**CURSO DE JAVASCRIPT**

# MÓDULO A

# AULA01- O QUE JAVASCRIPT É CAPAZ DE FAZER?

Faz praticamente tudo

Cliente x Servidor

Seu dispositivo e um cliente e o site que você está acessando e o servidor.

O servidor tem um determinado arquivo e manda uma cópia para a sua máquina que usa as fermentas para lê-lo e mostrá-lo em sua tela.

O Web site e composto por três tecnologias:

* Html
* Css
* JavaScript

Diferente de Html e Css, JavaScript é sim uma linguagem de programação.

Digitar uma frase- alert(‘Digite aqui’)

document.querySelector('img#hplogo').style.filter = 'grayscale(100%)'

document.querySelector('img#hplogo').style.visibility = 'hidden

document.querySelector('h1.content-head\_\_title').innerText = 'Vitinho e brabo no JavaScript'

**AULA02- JAVASCRIPT: COMO CHEGAMOS AQUI?**

Em 1960 a arpanet foi criada para fins militares pela darpa (dos estados unidos).

Em 1993, Tim bernnes lee, criou a linguagem html, o protocolo http e fundou a word wide web(www).

Surgiu o primeiro navegador que foi o mosaic, criado por Maike Wanderson.

Em 1994, Mike e Jim Clark fundaram a empresa Netscape.

Em 1995, Brendan criou a linguagem moka e ao mesmo tempo surgiu a linguagem Java

Depois renomearam a linguagem Moka de Javascript.

Em 1997, a Netscape cedeu o Javascript para ecma padronizar a linguagem.

Em 2002 , a Mozilla criou o Firefox.

Em 2008, a empresa google criou o navegador o google Chrome.

O Chrome tem um motor embutido que roda o Javascript que se chama v8.

Em 2010, foi criado o node que roda Javascript fora do navegador.

ECMAScript- padronização do Javascript.

1.0

2.0

3.0

Es5

Es6

Tecnologias que surgiram a partir do Javascript:

* jQuery- Conjunto de bibliotecas
* Angula- facilita as interações web
* Vue- framework
* Eléctron- criação de interfaces gráficas
* Ionic- criação para aplicações moveis.
* Cordova

**AULA03- DANDO OS PRIMEIROS PASSOS NA LINGUAGEM**

Dicas para aprender:

* Ver vídeos
* Praticar no pc
* Ler livros sobre o assunto
* Anotar informações relevantes
* Estudar sobre o assunto com seus amigos
* E tentar aplicar o conhecimento em situações do seu dia dia

Bibliografia do Javascript:

* Java script o guia definitivo
* Java script o guia do programador
* Guia de referencia do mozilla e do ecma

<https://www.mozilla.org/pt-BR/>

https://www.ecma-international.org/publications-and-standards/standards/ecma-262/

Software:

* Chrome
* Vs Code
* Node

**AULA04- CRIANDO SEU PRIMEIRO SCRIPT**

Alerta do navegador- window.alert(Digite aqui')

Confirmação- confirm

Perguntas- prompt

**AULA05- VARIÁVEIS E TIPOS PRIMITIVOS**

Comentários:

//- comentários para só uma linha

/\* \*/- para mais de uma linha

Sinal (=) no Js, significa Recebe.

Variável- e um espaço na memória onde são guarde dados. Essas variáveis são chamadas de VR.

Existem variáveis de tamanhos diferentes.

