repositório github - <https://github.com/luizomf/curso-js>

* Aula 03

console.log("Rodrigo de 'Souza'"); //String

console.log('Rodrigo de "Souza"'); //String

console.log(`Rodrigo de Souza`);   //String

console.log(123456, 15.85);        //Number

texto – aspas simples ou duplas

entre as crases(templates strings) - console.log(`tem ${altura}`);

números – sem aspas

ponto e vírgula opcional

* Aula 06

let nome; //underfined

let nome2 = 'Hayanne'

let filho = 'Valentina'

nome = 'Rodrigo'

// Não podemos criar variáveis com palavras reservadas

(let, console, if)

// Variáveis precisam ter nomes significativos(nomeDoAluno, nome\_cidade)

// Não pode começar o nome de uma variável com um número (1nome, 20idade)

// Não podem conter espaços ou traços (nome cliente, nome-cliente)

// CERTO seria (nome\_cliente ou nomeCliente)

// Case-sensitive, nome e diferente que Nome

// Não podemos redeclarar variáveis com let

* Aula 07

// Não podemos criar constantes com palavras reservadas

(let, console, if)

// constantes precisam ter nomes significativos(nomeDoAluno, nome\_cidade)

// Não pode começar o nome de uma constantes com um número (1nome, 20idade)

// Não podem conter espaços ou traços (nome cliente, nome-cliente)

// CERTO seria (nome\_cliente ou nomeCliente)

// Case-sensitive, nome e diferente que Nome

// Não pode modificar um valor de uma constantes

const primeiroNumero = 5

const segundoNumero = 10

const resultado = primeiroNumero + segundoNumero

console.log(resultado);

* Aula 09

// String, number, underfined, null, boolean, symbol

const nome = 'Rodrigo' // String

const nome1 = "Rodrigo" // String

const nome2 = `Rodrigo` // String

const num1 = 10 // Number

const num2 = 10.52 // Number

let nomeAluno // underfined -> não aponta pra local nenhum na memória

const sobreNomeAluno = null // Nulo -> não aponta pra local nenhum na memória

const aprovado = true // Boolean = true ou false (lógico)

* Aula 10

// Aritméticos \* / + - \*\* %

/\*

Ordem de precedência () \*\* \* / % + -

Incremento = ++

decremento = --

NaN - Not a number

parseInt (inteiro)

parseFloat(decimais)

Number(irá converte para número tanto inteiro quanto float)

\*/

 const num1 = 5

 const num2 = parseInt('2')

 const num3 = parseFloat(10.5)

 const num4 = Number(2.78)

 console.log(num1 + num2 + num3 + num4);

* Aula 11

alert('Hello world!!!')

confirm('Deseja realmente cancelar')

prompt('Digite seu nome:')



funções do window, entre outras, window.alert, window.confirm, window.prompt



mostra caixa de texto



mostra caixa de escolha entre ok ou cancelar

pede para digitar algo

* Aula 13

//               01234567

let umaString = "Um texto"

console.log(umaString[4]);

irá mostrar o valor do índice informado, esse caso a string e

console.log(umaString.charAt(6));

outra maneira de recuperar o valor.

console.log(umaString.concat(' outra string'));

para concatenar uma string

console.log(umaString.indexOf('texto'));

para saber em qual índice começa a palavra texto, caso não encontre nada ele retornará o índice -1.

console.log(umaString.indexOf('Um', 3));

aqui ele está procurando a palavra Um a partir do índice 3

console.log(umaString.lastIndexOf('S'));

procura onde o aparece a letra por ultimo

console.log(umaString.replace('Um', 'Outra coisa'));

substituir uma palavra por outra

console.log(umaString.replaceAll("a", "b"));

nesse modelo ele troca todos os a por b na frase

console.log(umaString.length);

ver o tamanho total do texto, o tamanho não o índice

console.log(umaString.slice(3, 7));

ele fatia o texto, começa e ler do 3 índice ao 7, porém não mostrará o que está no índice 7

console.log(umaString.slice(-3));

com valor negativo ele irá ler a partir do final.

console.log(umaString.slice(-3,-1));

começo e parada para leitura de uma string, começando do -3 e parando no -1 de trás pra frente.

