LISTA DEI LINGUAGGI DI PROGRAMMAZIONE

Molte delle informazioni che vedremo riguardante JavaScript, sono strettamente legate al mondo dei linguaggi di programmazione HTML e CSS. Immaginiamo come se fosse una **TORTA** composta da 3 strati.

-**JavaScript** 🡪 si occupa della logica che permette di compiere delle azioni dinamiche, può inviare delle richieste e rimandare indietro…

un elemento di JavaScrpit vive all’interno di uno <script> all’interno della struttura dell’HTML.

-**ECMAScript** 🡪 rappresenta lo standard su cui si basa JavaScript, e le diverse versioni aggiungono nuove funzionalità (mantenuto dall'organizzazione **Ecma International**).

-**TypeScript** 🡪 aggiunge alcune funzionalità in più a JavaScript, in particolare il supporto per il typing statico. (serve principalmente per REACT e ANGULAR).

INTRODUZIONE AL LINGUAGGIO DI PROGRAMM. JAVASCRIPT

si occupa della logica che permette di compiere delle azioni dinamiche , è un **linguaggio versatile**, utilizzato per sviluppare il lato **client** (nel browser)

JavaScript è responsabile per il comportamento nel sito! ci permette di implementatare delle funzionalità complesse nelle nostre pagine, tra le principali abbiamo:

* aggiornare il conenuto del sito con nuovi contenuti.
* disegni, animazioni, gallerie di immagini…
* tenere traccia degli utenti tramite i **COOKIE.**
* Elementi intereattivi (schede, cursori..)

INTRODUZIONE A NODE.JS

Node.js è un ambiente di **runtime open-source** JavaScript basato sul motore di Google Chrome (V8)

consente di eseguire codice JavaScript **lato server**, consentendo agli sviluppatori di creare applicazioni web scalabili, veloci ed ad alte prestazioni; inoltre semplifica lo sviluppo e la manutenzione del codice.

Node.js utilizza un modello basato sugli eventi che lo rende possibile Leggero ed efficiente, perfetto per applicazioni in tempo reale ad alta intensità di dati che vengono eseguiti su dispositivi distribuiti.

LINGUAGGIO COMPILATO E INTERPRETATO

il linguaggio compilato ed interpretato sono caratterizzati e dipendono da dei processi di implementazione che contano molto di + delle proprietà del linguaggio di programmazione.

dicendo questo, QUALSIASI linguaggio potrebbe essere COMPILATO oppure INTERPRETATO a seconda dell’implementazione utilizzata.

- **LINGUAGGIO COMPILATO** 🡪 cosa è la COMPILAZIONE? (C, C++, Pascal…)

Il compilatore converte il programma direttamente in **codice macchina**,

riferendosi al codice progettato per un determinato processore e OS,

successivamente **il computer indipendetemente** **esegue il codice macchina.**

comporta l’effetto negativo, che la compilazione sia lenta.

****VANTAGGI:

* Esecuzione rapida.
* Ottimizzato per l’hardware di destinazione.

SVANTAGGI:

* Richiede un compilatore.
* Modifiche e Distribuzione del codice sono + lenti dell’interprete.

- **LINGUAGGIO INTERPRETATO** 🡪 cosa è l’INTERPRETAZIONE? (Python, PHP, Ruby, JavaScript…)

qui il codice sorgente NON VIENE eseguito direttamente dalla macchina,

c’è la presenza di un altro programma (**interprete**) che lo legge ed esegue

l’interprete è personalizzato per la macchina nativa.

VANTAGGI:

* Facile da imparare, usare.
* Consente di eseguire operazioni + complesse in pochi passaggi.
* Permettono l'aggiunta di attività dinamiche e interattive alle web.
* Modifca e Esecuzione del codice è veloce.

SVANTAGGI:

* Di solito funziona abbastanza lentamente.
* Comandi limitati per eseguire operazioni dettagliate sulla grafica.
* Accesso limitato al codice di basso livello e di ottimizzazione della velocità.
* Accesso limitato al dispositivo