

```
console.log('Alexandre Paes')

let idade = 5
console.log('idade: ', idade)

let enderecoAluno = 'Avenida Principal'
let idadeAluno = 15
let nomeDaMae = 'Mae do Aluno'

dados = [
  enderecoAluno,
  idadeAluno,
  nomeDaMae,
  outros = {professora: 'Maria'},
  professor = 'Carlos'
]

console.log(dados)

let diretora = {
  nome: 'Regina',
  escola: 'Dolor'
}
console.log(diretora)

// %c Applies CSS style rules to the output string as specified by the second
parameter

console.log('%cHello world!', 'color: blue; font-size: xx-large')

const valorIngressoAdulto = 20
console.log(valorIngressoAdulto)

const fruits = ['apple', 'banana', 'orange']
for (let i = 0; i < fruits.length; i++) {
  console.log(fruits[i])
}

let idadePessoa = 1
if (idadePessoa <= 0){
  idadePessoa += 2
  console.log(idadePessoa)
}
else
  console.log('Could not identify')

let i = 0
if (i < 1) {
  console.log("i is smaller than 1");
} else {
  console.log("i was not smaller than 1");
}
```

```
let idadeBebe = 10
while (idadeBebe >= 5){
  console.log(idadeBebe)
  idadeBebe--
}
```

```
// function names === Verbo + Substantivo
let corSite = 'azul'
function resCor(){
  corSite = ''
}
console.log(corSite)
resCor()
console.log(corSite)
```

```
let barkDogColor = 'white'
function resetColor(color, intensity){
  barkDogColor = color + ' ' + intensity
}
console.log(barkDogColor)
resetColor('blue', 'and green')
console.log(barkDogColor)
```

```
function sayMyName(){
  console.log('Alexandre')
}
sayMyName()
```

```
function multiplyByTwo(num){
  return num * 2 + ' contos'
}
let result = multiplyByTwo(10)
console.log(result)
```

```
let pontos = 200
let tipo = pontos > 100 ? 'Premium' : 'Basic'
console.log(tipo)
```

```
console.log(true && true)
```

```
let maiorDeIdade = true
let possuiCarteiraDeTrabalho = false
let elegivel = maiorDeIdade && possuiCarteiraDeTrabalho
let podeAplicar = maiorDeIdade || possuiCarteiraDeTrabalho
candidatoRecusado = !podeAplicar
```

```
console.log(elegível)
console.log(podeAplicar)
console.log(candidatoRecusado)

//personalizedColor = 'green'
personalizedColor = ''
defaultColor = 'yellow'
primeColor = personalizedColor || defaultColor

console.log(primeColor)

let a = 'A1'
let b = 'B2'

console.log(a)
console.log(b)

// Minha tentativa baseado em Python
// a, b = 'B2', 'A2'
// a2 = a
// b2 = b

// console.log(a2)
// console.log(b2)

// Solução p trocar valores entre variáveis

let c = a
a = b
b = c

console.log(a)
console.log(b)

let hora = 22

if (hora > 5 && hora < 12) {
  console.log('Bom dia!')
}
else if (hora > 12 && hora < 18) {
  console.log('Boa tarde!')
}
else {
  console.log('Boa noite!')
}

let permissao = 'diretor'

switch (permissao) {
  case 'comum':
    console.log('usuário comum')
    break
```

```

    case 'gerente':
        console.log('usuário gerente')
        break

    case 'diretor':
        console.log('usuário diretor')
        break

    default:
        console.log('usuário não cadastrado.')
}

for (let i = 5; i >= 1; i--) {
    if (i % 2 !== 0) {
        console.log(i)
    }
}

let indice = 5
while (indice >= 1) {
    if (indice % 2 !== 0) {
        console.log(indice)
    }
    indice--
}

let n = 0

do {
    console.log('digitando...')
    //console.log('digitando...', n)
    n++
} while (n < 10)

const pessoa = {
    nome: 'Alex',
    idade: 38
}

for(let chave in pessoa) {
    //console.log(pessoa.nome)
    //console.log(pessoa.idade)
    //console.log(pessoa)
    console.log(chave, pessoa[chave])
}

const colors = ['blue', 'yellow', 'green']

for (let c in colors) {
    console.log(c, colors[c])
}

