Java Script

Para fazer comentários em JS é só usar :  
// única linha

/\* mais de uma linha de código\*/

Brendan Eich foi o criador de Java script.

Bibliografia:

- JavaScript – Guia definitivo

-JavaScript- guia do programador

Mozila – guia de referencia ... /\*interessante esse \*/

Ecma- guia de referencias

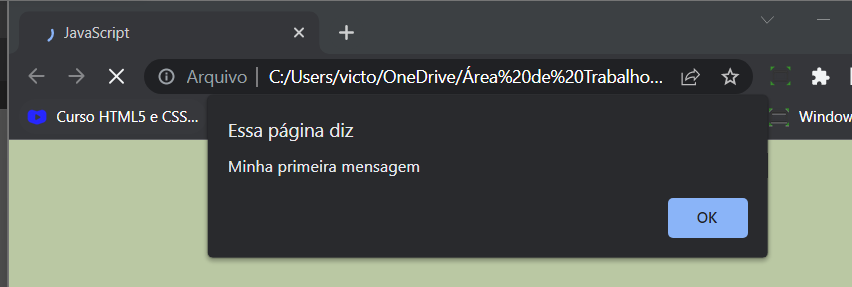
Primeiro script

<script>

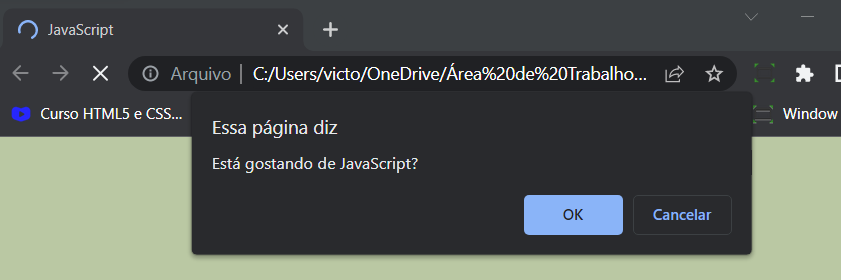
window.alert (‘Minha Primeira Mensagem’)

</script>

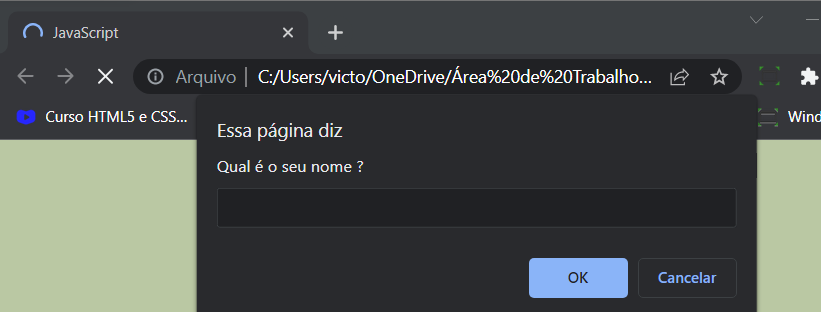
Colocando esse script aparece uma mensagem de alertar no servidor, igual a imagem abaixo :



O “Window.confirm” serve para pedirmos uma confirmação, na tela ele aparecerá assim:



“Window.prompt”, serve para fazer uma pergunta e a pessoa digitar a resposta. Na Tela aparecerá assim :