O nome de uma var chama-se identificador

Identificadores podem começar:

* Letra
* $
* \_
* Não podem começar com números
* Possível usar letras e números
* Pode-se usar acentos e símbolos
* Não podem conter espaços
* Não podem ser palavras reservadas como as usadas em comandos
* Nas var, maiúsculas e minúsculas fazem diferença

Ex: salario = ‘1500.00’

Para sair .exit

Entre no vs code

Acesse o terminal- ctrl shift ‘

Digite:

Node

Escreva o código

.exit- par sair do node

Exit- para sair do terminal

Limpar tela do código- ctrl+ n

Representações ou data types: (como o Js vai identificar)

1. Number- números
2. Strings- palavras
3. Boolean- valores
4. Null
5. Undefined
6. Object
7. Function

Comando typeof

**MÓDULO B**

**AULA07- ${OPERADORES DO JS}**

Famílias:

* Aritméticos
* Atribuição
* Relacionais
* Lógicos
* Ternário

Aritméticos:

Todos os operadores precisão ter dois operados.

+ (adição)

- (subtração)

\* (multiplicação)

/ (ivisão)

% (resto da divisão inteira) – oque sobra de uma divisão inteira para ela não ficar quebrada

\*\* (potenciação)

Ordem dos sinais:

**parênteses, potências, multiplicações e divisões e adição e subtração**

ou seja

**()**

**\*\***

**\* / % (da esquerda para direita)**

**+ -**

Operadores de Atribuição

var

ex: 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

auto atribuições

Ex: var n = 3 (n vai valer 3)

Var n = n+4 (n vai valer 7)

Simplificando:

Ex: n = n+4 🡪 n += 4

Operadores de incremento: um operador ainda mais simplificado

Ex: var x = 5

x = x+1 🡪 x += 1🡪 x++ ou ++x

## AULA08- OPERADORES PARTE 2

Operadores:

1. Relacionais;

* >
* <
* >=
* <=
* ==
* != : diferente
* Os resultados serão sempre bolivianos, ou seja, verdadeiro ou falso.

1. Lógicos;

* ! = negação:
* && = conjunção
* || = distinção

1. Ternário;

* ?
* :

5 == ‘5’ = true pq o sinal relacional relaciona a grandeza

5 === (operador de identidade) ‘5’ = não porque é do mesmo valor, mas não do mesmo tipo

5 != ‘5’ = false

5 !== ‘5’ = true

! = negação

True

False

&& = conjunção

True true 🡪true

True false 🡪false

&&

False true 🡪 false

False false 🡪 false

|| = dijunção

True true 🡪 true

True false 🡪 true

||

False true 🡪 true

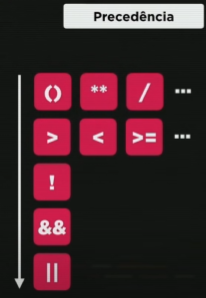
False false 🡪 false

Operador ternário

Tem 3 operando

Teste?true:false

Ex: media >= 7 ? “Aprovado” : “Reprovado”



### MÓDULO C

**AULA09- INTRODUÇÃO AO DOM**

Extenções

No vscode:

* Node exec
* Live server

No chrome:

* Live server

Procure o live server no ctrl shift p ou use o comando (alt L + alt O)

Oque é DOM?

É um Acrônimo de Document Object Model ou Modelo de Objetos para Documentos

É um conjunto de objetos que vai dar acesso aos componentes internos do seu web site

Presente no uso do JS no navegador

Arvore DOM:

Raíz: window:

* Location
* Document
  + Html
    - Head
      * Meta
      * Title
    - Body
      * H1
      * P
        + Strong
      * Div
* History

Quem está em baixo e children e quem está em cima e um parent

São chamados de elementos e podem ser selecionados por:

* Marca
  + getElementByTagName()
* Id
  + getElementById()
* Nome
  + getElementsByName()
* Classe
  + getElementsByClassName()
* Seletor
  + querySelector()
  + querySeletorAll()

Marca:

    <script>

        var corpo = window.document.body

        var p1 = window.document.getElementsByTagName('p')[1]

        corpo.style.background = 'black'

        window.document.write('Está escrito assim: ' + p1.innerText)

        window.document.write('Está escrito assim: ' + p1.innerHTML)

        window.alert('Está escrito assim: ' + p1.innerText)

    </script>

Id:

No html

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

No JS:

 <script>

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

        d.style.background = 'green'

        d.innerText = 'Estou aguardando...'

