Javascript:

* **Observações:**

- Dentro do VS Code, usa o terminal, e la dentro escrevemos “node”, para assim abrir o node no próprio VS Code

- ctrl + alt + n : executa o código no terminal

- Site que roda HTML CSS e JAVASCRIPT: <https://jsfiddle.net/>

* **Erro no código:**

- botão direito do mouse no google e clica em inspecionar

* **Como debugar o código no VS Code:**

I) No roda pe, vai na opção “run”

II)seleciona “Start Debugging”

III) Vai aparecer uma barra para ir debugando

* Modulo B: (Variáveis e tipos primitivos)

**Usamos sempre “var”:**

Var nome = “Thiago”

Var idade= 18

Var maiorDeIdade= True

**Obs**: typeof nomeVariavel : diz qual tipo de variável.

**Transformando String em numero:**

Number.parseInt(n)

Number.parseFloat(n)

Number(n) // o próprio javaScript identifica o que é

**Transformando numero em String:**

String(n)

**Formação String:**

s.length // quantos caracteres tem a string

s.toUpperCase() // transformar tudo em maiúsculo

s.toLowerCase() // transformar tudo em minúsculo

**Como digitar:**

window.alert(`A soma ente ${n1} e ${n2} é igual a ${soma}`) // entre crases

Numeros:

n1.tofixed(2) // ate 2 números depois da virgula

n1.toLocaleString(‘pt-BR’, {style: ‘currency’ , currency: ‘BRL’}) // mostra o dinheiro em real

n1.toLocaleString(‘pt-BR’, {style: ‘currency’ , currency: ‘USD’}) // mostra o dinheiro em dólar

* **Modulo B: (Operadores)**

**- Ordem de Precedência dos operadores:**

()

\*\*

\* / %

+ -

**- Identidade:**

5 == ‘5’ True (mesmo sendo um numero e uma string, ele da como verdadeiro)

5 === ‘5’ False (compara se é idêntico tanto o valor, como o tipo)

5 === 5 True

**- Logico:**

! Negação

&& Conjunção (E)

|| Disjunção (OU)

**- Ternario:**

Teste ? True : False

Exemplo:

Media >= 7.0 ? “Aprovado” : “Reprovado”

* **Modulo C: (DOM) D**ocument **O**bject **M**odel

Selecionamos os elementos de forma a uma arvore.

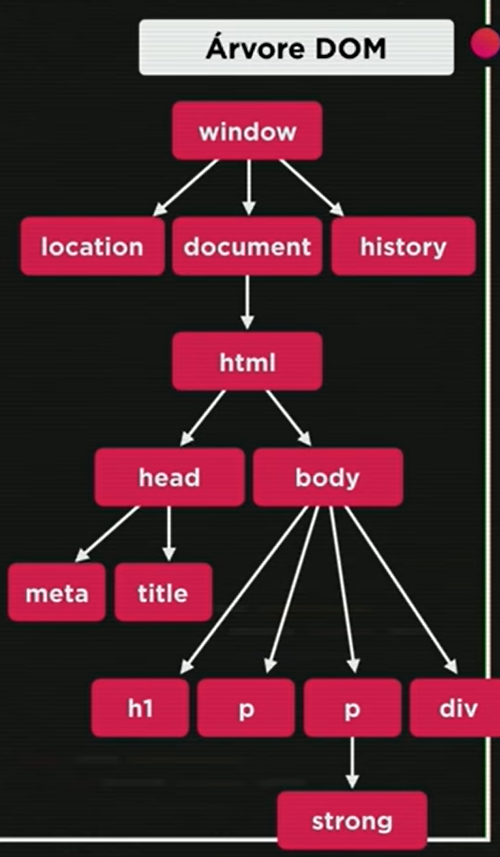
- Por Marca

- Por ID

- Por Nome

- Por Classe

- Por Seletor

****

**I)Nome:**

var p1 = window.document.getElementsByTagName('p')[0]

// p1 = primeiro paragrafo do html

 window.document.write('Esta escrito Assim: '+ p1.innerText)

// p1.innerText : Escreve o que temos no paragrafo 1

// p1.innerHTML : Escreve o paragrafo 1 com toda a formatação do html

**II) ID:**

<div id="msg">Clique em mim</div>

var d = window.document.getElementById('msg')

        d.style.background = 'green'

        d.innerText = 'Estou Aguardando...'

// Inicia uma variável com o ‘ID’ que quer

// depois com a variável edita o que deseja

**III) Seletor:**

var d = window.document.querySelector('div#msg')

        d.style.background = 'blue'

        d.innerText = 'Estou Aguardando....'

