Java Scriptdo do Curso em Video

No VsCode > escreva > html:5 > enter > Faz o html normal

No final do body adiciona <script></scricpt>

Aula 4: ex001:

<script>

        window.alert('Minha primeira mensagem.')

</script>



O conteúdo some

<script>

        window.confirm('Esta gostando de Java Script?')

</script>



<script>

        window.prompt('Qual é o seu nome?')

</script>

****

<script>

        window.alert('Minha primeira mensagem.')

        window.confirm('Esta gostando de Java Script?')

        window.prompt('Qual é o seu nome?')

</script>

Variáveis: serve para guardar dados

Comentários: // - serve para uma única linha e /\*\*/ - mais de uma linha

##### Identificadores das variáveis:

* Podem começar com letra, $ ou \_
* Não podem começar com números
* É possível usar letras ou números
* É possível usar acentos e símbolos
* Não podem conter espaços
* Não podem ser palavras reservadas (comandos do JS)

No Node.js ou no VsCode > Terminal > New terminal



Para limpar a tela: ctrl + L

Para sair do programa: .exit

Para mostrar qual o tipo da variável no terminal ou Node.js se coloca typeof

Diferença entre VAR  X  LET  X CONST Aula html > udemy > Var.Let.Const.js

//Usando Var - pouco usado

var value; //pode apenas declarar sem descrição de primeira

value = '1232';

//Usando Let

let color = 'Red'; //pode apenas declarar sem descrição de primeira

color = 'Blue'

//Usando Const - NÂO pode alterar, redefinir, da erro - mais usado

const fullName = 'Larytgs' //sempre tem q ter uma descrição



console.log(value);

console.log(color);

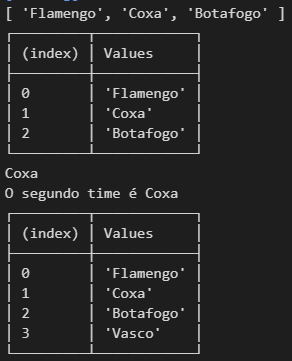
console.log(fullName);

#### Tipos Primitivos Primordiais

|  |  |
| --- | --- |
| Number | 5 18 -12 0.5 -15.9 3.14 8.0  Infinity NaN(not a number) |
| String | “Google” ‘JavaScript’ `Maria` “098-97.09” |
| Boolean | True e False |
| Null | Valor vazio |
| Undefined | Quando não tá recebendo nenhum valor |
| Object | [ ] { } Array(vetor) |
| Function | Funções |

## Uso de caracteres especiais em string

|  |  |
| --- | --- |
| \0 | Byte nulo |
| \b | Backspace |
| \f | Alimentador de formulário |
| \n | Nova linha |
| \” | Aspas duplas |
| \’ | Apóstrofo ou aspas simples |

 Array – serve para fazer uma lista Aula html > udemy > array.js

let times = ['Flamengo', 'Coxa', 'Botafogo'];

console.log(times) //mostrar em lista

console.table(times) //mostrar em tabela

console.log(times [1]) //mostrar o segundo time

let msg = 'O segundo time é ' + (times[1])

console.log(msg)

times.push('Vasco') //para adicionar mais no final

console.table(times)

//adicionar um elemento no final do seu array, use o método push()

//no início do seu array, use o método unshift()

//Se você deseja adicionar um elemento em uma posição específica do seu array, use o método splice()

## Conversão de string para numero: (parse-conversao) Aula06-Ex003

* Number.parseInt(n)
* Number.parseFloat(n)
* Number(n)

<script>

        var n1 = Number.parseFloat(window.prompt("Digite um numero:"))

        var n2 = Number(window.prompt("Digite outro numero: "))

        var s = n1 + n2

        window.alert('A soma dos dois numero é ' + s)

</script>

#### Conversão de numero para string:

* Sting(n)
* n.toSting( )

#### Template de String – **Formatação de strings** **– utilizando `crases` e ${n}** – no node

## Aula06-Ex004:

* s.length : quantos caracteres a string tem
* s.toUpperCase() : tudo em letra ‘MAIUSCULAS’
* s.toLowerCas() : tudo em letra ‘minusculas’

