LINGUAGGIO COMPILATO E INTERPRETATO DIFFERENZE IMP.

-**un compilatore** converte l'intero codice sorgente in codice oggetto, e lo salva come file prima di eseguirlo, tra i linguaggi più noti di questo tipo abbiamo ( C, C++, COBOL, Fortran).

Come funziona un compilatore? :

* **Creazione del codice sorgente**: il codice sorgente è un pezzo di codice composto in un editor di testo e l'estensione del file per il codice sorgente.
* **Pre-elaborazione**: questo codice sorgente viene inizialmente trasmesso al preprocessore, che lo espande. Il codice ingrandito verrà fornito al compilatore dopo l'espansione.
* **Compilazione**: il codice espanso dal preprocessore viene passato al compilatore, che lo converte in codice assembly.
* **Conversione in codice oggetto** da parte di un assembler: utilizzando un assembler, il codice assembly viene trasformato in codice oggetto. Il file oggetto creato da un assembler ha lo stesso nome del file di origine.
* **Collegamento**: nel flusso di lavoro del compilatore, la funzione principale di un linker consiste nel connettere il codice oggetto dei dati della libreria di codifica con il codice oggetto di un programma.
* **Esecuzione**: il file eseguibile è il prodotto finale del linker.

-**un interprete** trasforma ed esegue il codice sorgente riga per riga, senza salvarlo e segnala gli errori lungo

il percorso. tra i linguaggi più noti di questo tipo abbiamo (Python, JavaScript, Perl e BASIC )

Come funziona un interprete? :

* **Creazione del codice sorgente**: questo passaggio della funzionalità è lo stesso di un compilatore. Durante il runtime, però, l'interprete trasforma il codice sorgente una riga alla volta.
* **Interpretazione diretta**: un interprete traduce un programma linguistico di alto livello in un linguaggio a livello di macchina.
* **Modifica del codice sorgente**: l'interprete consente la valutazione e la modifica del programma durante l'esecuzione in una finestra affiancata.
* **Esecuzione**:  l'esecuzione del programma è moderatamente lenta poiché tutti i collegamenti vengono eseguiti in fase di esecuzione senza un linker separato.

LIVELLI DEI LINGUAGGI DI PROGRAMMAZIONE

-**LINGUAGGI MACCHINA**: la lingua + vicina al computer stesso, un programma in linguaggio macchina è **costituito da una serie di modelli binari** (EX: 01011100), che rappresentano semplici operazioni che possono essere eseguite dal computer.

I programmi in linguaggio macchina sono **eseguibili***,* significa che possono essere eseguiti direttamente.

-**LINGUAGGI ASSEMBLY**: rendono la programmazione +facile per l'uomo. **Le istruzioni in linguaggio macchina vengono sostituite da semplici abbreviazioni  (EX: ADD, MOV),** prima dell'esecuzione, un programma in linguaggio assembly richiede la **traduzione in linguaggio macchina**.

traduzione viene eseguita da un programma per computer noto come **Assembler**.

-**LINGUAGGI DI ALTO LIVELLO**: I linguaggi di alto livello (EX: **C C++, JAVA…)** sono più simili all'inglese, questo rende più facile ai programmatori pensare nel linguaggio di programmazione. Anch’essi, richiedono anche la traduzione in linguaggio macchina prima dell'esecuzione.

Traduzione viene eseguita da un c**ompilatore** o da un **interprete**.