('div#msg') // div = tag // ‘#’ => quando id // Se ‘.’ => qnd é classe // ‘msg = id

* **DIV:**

-  <**div**> é um container genérico para conteúdo de fluxo.

Com essa tag podemos ter um ID nela, podemos modificar tudo dentro dela.

**- O Que podemos fazer com Eventos na DIV:**

**Link:** <https://developer.mozilla.org/en-US/docs/Web/Events>

**- Eventos DOM:**

Funcionalidades que podemos ter com minha tag, como a <div>

Por exemplo: passar o mouse, tirar o mouse, clicar, dendre outros, sempre gerando algum evento.

**- Function:**

Funções são utilizadas para uma execução ocorrer so quando for solicitada, por exemplo apertando um botão.

Exemplo:

<input type="button" value="Somar" onclick="somar()"> // botão que chamma a função somar

        function somar(params) {

        }

// assim, sempre que apertarmos o botão “somar” , vai ter uma chamada para a função somar e será executado

* **Modulo D: (Condições) IF e Else**

**Escrevendo em JS:**

console.log(`A velocidade do seu carro é de ${vel} km/h`)

// usamos console.log e o que escrevemos entre crases

**Exemplo condicional:**

var pais = 'França'

console.log(`Vivendo em ${pais}`)

if(pais == 'Brasil'){

    console.log(`Voce é Brasileiro`)

}

else{

    console.log(`Voce é estrangeiro`)

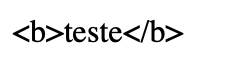
}

**InnerText vs InnerHTML:**

O innerText altera o conteúdo de um elemento de sua página (DOM) com o conteúdo tratado apenas como texto. Por exemplo:

document.getElementById('Teste').**innerText** = '<b>teste</b>'

Irá exibir:

[](https://i.stack.imgur.com/AuSYN.png)

document.getElementById('Teste').**innerHTML** = '<b>teste</b>'

Será exibido dessa maneira:

[inserir a descrição da imagem aqui](https://i.stack.imgur.com/h4tpu.png)

- **Condições:**

Varias condições:

**If:**

if (condition) {

}

else if (condition) {

}

**Switch:**

Exemplo:

switch (diaSem) {

    case 0:

        console.log('Domingo')

        break;

    case 1:

        console.log('Segunda')

        break;

default:

        console.log('Dia invalido')

        break;

// OBS: Switch só funciona com **numero** ou **caractere**

**#Criando Imagem Dinamicamente:**

var img = document.createElement('img') // criando uma tag "img"

    img.setAttribute('id', 'foto')  // id da img = "foto"

* **Modulo E: (Repetições)**

**Exemplo:**

var c = 1

while (c <= 5) {

    console.log(`Passo [${c}]`)

    c++

}

do{

    console.log(`Ola Mundo ${c}`)

}while(c <= 5)

for (let c = 1; c <= 5; c++) {

    console.log(`Passo ${c}`)

}

**#Apredizado Com Exercícios:**

- Let: são variáveis, como “var”, porem são temporárias. Exemplo, uma variável usada só dentro de uma função

- Numero.value.length == 0 // verificar se “numero” esta vazio. Exemplo:

if(num.value.length == 0)

- Criando um objeto html dentro do JavaScript

var tab = window.document.querySelector('select#tab')

let item = document.createElement('option') // criando um objeto html no javaScript

tab.appendChild(item) // adiciona um elemento filho "item" dentro do pai "select”

// Nesse exemplo temos o pai “select”, e estou adicionando filhos “option” dentro do pai, ficaria assim:

<select>

<option>

</option>

<\select>

* **Modulo f: (Vetores)**

let num = [0, 0, 7, 6, 5] // construindo um Array

num[3] = 7 // adicionando '7' na posição 3

num.push(5) // adicionando '5' no final do vetor

num.length // quantos elementos tem no meu vetor

num[0] // lendo um elemento na posição 0

num.sort() // transforma o vetor em ordem crescente

let pos = num.indexOf(7) // busca em que posição esta o valor "7"

// obs: se não encontrar, retorna o valor “-1”

**- Usando o for:**

for(let pos=0;pos<num.length;pos++){

    console.log(`A posição ${pos} tem valor ${num[pos]}`)

}

**- Usando o For simplificado:**

for(let pos in num){

    console.log(`A posição ${pos} tem valor ${num[pos]}`)

} // obs so funciona com array e objetos

// Para cada “posição” dentro de “num”, Execute