```

// Exercício: Escreva uma função que contenha 2 números e retorna o maior entre

eles!

```
function minMaxNum(){
  let num1 = 1
  let num2 = 2
  if (num1 > num2) {
    console.log('num1 is the max num.')
  }
  else {
    console.log('num2 is the max num.')
  }
}
minMaxNum()
```

// A mesma função da de cima mas em arrow function

```
let minMaxNumVar = () => {
  let num1 = 4
  let num2 = 2
  if (num1 > num2) {
    console.log('num1 is the max num.')
  }
  else {
    console.log('num2 is the max num.')
  }
}
console.log(minMaxNumVar())
```

// Solução do professor para o exercício imediato acima

```
let valorMaior = max(10, 5)
console.log(valorMaior)
```

```
function max(numero1, numero2){
  if (numero1 > numero2)
    return numero1
  else return numero2
}
```

// Exemplo função 2

```
function max(numero1, numero2){
  if (numero1 > numero2)
    return numero1
  return numero2
}
```

// Exemplo função 3

```
function max(numero1, numero2){
  return numero1 > numero2 ? numero1 : numero2
}
```

// Exercício fizzbuzz:

// Divisível por 3 === fizz

```

// Divisível por 5 === buzz
// Divisível por 3 e 5 === fizzbuzz
// Divisível por 3 ou 5 === num
// Divisível por nenhum = 'Não Divisível'

let num = 35 //prompt('Digite a number here: ')

// FizzBuzz com switch não funciona/funcionou

// switch (num) {
//   case num % 3 === 0 && num % 5 === 0:
//     console.log('fizzbuzz')
//     break

//   case num % 5 === 0:
//     console.log('buzz')
//     break

//   case num % 3 === 0:
//     console.log('fizz')
//     break

//   case num % 3 === 0 || num % 5 === 0:
//     console.log(num)
//     break

//   default: console.log('Não divisível.')
// }

// FizzBuzz com if else

if (num % 3 === 0 && num % 5 === 0){
  console.log('fizzbuzz')
}
// else if (num % 3 === 0 || num % 5 === 0) {
//   console.log(num)
// }
else if (num % 3 === 0 ) {
  console.log('fizz')
}
else if (num % 5 === 0) {
  console.log('buzz')
}
else {
  console.log('Non divisible.')
}

// FizzBuzz Solução do Professor:

const resultado = fizzBuzz(3)
console.log(resultado)

function fizzBuzz(entrada){
  if (typeof entrada !== 'number')

```

```

    return 'Não é número.'
  if((entrada % 3 === 0) && (entrada % 5 === 0))
    return 'FizzBuzz'
  if (entrada % 3 === 0)
    return "Fizz"
  if (entrada % 5 === 0)
    return 'Buzz'
  return entrada
}

```

// Exercício verificar velocidade

```

// Velocidade max de 70
// Cada 5km acima do limite ganha 1 ponto
// Utilizar Math.Floor() para arredondar números
// Caso pontos maior que 12 = 'Carteira Suspensa'

```

verificarVelocidade(120)

```

function verificarVelocidade(velocidade) {
  const velocidadeMaxima = 70
  const kmPorPonto = 5
  if (velocidade <= velocidadeMaxima)
    console.log('Ok!')
  else {
    const pontos = Math.floor(((velocidade - velocidadeMaxima) / kmPorPonto))
    if(pontos >= 12)
      console.log('Carteira Suspensa.')
    else
      console.log('Pontos: ', pontos)
  }
}

```

// Exercício Par ou Ímpar

```

// Receber uma quantidade de valores para avaliar
// função exibe se cada valor é par ou ímpar

```

exibirTipo(5)

```

function exibirTipo(limite) {
  for (let i = 0; i < limite; i++) {
    if (i % 2 === 0)
      console.log(i, 'Par')
    else
      console.log(i, 'Ímpar')
  }
}

```

```

// Criar um método para ler propriedades de um objeto e exibir
// somente as propriedades que são do tipo string nesse objeto

