

Introduzione al linguaggio c++

Edoardo Lo Iacono

Indice

1	Perché studiamo c++?	3
2	Installazione su visual studio code	3
3	Sintassi di base	3
3.1	std::cout	4

1 Perché studiamo c++?

C++ è un linguaggio di programmazione creato da *Bjarne Stroustrup*, entrato in vigore nel 1985. Nasce come estensione del linguaggio c e con il tempo si espande notevolmente rispetto ad esso. Nonostante sia un po' datato è uno dei linguaggi di programmazione più completi ed utilizzati ancora al giorno d'oggi, senza saperlo lo troviamo nella maggior parte delle applicazioni che usiamo quotidianamente; sono infatti sviluppati (completamente o in parte) in c++ applicazioni come:

- I principali web browser (Chrome, Firefox, Edge)
- I programmi del pacchetto office
- Photoshop
- Diversi videogiochi con il motore grafico unreal engine

2 Installazione su visual studio code

Per lavorare con c++ possiamo usare diversi IDE, alcuni sono dedicati ai linguaggi c come per esempio codeblocks, nulla vieta di usare questo o altri ambienti di sviluppo. Vediamo però di seguito come fare programmi in c++ tramite visual studio code: IDE molto versatile e professionale grazie alla vastità di estensioni disponibili.

1. Installare visual studio code al link: [vscode](#)
2. Scaricare il compilatore seguendo le istruzioni al seguente link: [link](#)
3. Una volta installato aprire a sinistra la barra delle estensioni
4. Cercare nella barra di navigazione "c++" ed installare le estensioni "c++" e "c++ Extension Pack" le quali ci permetteranno non solo di eseguire il nostro codice ma anche di vedere a schermo eventuali errori e, come negli altri linguaggi un color coding che ci permetta di capire l'entità di tutti i nostri elementi.
5. Cercare nella barra di navigazione "code runner" e scaricarlo

3 Sintassi di base

Prendiamo in esame il più basilare dei codici in c++ per iniziare a parlare della sintassi di base.

```
1 #include <iostream>
2
3 int main() {
4     std::cout << "Hello World!";
5     return 0;
6 }
```

Lasciamo per un momento da parte la riga `#include <iostream>`, che approfondiremo meglio in seguito parlando di librerie.

La funzione `main()` viene comunemente chiamata **entry point**, e racchiude le prime istruzioni che verranno eseguite quando viene fatto partire il programma, è importante specificare il fatto che sia la prima in quanto potremmo avere (come vedremo meglio in seguito) diverse funzioni nel nostro programma, il `main` viene sempre eseguiti prima di ogni altra funzione!

3.1 `std::cout`

Iniziamo a parlare del codice vero e proprio, il primo comando che vediamo è **`cout`**. `cout` è un oggetto che fa parte della libreria standard (`std`), per questo andiamo a indicarla, ci permette di mandare a schermo un testo. Lo "schermo" su cui viene stampato il nostro testo prende il nome di *console*, infatti `cout` non è altro che l'acronimo per Console Out.

Nulla di trascendentale come implicazione logica ma si noti bene una cosa che tornerà nella maggior parte dei comandi che vedremo, quando il codice *esegue* in maniera pragmatica qualcosa (manda a schermo, salva in memoria) è necessario terminare il comando con `;`, questa è una regola sempreverde nei linguaggi di programmazione basati su `c` ma non in tutti i linguaggi, il `;` serve al compilatore per capire che quell'operazione è finita e passare alla prossima.