console.log(umaString.split(' '));

ira dividir o texto em índice de uma array cada vez que achar o espaço

console.log(umaString.split(' ', 2));

para que esse processo só aconteça duas vezes

console.log(umaString.toUpperCase());

console.log(umaString.toLowerCase());

colocar em maiúsculo e minúsculo

* Aula 14

let num1 = 1500

console.log(num1.toString());

passando o tipo número para uma string

console.log(num1.toString(2));

para transformar em representação binária

console.log(num1.toFixed(3));

ele coloca quantas casas decimais queremos, nesse caso 3, e faz o arredondamento sé necessário

console.log(Number.isInteger(num1));

verifica se o número e inteiro

let temp = num1 \* 'Olá'

console.log(Number.isNaN(temp));

verifica se a conta e invalido, retornará sempre que a conta for invalida

retorna true quando e NaN, 3 + ‘5’ = 35, retornará true.

* Aula 15

let num1 = 9.49

let num2 = 25

console.log(Math.floor(num1));

//arredondando número para baixo

console.log(Math.ceil(num1));

//arredondando número para cima

console.log(Math.round(num1));

//arredonda para o mais próximo

console.log(Math.max(1,2,3,4,5,6-10,50,1500,9,8,7,6));

// pega o maior número da sequência

console.log(Math.min(1,2,3,4,5,6-10,50,1500,9,8,7,6));

// pega o menor número da sequência

const aleatorio = Math.random() \* (10 - 5) + 5

console.log(aleatorio);

// gerar número aleatório

console.log(Math.round(aleatorio));

// arredondando o número aleatório para o mais perto

console.log(num2 \*\* 0.5);

// tirar a raiz quadrada do número

console.log(Math.sqrt(num2));

// outra forma de usar

console.log(100 / 0);

// dado númerico INFINITY, aproxima muito do ZERO, faz a conta mais NÃO devéria

Selecionado elementos

const numero = prompt('Digite um número:')

const numeroTitulo = document.getElementById('numero-titulo')

numeroTitulo.innerHTML = numero

está pegando todo o elemento da pagina HTML que tenha o id ‘numero-titulo’ e jogando em uma variável/constante.

Em seguida está modificando o HTML somente desse e id que vou selecionado.

* Aula 16 Array

let alunos = ['Luiz', 'Maria', 'João', 1, true, null];

tente deixar suas listas coesos com informações de grupos iguais

let alunos = ['Luiz', 'Maria', 'João'];

//              0       1        2

let alunos = ['Luiz', 'Maria', 'João'];

alunos[0] = 'Eduardo'

//alterando

alunos[3] = 'Luiza'

//inserindo

alunos.push('Luiza')

alunos.push('Otávio')

//inserindo pelo push, insere no final da arrey

alunos.unshift('Luiza')

//inserindo no começo da arrey, no indice 0

alunos.pop()

// remove o último item do array

alunos.shift()

// remove do começo da array

delete alunos[1];

// apaga o item do indice, mais não altera o valor dos índices

console.log(alunos.slice(0,3));

//para fatiar os indices como um string, começa no 0 e vai até o 2

console.log(alunos instanceof Array);

// verificando se isso e uma instancia de arrey

* Aula 17 Funções

function saudacao() {



    console.log('Bom dia!');

}

saudacao()



cria uma função



chama a função

function saudacao(nome) {



    console.log(`Bom dia ${nome}!`);



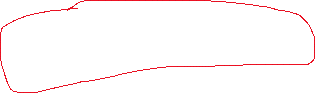
}

saudacao('Rodrigo')

passando um parâmetro para a função

function saudacao(nome) {

    return `Bom dia ${nome}!`



}



const variavel = saudacao('Rodrigo')



console.log(variavel);

sua função agora está retornando algo e enviando para uma variavel, que poderá ser acessada depois, essa função não vai exibir nada, quando chamada irá somente fazer o return para a variavel

function soma(x=2, y=0) {

    const resultado = x + y

    return resultado

}

console.log(soma());

colocando valores como padrão, x=2 e y=0, caso não passe nenhum valor

const raiz = function (n) {

    return n \*\* 0.5

};

console.log(raiz(9));

criando funções anônimas, está função está sendo chamada pela variavel raiz

const raiz1 = (n) => n \*\* 0.5

// arrow function

Arrow function que a mesma coisa que uma função só que resumida.

