

JAVASCRIPT

MANIPULANDO OBJETOS COM DOM

ANTES DE INICIARMOS...

- CRIAR UM ARQUIVO INDEX.HTML
- CRIAR UM ARQUIVO SCRIPT.JS
- LINKAR O SCRIPT.JS NO HTML

VARIÁVEIS

- SÃO RESPONSÁVEIS POR GUARDAR DADOS NA MEMÓRIA
- VAR, LET E CONST: QUAIS AS DIFERENÇAS?
- UTILIZEI A VÍRGULA PARA CRIAR MAIS DE UMA VARIÁVEL, SEM REPETIR A PALAVRA CHAVE VAR.

```
var nome = 'SECIT';  
let idade = 17;  
const estaMatriculado = true;
```

```
let nome = 'SECIT',  
    idade = 17,  
    estaMatriculado = true;
```

VARIÁVEIS

- PODE DECLARAR ELA E NÃO ATRIBUIR VALOR INICIALMENTE, RETORNANDO UNDEFINED
- OS NOMES PODEM INICIAR COM \$, _, OU LETRAS.
 - PODEM CONTER NÚMEROS MAS NÃO INICIAR COM ELES
- CASE SENSITIVE (MAIUSCULAS E MINUSCULAS)
- NÃO UTILIZAR PALAVRAS RESERVADAS - [HTTPS://WWW.W3SCHOOLS.COM/JS/JS_RESERVED.ASP](https://www.w3schools.com/js/js_reserved.asp)
- CAMELCASE - PALAVRAS COMEÇAM COM MAIUSCULA MENOS A PRIMEIRA

EXERCÍCIO

- DECLARAR UMA VARIÁVEL COM O SEU NOME
- DECLARAR UMA VARIÁVEL COM A SUA IDADE
- DECLARAR UMA VARIÁVEL COM A SUA COMIDA FAVORITA E NÃO ATRIBUIR VALOR
- ATRIBUIR VALOR A SUA COMIDA FAVORITA
- DECLARAR 5 VARIÁVEIS DIFERENTES SEM VALORES

TIPOS DE DADOS

TODOS SÃO PRIMITIVOS EXCETO OS OBJETOS.

```
var nome = 'SECIT'; // String
var idade = 17; // Number
var estaInscrito = true; // Boolean
var time; // Undefined
var comida = null; // Null
var simbolo = Symbol() // Symbol
var novoObjeto = {} // Object
```

É UM DADO QUE NÃO É REPRESENTADO ATRAVÉS DE UM OBJETO E, POR CONSEQUÊNCIA, NÃO POSSUI MÉTODOS.

VERIFICAR TIPO DE DADO

```
let nome = 'SECIT';  
console.log(typeof nome);  
// retorna string
```

STRING

VOCÊ PODE SOMAR UMA STRING E ASSIM CONCATENAR AS PALAVRAS.

```
var nome = 'SECIT';  
var sobrenome = 'IFZN';  
var nomeCompleto = nome + ' ' + sobrenome;  
// retorna SECIT IFZN
```

VOCÊ PODE SOMAR NÚMEROS COM STRINGS, O RESULTADO FINAL É SEMPRE UMA STRING.

```
var gols = 1000;  
var frase = 'Romário fez ' + gols + ' gols';
```


TEMPLATE STRING

VOCÊ DEVE PASSAR EXPRESSÕES / VARIÁVEIS DENTRO DE \${}

```
let gols = 1000  
let frase1 = 'Romário fez ' + gols + ' gols'  
let frase2 = `Romário fez ${gols} gols` // Utilizando Template String
```

EXERCÍCIO

- *DECLARE UMA VARIÁVEL CONTENDO UMA STRING*
- *DECLARE UMA VARIÁVEL CONTENDO UM NÚMERO DENTRO DE UMA STRING*
- *DECLARE UMA VARIÁVEL COM A SUA IDADE*
- *DECLARE DUAS VARIÁVEIS, UMA COM SEU NOME E OUTRA COM SEU SOBRENOME E SOME AS MESMAS*
- *VERIFIQUE O TIPO DA VARIÁVEL QUE CONTÉM O SEU NOME*

