Curso de JavaScript

Modulo A

Assistir novamente aulas 1 e 2 e fazer anotações

Referência Biblioteca de javaScrip aula 3-

Aula-4

Comandos em JS window.alert (Uma janela que da um aviso)

Window.confirm(semelhante ao alert que da opção de ok ou confirmar)

Window.prompt(interação com o usuário dando a opção de ele interagir)

Modulo B-Aula 5

Tipos de dados e Variáveis – variáveis em JS tem uma pequena diferença.

Usar comentários na codificação para que ele seja bem documentado, com o passar do tempo vc não esquece.

No JS usa 2 simbologias (2 barras //) 2 barras 1 linha ou (/\* \*/) barra e asterisco mais de uma linha.

Comentário também ajuda outra pessoa a ver seu código de maneira mais claro...

Depuração= usa se para ver como seu comando está funcionando

VARIÁVEIS= imagina um estacionamento de carros, vc não pode simplesmente colocar todo tipo de automóveis lá se não congestiona, vc tem que delimitar o espaço criando vagas e numera-las, identificando elas, se vc cria 6 vagas para carros numere as, de 1 a 6, se criar 6 vagas pra caminhonetes identifica c1 a c6 e se for 6 para motos identifica com m1 a m 6. Vc organizou espaços específicos para diferentes tipos de automóveis...

Vaga 1= carro 1 (sinal de = igual é recebe em Js)

Fisicamente não tem como 2 carros usar o mesmo espaço, tem que tirar um e colocar o outro, vaga 1 = 2 ou também vc pode deixar a vaga vazia...vaga1 = null (nula, vazia, desocupada) O mesmo se dá para computadores o estacionamento é a memória, os veículos são os dados, que no espaço em que estão também precisam ser delimitados, identificados como (variáveis) var1, var 2 e var 3 assim por diante e esses espaços ou variáveis recebem = valores em cada espaço ...5, 8,6, 15 esses valores recebem o nome de atribuição.

Essas delimitações também recebem tamanhos diferentes como nos estacionamentos.

Recebem veículos de tamanhos diferentes que são chamadas de string s1, s2, s3 que recebem palavras (caracteres)

S1 = JavaScript

S2 = Sérgio dos Santos

S2 = aula 5 variáveis

No Js a string tem que estar entre aspas (””) que tem 3 tipos, aspas simples, aspas duplas e apócrifos.

O nome das variáveis chama se de identificador que tem regras de nomeação

1. Pode começar com letra ou cifrão

b-não pode começar com número

c- É possível começar com letra e número

d- É possível usar acentos e símbolos

e- Não podem conter espaços

f- Não podem ser palavras reservadas- palavras que o Js usa como comando

Tipos primitivos principais

Números

Strings

Boolean

Armazenamento de dados

Operações com dados

Aula 6

Tratamento dados

Data types (tipo de banco de dados)

Number

-Infinity

-NaN

Strings

Boolean

Null

Undefined

Object

-Array

Function

<script> window.alert // este comando emite uma mensagem!

window.confirm // este comando faz uma pergunta de confirmação

window.prompt // este comando faz um pergunta de resposta!')

 //O sinal de + é concatenação...

1. No Js se vc colocar 4+2 o resultado será 42 porque o sinal de + é uma concatenação
2. (Relacionamento de ideias, fatos ou coisas entre si; ligação, encadeamento. Harmonização, conciliação.)
3. O Js sempre te fornecera uma string. Nesse caso vc tem que transformar a strings para número

Como converter string para número? Ou Conversão de tipos

Existem várias formas...

1 – Number.parseInt(n) ou só parseint funciona igual. Para número inteiro.

2- Number.parseFloat(n) para números reais com virgula, float = ponto Flutuante o ponto que separa os números, 5.5. 10.4 e parse é converter, parear

Quando o valor que vc pode receber é relativo ou opcional vc simplifica usando somente o Number (n) porem quando vc sabe qual o tipo exclusivo que vc quer use os...