<script>

        var nome = window.prompt("Qual é oseu nome? ")

        document.write(`Ola ${nome}. Seu nome ${nome.length} tem letras. <br>`)

        document.write(`Seu nome em maiusculas fica ${nome.toUpperCase()} <br>`)

        document.write(`Seu nome em minusculas fica ${nome. toLowerCase()}`)

</script>

#### Formatação de números – no node

* n1.toFixed(2) : para colocar duas casas depois do ponto
* n1.toFixed(2).replace(‘.’ , ‘,’) : para substituir oponto por virgula
* n1.toLocalesString(‘pt=BR’, {style: ‘currency’,

currency: ‘BRL’}) : colocando R$

* n1.toLocalesString(‘pt=BR’, {style: ‘currency’,

currency: ‘USD’}) : colocando US$

* n1.toLocalesString(‘pt=BR’, {style: ‘currency’,

currency: ‘EUR’}) : colocando €

## **Operadores de Java Script**

|  |  |  |
| --- | --- | --- |
| * Aritméticos     Ordem de precedência:  ( )  \*\*  \* / %  + -  .  5 % 2 = é o resto da divisão  .  5 \*\* 2 = é a potencia | * Atribuição     Atribuiçao simples <- | |
| * Relacionais     Identidade:  .  5 é igual a 5  5 é iguala ‘5’  5 não é idêntico a ‘5’  5 é idêntico a 5 | * Lógicos     && significa “e”  || significa “ou”   * Ordem de precedência:   1º Aritméticos  2º Relacionais  3º Lógicos | |
| * Ternário     O teste logico que pode dar verdadeiro ou falso    Vai aparecer aprovado ou Reprovado dependendo do teste  Ex1:  A média é 5.5  A média é maior que 7?  Se sim, Aprovado, se não Reprovado  Ex2:  X é 8  8 % 2 == 0  Resto de 8 dividido por 2 é 0 então 0 é igual a 0  Se a resposta for True = 5 se for False = 9  Res = 5  Ex3: | | |

## **DOM document Object Model**

Aula 9 ex 05:

Arvore do DOM

3 dos objetos:

Location : URL

Document: o doc atual

History: de onde vim p onde vou

**Ver mais sobre na Aula 9 – Introdução ao DOM**

5 tipos para selecionar:

* Por marca

|  |
| --- |
| <script>          var p1 = window.document.getElementsByTagName('p')[1] //seleciona o 1º paragrafo          window.document.write("<hr>Esta escrito assim: &nbsp" + p1.innerText)  /\*innerHTML: mostra com as formatações\*/  </script> |

* Por ID : colocando: id=”msg” la na <div>

|  |
| --- |
| <script>          var d = window.document.getElementById('msg')          window.document.write('<hr>' + d.innerHTML)          d.style.background = 'green' //vai mudar a cor dele em cima          d.innerText = 'Mudando o texto da div' //vai mudar em cima  </script> |

* Por nome colocando: name=”msg” la na <div>

|  |
| --- |
| <script>        var d = window.document.getElementsByName('msg')[0]  d.style.textAlign = 'center' //colocando o texto no centro        d.innerText = 'OLAAA' //vai mudar em cima  </script> |

* Por classe colocando: class=”msg” la na <div>

|  |
| --- |
| <script>       var d = window.document.getElementsByClassName('msg')[0]       d.style.background = 'orange'       d.innerText = 'Mudando agora com o Class'  </script> |

* Por seletor colocando por id=”msg” la na <div>

|  |
| --- |
| <script>       var d = window.document.querySelector('div#msg')       d.style.color = 'blue'       d.innerText = 'Mudando agora por QuerySelector'  </script>  Por class=”msg”  Apenas mudar no var no <div>:  ('div.msg')  querySelector : singular  querySelectorAll : plural |

