

Pros e Contras das linguagens:

C/C++:

Prós:

- \*Linguagem Compilada ---> Extremamente Rápida (Próxima do que o computador entende)
- \*Linguagem Compilada ---> O compilador, o programa que faz a tradução entre o que a gente entende e escreve e o que a máquina entende(binário), ajuda com erros e sugestões
- \*Linguagem Compilada ---> Nada compila se não tiver uma sintaxe 100% correta
- \*Linguagem que suporta objetos ---> Permite programar da forma mais poderosa e desde os anos 80 definida como a filosofia de programação mais completa e eficiente contra erros
- \*Linguagem antiga ---> Muita documentação na internet a explicar como fazer as coisas
- \*Bibliotecas de Sistema ---> Todo o software que precise de trabalhar com os recursos I/O do computador eficientemente e de forma complexa está praticamente sequestrada ao C++

Contras:

- \*Linguagem Compilada ---> Exige ter 100% sintaticamente correto para correr.
- \*Linguagem Compilada ---> Exige a instalação de um programa, o compilador para funcionar.
- \*Linguagem Compilada ---> Exige o processo, não automático, de correr o compilador
- \*Linguagem Compilada ---> Existem emuladores que permitem simular o processo de compilação usando ferramentas na net, mas são muito limitadas pois o compilador nao é feito para correr no browser
- \*Linguagem antiga ---> Tem sido na industria marginalizada dada a complexidade que ela pode atingir caso seja explorada ao máximo.
- \*Linguagem que suporta objetos de forma não nativa ---> C é anterior à filosofia dos objetos, c++ é um remendo para fazer de c algo que suporte objetos. Há linguagens que naturalmente logo de inicio foram feitas para objetos sem ser com remendos.

Exemplos de uso de C/C++:

- \*Sistemas operativos, o Windows, o Linux são escritos em c++.
- \*O Google Chrome é muito rápido face ao que existia pois foi escrito em c++
- \*Todos os jogos de playstation com exelente realismo são feitos quase na totalidade em c++

```
int main(){
```

```
printf("Hello world");  
}  
////////////////////////////////////
```

Python:

Prós:

\*Linguagem recente ---> A industria tem vindo a adopta-la pois ela é recente e foi feita de pensada precisamente com as necessidades atuais

\*Bibliotecas práticas e visualmente orientadas ---> Python segue as tendencias modernas, atualmente tudo tem uma interface gráfica e deixa de ter importancia a linha de comandos. Todas as Bibliotecas são muito ligadas a ter uma interface grafica sem complexidade adicional, inves de "sequestrado" a uma linha de comandos.

\*Suporta Objetos --->(Mesma coisa que c++)

\*Linguagem Interpretada ----> Muito rápida de ver o resultado automaticamente. Não obriga a invocar um compilador.

\*Linguagem Interpretada ----> Exige instalar um interpretador que é mais simples em windows que o compilador

\*Bibliotecas para representar gráficos e fazer calculo economico muito completas ----> Para os jovens da economia

\*Linguagem Interpretada ----> Mais facil de correr no browser em sites que permitem simular o interpretador de Python, mas mesmo assim não é tão facil como JavaScript

\*Linguagem Compacta ----> Geralmente tem um sintaxe mais "light", permite fazer a mesma coisa com menos codigo

\*Linguagem modernas ----> Imenso material moderno na web e facil ter cursos pq está na moda

\*Linguagem Interpretada ----> Não exige decorar determinadas convenções para funcionar

Contras:

\*Linguagem Interpretada ---> Podemos correr um programa sem estar 100% sintaticamente correto e o programa só estoura quando está a correr

\*Linguagem Interpretada ---> Mais liberdade de fazer erros

\*Linguagem Interpretada ---> Mais lenta a executar

\*Linguagem não baseada na sintaxe de C ---> Face às restantes Python não tem similitudes com a sintaxe de C. O que faz com que quem que saiba Python bem, pode ter deficuldade a

interpretar código de Java/JavaScript/C/C++ que são baseadas na sintaxe de C. No entanto ambas executam exatamente o mesmo

\*Linguagem não baseada na sintaxe de C ---> Sobretudo na parte dos objetos, faz as mesmas coisas numa filosofia ligeiramente diferente, exigindo entender o que está por detrás, mais teórico, o que vem no papel, e as diferentes linguagens põe na prática, para poder fazer a ponte entre as baseadas em C e Python

\*Linguagem não baseada na sintaxe de C ---> Python é moderno, nenhuma outra, para já, se baseia na sintaxe da mesma.

def():

```
print("Hello world")
```

def()

```
////////////////////////////////////
```

Java:

Prós:

\*Linguagem somente orientada a objetos ---> Quem sabe Java, sabe a filosofia dos objetos. Pois em Java 99% das coisas são objetos, não conseguimos fugir deles

\*Linguagem com o compilador mais completo para ajudar quem conhece ---> Ajuda-nos imenso a resolver os erros e com sugestões que C/C++ não dá tanto

\*Linguagem Compilada ---> Avisa-nos dos nossos erros e dá sugestões

\*Linguagem Mais usada ---> Apesar de estar a perder terreno, porque é dose de cavalo, é a linguagem mais usada em todo o mundo. Desde os PCs até aos cartões de crédito, ao microondas, Java é altamente provável de estar lá

Contras:

\*É dose de cavalo ---> Se quiserem ir por aqui, precisam de aulas personalizadas sobre Java e sobre objetos já invés de ser a última cena que eu vou mencionar, porque não dá para fazer nada sem ser com objetos

\*Não dá para simular no browser ---> Exige mesmo instalar coisas no PC

```
class Programa{
```

```

public static void main(){
    system.out.print("Hello World");
}

}

```

////////////////////////////////////

JavaScript:

Prós:

- \*Não exige instalar nada ---> já temos o browser que é o interpretador de JavaScript
- \*Moderna ---> JavaScript está a tornar-se imensamente importante
- \*Super Compacta e Straightfoward ---> É feita para trabalhar nos browsers para ser rapida como nos queremos os sites, rápidos, feita para ser compacta e nao andar com syntaxes
- \*Super fácil de encontrar interpretadores na net ----> Não é necessario instalar nada, para coisas pequenas podemos usar sites que nos correm o JavaScript por nós
- \*Linguagem Interpretada ----> Mesmas vantagens de Python, mas esta é baseada na sintaxe de C

Contras:

- \*Não é nativa a objetos ---> Foi lá com remendos
- \*Foi criada para só servir na programação de sites ---> Mas a industria está a adora-la e cada vez faz mais uso dela noutros locais~
- \*Não tem bibliotecas muito uteis fora da ideia de programação de paginas web ----> Tudo está "sequestrado" pelo browser
- \*Exige um template extremamente chato de HTML se não for corrido num site de outra pessoa ---> A maneira mais facil de correr JavaScript exige andar com um ficheiro HTML atrás
- \*Linguagem Interpretada ----> Mesmas desvantagens de Python, não tem um compilador para nos ajudar, por isso pode estourar só quando está a correr, ou seja corre programas mesmo com a sintaxe não 100% correta

```

console.log("Hello World");

```