minhas anotações ate agora:

Comandos em JS:

<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!')

</script>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Variáveis:

Como Criar variáveis, Ex: var nome ou let nome

para uma variável receber um valor usamos:

var nome = Gustavo // desta forma criamos uma variável e ao mesmo tempo demos uma valor a ela

nome = Gustavo //desta forma apenas demos um valor para a variável que ja foi criada

Regras 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 pode conter espaços

Não pode usar palavras que são comandos

Dicas para nomes das variáveis:

Maiúsculas e Minúsculas fazem a diferença!

Tente escolher nomes coerentes a função da variável. Ex: Variável que vai armazenar a idade, coloca o nome dela de “idade”

Evite se tornar um “Programador Alfabeto” ou um “Programador Numérico”. Ex: Não usar os nomes das variáveis como “a”, “b”, “c”, etc; ou “a1”, “a2”, “a3”, etc.

Tipos de Dados das Variáveis: (Tipos Primitivos, lembrando que existem muitas outras!)

Numbers;

Strings;

Boolean;

// Numbers: 1; -2; 4.5; 6.555 -> Basicamente números

// Strings: Maria, Google, Joao, pedreiro, (seu CPF) -> Basicamente cadeia de caracteres

// Boolean: True; False

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Transformando uma string em um number

var n1 = Number.parseInt (window.prompt ('digite aqui um numero!'))

var numero1 = Number.parseFloat (window.prompt ('digite aqui um numero!'))

var numero1 = Number (window.prompt ('digite aqui um numero!’))

Mas qual é a diferença entra “Number.parseInt”, “Number.parseFloat” e Number?

// Number.parseInt: Numero Inteiro

// Number.parseFloat: Numero com virgula

// Number: Js vai decidir qual é

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Transformando um number em uma string

window.alert ('a soma dos numeros é: ' + soma.toString()) // Jeito mais antigo

ou

window.alert ('a soma dos numeros é: ' + String(soma)) // Jeito mais simples

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Formatando Strings:

var teste = 'java script’

‘eu estou aprendendo’ + teste

`eu estou aprendendo ${teste}` -> não esqueça de usar crase!

teste.length // conta quantos caracteres tem na variável

teste.toUpperCase // coloca tudo em caixa alta

teste.toLowerCase // coloca tudo em minúsculo

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Formatando números:

Var n1 = 1543.5

n1.toFixed(2) // Coloca em duas casas decimais (para colocar em mais ou menos casas troque o numero entre parênteses)

n1.toLocaleString( ‘pt-BR’,{style: ‘currecy’, currency: ‘BRL’} ) // Coloca o R$ na frente do numero (pode trocar entre outras moedas)

n1.replace (‘.’, ‘,’) // Troca o ponto pela virgula

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Operadores:

Tipos de operadores que vamos estudar:

Aritméticos

Atribuição

relacionais

lógicos

ternarios

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Operadores Aritméticos:

+ // Somar

- // Subtrair

\* // Multiplicação

/ // divisão

% // Resto de uma divisão

\*\* // Potencia do primeiro numero elevado ao segundo

5 + 3 = 8

5 - 3 = 2

5 \* 3 = 15

5 / 3 = 1,6

5 % 3 = 2

5 \*\* 3 = 125

Precedencia dos operadores↓

( )

\*\*

/ \* %

+ -

Auto Atribuições: Forma Simplificada:

var n = 3 var n = 3

n = n + 4 // ele vai somar ele mesmo a 4 | n += 4

n = n - 5 //ele vai pegar ele mesmo e subtrair 5 | n -= 2

n = n \* 4 //ele vai pegar ele mesmo e multiplicar por 4 | n \*= 5

n = n / 2 //ele vai pegar ele mesmo e dividir por 2 | n /= 2

n = n \*\* 2 //ele vai pegar ele mesmo e elevar a 2 potência | n \*\*= 2

n = n % 5 //ele vai pegar ele mesmo, dividir por 5 e dar o resto | n %= 5

outra simplificação:

n++ // é a mesma coisa que n = n + 1 ou n += 1

n— //é a mesma coisa que n = n - 1 ou n -= 1

++n // ele vai somar antes

—n // ele vai diminuir antes

Operadores Lógicos

! //representa negação

&& //representa conjunção ‘e’

|| //representa disjunção ‘ou’

Operador ternário

‘Teste’ ? ‘ação se True’ : ‘ação se false’

O que é DOM?

DOM (Document Object Model), Modelo de Objeto para Documento, é uma \*INTERFACE padronizada utilizada para REPRESENTAR um DOCUMENTO HTML. Este modelo utiliza uma linguagem neutra que, por meio de uma árvore hierárquica de diretórios, permite você ACESSAR e INTERAGIR com QUALQUER ELEMENTO de seu DOCUMENTO de maneira DINÂMICA com o uso do JavaScript. Desta forma, o JS tem o poder de:

alterar, excluir e adicionar: elementos HTML, atributos HTML, eventos HTML, estilos CSS.

\*Interface, do latim INTER(entre) + FACIES(faces), significa: ligação física ou lógica entre 2 sistemas ou partes que não poderiam ser conectados diretamente.

Ou seja, o DOM é o INTERMEDIADOR da COMUNICAÇÃO da linguagem JavaScript com as tecnologias HTML5 e CSS3.