Número para String

Na conversão de número para string usa se os comandos

String (n)

n.tostring() coloque o que deseja converter com essa função

Formatação de Strings

Var s = ‘JavaScript’

‘Eu estou aprendendo s - Errado’//Não faz interpolação

‘Eu estou estudando’ + s – certo //Usa se a concatenação

Nome = ‘Sérgio’

Idade = 40

Nota = 5.5

‘O aluno’ + nome + ‘com’ + idade + ‘anos, tirou a’ + nota

Porem nas formais recente de JavaScript tem formas mais legal de trabalhar...

Técnica Template String

Eu estou aprendendo ${s} //${s} nomenclatura placeholder //Usa se formatador de String, no Template string se usa crase em vez de aspas simples

‘O aluno’ + nome + ‘com’ + idade + ‘anos, tirou a’ + nota

//Modo antigo e mais complexo

`O aluno ${nome} com ${idade} anos tirou a nota, ` ${nota}` //com placeholder mais simplificado.

aqui no node

Window.alert(`A soma entre ${n1} e ${n2} e igual a ${s}`)

Aqui no VsCode

<script>

       var nome = window.prompt('Qual é o seu nome')

document.write(`Seu nome tem ${nome.length} letras <h2>`)

//Esse comando length diz o tamanha, quantas letras tem...vc também pode colocar tegs como representado o <h2> o que não e muito legal mas pode

    </script>

<style>

        body{

            font: normal 20pt Arial;

        }

    </style> /\*Essa é a maneira correta de se configurar fonte no css\*/

 document.write(`Olá, ${nome}. Seu tem ${nome.length} letras! `) //Esse comando diz o tamanha ou a quantidade de letras

 document.write(`Seu nome em maiúsculo é ${nome.toUpperCase()}`)

Olá, Sergio. Seu nome tem 6 letras!

 Seu nome em maiúsculo é function toUpperCase () {[código nativo]}Esse comando da esse resultado

Usa se o <br> para quebra de linha e também pode colocar <strong>${nome}</strong> para por em negrito

<script>

       var nome = window.prompt('Qual é o seu nome')

        document.write(`Olá, <strong>${nome}</strong>! Seu nome tem ${nome.length} letras.<br>`)

        document.write(`Seu nome em maiúsculas é ${nome.toUpperCase()}<br/>`)

    </script>

Olá, **Sergio**! Seu nome tem 6 letras.

Seu nome em maiúsculas é SÉRGIO

Com a tag <br/> vc quebra a linha ao termino da frase

${nome.toLowerCase()}

Seu nome em minúsculas é: SÉRGIO

Para abri o node, ctrl + Shift + ‘

Formatação de números

**Node-**

var n1 =1545.5

N1

1545.5

Se querer escrever para 2 casas decimais

N1.toFixed(2)

1545.50

Se querer mudar ponto para virgula

N1. toFixed(2).replace(‘.’ , `,`)

Como fazer um comando que mostra o salário de uma pessoa?

N1.toLocaleString(‘pt-BR’ , {style: ‘currency’, currency: ‘BRL’})

R$ 1545,50

N1.toLocaleString(‘pt-BR’ , {style: ‘currency’, currency: ‘USD’}) Em dólar

U$$ 1545.50

> n1.toLocaleString('pt-BR', {style: 'currency', currency: 'EUR'})

'€ 1,545.50'

Operadores em Js.parte 1 –Aula 7 e 8

**Operadores**

1 Aritméticos (+, -) ,

Multiplicação ( \* )

Divisão real ( / )

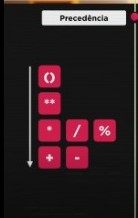
Resto da divisão (%) exp. 5%2 = 1🡨 resto da divisão Potencia(\*\*)



Na programação a divisão vem primeiro 5+3/2 = 6.5, se você quer a soma primeiro (5+3)/2=4, tem que colocar entre parentes

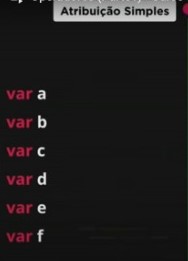
Quando vc programa a linguagem só mostro o erro se for erro sintático de escrita, as vezes quando vc não recebe mensagem de erro não significa que este certo, cuidar pra não dar o comando errado

Com isso vamos a ordem de precedência dos operadores...

