**linguaggio interpretato e linguaggio compilato**

1. **In generale**

il linguaggio può essere sia interpretato che compilato, indipendentemente dalle implementazioni del linguaggio; infatti, risulta scorretto dire che un linguaggio è esclusivamente interpretato o compilato, la cosa importante che determina se un linguaggio è interpretato o compilato sono le implementazioni utilizzate.

1. **Cos’è esattamente una compilazione**

La compilazione ha dei passaggi molto precisi di comunicazione che noi possiamo semplificare in dei passaggi fra 5 interlocutori.

Facciamo un’ipotesi in cui cerco eseguire un linguaggio come Java, per creare un eseguibile (file.exe), avrò bisogno di un compilatore che si occuperà di tutto il progetto: leggerà tutti i file, cercando di compilarli, non ci riuscirà nel momento in cui ci sia un errore nel codice.

Nel caso in cui non ci siano errori, viene costruito un eseguibile, conterrà in sostanza linguaggio macchina (un insieme di codice binario), creato appositamente per la mia architettura: un insieme di due elementi, un comunicatore detto Sistema Operativo che si occupa di, appunto, comunicare con un processore.

**PRO**

È difficile ritornare al codice di partenza avendo un solo eseguibile.

Sarà ottimizzato al meglio per il mio sistema operativo.

**CONTRO**

Avrò bisogno di un compilatore diverso per ottenere un eseguibile diverso per ottenere uno stesso risultato su architetture diverse.

Il codice dovrà essere aggiornato ogni volta che l’architettura viene aggiornata o avrò bisogno di implementare delle funzioni.

**Linguaggio Compilato**

c++, Java, c

1. **Cosa vuol dire interpretazione?**

Il linguaggio interpretato funziona in maniera simile ma sensibilmente diversa, l’interprete al contrario del compilatore, non leggerà tutto il progetto ma verrà letto riga per riga, nel momento in cui incontrerà un errore allora smetterà di eseguire il codice. Il codice verrà sempre tradotto in linguaggio macchina per comunicare con l’architettura.

Nel linguaggio del web, l’eseguibile verrà spedito tramite il web per essere eseguito sulla macchina ogni singola volta che io vorrò accedere al URL.

PRO

Non ci sarà bisogno di aggiornare eventuali clienti della versione aggiornata del programma poiché verrà inviata ogni volta nella versione che io ho implementato.

Qualcuno al posto nostro si è impegnato a interpretare il codice al meglio per ogni architettura immaginabile.

CONTRO

Non potrà mai avere la certezza che il codice funzionerà al 100%, dovrà testare il codice frequentemente su ogni dispositivo e Browser differente.

Il codice potrebbe avere degli errori non facilmente individuabili, bisogna testare ogni singola possibilità del codice.

Es: in una pagina in cui viene richiesta la cittadinanza, io dimentico di implantare gli americani fino a quando un americano non proverà il mio codice per me (non deve succedere).

**Linguaggio Interpretato**

PHP, Ruby, Python, e JavaScript