    </script>

Seletor

Id = #

Class = .

Ex:

No Html:

 <div class="mensagem">Clique em mim</div>

No JS:

 <script>

        var d = window.document.querySelector('div.mensagem')

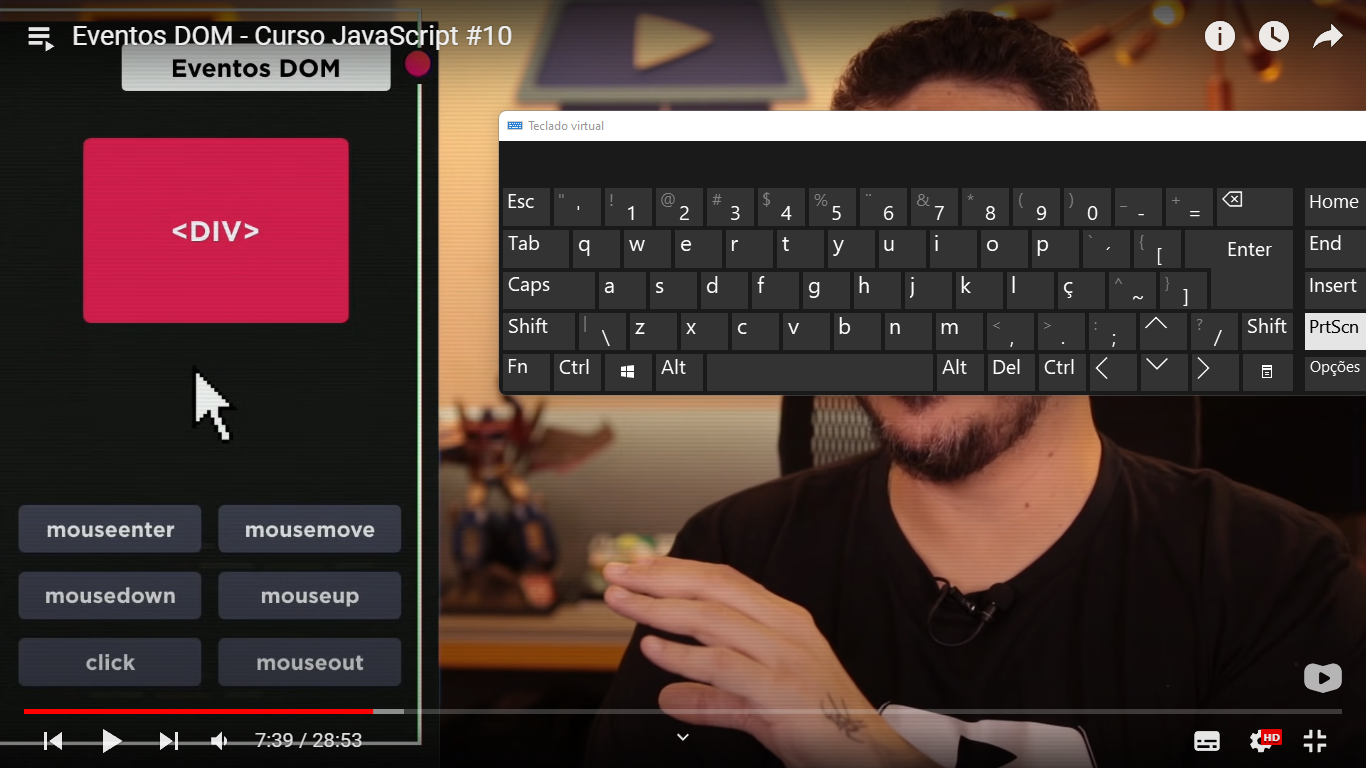
        d.style.background = 'red'

    </script>

## AULA10- EVENTOS DOM

Ao interagir com o site estará ocorrendo vários evento dom:

Ex: quando você vai passar o mouse em uma div :



Funções: serão feitas em bom bloco e só serão executadas a partir de determinados acontecimentos.