```

```

const filme = {
  título: 'Vingadores',
  ano: 2018,
  diretor: 'Robin Fake',
  personagem: 'Thor'
}
exibirPropriedades(filme)
function exibirPropriedades(obj) {
  for (prop in obj)
    if (typeof obj[prop] === 'string')
      console.log(prop, obj[prop])
}

// Exercício Soma Múltiplos de 3 e 5

// Criar a função somar que retorna a soma de todos os
// múltiplos de 3 e 5

// Multiplos de 3: 3, 6, 9...
// Multiplos de 5: 5, 10, 15...

// Armazenar os multiplos de 3
// Armazenar os multiplos de 5
// somar os dois

somar(10)
function somar(limite) {
  let multiplosDe3 = 0
  let multiplosDe5 = 0
  for (i = 0; i <= limite; i++) {
    if (i % 3 === 0)
      multiplosDe3 += i
    if (i % 5 === 0)
      multiplosDe5 += i
  }
  console.log(multiplosDe3 + multiplosDe5)
}

// Exercício Nota Escolar

// Obter a média a partir de um array
// 0-59: E
// 60-69: D
// 70-79: C
// 80-89: B
// 90-100: A

const array = [70, 70, 80]
// 75

function mediaDoAluno(notas) {
  const media = calcularMedia(notas)

```



```
    if (media < 59) return 'E'
    if (media < 69) return 'D'
    if (media < 79) return 'C'
    if (media < 89) return 'B'
    return 'A'
}
```

```
function calcularMedia(array) {
    let soma = 0
    for (let valor of array) {
        soma += valor
    }
    return soma / (array.length)
}
```

```
console.log(mediaDoAluno(array))
```

```
// Criar uma função que exibe a quantidade de Asteriscos (*)
// que cada linha possui
```

```
function exibirAsteriscos(linhas) {
    // let padrao = ''
    // for (let linha = 1; linha <= linhas; linha++){
    //     padrao += '*'
    //     console.log(padrao)
    // }
```

```
    // Exemplo 2
    for(let linha = 1; linha <= linhas; linha++){
        let padrao = ''
        for(let i = 0; i < linha; i++){
            padrao += '*'
        }
        console.log(padrao)
    }
}
exibirAsteriscos(5)
```

```
// Exercício: Criar uma função que exiba números primos (divisível por 1 ou por
// ele mesmo apenas)
```

```
// 11 = número primo (divisível por 1 e 11 apenas)
// 10 = número composto (divisível por 1, 2, 5, 10)
```

```
exibirNumerosPrimos(10);
```

```
// function exibirNumerosPrimos(limite){
//     for(let num = 2; num <= limite; num++){
//         let ehPrimo = true;
//         for(let divisor = 2; divisor < num; num++){
//             if(num % divisor === 0){
```

```

//      ehPrimo = false;
//      break;
//    }
//  }
//  if (ehPrimo) console.log(num);
// }
// }

// Exemplo 2 exibirNumerosPrimos

function exibirNumerosPrimos(limite){
  for(let num = 2; num <= limite; num++){
    if(NúmeroPrimo(num)) console.log(num)
  }
}

function NúmeroPrimo(num){
  for(let divisor = 2; divisor < num; divisor++){
    if(num % divisor === 0){
      return false;
    }
    //return true; // returns 3,5,7,9
  }
  return true; // returns 2,3,5,7 correctly
}

// Factory functions

const celular = {
  marcaCelular: 'ASUS',
  tamanhoTela: {
    vertical: 155,
    horizontal: 75
  },
  capacidadeDaBateria: 5000,
  ligar: function () {
    console.log('Fazendo ligação...')
  }
}
console.log(celular)
console.log(celular.ligar)
celular.ligar()

// Exemplo 2: Refatorado

function criarCelular(
  marcaCelular,
  tamanhoTela,
  capacidadeDaBateria
) {
  return {
    marcaCelular,
    tamanhoTela,
    capacidadeDaBateria,
  }
}

