

Caratteristiche di JavaScript

Indice

1. Introduzione a JavaScript
2. Variabili e tipi di dati
3. Operatori
4. Funzioni
5. Oggetti e Array
6. Operatore Spread e Rest
7. Async/Await e Promises
8. Classi e Prototipi
9. Moduli
10. Conclusioni

1. Introduzione a JavaScript

- Linguaggio di scripting interpretato, orientato agli oggetti.
- **Single-threaded** e **asincrono**.
- Supporta la programmazione **event-driven** e **funzionale**.
- Eseguito nel browser e su server (Node.js).

2. Variabili e tipi di dati

Dichiarazione:

- `var` (scope globale/funzione, hoisting)
- `let` (scope di blocco, riassegnabile)
- `const` (scope di blocco, costante)

Tipi di dati primitivi:

- `String`, `Number`, `Boolean`, `Null`, `Undefined`, `Symbol`, `BigInt`

3. Operatori

```
let x = 10 + 5; // +, -, *, /, %, **
```

```
"5" == 5; // true (coercizione)  
"5" === 5; // false (strict equality)
```

```
true && false; // AND  
true || false; // OR  
!true; // NOT
```

4. Funzioni

Dichiarazione:

```
function somma(a, b) {  
    return a + b;  
}
```

Funzioni

Arrow function (ES6+):

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

Funzioni

Parametri predefiniti:

```
function greet(name = "Guest") {  
  return `Hello, ${name}!`;  
}
```


5. Oggetti e Array

Oggetti:

```
const persona = {  
  nome: "Mario",  
  saluta() {  
    console.log(`Ciao, sono ${this.nome}`);  
  }  
};
```

Oggetti e Array

Array:

```
const numeri = [1, 2, 3];  
numeri.push(4); // Aggiunge un elemento
```

6. Operatore Spread

Spread (`...`):

```
const arr1 = [1, 2];  
const arr2 = [...arr1, 3]; // [1, 2, 3]  
  
const obj1 = { a: 1 };  
const obj2 = { ...obj1, b: 2 }; // { a: 1, b: 2 }
```

Rest parameters:

```
function logArgs(...args) {  
  console.log(args); // Array di tutti gli argomenti  
}
```

7. Async/Await e Promises

Promises:

```
fetch("https://api.example.com")  
  .then(response => response.json())  
  .then(data => console.log(data));
```

Async/Await:

```
async function getData() {  
  const response = await fetch("https://api.example.com");  
  const data = await response.json();  
  return data;  
}
```

8. Classi e Prototipi

Classi (ES6):

```
class Persona {  
  constructor(nome) {  
    this.nome = nome;  
  }  
  
  saluta() {  
    console.log(`Ciao, sono ${this.nome}`);  
  }  
}
```

Ereditarietà:

```
class Studente extends Persona {  
    constructor(nome, corso) {  
        super(nome);  
        this.corso = corso;  
    }  
}
```

9. Moduli (ES6)

Export:

```
// module.js  
export const pi = 3.14;  
export function raddoppia(x) { return x * 2; }
```

Import:

```
import { pi, raddoppia } from './module.js';  
console.log(raddoppia(pi)); // 6.28
```


10. Conclusione

- JavaScript è un linguaggio versatile e potente.
- Continua a evolversi con nuove funzionalità (ES6+).
- Fondamentale per lo sviluppo web moderno.

Grazie!