Ex:

Function ação(parametros) {

Bloco

}

Ex:

 <div id="area" onclick="clicar()" onmouseenter="entrar()" onmouseout="sair()">

        Interaja...

    </div>

    <script>

        var area = window.document.getElementById('area')

        function clicar() {

            area.innerText = 'Clicou!'

            area.style.background = 'red'

        }

        function entrar() {

            area.innerText = 'Entrou!'

        }

        function sair() {

            area.innerText = 'Saiu!'

            area.style.background = 'blue'

        }

    </script>

Ou

Ex:

 <div id="area">

        Interaja...

    </div>

    <script>

        var area = window.document.getElementById('area')

        area.addEventListener('click', clicar)

        area.addEventListener("mouseenter", entrar)

        area.addEventListener('mouseout', sair)

        function clicar() {

            area.innerText = 'Clicou!'

            area.style.background = 'red'

        }

        function entrar() {

            area.innerText = 'Entrou!'

        }

        function sair() {

            area.innerText = 'Saiu!'

            area.style.background = 'blue'

        }

    </script>

Para descobrir erros em JS

Use o devtolls

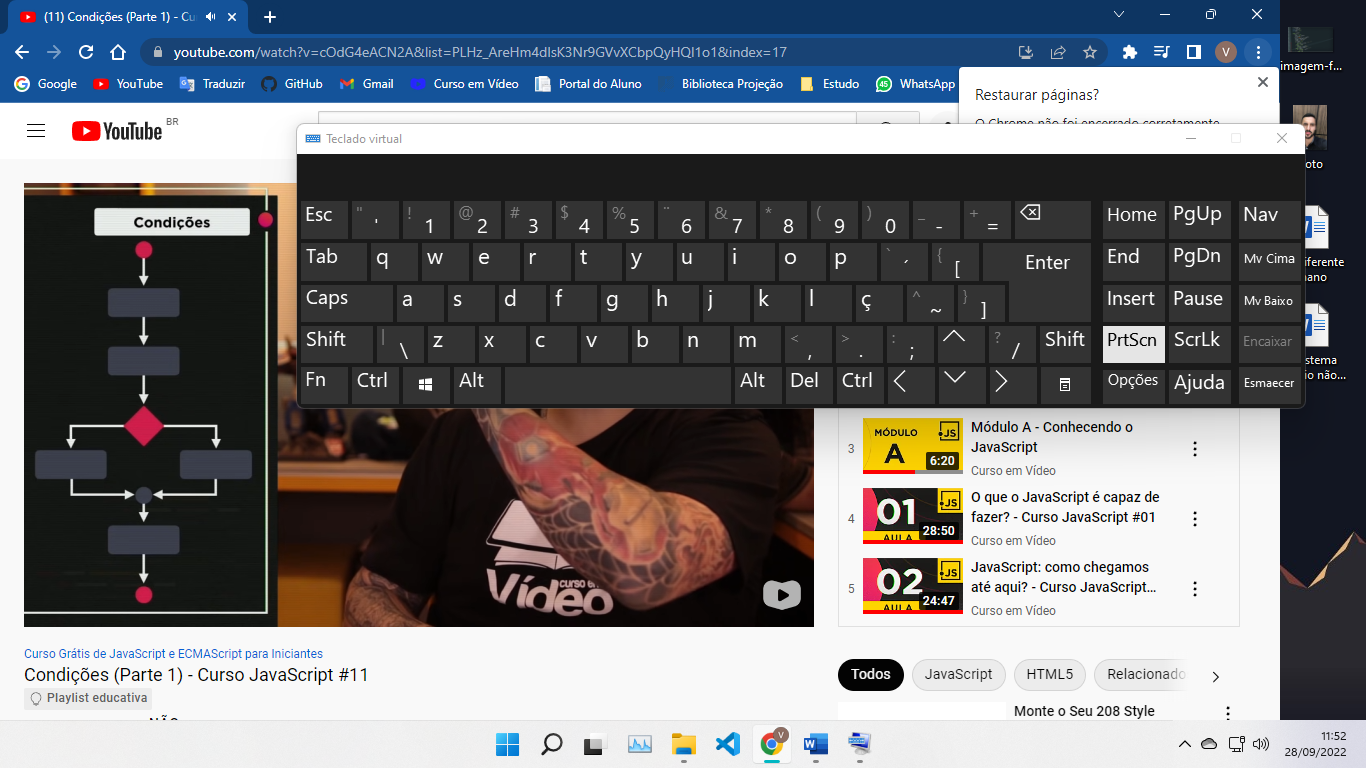
## MÓDULO D

## AULA11-` ${CONDIÇÕES EM JS}` PARTE 1

Sequencias:

* Var n = 3
* N +=2
* window.alert(n)

condições:



desvio condicional

if(condiição)

te da duas possibilidades

faça um bloco para if

e um para else

tipos de condição:

* simples:

if(condição) {

true

}

* Composta:

If(condição){

true

} else {

False

}

<script>

       function calcular(){

        var txtv = window.document.querySelector('input#txtvel')

        var res = window.document.querySelector('div#res')

        var vel = Number(txtv.value)

        res.innerHTML = `<p>Sua velocidade é de <strong>${vel} km/h</strong>.</p>`

        if(vel > 60) {

            res.innerHTML += '<p>Você está <strong>MULTADO</strong> por excesso de velocidade.</p>'

        }

            res.innerHTML += '<p>Dirija sempre com cinto de segurança.</p>'