```

```

        religar() {
            console.log('Rediscando...')
        }
    }
}
const celularDois = criarCelular('Moto-G5',5,3000,criarCelular().religar())
console.log(celularDois)
criarCelular().religar()

// Constructor functions

//PascalCase
function Celular(
    marcaCelular,
    tamanhoTela,
    capacidadeDaBateria
) {
    this.marcaCelular = marcaCelular,
    this.tamanhoTela = tamanhoTela,
    this.capacidadeDaBateria = capacidadeDaBateria,
    this.atende = function() {
        console.log('Atende pow!')
    }
}
const mobile = new Celular('iPobre',4,2000)
console.log(mobile)
mobile.atende()

// Natueza Dinâmica de Objetos

const mouse = {
    cor: 'yellow',
    marca: 'multilaser'
}
mouse.velocidade = 5000
mouse.trocarDPI = function(){
    console.log('mudando DPI')
}
//delete mouse.velocidade
//delete mouse.trocarDPI

console.log(mouse)

// Clonando Objetos sem e com spread...

// Reutilizar o objeto acima, mouse

const novoObjeto = Object.assign({
    teclado: 'mecanico'
}, mouse)
console.log(novoObjeto)

const obj = {...mouse, tela: 'Dell 23 inches'}
console.log(obj)

```

```

const obj2 = {...novoObjeto, tela: 'Dell 23 inches'}
console.log(obj2)

// Math

//
https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Math/max

a = Math.random()
console.log(a)

let numMax = [1,3,5,7]
let max2 = Math.max(...numMax)
console.log(max2)

let numMin = [1,3,5,7]
let min2 = Math.min(...numMin)
console.log(min2)

// String methods

// Tipo primitivo
const mensagem = 'uma primeira msg '
// Tipo objeto
const outraMensagem = new String('Bom dia')

console.log(typeof mensagem + ' here')
console.log(typeof outraMensagem)

console.log(outraMensagem.length)
console.log(outraMensagem[0])
console.log(mensagem.includes('again'))
console.log(mensagem.includes('msg'))
console.log(mensagem.startsWith('uma'))
console.log(mensagem.endsWith('tang'))
console.log(mensagem.indexOf('primeira'))
console.log(mensagem.replace('primeira', 'segunda'))
console.log(mensagem.trim())
console.log(mensagem.split(" "))

// Template Literal = ``
// object = {}
// string = '', ''
// boolean = true, false

let nome = 'Alex'
let idade = 39

const msg = `Isso é uma msg para ${nome}
e sua idade é ${idade}`

```

e essa msg usa template literal
para não precisar de usar 'escape \n nem' outros tipos de concatenações que
poluem
a leitura do código.``

```
console.log(msg)
```

```
// Date formats
```

```
const data1 = new Date()  
console.log(data1)
```

```
const data2 = new Date('May 14 2022 09:43')  
console.log(data2)  
data2B = data2.toTimeString()  
console.log(data2B)
```

```
const data3 = new Date(2022,04,14,9,51)  
console.log(data3)  
data3B = data3.toDateString()  
console.log(data3B)
```

```
data3C = data3.toISOString()  
console.log(data3C)
```

```
// ??? não funcionou como esperado  
data4 = data2.setFullYear(2030)  
console.log(data4)
```

```
// Exercício Montando Endereço
```

```
// Criar um objeto endereço que contém:  
// Rua; Cidade; CEP; e uma função exibirEndereco(endereco)
```

```
const enderecoCliente = {  
  rua: 'Av O Daros',  
  cidade: 'cg',  
  cep: 70000000,  
  exibirEndereco() {  
    console.log({enderecoCliente})  
  }  
}  
enderecoCliente.exibirEndereco()  
  
console.log('-----*-----*-----')
```

```
// Solução do professor
```

```
let endereco = {  
  rua: 'a',  
  cidade: 'b',
```

```

    cep: 1
  };

function exibirEndereco(endereco) {
  for (let chave in endereco)
    console.log(chave, endereco[chave]);
}

exibirEndereco(endereco);

```

// Exercício Encontre a Igualdade

```

function Endereco(rua, cidade, cep){
  this.rua,
  this.cidade,
  this.cep
}
const endereco1 = new Endereco('a','b','c');
const endereco2 = new Endereco('a','b','c');
//const endereco3 = endereco1

function saoIguais(endereco1, endereco2) {
  //comparar as propriedades
  return endereco1.rua === endereco2.rua &&
    endereco1.cidade === endereco2.cidade &&
    endereco1.cep === endereco2.cep
}
console.log(saoIguais(
  endereco1, endereco2
));