CONDICIONAIS

VERIFICAR SE UMA EXPRESSÃO É VERDADEIRA COM IF, CASO CONTRÁRIO O ELSE SERÁ ATIVADO.

```
let possuiMatricula = true;

if(possuiMatricula) {
  console.log('Possui matrícula');
} else {
  console.log('Não possui matrícula');
}

// retorna Possui Matrícula e não executa o else
```

CONDICIONAIS

ELSE IF

SE O IF NÃO FOR VERDADEIRO, ELE TESTA O ELSE IF

```
let possuiGraduacao = true;
let possuiDoutorado = false;

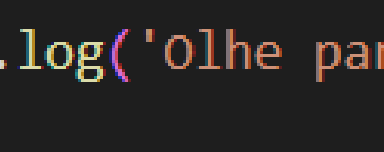
if(possuiDoutorado) {
  console.log('Possui graduação e doutorado');
} else if(possuiGraduacao) {
  console.log('Possui graduação, mas não possui doutorado');
} else {
  console.log('Não possui graduação');
}
// retorna Possui Graduação, mas não possui doutorado
```

SWITCH

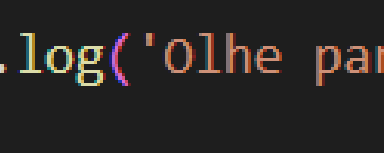
COM O SWITCH VOCÊ PODE VERIFICAR SE UMA VARIÁVEL É IGUAL À DIFERENTES VALORES UTILIZANDO O CASE. CASO ELA SEJA IGUAL, VOCÊ PODE FAZER ALGUMA COISA E UTILIZAR A PALAVRA CHAVE BREAK; PARA CANCELAR A CONTINUAÇÃO. O VALOR DE DEFAULT OCORRERÁ CASO NENHUMA DAS ANTERIORES SEJA VERDADEIRA.

```
let corFavorita = 'Azul';

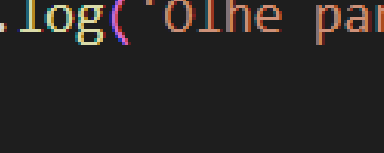
switch (corFavorita) {
  case 'Azul':
    console.log('Olhe para o céu.');
```



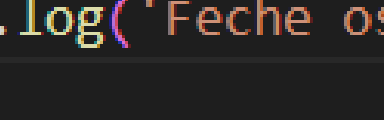
```
    break;
  case 'Vermelho':
    console.log('Olhe para rosas.');
```



```
    break;
  case 'Amarelo':
    console.log('Olhe para o sol.');
```



```
    break;
  default:
    console.log('Feche os olhos');
```



```
}
```

OPERADORES LÓGICOS – NEGAÇÃO

O OPERADOR !, NEGA UMA OPERAÇÃO BOOLEANA. OU SEJA, !TRUE É IGUAL A FALSE
VOCÊ PODE UTILIZAR O !! PARA VERIFICAR SE UMA EXPRESSÃO É TRUTHY OU FALSY

```
if(!true) // false  
if(!1) // false  
if('') // true  
if(!undefined) // true  
if(!!' ') // true  
if(!!'') // false
```

OPERADORES DE COMPARAÇÃO

VÃO SEMPRE RETORNAR UM VALOR BOOLEANO

```
10 > 5; // true
5 > 10; // false
20 < 10; // false
10 <= 10 // true
10 >= 11 // false
```

OPERADORES DE COMPARAÇÃO

O == FAZ UMA COMPARAÇÃO NÃO TÃO ESTRITA E O === FAZ UMA COMPARAÇÃO ESTRITA, OU SEJA, O TIPO DE DADO DEVE SER O MESMO QUANDO USAMOS ===

```
10 == '10'; // true
10 == 10; // true
10 === '10'; // false
10 === 10 // true
10 != 15 // true
10 != '10' // false
10 !== '10' // true
```


