

# ALGORITMO

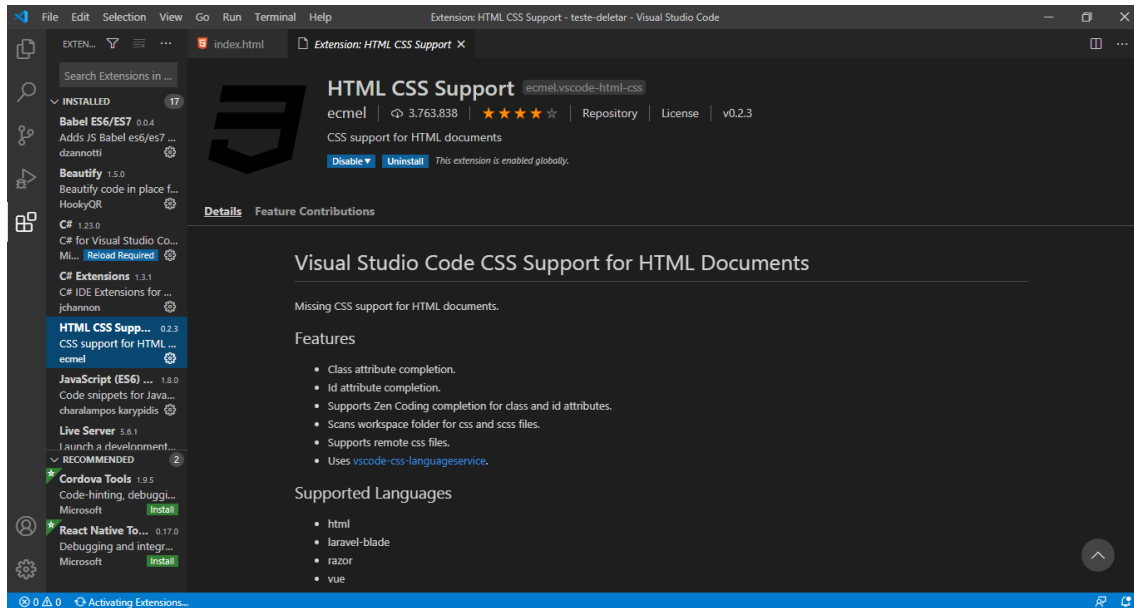
Não é a linguagem de programação que define o programador, mas sim sua lógica.

David Ribeiro Guilherme  
([https://www.pensador.com/frases\\_de\\_programador/](https://www.pensador.com/frases_de_programador/))

*Lógica de  
Programação*



## Instalando o VS Code



## Instalando as extensões

- vscode-html-css
- auto-rename-tag
- LiveServer
- vscode-icons
- HTML CSS Support
- HTML end Tag Labels
- JavaScript (ES6) code snippets

## Node.JS



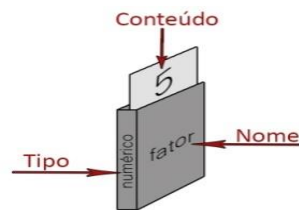
## Variáveis

Toda vez que criamos um programa / algoritmo estamos planejando quais instruções o computador vai executar.

Podemos entender a programação como um planejamento.

Muitas vezes, ao iniciar um programa é necessário declarar as variáveis que serão utilizadas neste algoritmo.

Quando declaramos uma variável estamos reservando um local na Memória Principal do Computador, isto é, um endereço para armazenar o conteúdo de uma variável.



## Saída de dados – Console

```
console.log('Olá mundo!!!');  
console.log('Um exemplo\nde texto\nquebrando linhas');  
console.log('Um exemplo\n\tde texto\n\t\tquebrando linhas');
```

```
console.log('Posso escrever com aspas simples');  
console.log("Posso escrever com aspas duplas");
```

## Comentário

```
// Comentário de uma linha
```

```
/*  
Comentário  
de  
várias  
linhas  
*/
```

## Declaração de variáveis

Não crie variáveis que se diferenciem apenas por sua forma.

Por exemplo: **seuNome** e outra chamada **SeuNome**.

O nome da variável deve representar sua função. Por exemplo: **idade** representa a idade de uma pessoa ou alguma coisa, **cont** representa um contador.

Inicie o nome da variável com uma letra minúscula

Não utilize todas as letras maiúsculas.

Quando o nome tiver mais que uma palavra, a primeira letra de cada palavra após a primeira deve ser maiúscula (notação **Camel Case**);

Não utilize acentos ou caracteres especiais para declarar os nomes das variáveis.

Variáveis não podem ter nomes iguais.

JavaScript é case sensitive.