       }

#### AULA12- CONDIÇÕES PARTE 2

Condições aninhadas: uma condição dentro da outra

var idade = 22

if(idade < 16) {

    console.log('Não vota.')

}

else{

    if(idade < 18){

        console.log ('Voto opicional.')

    }

}

Ou simplifique colocando o if no else

Ex:

var idade = 20

if(idade < 16) {

    console.log('Não vota.')

}

else if(idade < 18 || idade > 65){

        console.log ('Voto opicional.')

}

else{

    console.log('Voto obrigatório')

}

Como pegar hora atual:

var agora = new Date()

var hora = agora.getHours()

Condição Multipla:

Commando – swith(expressão){

case valo1:

codigo

break - obrigatório

case valor2:

código

break

case valoe3:

codigo

break

defalt:

codigo

break

}

Ex

var agora = new Date()

var diaSem = agora.getDay()

    /\*

    0 - domingo

    1 - segunda

    2 - terça

    3 - quarta

    4 - quinta

    5 - sexta

    6 -sábado

    \*/

    switch(diaSem){

        case 0:

        console.log('Domingo')

            break

        case 1:

        console.log('Segunda')

            break

        case 2:

        console.log('terça')

            break

        case 3:

        console.log('Quarta')

            break

        case 4 :

        console.log('Quinta')

            break

        case 5:

        console.log('sexta')

            break

        case 6:

        console.log('Sábado')

            break

        default:

            console.log('[ERRO] Dia Inválido!!!')

            break

    }

# EX001

Alguns commandos JS:

Imagem -  img.setAttribute('src',’imagem.formato’)

res.appendChild(img)

res.style.textAlign = 'center'

criar um id ou class

 var img = document.createElement('img')

        img.setAttribute('id', 'foto')/\*Esse comando acaba de criar um img com o id foto.\*/

var data = new Date()

    var ano = data.hora/dia/ano/etc ()

ex ano atual:

var data = new Date()

    var ano = data.getFullYear()

]

img.src = 'imagens/manha-redonda-p.png'

document.body.style.background = '#4F4066'

commando para falar no node.js

console.log('tudo bem?')