OPERADORES LÓGICOS &&

&& COMPARA SE UMA EXPRESSÃO E A OUTRA É VERDADEIRA

```
true && true; // true
true && false; // false
false && true; // false
'Gato' && 'Cão'; // 'Cão'
(5 - 5) && (5 + 5); // 0
'Gato' && false; // false
(5 >= 5) && (3 < 6); // true
```

SE AMBOS OS VALORES FOREM TRUE ELE IRÁ RETORNAR O ÚLTIMO VALOR VERIFICADO SE ALGUM VALOR FOR FALSE ELE IRÁ RETORNAR O MESMO E NÃO IRÁ CONTINUAR A VERIFICAR OS PRÓXIMOS

OPERADORES LÓGICOS ||

|| COMPARA SE UMA EXPRESSÃO **OU** OUTRA É VERDADEIRA

```
true || true; // true
true || false; // true
false || true; // true
'Gato' || 'Cão'; // 'Gato'
(5 - 5) || (5 + 5); // 10
'Gato' || false; // Gato
(5 >= 5) || (3 < 6); // true
```

RETORNA O PRIMEIRO VALOR TRUE QUE ENCONTRAR

EXERCÍCIO

- *VERIFIQUE SE A SUA IDADE É MAIOR DO QUE A DA PESSOA AO SEU LADO, DEPENDENDO DO RESULTADO COLOQUE NO CONSOLE 'É MAIOR', 'É IGUAL' OU 'É MENOR'*

FUNÇÕES

BLOCO DE CÓDIGO QUE PODE SER EXECUTADO E REUTILIZADO. VALORES PODEM SER PASSADOS POR UMA FUNÇÃO E A MESMA RETORNA OUTRO VALOR.

```
function areaQuadrado(lado) {  
    return lado * lado;  
}  
  
areaQuadrado(4) // 16  
areaQuadrado(5) // 25  
areaQuadrado(2) // 4
```

FUNÇÕES

PARÊNTESES () EXECUTAM UMA FUNÇÃO

```
function pi() {  
  return 3.14;  
}
```

```
let total = 5 * pi(); // 15.7
```

PARÂMETROS E ARGUMENTOS

AO CRIAR UMA FUNÇÃO, VOCÊ PODE DEFINIR PARÂMETROS.
AO EXECUTAR UMA FUNÇÃO, VOCÊ PODE PASSAR ARGUMENTOS.

```
// peso e altura são os parâmetros
function imc(peso, altura) {
    const imc = peso / (altura ** 2);
    return imc;
}

imc(80, 1.80) // 80 e 1.80 são os argumentos
imc(60, 1.70) // 60 e 1.70 são os argumentos
```

PARÊNTESES EXECUTA A FUNÇÃO

SE APENAS DEFINIRMOS A FUNÇÃO COM O FUNCTION E NÃO EXECUTARMOS A MESMA, NADA QUE ESTIVER DENTRO DELA IRÁ ACONTECER

```
function corFavorita(cor) {  
  if(cor === 'azul') {  
    return 'Você gosta do céu';  
  } else if(cor === 'verde') {  
    return 'Você gosta de mato';  
  } else {  
    return 'Você não gosta de nada';  
  }  
}  
corFavorita(); // retorna 'Você não gosta de nada'
```

ARGUMENTOS PODEM SER FUNÇÕES

FUNÇÕES ANÔNIMAS SÃO AQUELAS EM QUE O NOME DA FUNÇÃO NÃO É DEFINIDO, ESCRITAS COMO `FUNCTION() {}` OU `() => {}`

```
addEventListener('click', function() {  
    console.log('Clicou');  
});  
// A função possui dois argumentos  
// Primeiro é a string 'click'  
// Segundo é uma função anônima
```