## **Eventos DOM – o que pode acontecer com as tags Aula 9 ex 06**

* Mouseenter – entrar com o mouse em cima da tag
* Mouseout – tirar o mouse de cima

|  |
| --- |
| <div id="area" onmouseenter="entrar()" onmouseout="sair()">          Interaja...  </div>  <script>  *var* a = window.document.getElementById('area') //colocando p fora, serve para todos          a.addEventListener('click', clicar)          a.addEventListener('mouseenter', entrar)          a.addEventListener('mouseout', sair)    *function* clicar() {              a.innerText = 'Voce clicou!!'              a.style.background = 'green'              a.style.color = 'yellow'          }  *function* entrar() {              a.innerText = 'entrou'          }  *function* sair() {              a.innerText = 'saiu'              a.style.background = 'purple'          }  </script> |

* Mousemove – mexer o mouse dentro da tag
* Mousedown – pressionar o mouse
* Mouseup – soltar o mouse
* Click – clicar em cima

Na div pode tirar tudo aquilo pra não ficar poluído, e adicionar no script

a.addEventListener('click', clicar)

a.addEventListener('mouseenter', entrar)

a.addEventListener('mouseout', sair)

Mais tipos acessa 🡪 <https://developer.mozilla.org/en-US/docs/Web/Events>

## **Condições**

* Condições Simples aula11 ex08

|  |
| --- |
| var velocidade = 80.5  console.log (`A velocidade do seu carro é ${velocidade}`)  if (velocidade > 70){      console.log (`Voce ultrapassou o limite de velocidade! Multado!`)  }  console.log(`Use sempre o cinto de segurança!`) |

* Condições Compostas aula11 ex09

|  |
| --- |
| aula11 ex09  var pais = "EUA"  console.log(`Vivendo em ${pais}`)  if (pais == "França"){      console.log('Voce é Brasileiro');  }else{      console.log('Voce é Estrangeiro')  }  aula11 ex10 em html  <body>      <h1>Paises</h1>      Em que país voce vive? <input type="text" name="nome" id="txtnome" >      <input type="button" value="Verificar" onclick="verificar()">      <div id="vive"></div>      <script>          function verificar(){              var pais = window.document.getElementById('txtnome')              var vive = window.document.querySelector('div#vive')              var nac = String(pais.value)              vive.innerHTML = `Vivendo em ${nac}`              if (nac == 'Brasil' || nac == 'brasil'){                  vive.innerHTML += "<p>Voce é Brasileiro.</p>"              }else{                  vive.innerHTML += "<p>Voce é Estrangeiro.</p>"              }          }      </script>  </body> |

## Elemento: propriedade: ClassList aula11 ex10 em html

Usar classList: uma alternativa conveniente para acessar a lista de classes de um elemento como uma string delimitada por espaço via [element.className](https://developer.mozilla.org/en-US/docs/Web/API/Element/className).

Embora a classListpropriedade em si seja somente leitura, você pode modificá-la DOMTokenListusando os métodos [add()](https://developer.mozilla.org/en-US/docs/Web/API/DOMTokenList/add" \o "adicionar()), [remove()](https://developer.mozilla.org/en-US/docs/Web/API/DOMTokenList/remove), [replace()](https://developer.mozilla.org/en-US/docs/Web/API/DOMTokenList/replace" \o "substituir())e [toggle()](https://developer.mozilla.org/en-US/docs/Web/API/DOMTokenList/toggle).

<style>

        .blue{

*color*: blue; }

        .red{

*color*: red; }

</style>

<body>

<p id="text" class="texto">

        Lorem ipsum dolor sit amet consectetur adipisicing elit. Natus quis nobis.