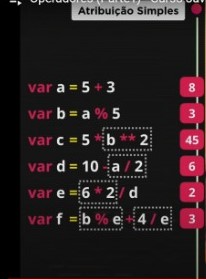


Quando \* / % estiverem numa mesma conta se resolva da esquerda para a direita

2 Atribuição



Ordem de precedência com atribuição simples

-

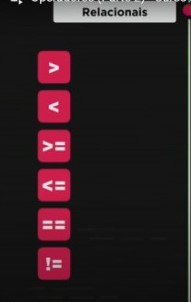
Pode se fazer também as auto atribuições exp. Var n = 3

N = n + 4 == 7 o valor da soma passa a ocupar o valor da variável, isso chama se auto atribuição...n = n – 5 == 2.

Pode também simplificar essas auto atribuições reescrevendo assim var n = 3, n += 4 mas vc só pode fazer desde que a variável receba ela mesmo, o próprio valor se for outro não consegue a maioria das linguagens aceita esse termo... e todos os termos aritmético da mesma forma n -=5, n\*= 3, n/=2...

Outro auto atribuição ainda simplificadora ainda maior que é chamada de **incremento** var x = 5, x = x +1==6, x = x – 1 == 5 simplificando x+= 1 ==6, v=v-1==5 INCREMENTANDO VAR X = 5 X++, X--

3 relacionais



5 > 2 == True

7 < 4 ==false

8 >= 8==true

9 <= ==7false

5 == 5 == true

4 != 4 ==false

De todas essas expressões com relacionais o resultado será sempre booleano verdadeiro (V) ou falso (F) (!=)esse símbolo significa não igual ou diferente

Exemplos

Preço == 200.50

Idade < 18

Curso == JavaScript

N1 != n2

Quando houver operadores aritméticos e relacionais na mesma linha primeiro se resolve os aritméticos

Var a = 8

Var b = 15

A < b ==true

a <= b +10

== false

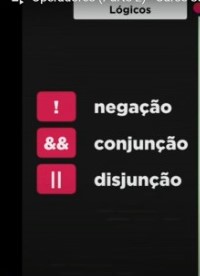
Operadores relacionais de Identidade o JavaScript não compara o tipo 5 == 5 true o que o script faz e analisar se o 5 tem o mesmo valor que o outro, não o tipo, mas a grandeza, valor...

5 ==´5’ true

No Caso de comparação de identidade restrita usa se o símbolo === desigual restrito, ele analisa o valor e o tipo 5 === ‘5’ false

5===5 true

**4 Lógicos**

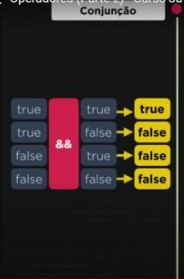


Os lógicos são 3

! Exclamação que representa negação (não),Tratado como operador unário - !true, !false. Algo que não é verdadeiro e falso e algo que não é falso é verdadeiro...

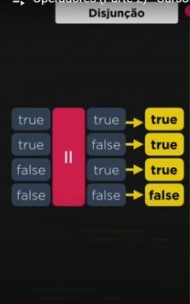


2 && e comerciais, significa conjunção (e) Operadores binário com 2 operadores lógicos um de cada lado dando o resultado lógico de cada expressão,



Só satisfaz se as duas expressões forem verdadeiras

|| papy disjunção(ou) também é um operador binário com 2 operadores lógicos um de cada lado, dependendo das variações também se obtém um valor lógico



Basta de tenha um valor verdadeiro que os outros são verdadeiros a menos que os dois sejam falsos...