Módulo B  
-Armazenar dados

-Tratamentos de dados

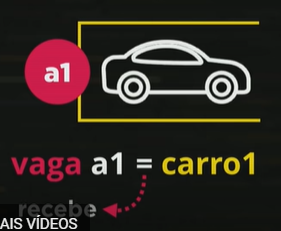
-Operações com dados

Variáveis e tipos primitivos

variais serve para guardarmos dados

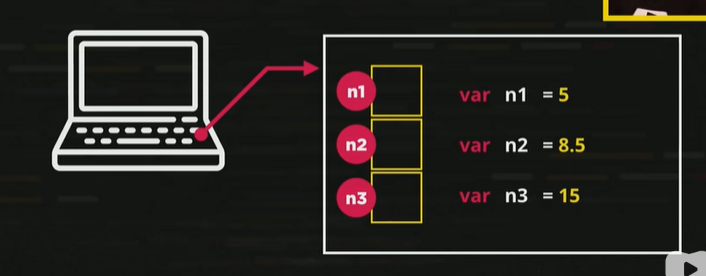
= Recebe

Ex: a vaga a1 recebe o carro1

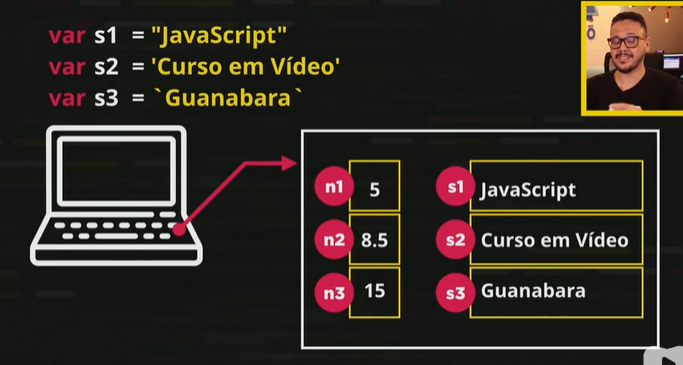


Variavel só pode receber um dado, precisaria tirar uma variável e colocar outra. Ex do estacionamento   
Para o carro 2 estacionar na vaga1 precisaria tirar o carro 1 de la. Caso eu não queria colocar nenhum carro seria só colocar   
a1=null

No JS usamos a palavra “var” ou “let” para identificar uma variável



Existem variáveis de tamanhos diferentes



Sempre que colocar uma palavra no string precisa usar as aspas e existem os três modelos de aspas

Identificadores:  
Chamamos as variáveis de identificadores  
- Podemos começar com letra, $ ou \_  
-Não podem começar com números

-È possível usar letras ou números

-è possível usar acentos e símbolos   
- Não podem conter espaços

-Não podem ser palavras reservadas (palavras reservadas são palavras que o javascript usa como comandos, ex: “var” que é usado para indicar variais

Dicas :  
-Maiúscula e minúsculas fazem diferença, se eu criar uma variável “A” e eu pedir para mostrar “a” não vai mostrar, vai dizer que não foi definido.

-Tente escolher nomes coerentes para as variáveis

Tipos Primitivos – data types

String: Conjunto de caractéries   
Boolean: True ou false ( verdadeiro ou falso)

Number: Números

Para trabalhar com esses tipos primitivos é só usar “ typeof”

Para fazer uma conversão de string -🡪 para Número

Number (n) para números reais e inteiros  
- Number.parseInt(n) ( fara a conversão para um numero inteiro)  
- Number.parseFloat(n) (fará a conversão para um número real)

Conversão de Número -🡪 string

String(n)  
n.toString()

Formatando strings

‘Eu estou aprendendo s’ // não faz interpolação

‘eu estou aprendendo’ + s //usa concatenação

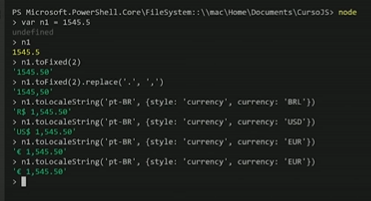
`Eu estou aprendendo ${s}` //usa template string ( essa é a forma mais “bonita” de se escrever.

s.length // quantos caracteres a string tem

s.toUpperCase() //tudo para ‘MAIÚSCULAS’

s.tpLowerCase() // tudo para ‘minúsculas’

String para valores em real, dólar , euro



Operadores :

Tome cuidado para usar operadores, com a ordem de precedência exemplo

5 + 3 / 2 na teoria daria 4 mas usando operadores dá 6.5 , isso porque em JS primeira se faz divisão e depois mais . Ou seja 3/2 = 1.5 + 5 = 6.5

Caso queira somar o 5+3 primeiro precisaria coloca-los entre parênteses.

(5+3) / 2

A ordem de Precendência é sempre

( )

\*\*

\*/ % ( estão na mesma linha porque tem a mesma ordem de precedência )  
+ -

-Aritméticos: + - \* / % \*\* (operadores usados para fazer cálculos)  
 5 + 2 🡪 7  
 5 – 2 🡪 3  
 5\* 2 🡪10

5/2 🡪2.5

5%2 🡪 1 (Esse operador não serve para porcentagem, ele é o operador para resto de divisão inteira) .

