**Programando com JavaScript**

Um IDE (ou Ambiente de Desenvolvimento Integrado) é um software que permite programar utilizando recursos extras.

Com IDEs, você pode editar, modificar, debugar, executar e implantar seu código. (Sem um IDE, um desenvolvedor deve selecionar, implementar, integrar e gerenciar todas essas ferramentas separadamente.)

IDE: (mais utilizados)

* Eclipse
* Visual Studio
* **Visual Studio Code (que vamos usar)**
* Netbeans
* Atom
* WebStorm

\*com o console do navegador tem como testar alguns dos códigos.

JavaScript pode ser usado de varias formas (tanto no lado do cliente como servidor).

* Para o lado do servidor iremos usar o NodeJS.
* No lado do cliente iremos usar com HTML.

Os documentos com JavaScript são feitos com (.js) ex: script1.js

Sempre que quiser inserir/informar algo (número, nome...):

Let nome =

Se eu colocar Console.log ele imprime as informações:

Console.log (“seu nome é ” + nome + “ e sua idade é “ + idade);



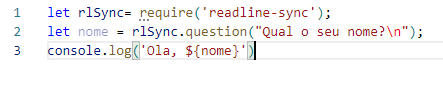
**Instalando um modulo que permite que faça input pelo CMD:**

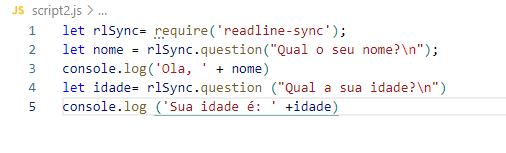
Vou em terminal e escrevo *“npm install readline-sync –save”*

Se eu quiser usar ele coloca require (‘readline-sync’);

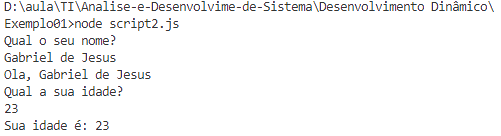
Isso permite que eu faça pergunta para quem vai utilizar! (obs: **\n** para soltar espaço no js.)

* Se eu não quiser usar +nome... posso usar ${nome})





Para executar eu vou no terminal e coloco node nome do arquivo.js



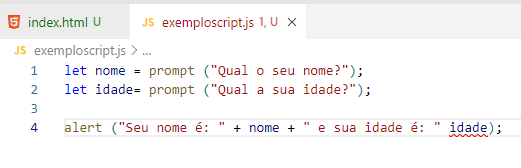
**Posso usar com Html**

Faz um index.html padrão. Na cabeça eu crio um script src e coloco **“script.js”** e criar

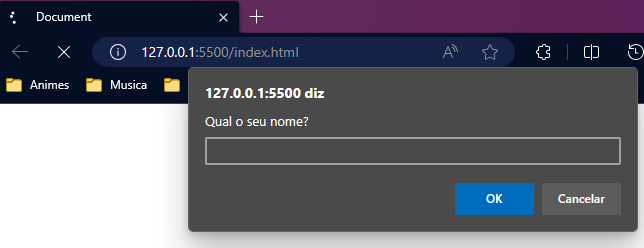
Nele posso fazer meus let nome; let idade e etc

para eu solicitar para alguém posso colocar o comando prompt

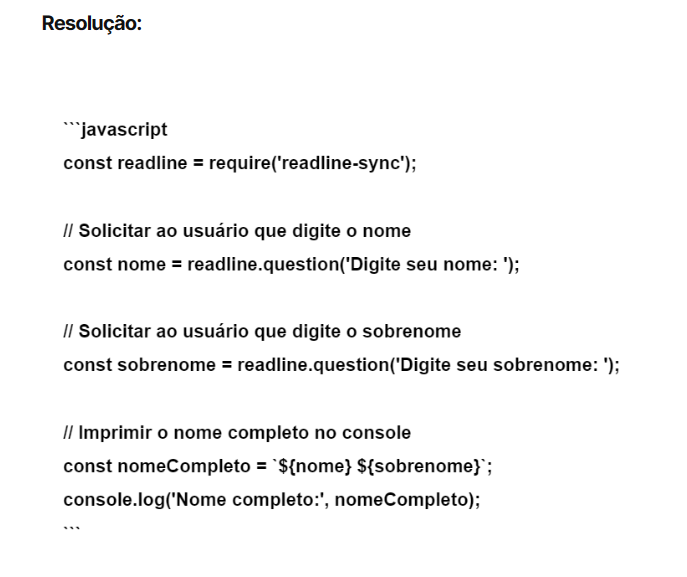
e se eu quiser mostrar alert



E assim que executar o html ele aparece:



Desafio 01: perguntar nome, sobrenome e falar



**Conceitos gerais do JS**

Linguagem de script orientada a objetos, multiplataforma.

