Auto spenta!");
    },

    info: function() {
        return this.marca+' '+this.modello+' '+ del +' '+this.anno;
    }
};

// Utilizzo dei metodi
automobile.accendi();      // "Auto accesa!"
console.log(automobile.info()); // "Toyota Corolla del 2020"
```

Costruttori di Oggetti

I costruttori sono funzioni speciali che permettono di creare più istanze di oggetti con la stessa struttura.

Funzioni Costruttore (ES5)

Acronimo di ECMAScript 5, è la quinta edizione dello standard JavaScript, pubblicata nel 2009

```
// Definizione del costruttore
function Persona(nome, cognome, eta, citta) {
    this.nome = nome;
    this.cognome = cognome;
    this.eta = eta;
    this.citta = citta;
    this.sposato = false;

    this.presentati = function() {
        return 'Ciao, sono '+this.nome + this.cognome+', ho '+this.eta+'anni e
vivo a '+this.citta';
    };
```

```
}

// Creazione di oggetti usando il costruttore
let persona1 = new Persona("Mario", "Rossi", 30, "Roma");
let persona2 = new Persona("Lucia", "Bianchi", 25, "Milano");
```

```
// VERSIONE MODERNA CON CLASSI (ES6+)
// Stesso concetto, sintassi più pulita
class PersonaModerna {
    constructor(nome, cognome, eta, citta) {
        this.nome = nome;
        this.cognome = cognome;
        this.eta = eta;
        this.citta = citta;
        this.sposato = false;
    }

    presentati() {
            return 'Ciao, sono '+this.nome + this.cognome+', ho '+this.eta+'anni e
    }


}

let persona1 = new Persona("Mario", "Rossi", 30, "Roma");
let persona2 = new Persona("Lucia", "Bianchi", 25, "Milano");
```

Esercizi

Esercizio 9.1: Crea 1 oggetto "libro" con proprietà (titolo, autore, pagine, letto) e metodi per: mostrare info del libro, segnare come letto/non letto, calcolare tempo di lettura stimato (assumendo 1 pagina = 2 minuti)

Esercizio 9.2

Crea una classe "ContoBancario" con un Costruttore e istanzia 3 conti:

- Proprietà: numero conto, intestatario, saldo
- Metodi: deposita, preleva, visualizzaSaldo, calcolaInteressi (1% annuo)

- Crea almeno 3 conti diversi e simula alcune operazioni