Como só tem um parâmetro o n poderia ficar sem os parênteses

Const raiz1 = n => n \*\* 0.5

* Aula 18 objetos

const nome01 = 'Luiz'

const sobrenome01 = 'Miranda'

const idade01 = 25

const nome01 = 'Maria'

const sobrenome01 = 'Oliveira'

const idade01 = 55

em objeto ficaria



const pessoa1 = {

    nome: 'Luiz',

    sobrenome: 'Miranda', // argumentos

    idade: 25

}

const pessoa2 = {

    nome: 'Maria',

    sobrenome: 'Oliveira',

    idade: 55

}

Porem caso tivesse que adicionar 1000 pessoas esse processo ficária muito demorado.

parâmetros



function criaPessoa(nome, sobrenome, idade) {

    return {

        nome,       // nome: nome

        sobrenome,  //sobrenome: sobrenome

        idade       // idade: idade

    }

}

const pessoa1 = criaPessoa('Luiz', 'Otavio', 25)

const pessoa2 = criaPessoa('Maria', 'Miranda', 32)



argumentos

const pessoa1 = {

    nome: 'Luiz',

    sobrenome: 'Miranda',

    idade: 25,

    fala() {

      console.log('Olá mundo');

    }

}

pessoa1.fala()

fala() = método

const pessoa1 = {

    nome: 'Luiz',

    sobrenome: 'Miranda',

    idade: 25,

    fala() {

   console.log(`${this.nome} ${this.sobrenome} falando oi`);

    }

}

pessoa1.fala()

this. = neste objeto

const pessoa1 = {

    nome: 'Luiz',

    sobrenome: 'Miranda',

    idade: 25,

    fala() {

      console.log(`A minha idade atual é ${this.idade}`);

    },

    incrementaIdade() {

        this.idade++

    }

}

pessoa1.fala() //25

pessoa1.incrementaIdade() //adiciona mais 1 a idade

pessoa1.fala() //26

* Aula 19 Valores primitivos e valores por referência

/\*

Primitivos (imutáveis)

string, number, boolean, underfined, null (bigint, symbol) =

VALORES COPIADOS

Referência (mutável)

APONTA PARA O ESPAÇO DA MEMÓRIA QUE ELE FAZ REFERÊNCIA

Array, object, function

\*/

let a = [1, 2, 3]

let b = a       //referencia b=a e a=b

let c = [...a]  // para que o valor de a seja puxado para o c,

fez a cópia e não a referência

console.log(a, b);

a.push(4)  // a e b  serão afetados

console.log(a, b);

b.pop()    // a e b serão afetados

a.pop()    // a e b serão afetados

console.log(a, b, c);

const nome1 = {

    nome: 'Luiz',

    sobrenome: 'Otavio'

}

const nome2 = {...nome1} // nome2 recebe os valores de a espalhados, não a referência

nome1.nome = 'Maria' // somente nome1 será alterado

console.log(nome1, nome2);

* Aula 19 exercícios

    const form = document.querySelector('.form')

    const texto = document.querySelector('#texto\_1')

forma de selecionar tanto ID quanto nome de classe

const nome = form.querySelector('.nome').value;

para pegar o valor que está escrito dentro do input,

o caminho será:

.form -> .nome -> oque foi digitado

Exercício completo:

function meuEscopo () {

    const form = document.querySelector('.form');

    const resultado = document.querySelector('.resultado');

    const pessoas = [];

    function recebeEventoForm (evento) {

        evento.preventDefault();  //evento que impede ao clique do botão a pagina atualize

        const nome = form.querySelector('.nome').value;

        const sobrenome = form.querySelector('.sobrenome').value;

        const peso = form.querySelector('.peso').value;

        const altura = form.querySelector('.altura').value;

        const individuo = {

            nome: nome,

            sobrenome: sobrenome,

            peso: peso,

            altura: altura

        }

        pessoas.push(individuo)

        console.log(pessoas);

        resultado.innerHTML += `<p>${individuo.nome} ${individuo.sobrenome} ${individuo.peso} ${individuo.altura}</p>`

    }

    form.addEventListener('submit', recebeEventoForm)

}

meuEscopo()