##### AULA13- REPETIÇÕES EM JS

Sequências :

Repetição 1

Repetição 2

Repetição 3

Ex:

var  c = 1

console.log(`Passo ${c}.`)

c++

console.log(`Passo ${c}.`)

c++

console.log(`Passo ${c}.`)

c++

console.log(`Passo ${c}.`)

c++

console.log(`Passo ${c}.`)

c++

console.log(`Passo ${c}.`)

c++

Partirá do ponto A, o chegar na condição, se for verdadeira, ele vai criar um (laço) que vai se repetir enquanto a condição for verdadeira, quando ela der falsa, ele irá partir para o ponto B.

Enquanto for verdade = while (condição) {

Bloco

}

Ex:

var c = 1

while (c <= 6){

    console.log(`Passo ${c}.`)

    c++

}

Ou ao contrário:

Ele lê o bloco e depois faz o teste.

Faça = Do{

Bloco

}while (condição)

Ex:

var c = 1

do{

    console.log(`Passo ${c}.`)

    c++

}while (c <= 6)

**AULA14- REPETIÇÕES JS (PARTE2)**

Estrutura de controle:

Representado por um hexágono irregular, faz quase a mesma coisa que o while mas tem o incremento.

For (inicio;teste;incremento) {

bloco

}

Ex:

for (var c = 1; c <=10; c++){

    console.log(`Passo ${c}`)

}

# EXX 4

**AULA15- VARIÁVEIS COMPOSTAS**

Variáveis simples conseguem armazenar apenas um valor por vez ao contrario da composta que guarda vários valores em uma mesma estrutura.

Ex:

Var a = [valor1 , valor2, valor3]

0, 1, 2

a – array vetor

valor – Elemento

0 – Índice ou chave

Oq esta dentro do valor – conteúdo valor

Um array é uma variável que tem vários elementos, cada elemento é composto por seu valor e por uma chave de identificação.

Ex:

let num = [5,8,2,9,3]

para adicionar uma variável:

num [3] = 6

adicionar elemento seis no espaço 3

para adicionar um elemento na última posição do array:

num.push(7)

adicionar o elemento 7 na ultima posição do array.

Num.length – para ler o comprimento do array.

num.sort() – ele pega todos os elementos e coloca em ordem crescente

ex:

let num = [5,8,2,9,3]

    num[3] = 6

    num.push(7)

console.log(`Nosso vetor é o ${num}`)

console.log(`O vetor tem ${num.length} posições.`)

console.log(`O primeiro valor do vetor é ${num[0]}.`)

    num.sort()

console.log(`Nosso vetor é o ${num}`)

console.log(`O primeiro valor do vetor é ${num[0]}.`)

aplicando repetição:

let valores = [3,6,4,9,8,2,10,15]

/\*console.log(valores[0])

console.log(valores[1])

console.log(valores[2])

console.log(valores[3])

console.log(valores[4])

console.log(valores[5])

console.log(valores[6])

console.log(valores[7]) \*/

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

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

}

Utilizando for in:

For ( in ) {

Bloco

}

for (let pos in valores) {

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

}

Para buscar um valor dentro do vetor:

num.indexOf(7)

Ele mostrara a posição do 7 no caso do ex, é o 3.

Num.indexOf(3)

Ele mostrara o número -1, porque no exemplo não tem nenhum valor 3.

let pos = num.indexOf(8)

console.log(`O valor 8 está na posição ${pos}`)

no caso e não encontrar:

let poss = num.indexOf(4)

if (poss == -1) {

    console.log('O valor não foi encontrado')

} else {

    console.log(`O valor está na posição ${poss}`)

}

# AULA16- FUNÇÕES JS

Vamos supor que sua mãe, te entrega um dinheiro, fala pra você ir na mercearia e comprar um leite, você faz o caminho, compra o leite e entrega para ela.