ESCOPO

VARIÁVEIS E FUNÇÕES DEFINIDAS DENTRO DE UM BLOCO {}, NÃO SÃO VISÍVEIS FORA DELE.

```
function precisoVisitar(paisesVisitados) {  
  var totalPaises = 193;  
  return `Ainda faltam ${totalPaises - paisesVisitados} paises para visitar`  
}  
console.log(totalPaises); // erro, totalPaises não definido
```

ESCOPO LÉXICO

FUNÇÕES CONSEGUEM ACESSAR VARIÁVEIS QUE FORAM CRIADAS NO CONTEXTO PAI

```
let profissao = 'Programador';

function dados() {
  var nome = 'Maria';
  var idade = 17;
  function outrosDados() {
    var endereco = 'Natal';
    var idade = 16;
    return `${nome}, ${idade}, ${endereco}, ${profissao}`;
  }
  return outrosDados();
}

dados(); // Retorna 'Maria, 16, Natal, Programador'
outrosDados(); // retorna um erro
```

EXERCÍCIO

- CRIE UMA FUNÇÃO MATEMÁTICA QUE RETORNE O PERÍMETRO DE UM QUADRADO, LEMBRANDO: PERÍMETRO É A SOMA DOS QUATRO LADOS DO QUADRADO
- CRIE UMA FUNÇÃO QUE RETORNE O SEU NOME COMPLETO, ELÁ DEVE POSSUIR OS PARÂMETROS: NOME E SOBRENOME

OBJETOS

CONJUNTO DE VARIÁVEIS E FUNÇÕES, QUE SÃO CHAMADAS DE PROPRIEDADES E MÉTODOS.

```
let pessoa = {  
  nome: 'Joaozinho',  
  idade: 29,  
  profissao: 'Engenheiro',  
  possuiFaculdade: true,  
}  
  
pessoa.nome; // 'Joaozinho'  
pessoa.possuiFaculdade; // true
```

MÉTODOS

É UMA PROPRIEDADE QUE POSSUI UMA FUNÇÃO NO LOCAL DO SEU VALOR.

```
let quadrado = {  
  lados: 4,  
  area(lado) {  
    return lado * lado;  
  },  
  perimetro(lado) {  
    return this.lados * lado;  
  },  
}
```

MÉTODOS

DADOS.NOMECompleto (NO CONSOLE)

```
let dados = {  
  nome: 'Letícia',  
  sobrenome: 'Ribeiro',  
  idade: 17,  
  cidade: 'Natal',  
}  
dados.nomeCompleto = function() {  
  return `${this.nome} ${this.sobrenome}`;  
}
```

OBJETOS SERVEM PARA ORGANIZAR O CÓDIGO EM PEQUENAS PARTES REUTILIZÁVEIS.

```
Math.PI; // 3.14  
Math.random(); // número aleatório  
  
let pi = Math.PI;  
console.log(pi); // 3.14
```

DOT NOTATION GET

ACESSE PROPRIEDADES DE UM OBJETO UTILIZANDO O PONTO .

```
var menu = {  
    width: 800,  
    height: 50,  
    backgroundColor: '#84E',  
}  
  
var bg = menu.backgroundColor; // '#84E'
```


DOT NOTATION SET

SUBSTITUA O VALOR DE UMA PROPRIEDADE UTILIZANDO . E O = APÓS O NOME DA MESMA.

```
var menu = {  
  width: 800,  
  height: 50,  
  backgroundColor: '#84E',  
}  
  
menu.backgroundColor = '#000';  
console.log(menu.backgroundColor); // '#000'
```

ADICIONAR PROPRIEDADES E MÉTODOS

BASTA ADICIONAR UM NOVO NOME E DEFINIR O VALOR.

```
var menu = {  
  width: 800,  
}  
  
menu.height = 50;  
menu.position = 'fixed';
```

PALAVRA-CHAVE THIS

THIS IRÁ FAZER UMA REFERÊNCIA AO PRÓPRIO OBJETO.

```
var height = 120;
var menu = {
  width: 800,
  height: 50,
  metadeHeight() {
    return this.height / 2;
  }
}

menu.metadeHeight(); // 25
// sem o this, seria 60
```