```

```

function temEnderecoMemIguais(endereco1, endereco2){
  // comparar referência do objeto na memória
  return endereco1 === endereco2;
  //return endereco1 === endereco3;
}
console.log(temEnderecoMemIguais(
  endereco1, endereco2
));

```

// Exercício Objeto Publicacao de Blog

// Criar um objeto de publicação de blog com os seguintes requisitos:
 // título; msg; visualizacoes; comentários(auto, msg); on/offline

```

let pub = {
  titulo: 'a',
  msg: 'b',
  autor: 'c',
  viu: 10,
  comentarios: [

```

```

        {autor: 'a', msg: 'b'},
        {autor: 'a', msg: 'b'}
    ],
    estaOnline: true
}
console.log(pub)

// Exercício Constructor function
// Criar um objeto publ com params acima

// função Construtor c inicial Maiúscula
function Publ(titulo, msg, autor){
    this.titulo = titulo,
    this.msg = msg,
    this.autor = autor,
    this.viu = 11,
    this.comentarios = [],
    this.estaOnline = false
}
let publ = new Publ('a', 'b', 'c')
console.log(publ)

// Exercício Faixa de Preço
// Criar um array de objetos de faixa de preços

let faixas = [
    {cadeira: 'até 700', mínimo: 0, máximo: 700},
    {cadeira: 'de 701 a 1k', mínimo: 701, máximo: 1000},
    {cadeira: '1,001k ou mais', mínimo: 1001, máximo: 999999}
]
console.log(faixas)

// Solução 2:

function criarFaixaDePreco(tooltip, mínimo, máximo){
    return {
        tooltip,
        mínimo,
        máximo,
    }
}
let faixas2 = [
    criarFaixaDePreco('a', 1, 100),
    criarFaixaDePreco('b', 101, 1000),
    criarFaixaDePreco('c', 1001, 10000)
]
console.log(faixas2)

// Solução 3:

function FaixaPreco(tooltip, mínimo, máximo) {
    this.tooltip = tooltip,
    this.mínimo = mínimo,
    this.máximo = máximo
}

```

```

}

let faixas3 = [
  new FaixaPreco('d',10,20),
  new FaixaPreco('d',10,20),
  new FaixaPreco('d',10,20)
]
console.log(faixas3)

// Introdução a Arrays

// Add new elements
// Find elements
// Remove elements
// Divide elements
// Divide Arrays
// Combine Arrays

const nums = [1,2,3]

nums.unshift(0) //begin
console.log(nums)
nums.splice(0,1,'a') //middle
console.log(nums)
nums.push(5) //end
console.log(nums)
nums.indexOf(1)
console.log(nums)

console.log(nums.indexOf(2) !== -1)
console.log(nums.includes(4))

// Removendo elementos de um array
nums.shift(0) //begin
console.log(nums)
nums.splice(2,1) //middle
console.log(nums)

console.log(nums.pop()) // end

const marcas = [
  {id: 1, nome: 'a'},
  {id: 2, nome: 'b'}
]

const marca = marcas.find(function(marca){
  return marca.nome === 'a'
})
console.log(marca)

// Arrow functions
// Reutilizando const marcas acima

```



```
console.log(marcas.find((marca) => marca.nome === 'b'))

// Esvaziando um Array

const nums2 = [1,2,3,4,5,6]

nums2.length = 0
console.log(nums2)

// Combinar e dividir arrays
// Funciona somente para números primos/memória e não para objetos/referências

const primeiro = [1,2,3]
const segundo = [4,5,6]

const combinado = primeiro.concat(segundo)
console.log(combinado)

const dividido = combinado.slice()
console.log(dividido)

// Spread Operator

const combinadoSpread = [...primeiro, 'a',...segundo, 22]
console.log(combinadoSpread)

const clonado = [...combinadoSpread]
console.log(clonado)

// Iterando Arrays with Foreach

clonado.forEach((n,i) => console.log(n,i))

// Join

const juntado = clonado.join('-')
console.log(juntado)

// Split

console.log(juntado.split(' '))
```