Tipos de dados e variáveis

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Antes: COMENTÁRIOS

// -> uma linha

/\*\*/ -> várias linhas

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Um sinal de igual ( = ) significa recebe.

Dois sinais de igual ( == ) significa igual.

Se uma variável receber um valor *NULL*, significa que essa variável fica desocupada.

Quando for declarar uma variável, utiliza-se a palavra “ var ”. Podemos utilizar “ let ” também.

Existem 3 formas de delimitar uma String no JS. Aspas duplas apóstrofe e crase. Vamos ver isso melhor mais tarde, mas cada uma tem uma função.

Identificadores:

- Podem começar com **letras**, **$** ou **\_**

- Não podem começar com **números**

- É possível usar **letras** e **números**

- É possível usar **acentos** e **símbolos**

- Não podem conter **espaços**

- Não podem ser **palavras reservadas**

Dicas:

- **Maiúsculas** e **minúsculas** fazem diferença

- Tente escolher **nomes coerentes** para as variáveis

Dá para utilizar o node no vs code abrindo um terminal e digitando “node”.

Para fechar o node, basta digitar “.exit” e para sair do terminal basta digitar “exit”.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

TIPOS

number -> números em geral

string -> textos em geral

boolean -> verdadeiro ou falso

Para saber qual é o tipo da variável, basta utilizar o comando “ typeof ”

Interface gráfica do usuário

Descrição gerada automaticamente com confiança baixa

Texto

Descrição gerada automaticamente com confiança baixa

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Manipulação de dados

Dá para abrir uma variável usando var, let ou const. Há diferença entre eles, mas veremos depois.

Para concatenar, funciona igual Java -> usa-se o sinal de mais “ + ”

Existe o “+” para adição e “+” para concatenação

// (number + number) para adição

// (string + string) para concatenação

O window.prompt retorna um valor string.

Para resolvermos isso, precisamos fazer uma **conversão de tipos**

**String -> Número**

Number.parseInt(n) -> converte para número Inteiro

Number.parseFloat(n) -> converte para número Float

Number() -> converte para Inteiro ou Float, dependendo de que tipo de número será o resultado

**Número -> String**

String(n) -> Converte para String

n.toString() -> Converte para String

'O aluno ' + nome + ' tem ' + idade + ' anos e tirou ' + nota + ' na prova.'

- Para não precisar utilizar todas essas concatenações e ficar se confundindo e tendo dificuldade para entender o que está acontecendo, utiliza-se “template string” ->>>> ${}t

OBS: a frase vai começar com aspas.

EX: `O aluno ${nome} com ${idade} anos tirou a nota ${nota}`