THIS IRÁ RETORNAR O PRÓPRIO OBJETO

PROTÓTIPO E HERANÇA

O OBJETO HERDA PROPRIEDADES E MÉTODOS DO OBJETO QUE FOI UTILIZADO PARA CRIAR O MESMO.

```
var menu = {  
  width: 800,  
}  
  
menu.hasOwnProperty('width') // true  
menu.hasOwnProperty('height') // false
```

HASOWNPROPERTY É UM MÉTODO DE OBJECT

EXERCÍCIO

- CRIE UM OBJETO COM OS SEUS DADOS PESSOAIS, DEVE POSSUI PELO MENOS DUAS PROPRIEDADES NOME E SOBRENOME
- CRIE UM MÉTODO NO OBJETO ANTERIOR, QUE MOSTRE O SEU NOME COMPLETO

ARRAY

É UM GRUPO DE VALORES GERALMENTE RELACIONADOS. SERVEM PARA GUARDARMOS DIFERENTES VALORES EM UMA ÚNICA VARIÁVEL.

```
let videoGames = ['Switch', 'PS4', 'XBox'];  
  
videoGames[0] // Switch  
videoGames[2] // Xbox
```

ACESSE UM ELEMENTO DA ARRAY UTILIZANDO [N]

MÉTODOS E PROPRIEDADES DE UMA ARRAY

```
let videoGames = ['Switch', 'PS4', 'XBox'];  
  
videoGames.pop(); // Remove o último item e retorna ele  
videoGames.push('3DS'); // Adiciona ao final da array  
videoGames.length; // 3
```

FOR LOOP

FAZEM ALGO REPETIDAMENTE ATÉ QUE UMA CONDIÇÃO SEJA ATINGIDA.

```
for (let numero = 0; numero < 10; numero++) {  
  console.log(numero);  
}  
// Retorna de 0 a 9 no console
```

O FOR LOOP POSSUI 3 PARTES, INÍCIO, CONDIÇÃO E INCREMENTO

WHILE LOOP

```
let i = 0;
while (i < 10) {
  console.log(i);
  i++;
}
// Retorna de 0 a 9 no console
```

BREAK

O LOOP IRÁ PARAR CASO ENCONTRO E PALAVRA BREAK

```
let videoGames = ['Switch', 'PS4', 'XBox', '3DS'];  
for (let i = 0; i < videoGames.length; i++) {  
  console.log(videoGames[i]);  
  if(videoGames[i] === 'PS4') {  
    break;  
  }  
}
```

ELE IRÁ PARAR QUANDO ACHAR PS4 NO ARRAY

FOREACH

FOREACH É UM MÉTODO QUE EXECUTA UMA FUNÇÃO PARA CADA ITEM DA ARRAY. É UMA FORMA MAIS SIMPLES DE UTILIZARMOS UM LOOP COM ARRAYS (OU ARRAY-LIKE)

```
let videoGames = ['Switch', 'PS4', 'XBox', '3DS'];
videoGames.forEach(function(item) {
|   console.log(item);
});
// O argumento item será atribuído dinamicamente
```

PODEMOS PASSAR OS SEGUINTE PARÂMETROS ITEM, INDEX E ARRAY

FOREACH

```
let numeros = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
```

```
function tabuadaDe2(item) {  
  console.log(item*2);  
}
```

```
numeros.forEach(tabuadaDe2)
```

```
//O resultado impresso no console será o resultado da tabuada de 2: 0 2 4 6 8 10 12 14 16 18 20
```

EXERCÍCIO

- CRIE UMA ARRAY COM OS ANOS QUE O BRASIL GANHOU A COPA// 1959, 1962, 1970, 1994, 2002
- INTERAJA COM UM LOOP NAS FRUTAS ABAIXO E PARE AO CHEGAR EM PERA

```
LET FRUTAS = ['BANANA', 'MAÇÃ', 'PERA', 'UVA',  
'MELÂNCIA']
```