Se numa mesma expressão tiver um e um não ou um ou a ordem de execução é 1º não depois e depois o ou...



Aritméticos primeiro depois de todos, os relacionais que não tem ordem de precedência quem aparecer primeiro será feito primeiro, leitura da esquerda para a direita e após os lógicos que tem ordem de precedência primeiro o !não, depois o (E&&) e após (|| OU)...

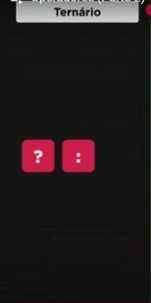
Exemplos:

Idade >= 15 && idade <= 17 //A idade está entre 15 e 17?

Estado ==’RJ’ || estado ==’SP’ //O estado é RJ ou É sp?

Salário > 1500 && sexo != ‘m’ //o salário é acima de 1500 e não é homem

5Ternário

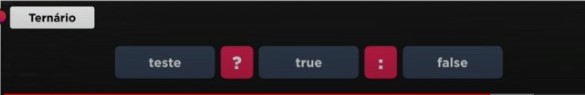


Ternários são compostos por 2 símbolos que aparecem na mesma expressão (?) (☺) isso é 2 pontos. Se chama ternário porque tem 3 operanos.

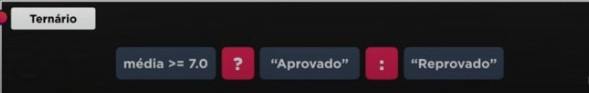
**Teste ? true : false**

Teste é um teste lógico que dá verdadeiro o falso

O símbolo de interrogação vai quando o teste for verdadeiro. E no final o que vai acontecer quando o teste lógico for falso...



Exemplo...



Var media = 5.5

Media > 7 ? ‘APROVADO’ : ‘REPROVADO

==REPROVADO

Var x = 8

Var res = x % 2 == 0 ? 5 : 8

Sintaxe da expressão, se a expressão receber 0, for verdadeira **então**: ela recebe 5 e **Se? Recebe 9**

X == 8

Res ==5

Var idade = 19

Var r = idade >= 18 ? ‘MAIOR’ : ‘MENOR’

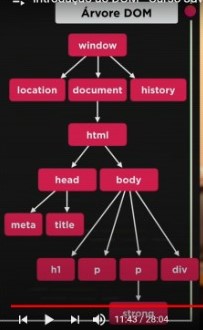
Expressão Se a idade for maior ou menor que 18 então ela recebe maior se não recebe menor...

Aula 9

**Modulo C-Entendendo o DOM**

Document Object Model == DOM / Modelo de objetos para documentos basicamente um conjunto de objetos dentro do seu navegador que dá acesso aos componentes internos do seu website ele só funciona quando rodamos JavaScript dentro do navegador...compreender e fazer a arvore DOM

Arvore DOM



A arvore **dom** começa pela raiz dentro do navegador essa raiz chama se window / janela. Tudo dentro do JavaScript está dentro do window, dentro do window contém vários outros objetos, exemplos **location. Document, history...**

**Location**-diz sobre a localização do website...

**Document** –dentro do document existe o html-dentro do html tem head e Bady quem está dentro ou a baixo é filho e quem está a baixo ou fora é pai...dentro de head tem meta, title e outros. Dentro de Bady tem h1,p, div e outros mais...

