

# Visual Studio Code come IDE

Visual Studio Code (VSC) è un IDE per quasi tutti i linguaggi di programmazione; è disponibile su ogni sistema operativo.

`Ctrl+Shift+P` apre la command palette, ossia una ricerca di tutti i possibili comandi, ed è l'unica vera shortcut da conoscere relativa a VSC; le altre shortcut sono simili a quelle di Visual Studio.

## Utilizzare VSC per il debugging

VSC essendo un IDE moderno possiede un debugger: uno strumento che permette di controllare cosa il programma (dal punto di vista della gestione delle variabili) sta facendo riga per riga.

Compilare in modalità di debug significa che vengono mandate al compilatore (nel nostro caso gcc) alcune flag che compilano il programma senza ottimizzazioni; a questo punto il debugger può utilizzare il sorgente così compilato per effettuare le vere e proprie operazioni di debug.

Nei casi di linguaggi compilati in un linguaggio intermedio (come l'esempio di Java) vengono anche inclusi nel compilato in modalità di debug alcune informazioni utili al debugger al fine di facilitargli il lavoro; quest'ultimo può usarle o meno in base alla situazione.

Il debugger, quando esegue il programma, permette di impostare alcuni punti chiamati **breakpoints** che permettono di bloccare l'esecuzione del codice in quella specifica riga. In alcuni casi i breakpoint possono essere condizionali, ossia che si attivano solo dopo certe condizioni.

Si possono anche controllare e modificare le variabili, lo stacktrace in utilizzo e calcolare il valore di espressioni arbitrarie.

## Build Systems & Gradle

Costruire software non è solo programmare, ma anche verificare la qualità del codice, compilarlo, testarlo e verificare se supera test sempre più difficili. Questo se eseguito a mano richiederebbe molto tempo e dunque si fa fare in automatico ad un **build automator**.

Utilizzeremo **Gradle** come build automator.

Solitamente i build automator sono configurati di default in modo pressoché ottimo; questo perché si preferisce la convenzione piuttosto che la configurazione.

In Gradle, il progetto è una directory che contiene un file speciale, `build.gradle.kts` ; nel nostro caso dobbiamo scrivere dentro:

```
plugins {  
    java  
}
```

Utilizzeremo un solo **plugin** utile per lo sviluppo in Java.

Dopo aver posto il file in `src/main/java` possiamo usare `gradle javaBuild` .

Tramite `gradle clean` si eliminerà la cartella `build` che gradle ha compilato in precedenza.