Função: São ações executadas assim q são chamadas ou em decorrência de algum evento.

* Chamada;
* Parâmetro;
* Ação;
* Retorno.

Function ação(parametro)🡪 parametro {

Bloco 🡪 ação

return res 🡪 retorno

}

ação(5) 🡪 chamada

ex:

let num = 156

function parimpar(n) {

    if (n%2==0) {

        return 'Par'

    } else {

        return 'Ímpar'

    }

}

let res = parimpar(num)

console.log(`O número ${num} é ${res}.`)

/\*

de outra forma

console.log(`O número ${num} é ` + parimpar(num) + `.`)

\*/

Outro exemplo: soma de números

function soma(n1, n2) {

    return n1 + n2

}

console.log(soma(2, 5))

caso coloque só um número:

function soma(n1=0, n2=0)

uma variável receber uma função:

let v = function (x){

    return x\*2

}

console.log(v(5))

calcular fatorial:

// 5! = 5 x 4 x 3 x 2 x 1 = 120

function fatorial(n) {

    let fat = 1

    for(let c = n; c > 1; c--) {

        fat \*= c

    }

    return fat

}

console.log(fatorial(5))

factorial dentro de factorial:

// RECURSIVIDADE

function fatorial(n) {

    if (n == 1) {

        return 1

    } else {

        return n \* fatorial(n-1)

    }

}

console.log(fatorial(5))

/\*

5! = 5 x 4 x 3 x 2 x 1

5! = 4!

n! = n x (n - 1)!

\*/

Modo depuração

# EXERCÍCIOS

##### EX001- IMAGEM DE MANHA TARDE E NOITE

Comandos para conseguir as datas atuais

var variavel1 = new Date()

 var variavel2 = variavel1.get-comando-desejado()

img.src = 'imagem.formato' // comando para adicionar imagens!

    document.body.style.background = '#897D75' //comando para adicionar um fundo

let variavel = '' // cria uma variável vazia para ser usada como duas opções como no if else

        let img = document.createElement('img')

            img.setAttribute('id', 'foto') // Cria uma imagem com o id foto ou seja, é a mesma coisa que adicionar o código <img id="foto"> no html

para puxar a imagem criada em js:

variavel.appendChild(img)

comandos de funções js:

<div id="botao" onclick="clicar()" onmousedown="apertar()" onmouseup="soltar()"> <!--

                        onclick = o clique do mouse

                        oumousedown = aperta o mouse

                        onmouseup = solta o mouse -->

                        <p>Gerar Tabuada</p> </div>

Comando de select html:

<select name="tabuada" id="seltab" size="10">

                <option>

                    Digite um número acima!

                </option>

            </select>

let p = Number(document.getElementById('txtpasso').value) /\*

    var variavel1 = document.getElementById('id')

    var variavel2 = Number(variave1.value) - Transformar variável 'string' em decimal

    \*/

variavel.innerHTML += `\u{1F3C1}` // \u{cod-emogi} - Para adicionarum emogi

let variavel1 = document.createElement('option') // cria um <option> no select do html

            variavel1.text = `${n} x ${c} = ${n\*c}` // adiciona o texto ao option

            variavel2.value = `tab${c}` // variavel2.value - transforma a variavel em decimal. Variavel2${c} dar identidade a cada option para um banco de dados no futuro

            tab.appendChild(variavel1) // Faz a variável criada pelo js aparecer na tela dentro do select como um option e seu texto definido no código acima.