**History**—diz se sobre de onde vc vem e para onde vc vai facilitando a navegação...

Pode se ter acesso a vários componentes utilizando diretamente dom dentro do JavaScript

Pode se navegar dentro da arvore DOM da maneira que achar melhor e por isso existe várias maneiras de navegar entre os **elementos** é todo e qualquer coisa que aparece na arvore DOM são ditos elementos e vc pode selecionar esses elementos para poder navegar dentro da arvore DOM, e existem vários métodos para isto...exemplos abaixo

window.document.write(window.document.charset)

UTF-8

indow.document.write(window.navigator.appName)

Netscape

 window.document.write(window.document.URL)

http://127.0.0.1:5500/aula6/ex005.html

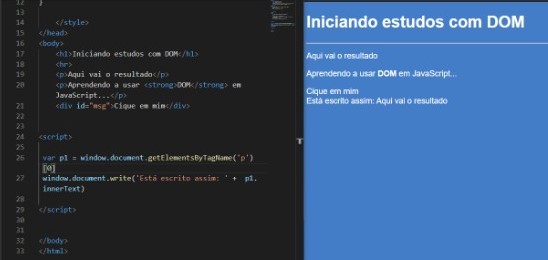
 window.document.write('Olá, Mundo!')

Olá, Mundo!



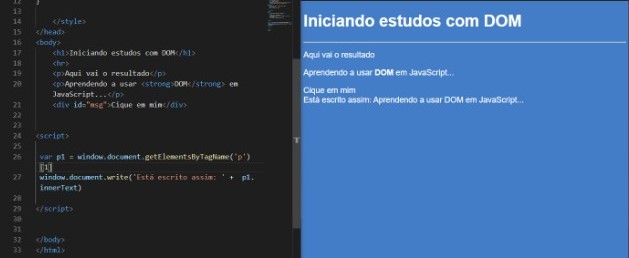
Manipulando DOM

Usando o elemento por marca, tag, você pode selecionar mais de 1 objeto...por exemplo



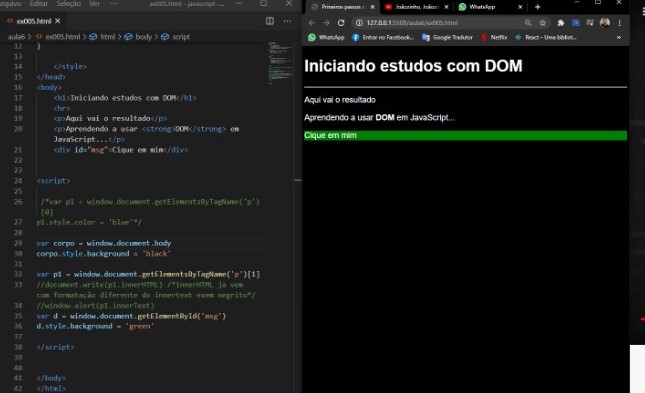
Como havia 2 parágrafos você pode selecionar qual deles chamar, começando pelo zero, veja o que acontece quando se troca o zero pelo 1 em colchetes, muda a escrita no navegador, o computador lê zero como um...

Innertext é o texto que esta dentro dos parágrafos...



Pode se fazer o acesso a todos os componentes utilizando getElementByTagName, mas essa não é a única técnica, também pode se utilizar objetos quando site for mais evoluído, por id, nome se tiver uma propriedade name, quando tiver uma propriedade no plural usar o [colchete] para selecionar o objeto

Por getElementById



d.innertext = ‘Estou aguardando...’ mudando a escrita dentro da div.

Também escrever o comando completo...

Window.document.getElementById(‘msg’).innertext = ‘Olá’

Se você trocar a id da div por nome ou classe também funciona.

<div name=’msg’>Clique em mim</div>

Var d = window.document.getElementByName(‘msg’)[0]

d.innertext = ‘Olá!’ ..também por classe, dependendo da situação do elemento...tudo mesmo princípio, a forma de acesso é a que você prefere

Aprendendo o querySelector()

QuerySelectorAll()

Nova forma, recomendada pela maioria dos manuais...

Toda div é representada por uma # hechtag e toda classe é representada por um( . ) Pontinho...

 <div id="msg">Cique em mim</div>

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

d.style.color = 'blue'

Cique em mim

O querySelector fica mais prático quando se faz alterações mais rápidas no site...querySelector é mais recente navegadores mais antigo talvez não terão suporte para ele...