Dentro de um host o JS pode ser ligado aos objetos deste ambiente para prover um controle programático sobre eles.

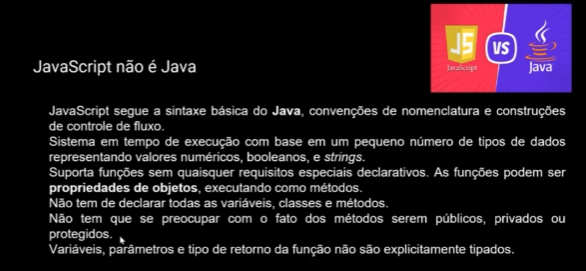
**Biblioteca padrão de objetos**: Array, Date e Math.

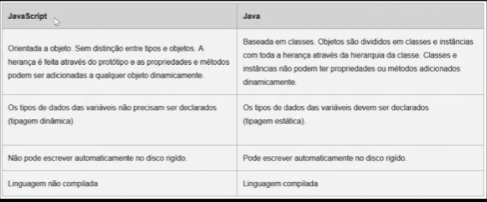
**Conjunto de elementos**: Operadores, estruturas de controle e declarações.

Pode usar do lado do:

* **Cliente:** fornece objetos para controlar um navegador web e seu Document Object Model (DOM), eventos do usuário. (Interação!)
* **Servidor:** fornece objetos relevantes à execução do JS em um servidor. Operacoes com BD e arquivos.

**Javascript não é java!**





JS tem versões! Padronizado pela Ecma Internacional.

**Estrutura da linguagem:**

Ele é case-sentive: Ele sabe que maiúscula é diferente de minúscula.

Espaços, tabulação e uma nova linha são considerados espaços em brancos.

É lido da esquerda para a direita.

Recomenda-se sempre adicionar ponto e virgula no final das declarações.

Escolha nomes coerentes para as variáveis> var **Nome**= “Maria”;

Adicionar comentários no código: \\ ou /\* e no final \*/

**Regras para identificadores de variáveis:**

Podem conter somente caracteres alfanuméricos (ou “$” ou “\_”).

Não podem iniciar com um digito. (ou seja, não começa com numero)

É “case sensitive”, Estado\_origem é diferente de estado\_origem

Ex valido: Nome\_funcionario, \_Salario, $cidade.

**Declarações:**

* **var**: declara uma variável e a inicializa com um valor. Ex: var Nome= “Maria”; *(global)*
* **let**: declara uma variável local, escopo do bloco, a inicialização do valor é opcional. Ex: let idade = 22; *(local)*
* **const**: declara uma constante de escopo do bloco, somente leitura.

**Variáveis global: (var)**

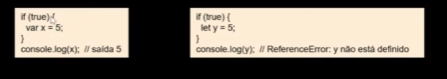
Declaradas fora de uma função.

Disponíveis para todos os blocos.

**Variáveis locais: (let)**

Declarada dentro de uma função.

Disponível apenas para a função onde foi declarada.



**Constantes: (const)**

Apenas leitura.

Sintaxe igual da variável.

Não pode alterar seu valor por meio de uma atribuição ou ser declarada novamente enquanto o script está em execução.

Deve ser inicializado com um valor.

Ex: const PI= 3.14;

**Diversas formas de trabalhar com JS:**

Node, HTML...

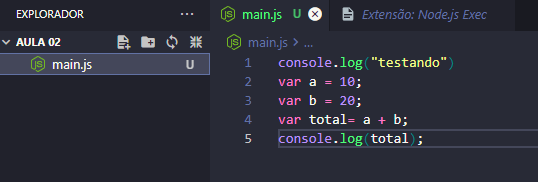
Abrir o navegador e abrir o console do navegador! (para testar).

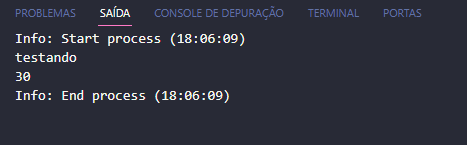
O comando console.log é para da uma informação no console!

Tipo console.log (“Olá”)

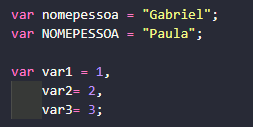
O arquivo javascript tem o “.js”

Para executar com a extensão só escrever o código e apertar “f8”





Posso declarar variáveis separadamente ou junto separando por “,”:



Ultimamente não se usa muito o var e sim o let!

O var não precisa iniciar com valor, posso declarar depois!

O let precisa iniciar com valor.