5 \*\* 2 🡪25 (esse representa a potência, ou seja 5 ao quadrado é 25)

-Atribuição simples:   
 var a = 5 + 3 🡪 8

var b = a % 5 🡪3

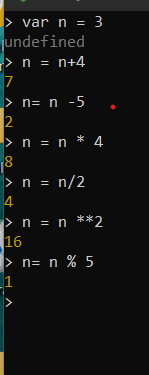
var c = 5 \* b \*\* 2 🡪 45

var d= 10 – a /2 🡪6

var e = 6\* 2 / d 🡪 2

var f= b % e + 4/ e 🡪 3

uma auto atribuição: Nesse exemplo a variável n = 3 passou a ser var n = 1



Incremento:  
 n= 5  
 n ++

n 🡪 6 ( isso porque ++ ou – ele soma mais um ou subtrai mais um)

-Relacionais

>

<

>= (maior e igual)

<= (menor ou igual)

== (igual)

!= (diferente)

5 > 2 🡪 true (verdadeiro)

7 < 4 🡪 false (falso)

8 >= 8 🡪 true

9<= 7 🡪 false

5 == 5 🡪 true

4 != 4 🡪 false

=== ( três sinais de igual significa que eu quero saber se a variavel x é idêntica a variável y

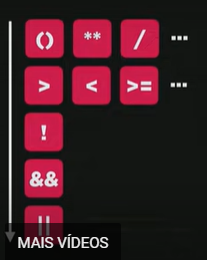
-Lógicos

! negação (não) . Esse operador é um operador unário. Ou é verdadeiro ou falso

&& Conjunção (e) . Esse operador é binário .

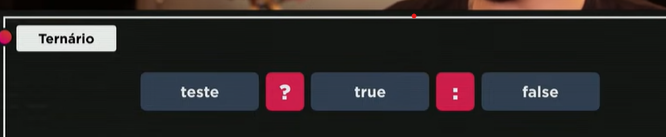
|| Disjunção (ou). Operador binário.

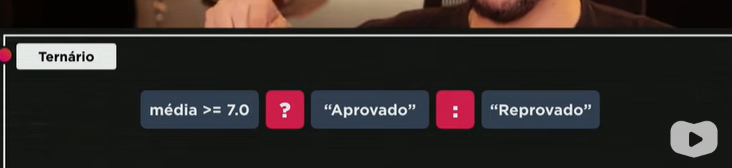
Ordem de precedência :



-Ternário

O operador ternário junta 3 operanos



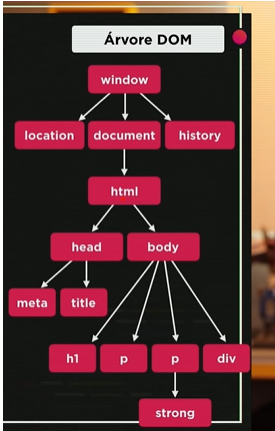


?

:

Entendendo o DOM

Dom é uma abreviação para Document Object Model  
modelo de documento, que dará acesso aos componentes internos no web site.



window- janela   
location – localização do site (url)

document- documento atual

history: de onde veio e pra onde vai

html- texto

head- cabeçalho

body- corpo

Pode acessar os elementos da arvore selecionando por marca, por ID, por nome, classe e por seletor (CSS).

Como selecionar por marca:   
getElementsByTagName()

ID:

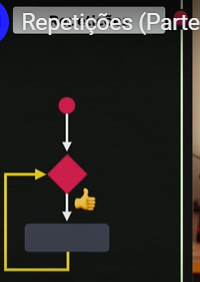
getElementsbyId()

por nome()  
getElementsByName()

por classe

getElementsByClassName()

repetição:



While(condição) : enquanto

Quando escreve while precisa colocar um bloco {

O comando sendo verdadeiro ira ser executado

} quando for falso sairá do fluxo.

C++ é igual c = c + 1

Do = faça {  
}

For (inicio ; teste ; incr (incremento ){

Na primeira passagem pelo for ele vai fazer a inicialização e o teste, sendo positivo ele vai executar o bloco, e voltar para cima e fazer o incremento . Quando for falso ele sai da estrutura

}