Evento em JavaScript - Os **eventos** são basicamente um conjunto de ações que são realizadas em um determinado elemento da página web, seja ele um texto, uma imagem, ou uma div, por exemplo. Muitas das interações do usuário que está visitando sua página com o conteúdo do seu site podem ser consideradas **eventos**.

**Funções** são blocos de construção fundamentais em **JavaScript**. Uma **função** é um procedimento de **JavaScript** - um conjunto de instruções que executa uma tarefa ou calcula um valor. Para usar uma **função**, você deve defini-la em algum lugar no escopo do qual você quiser chamá-la.

**Eventos DOM Aula 10**

Evento é todo aquilo que possa acontecer com um elemento, por exemplo uma div.-

Com apenas o movimento do mouse dentro de um elemento por exemplo uma div você pode disparar uma série de eventos (efeitos)



Pesquisar por Event referência na web...

<https://developer.mozilla.org/pt-BR/docs/Web/Events>

Uma função ou uma funcionalidade é um conjunto de códigos(conjunto de linhas) que serão executadas somente quando o evento ocorrer, por exemplo esse conjunto de linhas ou códigos são chamados de blocos...

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

        interaja...

    </div>

Ao invés de fazer isso no html vc pode puxar os elementos direto do script assim:

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

        a.addEventListenner('click', clicar)

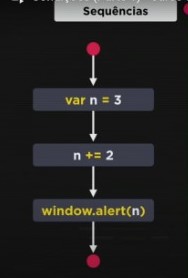
        a.addEventListenner('mouseover', entrar)

        a.addEventListenner('mouseout', sair)

Se você errar código em JS, olhar no devetools.

Dica: se você digitar div[#area](https://www.youtube.com/hashtag/area) e apertar tab ele cria automático <div id="area"></div>. Isso funciona tbm para classes e para tags personalizadas!

Modulo D Condições em JavaScript Aula 11



Essa é uma sequência em JS quando a função é executada de cima para baixo seguindo ordem de leitura...

Porem tem alguns momentos que vc não quer executar as funções na ordem de sequência, nesse caso tem as **condições**

Condição é quando uma função começa a ser executada e você precisa de opções para ir por um caminho ou outro, ele te dá a opção de escolha, caso aconteça uma coisa vc vai por um caminho, caso aconteça outra vc vai por outro caminho...esse disvio que acontece de resolver o problema e continuar o caminho chama se de **desvio condicional;** esse desvio faz parte dos fundamentos da programação, exitem vários tipos de desvio condicional.

Uma condição começa a ser representada por **if = se,** A condição if é criada por um bloco- um bloco é uma abertura e fechamento de chaves { } . Uma condição pode ter várias blocos ou comandos

**If** (condição) {

|  |
| --- |
| true |

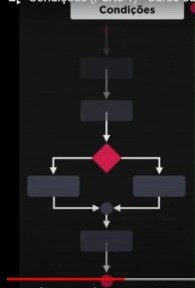
}

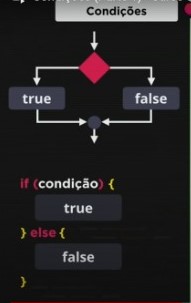
E o outro bloco é o :

**else = senão**{

|  |
| --- |
| false |

}





Condições é uma estrutura de controle muito importante pra programação...

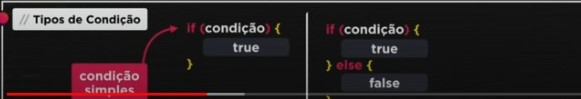
Existem vários tipos de condições, vamos ver uma segunda maneira:

A condição que só tem um tipo de bloco: if (condição) {

true

}

Essa estrutura por ser simples é chamada de **condição simples!**

****

Próxima aula 12, fazer exercício 9 em forma de site