</p>

<script>

        const texto = document.getElementById('text')

        texto.classList.add('blue') 

texto.classList.remove('texto') //vai remover a claas texto

      texto.classList.remove('blue') //vai tirar a cor azul

        texto.classList.add('red') // adicionando a cor vermelha



</script>

</body>

* Condições. Aninhadas

|  |
| --- |
| Aula12 ex12  var idade = 89  console.log(`Sua idade é ${idade}`)  if (idade < 16){      console.log('Voce NÂO VOTA.')  }else{      if (idade < 18 || idade >= 60){          console.log("Votação OPCIONAL")      }      else{          console.log("Voto OBRIGATÓRIO.")      }  }  Aula12 ex13  var agora = new Date //para mostrar as hora  var hora = agora.getHours() //exata q estiver executando  console.log(`Agora sao ${hora} horas`)  if (hora < 12){      console.log("Bom diaaa")  }else if (hora <= 17){      console.log("Boa tardee")  }else{      console.log("Boa noiteee")  } |

* Condições Múltiplas

|  |
| --- |
| Obrigatório o break  var agora = new Date() // dia atual  var diaSem = agora.getDay()  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("Sabado")          break  } |

Exercício de idades - aula12ex1

Para carregar o ano atual:

var data = new.data()

var ano = data.getFullYear() // ano atual

Exercício do horário - aula12ex2

Dicas:

Para adicionar imagens no JS:

img.src= ("fotos/manha.png")

ou

foto.innerHTML= '<img src="fotos/manha.png">'

Para carregar a hora atual:

var agora = new Date //para mostrar as hora

var hora = agora.getHours() //exata q estiver executando

Para mostrar o dia atual:

var agora = new Date() // dia atual

var diaSem = agora.getDay()

Para separar o codigo.js no final no body:

<script src="script.js"></script> <!--script separado tb-->

Para que o código JS execute assim que a página é carregada, usa-se Onload no HTML:

onload=" ()"

Colocando o nome do function do JS

function carregar()

Foi colocado no body: HTML:

<body onload="carregar()">

    <header>

        <h1>Hora do Dia</h1>

    </header>

    <section>

        <div id="msg">

            Aqui vai aparecera a hora

        </div>

        <div id="imagem">

            <img src="fotos/manha.png" alt="foto do dia">

        </div>

    </section>

    <footer>

        <p>&copy;Curso em Video</p>

    </footer>

    <script src="script.js"></script> <!--script separado tb-->

</body>

SCRIPT.JS :

function carregar() { //para carregar, coloquei no html(body)

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

    var foto = document.getElementById('imagem')

    var data = new Date()  //data atual

    var hora = data.getHours() //hora atual

    //var hora = 8

    if (hora >= 7 && hora < 12){

        msg.innerText = `Bom dia! Agora sao ${hora} horas.`

        img.src= ("fotos/manha.png")

        document.body.style.background = '#FEDC5A'

    } else if (hora >= 12 && hora <= 18){

        msg.innerHTML = `Boa tarde! Agora sao ${hora} horas.`

        foto.innerHTML= '<img src="fotos/tarde.png">'

        document.body.style.background = '#f59629'

    } else{

        msg.innerHTML = `Boa noite! Agora sao ${hora} horas.`

        foto.innerHTML= '<img src="fotos/noite.png">'

        document.body.style.background = '#1e4c5c'

    }

}

### **Repetições**

* Estrutura de while

|  |
| --- |
| .  //esse faz o teste antes e executa o comando  Aula13  var c = 1  while (c <= 6) { //enquanto o contador der 6..      console.log(`Passo ${c}`)      c++ //c = c + 1  } |

* Estrutura Do while

|  |
| --- |
| .  Aula13  //esse executa o comando e depois faz o teste  var c = 1  do {      console.log(`Passo ${s}`)      c++  } while (c <= 6) |

* Estrutura de for

|  |
| --- |
| Aula14    /\*var c = 1  while(c <= 5) {      console.log(c)      c++  }\*/  for (var c = 1 ; c<=10; c++){      console.log(c)  } |

Aula 14 ex 2

function tabuada() { //para carregar, coloquei no html

    var num = document.getElementById('txtn')

    var tab = document.getElementById('seltab')

    if (num.value.length == 0) {

        window.alert('Por favor, digite um numero!')

