Il mito del linguaggio definitivo

I CONSIGLI DELLO YOUTUBER

DI MARCO AGNESE

15/01/2026

ALGORITMI E PROGRAMMAZIONE | PRIMO BIENNIO

LICEO, TECNICO



"Ma perché nel 2025 dobbiamo ancora sudare su C++ quando esistono Python e persino Rust?"
È una domanda legittima, e se te la sei fatta non sei solo. Molti studenti la pensano così, soprattutto quando vedono quanto è rapido ottenere risultati con linguaggi moderni e ad alto livello.

Nel video, Marco - studente di ingegneria informatica - prova a dare una risposta chiara: **non esiste il linguaggio definitivo**, ma esistono strumenti diversi per scopi diversi. E C++ continua a essere uno di quelli fondamentali.

Python vs C++: comodità contro controllo

Il confronto più comune è quello con Python.
Python è come una city-car moderna: cambio automatico, sensori ovunque, mille accessori pronti all'uso. Apri l'editor, importi una libreria e in poche righe ottieni un risultato concreto, come un grafico o un'analisi dei dati.

C++ è l'opposto: una monoposto da Formula 1. Non ti aiuta se sbagli, non ti perdona le distrazioni. Ma ti dà accesso diretto a tutto ciò che conta davvero:
memoria, **CPU**, **cache**, **strutture dati**. Sei tu a decidere quando allocare e liberare memoria, che tipo di struttura usare, come ottimizzare le prestazioni.

Un'altra metafora efficace è quella del vestito: Python è un capo "taglia M" che sta bene un po' a tutti. C++ è un abito su misura: costa fatica, ma una volta finito è più leggero, più efficiente e calza alla perfezione.

Ma allora perché non passare tutti a Rust?

Qui entra in gioco Rust, uno dei linguaggi più discussi degli ultimi anni.
Rust nasce proprio per risolvere molti problemi storici di C e C++: riduce drasticamente gli errori legati alla gestione della memoria, evita i data race e garantisce alte prestazioni con maggiore sicurezza.

Il punto è che il **mondo reale non riparte da zero**.
Sistemi operativi, database, firmware, motori grafici e una quantità enorme di software critico sono stati scritti in C e C++ nel corso di decenni. Riscriverli tutti sarebbe come cambiare, in una sola notte, tutti i binari ferroviari del pianeta: teoricamente possibile, praticamente impossibile.

Per questo C++ continua a essere fondamentale, anche in un ecosistema dove Rust cresce rapidamente.

Il vero valore di C++

Studiare C e C++ non è nostalgia accademica. È un allenamento mentale.
Significa imparare come ragiona davvero la macchina: come gestisce la memoria, come esegue le istruzioni, dove nascono i colli di bottiglia nelle prestazioni.

Chi conosce C++ diventa, di fatto, **bilingue**: sa muoversi tra alto e basso livello. E questo rende più semplice anche imparare linguaggi moderni come Rust, perché la logica

sottostante è già chiara.

In conclusione

Non esiste il linguaggio perfetto che sostituisce tutti gli altri.
Python, Rust e C++ sono strumenti diversi, ognuno con un ruolo preciso. Ma il low-level resta una palestra insostituibile per allenare il pensiero su performance, memoria e architettura del software.

Ecco perché nel 2025 si studia ancora C++. Non perché sia “vecchio”, ma perché insegna cose che nessun linguaggio moderno può nascondere.

Il video

Il mito del linguaggio definitivo



Autori



MARCO AGNESE

Studia Ingegneria Informatica e ha un canale YouTube molto seguito. Nei suoi video affronta temi relativi alla tecnologia e alla vita da studente universitario.