

1. Cos'è Java?

Java è un linguaggio di programmazione ad alto livello, orientato agli oggetti e multiplatforma, famoso per il suo principio "scrivi una volta, esegui ovunque". È usato principalmente per lo sviluppo di applicazioni aziendali, Android, software per desktop e applicazioni server-side.

2. Ambiente di Lavoro

Per iniziare a programmare in Java, è necessario installare:

- **JDK (Java Development Kit):** Include gli strumenti necessari per compilare ed eseguire codice Java.
- **IDE (Integrated Development Environment):** Consigliati IntelliJ IDEA, Eclipse o NetBeans.

3. Il Primo Programma in Java

Scrivi il tuo primo programma in un file chiamato `HelloWorld.java`:

java

Copia codice

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

Compila ed esegui il programma:

bash

Copia codice

```
javac HelloWorld.java  
java HelloWorld
```

4. Sintassi di Base

Variabili e Tipi di Dati

Java è un linguaggio tipizzato staticamente:

java

Copia codice

```
int numero = 10;      // Intero
double decimale = 5.5; // Decimale
String testo = "Ciao"; // Stringa
boolean flag = true;  // Booleano
```

Condizioni

java

Copia codice

```
int numero = 10;
if (numero > 5) {
    System.out.println("Numero maggiore di 5");
} else {
    System.out.println("Numero minore o uguale a 5");
}
```

Cicli

For:

java

Copia codice

```
for (int i = 0; i < 5; i++) {
    System.out.println(i);
}
```

-

While:

java

Copia codice

```
int i = 0;
while (i < 5) {
    System.out.println(i);
    i++;
}
```

-

5. Classi e Oggetti

Java è orientato agli oggetti, quindi tutto si basa su classi e oggetti.

Definire una Classe

java

Copia codice

```
public class Studente {
    String nome;
    int età;

    public Studente(String nome, int età) {
        this.nome = nome;
        this.età = età;
    }

    public void saluta() {
        System.out.println("Ciao, sono " + nome);
    }
}
```

Creare un Oggetto

java

Copia codice

```
Studente studente = new Studente("Marco", 20);
studente.saluta();
```

6. Ereditarietà e Polimorfismo

L'ereditarietà permette di creare nuove classi a partire da classi esistenti.

java

Copia codice

```
public class Studente extends Persona {
    private String corso;

    public Studente(String nome, int età, String corso) {
        super(nome, età);
        this.corso = corso;
    }

    @Override
    public void saluta() {
        System.out.println("Ciao, sono uno studente del corso " +
        corso);
    }
}
```

}

Guida alla Programmazione in JavaScript

1. Cos'è JavaScript?

JavaScript è un linguaggio di scripting interpretato, eseguito principalmente nei browser web per aggiungere interattività alle pagine web. Oggi, grazie a **Node.js**, è usato anche per lo sviluppo di applicazioni server-side.

2. Ambiente di Lavoro

JavaScript può essere eseguito direttamente nel browser o tramite Node.js per esecuzioni lato server. Gli editor consigliati sono **Visual Studio Code** o **Atom**.

3. Il Primo Programma in JavaScript

Crea un file `index.html` e aggiungi il seguente codice JavaScript inline:

```
html
Copia codice
<!DOCTYPE html>
<html lang="it">
<head>
  <meta charset="UTF-8">
  <title>Hello World</title>
</head>
<body>
  <script>
    console.log("Hello, World!");
  </script>
</body>
</html>
```

Oppure crea un file JavaScript esterno `app.js` e includilo in `index.html`:

```
html
```

Copia codice

```
<script src="app.js"></script>
```

4. Sintassi di Base

Variabili e Tipi di Dati

JavaScript è un linguaggio dinamico e supporta `let`, `const`, e `var` per dichiarare variabili.

javascript

Copia codice

```
let numero = 10;           // Numero
let decimale = 5.5;       // Float
let testo = "Ciao";      // Stringa
let flag = true;         // Booleano
```

Condizioni

javascript

Copia codice

```
let numero = 10;
if (numero > 5) {
    console.log("Numero maggiore di 5");
} else {
    console.log("Numero minore o uguale a 5");
}
```

Cicli

For:

javascript

Copia codice

```
for (let i = 0; i < 5; i++) {
    console.log(i);
}
```

-

While:

javascript

Copia codice

```
let i = 0;
while (i < 5) {
```

```
    console.log(i);
    i++;
}
```

-

5. Funzioni

JavaScript supporta funzioni tradizionali e funzioni freccia (arrow functions).

javascript

Copia codice

```
function saluto(nome) {
    return "Ciao, " + nome;
}
```

```
console.log(saluto("Marco"));
```

```
// Arrow Function
```

```
const saluto = (nome) => `Ciao, ${nome}`;
```

6. Array e Oggetti

Array

javascript

Copia codice

```
let numeri = [1, 2, 3, 4, 5];
console.log(numeri[0]); // Accesso al primo elemento
numeri.push(6); // Aggiunge un elemento
numeri.forEach(n => console.log(n));
```

Oggetto

javascript

Copia codice

```
let studente = {
    nome: "Alice",
    età: 25
};
console.log(studente.nome);
```

7. Classi e Oggetti

JavaScript supporta le classi (a partire da ES6):

javascript

Copia codice

```
class Studente {
    constructor(nome, età) {
        this.nome = nome;
        this.età = età;
    }

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

```
let studente = new Studente("Marco", 20);
studente.saluta();
```

8. Manipolazione del DOM

In JavaScript, puoi manipolare il DOM (Document Object Model) per aggiornare il contenuto della pagina web.

javascript

Copia codice

```
document.getElementById("id-elemento").innerText = "Nuovo Testo";
```

9. Eventi

JavaScript gestisce vari eventi come click, mouseover, ecc.

javascript

Copia codice

```
document.getElementById("pulsante").addEventListener("click",
function() {
    alert("Hai cliccato il pulsante!");
});
```

10. Asincronia: Callbacks, Promises e Async/Await

JavaScript è noto per la sua natura asincrona. Le promesse e le funzioni `async/await` semplificano la gestione delle operazioni asincrone.

Promessa

javascript

Copia codice

```
let promessa = new Promise((resolve, reject) => {
    setTimeout(() => resolve("Operazione completata!"), 1000);
});

promessa.then(result => console.log(result));
```

Async/Await

javascript

Copia codice

```
async function caricaDati() {
    let dati = await fetch("url-dati");
    let json = await dati.json();
    console.log(json);
}
```

11. Conclusione

Java e JavaScript sono potenti linguaggi, ciascuno con i propri punti di forza e ambiti di utilizzo. Mentre **Java** è ideale per applicazioni server-side e Android, **JavaScript** domina nel mondo del web, rendendo le pagine dinamiche e interattive.