Operadores lógicos e matemáticos:

< menor que

> maior que

>= maior ou igual que

<= menor ou igual que

Variáveis e constantes:

Let – variável de bloco

Var – variável de escopo e funções

Const – constantes

Estruturas de dados:

Array – let x = Array()

Object = let x = {nome:” jhon”, idade:12}

Condicionais:

If(condicao){verdadeira}else if(condicao){verdeiro}

Laços de repetição:

for(let i=1;i>10;i++){bloco}

For in - for(let chave in objeto){a chave é o acesso do object, não recomendável usar em array}

For of – for(let valor of array){Retorna o valor em vez da chave.}

While – while(condicao){Bloco a ser executado}

Do while – do{bloco a ser executado}while(condição)

DOCUMENT OBJECT MODEL

Seletores:

Document.getElementById(“id da tag”) – Pega o ID

Document.getElementsByTagName(“nome da tag”) – Pega todas as tags com o nome fornecido

Document.getElementsByClassName(“Nome da Classe”) – Pega todas as tags que contém a classe fornecida

Document.querySelector(‘tag, classe, id’) – pega com base ou na classe, id, ou a tag

Document.querySelectorAll(‘tag, classe, id’) – pega todos os elementos com base na classe, tag ou id e retorna um array.

MÉTODOS DOM

Array.map(elemento, indice) – Percorre todas as coleções, e é mais eficiente que o laço for, e pode utilizar uma função anônima como parâmetro.

Array.filter(valor, índice, array) – Percorre o array, e faz uma filtragem e retorna determinados elementos, com base em uma condição e pode utilizar uma função anônima.

FUNÇÕES:

Funções Factory: são funções que carregam com si um objeto literal e métodos.

Ex: let funcaoFactory = {nome: ‘jorge’, idade: ‘idade’, falar(método)}

Funções Construtoras: são funções que carregam o operador this.

Ex: let funcaoConstr = function(){ this.nome = ‘Jorge’ this.idade = ‘19’ this.falar = function(){this.funcao}}

Funções Anônimas: são funções que são declaradas dentro de variáveis ou dentro de outras funções.

Ex: let arrowFunction= ()=>{bloco de funçao}

Prototype: são heranças que podem ser herdados de outras funções, já que todas advem do mesmo object. Para criar um novo atributo > Object.prototype.atributo = valor.

Ex: let animal = {atributo: ‘a’}

Let vertebrado = {\_\_proto\_\_: animal, atributo: ‘b’}

Operador Spred: ele é utilizado para espalhar algum conteúdo dentro de um array, objeto.

Ex: let x = ‘olá mundo’ ...x > espalha o conteúdo do x

Operador Rest: ele é utilizado para unir algum conteúdo dentro de uma função.

Ex: function soma(...parametro){let resultado = 0 return resultado}

Operador Destructuring Assingnment: desestrutura um array ou um objeto

Ex Array: let frutas = [1, 2, 3, 4] let [a, b, c=5] = frutas > o a e b são variáveis.

Ex Objeto: let produtos = {a: ‘1’, b: ‘2’, detalhes: {fab: ‘4’}} let {a, b} = produtos