Ilustração de uma árvore DOM:

window

/ | \

location document history

|

html

/ \

head body

/ | / | | \

meta title h1 p p div

|

strong

Parent (pai) - é quem está ACIMA na árvore HIERARQUICA.

Child (filho) - é quem está ABAIXO na árvore HIERARQUICA.

É como se fosse uma árvore genealógica. Exemplos:

HEAD e BODY são FILHOS de HTML;

HTML é PAI de HEAD e BODY;

HTML é FILHO de DOCUMENT.

É importante ressaltar que: um DOCUMENTO HTML e sua REPRESENTAÇÃO DOM são a MESMA COISA. A única diferença é que estes estão REPRESENTADOS de FORMAS DIFERENTES. Ex:

("uma garrafa") e ("a bottle") são a MESMA COISA, só que representadas em línguas diferentes. É a mesma ideia.

Caso você altere o DOCUMENTO HTML você também alterará o seu DOM e vice-versa.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Como ACESSAR/SELECIONAR ELEMENTOS através do DOM:

Lista de 5 MÉTODOS de acesso:

by Tag - getElementsByTagName("")

by ID - getElementByID("")

by Name - getElementsByName("")

by Class - getElementsByClassName("")

by Selector - querySelector (""); querySelectorAll ("");

O método getElementsByTagName() SELECIONA ELEMENTOS via TAG.

O método getElementById() SELECIONA UM ELEMENTO via ID.

O método getElementsByName() SELECIONA ELEMENTOS via NOME.

O método getElementsByClassName("") SELECIONA ELEMENTOS via CLASSE.

O método querySelector() SELECIONA UM ELEMENTO via SELETOR CSS.

O método queySelectorAll() SELECIONA ELEMENTOS via SELETOR CSS.

\*Atente-se que alguns dos seletores estão no SINGULAR e outros no PLURAL (Element e Element\*S\*)

Caso utilize um seletor que está no PLURAL, para ESPECIFICAR qual TAG, NAME ou CLASS você deseja ACESSAR, você têm que usar, após os parênteses (), os colchetes [] e, dentro destes, você indicará qual a POSIÇÃO do elemento que você deseja ACESSAR.

Ex da aula:

document.getElementsByTagName('p')[1];

Neste exemplo selecionou-se, dentro dos parênteses, todas as tags <p>. E, dentro dos colchetes, selecionou-se a tag <p> que se encontra na 1° POSIÇÃO, ou seja, a 2° tag <p>.

\*Lembrando que a contagem começa em 0. Portanto:

a posição 0 é a 1° tag <p>

a posição 1 é a 2° tag <p>

a posição 2 é a 3° tag <p> ...

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Diferença entre .innerText e .innerHTML :

A propriedade .innerText serve para DEFINIR ou RETORNAR o CONTEÚDO de TEXTO de um ELEMENTO.

A propriedade .innerHTML serve para DEFINIR ou RETORNAR o CONTEÚDO HTML (texto+formatação) de um ELEMENTO.

ILUSTRAÇÃO (crie um outro documento, monte a estrutura básica HTML5 (!+enter) e insira este código dentro do body):

<body>

<h1>Diferença entre innerText e InnerHTML:</h1>

<p class="teste\_1"></p>

<p id="teste\_2"></p>

<script>

var variavel\_1 = document.querySelector('.teste\_1')

variavel\_1.innerText = "Teste do <strong>innerText</strong>"

var variavel\_2 = document.querySelector('#teste\_2')

variavel\_2.innerHTML = "Teste do <strong>innerHTML</strong>"

</script>

</body>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ALGUMAS DEFINIÇÕES:

\*Um OBJETO no JS é um CONTAINER de PROPRIEDADES e MÉTODOS.

\*Uma FUNÇÃO no JS é um subprograma que pode receber PARÂMETROS e RETORNAR um resultado. São AÇÕES executadas assim que são CHAMADAS ou em decorrência de algum EVENTO. É um bloco de códigos projetado para executar uma tarefa ou calcular um valor.

\*Um MÉTODO é uma FUNÇÃO que é ARMAZENADA como PROPRIEDADE de um OBJETO. Ou seja, MÉTODOS são AÇÕES que podem ser EXECUTADAS em OBJETOS.

\*Um EVENTO no JS é uma CONDIÇÃO que, quando verdadeira, fará a CHAMADA de uma FUNÇÃO.

\*O Seletor de Classe é representado por: .

\*O Seletor de ID é representado por: #

* Aula 11

Estrutura Condicional Simples: IF (se)

Se (if) condição1 for VERDADEIRA, execute o Bloco A.

Ilustração:

if (condição1) {

Bloco A

}

Se condição1 for TRUE, execute o bloco TRUE (Bloco A).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Estrutura Condicional Composta: IF/ELSE (se/senão)

Se (if) condição1 for VERDADEIRA, execute o Bloco A.

Senão (else), execute o Bloco B.

Ilustração:

if (condição1) {

Bloco A

} else {

Bloco B

}

Se condição1 for TRUE, execute o bloco TRUE (Bloco A).

Senão for TRUE, execute o bloco FALSE (Bloco B).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

O método console.log() serve para ESCREVER (log) uma MENSAGEM no console.