    } else {

        var n = Number(num.value)

        var c = 1

        tab.innerHTML = '' //antes de mostrar a tabuada, limpe a tabela antes

        while (c <= 10){

            var item = document.createElement('option') //criar um elemento de opçoes

            item.text = `${n} x ${c} = ${n\*c}`

            tab.appendChild(item) //adicionar os itens

            c++

        }

    }

}

## **Aula 15 – Variáveis Compostas usando colchetes []**

Uma variável composta tem vários elementos, cada elemento é composto por um valor e por uma chave de identificação.



let num = [8, 4, 3, 1, 9]

num [3] = 6 //adicionar na posição 3 o nº 6

num.push(7) //adicionar no final

console.log(num)

console.log(`Nosso vetor é ${num}`)

console.log(`O vetor tem ${num.length} elementos`) //para ver quantos elementos tem

console.log(`Deixando em ordem crescente ${num.sort()}`)

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

/\*for (let pos = 0; pos < num.length; pos++){ //enquanto ele n chegar no final do vetor..

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

}\*/

// Mais simplificado:

for(let pos in num) { //para cada posição dentro de num

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

}

var pos = num.indexOf(7) //vai mostrar em que posição esta

console.log(`O valor 7 esta na posição ${pos}`)

## **Aula 16 – Funções**

* São ações executadas assim que são chamadas ou em decorrência de algum evento.
* Uma função pode receber parâmetros e pode retornar um resultado.

function parimp(n) {

    if (n % 2 == 0) {

        return `O numero ${n} é PAR` EX1

    } else {

        return `O numero ${n} é IMPAR`

    }

}

let res = parimp(1127)

console.log(res)

let v = function(x) { //uma variavel recebe uma função EX2

    return x\*2 //o dobro

}

console.log('O dobro de 5 é', v(5))

console.log('O dobro de 7 é', v(7))

function fatorial(n){

    let fat = 1

    for (let c = n; c > 1; c--) {

        fat \*= c EX3

    }

    return fat

}

console.log(fatorial(5))

let msg = ('O fatorial de 5 é '+ fatorial(5))

console.log(msg)

## Colocando tabela pelo JS e aparecer no HTML **Ex: Documents\Projetos HTML e CSS\Musculacao**

var container = document.getElementById("container");

container.innerHTML = [

'<table>',

'<thead>',

'<tr>',

'<th>id</th>',

'<th>col1</th>',

'<th>col2</th>',

'<th>col3</th>',

'</tr>',

'</thead>',

'<tbody>',

'<tr>',

'<td>1</td>',

'<td>data</td>',

'<td>data</td>',

'<td>data</td>',

'</tr>',

'<tr>',

'<td>2</td>',

'<td>data</td>',

'<td>data</td>',

'<td>data</td>',

'</tr>',

'<tr>',

'<td>3</td>',

'<td>data</td>',

'<td>data</td>',

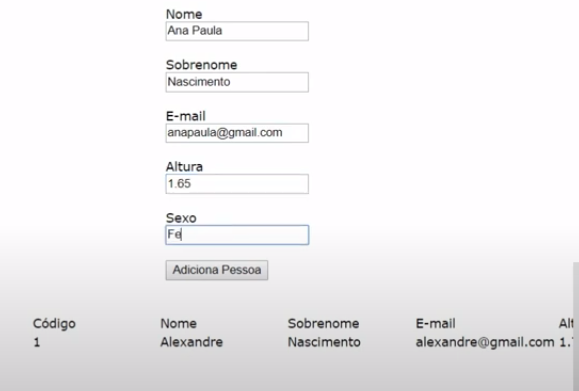
'<td>data</td>',

'</tr>',

'</tbody>',

'</table>'

].join("\n");



## Manipulando Tabelas com Javascript

<https://www.youtube.com/watch?v=ImkWbhXnISM>

Onde o usuário insere os dados solicitados no

formulário da pagina em HTML. Mostrando na tabela

quando clicar em adicionar.

## Busca em tabela com Javascript

<https://www.youtube.com/watch?v=HWZfqk6gSfI>

Onde o usuário quer fazer uma buscar de nom

na tabela