Ex:

let tab = document.getElementById('seltab')

 let c = 1

        while (c <= 10) {

            let item = document.createElement('option') // cria um <option> no select do html

            item.text = `${n} x ${c} = ${n\*c}` // adiciona o texto ao option

            tab.value = `tab${c}` // tab.value - transforma tab em decimal. tab${c} dar identidade a cada option para um banco de dados no futuro

            tab.appendChild(item) // Faz o item criado pelo js aparecer na tela dentro do select como um option e seu texto definido no código acima.

            c++

        }

// Funcionalidades dos Botões

function adap() {

    let botao = document.getElementById('botaoad')

    botao.style.background = 'var(--cor1)'

}

function adsl() {

    let botao = document.getElementById('botaoad')

    botao.style.background = 'var(--cor0)'

}

function fnap() {

    let botao = document.getElementById('botaofn')

    botao.style.background = 'var(--cor1)'

}

function fnsl() {

    let botao = document.getElementById('botaofn')

    botao.style.background = 'var(--cor0)'

}

// Funcionalidade do Analizador de números

let num = document.querySelector('input#txtvalor')

let res = document.getElementById('resultado')

let lista = document.querySelector('select#seltab')

let valores = []

// As duas próximas funções não estão presentes no html. Elas foram criadas para dar funcionalidade para a função (adicionar())

function isnumero(n) {

    if(Number(n) >= 1 && Number(n) <= 100) {

        return true

    } else {

        return false

    }

}

function inlista(n, l) {

    //O indexOf serve para procurar a posição de um elemento detro da variável composta, se o número não estiver em nenhuma posição da variável, será descrito o número -1

    if (l.indexOf(Number(n)) != -1) {

        return true

    } else {

        return false

    }

    /\*

    n == -1 significa que ele não esta na lista

    n != -1 significa que ele está dentro da lista

    \*/

}

/\*--------------------------------------------------\*/

function adicionar() {

    if(isnumero(num.value) && !inlista(num.value, valores)) {

        valores.push(Number(num.value))

        let item = document.createElement('option')

        item.text = `Valor ${num.value} adicionado.`

        lista.appendChild(item)

        res.innerHTML = ''

    } else {

        window.alert('Valor Inválido ou já encontrado na lista.')

    }

    num.value = '' /\*O Java sempre carregar o ultimo comando escrito no código, ou seja, ao colocar esse comando por ultimo o num ficar vazio\*/

    num.focus()  /\*comando usado para dar foco na variável, ou seja, o cursor vai ficar la mesmo depois de você clicar em um botao por acaso\*/

}

function finalizar() {

    if (valores.length == 0) {

        windoww.alert('Adicione Valores antes de finaizar.')

    } else {

       let tot = valores.length

       //quando a variavel e 1 ela é a maior e a menor por isso se dar os mesmos valores

       let maior = valores[0]

       let menor = valores[0]

       let soma = 0

       let media = 0

       for (let pos/\*var pos criada\*/ in valores) {

        soma += valores[pos] // a soma começa com 0 e vai somando com todos os valores até o for acaba

        if (valores[pos] > maior) // se o pos for maior que a variavel[posição 0] então ele será maior

        maior = valores[pos]

        if (valores[pos] < menor)

        menor = valores[pos]

       }

# AULA17- PRÓXIMOS PASSOS

Fazer curso de html e css;

Estudar mais sobre:

Functions;

Objetos;

Modularização;

RegEx;

JSON;

AJAX;

NodeJs;

Angula;

EcmaScript;

jQuery.

Introdução de objetos:

Num array você pode colocar diferentes tipos de variável como decimal, string, data, float entre outros.

O problema do array é que a chave não pode ser alterada

Para declarar um objeto é preciso colocar chaves

Ex:

Let amigo = {Nome:’Jose’}

Vai criar uma variável composta com a chave nome para a casa com a string Jose

Let amigo = (nome:’Jose’, sexo:’M’, peso: 85.4}

Significa:

Let amigo = {‘Jose’, ‘M’, 85.4}

Chave: Nome sexo peso

/\*Objeto\*/

let amigo = {nome:'José',sexo:'M',peso:84.5,engordar(p){}/\*função\*/}

para chamar pela chave

console.log(amigo.nome)

ex: