

Resumo Java Script

Catiusci P. C. Scheffer

Resumo JavaScript

JavaScript Developer - DIO ># »##### **VARIÁVEIS E OPERADORES - TIPOS DE VARIÁVEIS** - 'let' utilizado quando os valores podem variar; - 'const' utilizado quando os valores não mudam; - **OPERADORES DE COMPARAÇÃO** (return false ou true)

Operador	Ação
=	atribuir um valor à variável
==	verifica se os valores NÃO tipo
===	verifica se são iguais em TIPO e VALOR
!	operador de NEGAÇÃO

- OPERADORES LÓGICOS

Operador	Ação
&& OR	Seria a junção entre "isso + aquilo" Isso

ESTRUTURA CONDICIONAL

A estrutura condicional pode ser constituída de vários blocos condicionais, o primeiro a ser 'TRUE' é imediatamente executado.

Exemplo: ~~~javascript if('CONDIÇÃO A SER VERIFICADA'){ 'SE A CONDIÇÃO ACIMA FOR 'VERDADEIRA' EXECUTE O QUE ESTIVER NESTE BLOCO!' }else{ 'SE O BLOCO ACIMA NÃO FOR "VERDADEIRO" EXECUTE ESTE!' }~~ ># »##### **FUNÇÕES** As funções são objetos no Java script que podem ser invocadas no código infinitas vezes, sem necessidade de repetição de código, elas podem ou não ter parâmetros e retornos. As funções servem para organizar e deixar o código mais legível.

Exemplo: ~~~javascript function nome(parametro1, parametro2) { return parametro1 * parametro2; }~~~ Para invocar uma função basta colocar o nome dela e os parênteses com ou sem parâmetros, depende de cada função, desta maneira: **nome_função()**;

Existem também as funções anônimas e autoinvocáveis, ela são muito utilizadas quando devem ser executadas independente de serem invocadas.

Exemplo: ~~~javascript (function (parametro1, parametro2) { return parametro1 * parametro2; })()~~~ ># »##### **OBJETOS** É uma coleção de valores que tem chave e valor, o mesmo deve estar entre {}; O objeto pode ser a instância de uma classe. * **Para acessar os valores:** ~~~javascript console.log(pessoa.nome);~~
~~OU console.log(pessoa['nome']); //Maria~~~

- Criando um objeto:

```
const pessoa = {
  nome: 'Catiusci',
  idade: 25,
  altura: 1.56,
};
console.log(pessoa);
//{nome: 'Catiusci', idade: 25, altura: 1.56}
```

- Alterando valor da chave:

```
pessoa.nome = 'July';
OU
pessoa['nome'] = 'Jyly';
console.log(pessoa);
//{nome: 'July', idade: 25, altura: 1.56}
```

- Deletando uma chave e por consequência o seu valor:

```
delete pessoa.altura;
console.log(pessoa);
//{nome: 'July', idade: 25}
```

- Criando um Objeto com Método *Uma função dentro de um objeto é chamado de método.*

```
const pessoa = {
  nome: 'Maria',
  idade: 25,
  altura: 1.56,

  descrever: function(){
    console.log(`Meu nome é ${this.nome} e minha idade é ${this.idade} anos.`);
  }
};
// o "this" assume todos os valores do objeto ao qual ele pertence;
```

```
    pessoa.descrever();  
    //Meu nome é Maria e minha idade é 25 anos.
```

CLASSES

A classe é o contrato do objeto, ela define o mínimo de característica que o objeto que for instanciado a partir dela terá

Exemplo:

- **Classe sem construtor:** Uma classe criada sem construtor, eu consigo instanciá-la sem atribuir nenhum valor;

```
class Pessoa{  
    nome;  
    idade;  
    altura;  
  
    descrever(){  
        console.log(`Meu nome é ${this.nome} e minha idade é ${this.idade} anos.`);  
    }  
}  
  
//instanciando a classe, criando objeto  
const maria = new Pessoa();  
console.log(maria)  
//Pessoa {nome: undefined, idade: undefined, altura: undefined}, ficou assim pois falta def  
  
//atribuindo valores ao objeto maria  
maria.nome = 'Maria de Jesus';  
maria.idade = 26;  
maria.altura = 1.85;  
console.log(maria)  
//Pessoa {nome: 'Maria de Jesus', idade: 26, altura: 1.85}  
  
maria.descrever();  
//Meu nome é Maria de Jesus e minha idade é 26 anos.
```

- **Classe com construtor:** Esta classe já me obriga a definir valores no momento da instância, devido aos parâmetros do construtor;

```
class Pessoa{  
    nome;  
    idade;
```

```

    constructor(nome, idade){
        this.nome = nome;
        this.idade = idade;
        this.anoDeNascimento = 2022 - idade;
    }

    descrever(){
        console.log(`Meu nome é ${this.nome} e minha idade é ${this.idade} anos.`);
    }
}

//instanciando a classe, criando objeto
const joao = new Pessoa('João Paulo', 36);
console.log(joao)
//Pessoa {nome: 'João Paulo', idade: 36, anoDeNascimento: 1986}
joao.descrever();
//Meu nome é João Paulo e minha idade é 36 anos.

```

Listas - ARRAYS

Array assim como objeto é um conjunto de dados, mas o array tem apenas o valor sem chave como o objeto. Dentro da lista os valores podem ser tanto numérico quanto texto na mesma lista. A estrutura do array é caracterizada por ficar dentro de [], a mesma pode ser criada vazia. * **Inserir valores no Array PUSH:** O push insere novo item da lista na última posição.

```
listaCompras.push('Maria')
```

- **Removendo valores do Array, com POP ou SHIFT:** O POP remove o último item da lista enquanto o SHIFT remove o primeiro item.

```
listaCompras.pop()
```

OU

```
listaCompras.shift()
```

- **Contar quantos itens compõem a lista:**

```
listaCompras.length()
```

- **Encontrar o MAIOR valor dentro de um array:**

```

const listaMedias1 = [5, 4.5, 8, 6, 7, 7.49, 9, 9.9, 2];
var max = Math.max(...listaMedias1);
console.log(max)
//9.9

```

ESTRUTURA DE REPETIÇÃO

FOR

Utilizado por exemplo quando precisamos verificar todos os itens dentro de uma lista item a item, e a partir disso podemos ter várias operações com cada item ou com a lista inteira.

IMPORTANDO E EXPORTANDO

- Importação:
- Utilizando o **require**:

```
const funcoes = require('./funcoes-auxiliares');
```

- Exportação:
 - Utilizando **module.exports**: “javascript function gets() { return 10; }”

```
function print(texto) { console.log(texto); }
```

```
module.exports = { gets, print }; “
```

MÉTODOS/FUNÇÕES PRÉ-DEFINIDOS

//Exponenciação:

```
Math.pow(base, expoente);
```

//Arredondamento após a vírgula

```
variavel.toFixed('nº casas após a vírgula')
```

GERAL

Configuração CSS > NORMALIZER >

Para que o CSS se comporte da mesma maneira em qualquer navegador, mantenha margens e demais formatações é necessário utilizar o normalize, através deste site: “<https://cdnjs.com/libraries/normalize>”, copiando o código e colando no arquivo html.

`https://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.1/normalize.min.css`

PROTOCOLO HTTP {#protocolo-http} — ##### **API** {#api}
} Pode ser descrito como um servidor de dados. * **URL** A composição da URL é a seguinte: URL: `https://pokeapi.co/api/v2/pokemon/1?type=grass` DOMÍNIO/IP: `https://pokeapi.co` PATH: `/api/v2/pokemon/1` CARRY STRING: `?type=grass` Após o ponto de ? vem as condições, filtros de busca dentro da API, essas condições são compostas por chave e valor e podem ser várias concatenadas com &. * **REQUISIÇÃO** * **MÉTODOS**: * **GET**: para obter dados da API; * **POST**: para enviar dados à API; * **PUT**: para atualizar dados da API; * **DELETE**: para excluir dados da API; ... muitos outros * **REQUEST HEADERS** Configuração da solicitação da API. * **RESPONSE HEADERS** Configuração da resposta da API. * **REQUEST BODY** Dado que quero trafegar na requisição. O GET não tem BODY. * **RESPONSE** * **STATUS CODE** * 200 a 299 é família sucesso; * 300 a 399 é a família redirect; * 400 a 499 é a família erro por parte do cliente; * 500 a 599 é a família erro interno no servidor; * **RESPONSE HEADERS** * **RESPONSE BODY**

Exercícios da Aula:

```
/*  
Faça um programa para calcular o vaLor de uma viagem.  
Você terá 3 variáveis:  
1) Preço do combustível/lt;  
2) Rendimento do carro km/litro;  
3) Distância a ser percorrida em km;  
*/  
  
const kmLt = 13;  
const distanciaKm = 1420;  
const precoLt = 7.50;  
  
const custoViagem = (distanciaKm / kmLt) * precoLt;  
  
console.log('O custo da viagem será: R$', custoViagem.toFixed(2));  
  
/*  
Faça um programa para calcular o vaLor de uma viagem.  
Você terá 5 variáveis:  
1) Preço do ALCOOL/lt;
```

```

2) Preço da GASOLINA/lt;
3) Rendimento do carro km/litro;
4) Distância a ser percorrida em km;
5) Tipo de combustível, ALCOOL ou GASOLINA;
*/

const kmLt = 13;
const distanciaKm = 1420;
const precoLtGasolina = 7.50;
const precoLtAlcool = 4.50;
let tipoCombustivel = 'gasolina'

if (tipoCombustivel === 'alcool'){
    const custoViagem = (distanciaKm / kmLt) * precoLtAlcool;
    console.log('O custo da viagem será: R$', custoViagem.toFixed(2));
}else {
    const custoViagem = (distanciaKm / kmLt) * precoLtGasolina;
    console.log('O custo da viagem será: R$', custoViagem.toFixed(2));
}

/*Calcular a média usando array e for*/
const notas = [];

notas.push(8);
notas.push(6);
notas.push(7.4);
notas.push(4);

let soma = 0

for (let i = 0; i < notas.length; i++) {
    const nota = notas[i];
    soma += nota;
}

const media = soma/notas.length;
console.log(media);

```

Exercícios Propostos: ~~~javascript /*Faça um algoritmo que dado as 3 notas tiradas por um aluno em um semestre da faculdade calcule e

imprima a sua média e a sua classificação conforme a tabela abaixo.

Média = (nota 1 + nota 2 + nota 3) / 3;

Classificação: - Média menor que 5, reprovação; - Média entre 5 e 7, recuperação;
- Média acima de 7, passou de semestre; */

let nota1 = 7; let nota2 = 7; let nota3 = 6.9;

let mediaAluno = (nota1 + nota2 + nota3) / 3;

```
if(mediaAluno < 5){ console.log('Classificação do Aluno: REPROVADO')}else if  
(mediaAluno < 7){ console.log('Classificação do Aluno: EM RECUPERAÇÃO')}  
else{ console.log('Classificação do Aluno: APROVADO')} ~~~
```

*/*O IMC - Índice de Massa Corporal é um critério da Organização Mundial de Saúde para dar um*

Formula do IMC:

*IMC = peso / (altura * altura)*

Elabore um algoritmo que dado o peso e a altura de um adulto mostre sua condição de acordo com

IMC em adultos Condição:

*- Abaixo de 18.5 Abaixo do peso;
- Entre 18.5 e 25 Peso normal;
- Entre 25 e 30 Acima do peso;
- Entre 30 e 40 Obeso;
- Acima de 40 Obesidade Grave;
/

```
let pesoKg = 100;
```

```
let alturaMt = 1.57;
```

```
//let calculoIMC = pesoKg / (alturaMt * alturaMt);
```

```
let calculoIMC = pesoKg / Math.pow(alturaMt, 2);
```

```
if (calculoIMC < 18.5){  
  console.log('IMC: Abaixo do Peso')  
}else if (calculoIMC < 25){  
  console.log('IMC: Peso Normal')  
}else if(calculoIMC < 30){  
  console.log('IMC: Acima do Peso')  
}else if(calculoIMC < 40){  
  console.log('IMC: Obeso')  
}else{  
  console.log('IMC: Obesidade Grave')  
};
```


*/*Elabore um algoritmo que calcule o que deve ser pago por um produto, considerando o preço. Utilize os códigos da tabela a seguir para ler qual a condição de pagamento escolhida e efetue o pagamento adequado.*/*

Código Condição de pagamento:

- À vista Débito, recebe 10% de desconto;
 - À vista no Dinheiro ou PIX, recebe 15% de desconto;
 - Em duas vezes, preço normal de etiqueta sem juros;
 - Acima de duas vezes, preço normal de etiqueta mais juros de 10%;
- */*

```
let precoEtiqueta = 59.60;
let desconto10 = (precoEtiqueta * 0.90);
let desconto15 = (precoEtiqueta * 0.85);
let parcelamentoMais2x = (precoEtiqueta * 1.10);

const formasPagto = ['Débito', 'Dinheiro/Pix', 'Parcelado em até 2X', 'Parcelamento 3X ou +'];
let pagamento = 'Parcelamento 3X ou +';
let totalPagar;

if(pagamento === formasPagto[0]){
    totalPagar = desconto10;
    console.log(`Forma de pagamento escolhida foi:`, pagamento, '\nTotal a Pagar:', totalPagar);
}else if(pagamento === formasPagto[1]){
    totalPagar = desconto15;
    console.log(`Forma de pagamento escolhida foi:`, pagamento, '\nTotal a Pagar:', totalPagar);
}else if(pagamento === formasPagto[2]){
    totalPagar = precoEtiqueta;
    console.log(`Forma de pagamento escolhida foi:`, pagamento, '\nTotal a Pagar:', totalPagar);
}else{
    totalPagar = parcelamentoMais2x;
    console.log(`Forma de pagamento escolhida foi:`, pagamento, '\nTotal a Pagar:', totalPagar);
};
```

/ Crie uma classe para representar carros.*

Os carros possuem uma marca, uma cor e um gasto médio de combustível por Kilômetro rodado. Crie um método que dado a quantidade de quilômetros e o preço do combustível nos dê o valor gasto em reais para realizar este percurso./*

```
class Carro{
    marca;
    cor;
    rendPorKM;

    constructor(marca, cor, rendPorKM){
        this.cor = cor;
        this.marca = marca;
    }
}
```

```

        this.rendPorKM = rendPorKM;
    }

    metodo(distanciaKM, precoCombustivel){
        const custoTotal = (distanciaKM / this.rendPorKM * precoCombustivel);
        console.log(`O custo da viagem será de R$ ${custoTotal.toFixed(2)}`);
    }
}

const peugeot208 = new Carro('peugeot', 'vermelho', 13);
peugeot208.metodo(1420, 7.50);

/*Crie uma classe para representar pessoas.
Para cada pessoa teremos os atributos nome, peso e altura.
As pessoas devem ter a capacidade de dizer o valor do seu IMC (IMC = peso / (altura * altura)
Instancie uma pessoa chamada José que tenha 70kg de peso e 1,75 de altura e peça ao José para
dizer o seu IMC;*/
class Pessoa{
    nome;
    peso;
    altura;

    constructor(nome, peso, altura){
        this.nome = nome;
        this.peso = peso;
        this.altura = altura;
    }

    dizer(){
        return (this.peso / Math.pow(this.altura, this.altura)).toFixed(2);
    }
}

const jose = new Pessoa('José', 70, 1.75);
console.log(jose.dizer());

//Crie um programa que dado um número imprima a sua tabuada.

const numero = 7;

for (let i = 1; i <= 10; i++) {
    console.log(`${numero} X ${i} = `, i * numero);
}

//Crie um programa que seja capaz de percorrer uma lista de números e imprima cada número P

const listaNumeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17];

```

```

for (let i = 0; i < listaNumeros.length; i++) {

    if (listaNumeros[i] % 2 === 0){
        console.log(`Número par ${listaNumeros[i]}`);
    }
}

//Faça um programa que dado uma lista de nomes imprima os nomes que comecem com a letra "V"

const listaNomes = ['ANA', 'VALERIA', 'JOÃO', 'VITOR', 'LENIRA', 'VANDA'];

for (let i = 0; i < listaNomes.length; i++) {
    const element = listaNomes[i];
    if(listaNomes[i][0] === 'V'){
        console.log(element);
    }
}

//4) Faça um programa que preencha uma lista com todos os números pares contidos no intervalo de 10 a 50

let numerosPares10_50 = [];

for (let i = 10; i <= 50; i++) {
    if(i % 2 === 0){
        numerosPares10_50.push(i);
    }
}
console.log(numerosPares10_50);

/*Dado uma lista com as médias tiradas pelos alunos. Imprima todas as médias que ficaram de 5 em 5
Ex de lista = [2, 7, 3, 8, 10, 4]*/

const listaMedias = [2, 7, 3, 8, 10, 4]

for (let i = 0; i < listaMedias.length; i++) {
    const medialist = listaMedias[i];
    if(medialist < 5){
        console.log(medialist);
    }
}

/*6) Dado uma lista com as notas tiradas pelos alunos. imprima a maior nota.
Ex de lista de notas = [2, 7, 3, 8, 10, 4]*/

const listaMedias1 = [5, 4.5, 8, 6, 7, 7.49, 9, 9.9, 2];
var max = Math.max(...listaMedias1);
console.log(max)

```