## var

```
console.log(nome); // undefined
var nome = 'Rodrigo';
console.log(nome); // Rodrigo
```

## Let

```
// Quando utilizamos o 'let' devemos obedecer a lógica de maneira correta
// No exemplo abaixo teremos um erro, pois chamamos a variável 'sobrenome'
// e neste ponto ela ainda não foi declarada
// console.log(sobrenome);
```

```
let sobrenome = 'Dionisio';
console.log(sobrenome);
```

## Constantes

São como variáveis em que uma vez definido o valor, este não pode ser mudado pelo programa. Exemplo:

```
const PI = 3.1415;
```

Por convenção e para diferenciar uma variável normal de uma constante, ela deve ser escrita com letra maiúscula.

Outro ponto é que uma constante deve sempre ser inicializada na declaração, caso contrário, será gerado um erro.

```
const PI = 3.14;
console.log(PI);
// Erro
// Constantes não podem ser alteradas
// PI = 25;
// console.log(PI);
```

## Concatenação, Interpolação e Template String

```
let meuNome = 'Rodrigo';
let meuSobrenome = 'Dionisio';

console.log(meuNome + ' ' + meuSobrenome);
console.log(meuNome , meuSobrenome);
console.log(`${meuNome} ${meuSobrenome}`);
```

## Incorporando um arquivo JavaScript em uma página HTML

### HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>JavaScript</title>
</head>
<body>
  <h1>Algoritmo</h1>
  <h3>Lógica de programação</h3>

  <p>
    Fazendo o link entre um arquivo <strong>JavaScript</strong> e a página <strong>HTML</strong>.
  </p>

  <script src="arquivoJavaScript.js"></script>
</body>
</html>
```

### JavaScript

```
alert('Olá mundo!!!')
```

## Saída de dados – Document.Write

O arquivo JavaScript deve estar vinculado à um documento HTML.

```
document.write("O nome digitado foi: " + nome + '<br/>');
document.write("A idade digitada foi: " + idade);
```

## Entrada de Dados com PROMPT

```
let nome = prompt('Digite seu nome: ');
```

```
let idade = prompt('Digite a sua idade: ');
```

```
document.write("O nome digitado foi: " + nome + '<br/>');  
document.write("A idade digitada foi: " + idade);
```

## typeof

Exibe o tipo de dado.

```
let variavel = prompt("Digite alguma coisa: ")
```

```
// TODO VALOR QUE VEM DE UM FORMULÁRIO SERÁ SEMPRE UMA STRING  
document.write(typeof variavel, '</br>'); // string
```

```
variavel = 'Nome:';  
document.write(typeof variavel, '</br>'); // string
```

```
variavel = 22;  
document.write(typeof variavel, '</br>'); // number
```

```
variavel = 15.55;  
document.write(typeof variavel, '</br>'); // number
```

## Casting

### parseInt()

Faz a conversão para um número inteiro.

```
let numero1 = prompt('Digite um número: ');  
let numero2 = prompt('Digite outro número: ');
```

```
let soma = (parseInt(numero1) + parseInt(numero2))  
document.write(soma)
```

### parseFloat()

Faz a conversão para um número real.

```
let numero1 = prompt('Digite um número: ');  
let numero2 = prompt('Digite outro número: ');
```

```
let soma = (parseFloat(numero1) + parseFloat(numero2))  
document.write(soma)
```

## Operador de atribuição "="

### Exemplo:

```
let a = 10;  
let b = a;  
Console.log(a + " - " + b);
```

## Operadores Aritméticos

+	soma
-	subtração
*	multiplicação
/	divisão
%	módulo

## Lista de exercício 01

01 – Elaborar um algoritmo que imprima em modo console a frase abaixo:

“Aprendendo Algoritmo”

02 – Elabore um algoritmo que imprima a frase da maneira descrita abaixo, uma frase abaixo da outra (imprima no modo console):

Aprendendo Algoritmo  
e Fazendo muito Exercício  
Primeiro fazendo exercício em ‘JavaScript’

03 – Crie um arquivo HTML e vincule um arquivo JavaScript ao arquivo.

No arquivo JavaScript crie duas variáveis, uma para armazenar o seu nome e outra para armazenar sua idade.

Exiba estes dados no console da página Web.

04 – Crie uma aplicação que receba duas variáveis do tipo inteiro, exiba os valores digitados e posteriormente exiba a primeira variável acrescida de uma unidade e a segunda variável decrescida de uma unidade.

05 – Crie uma aplicação que receba 5 números e exiba a soma com a seguinte frase:

“Os números digitados foram ..., ..., ..., ... e sua soma é ... .

06 – Cria uma aplicação que receba dois números e exibir as seguintes mensagens:

O números digitados foram ... e ....

A soma dos números ... e ... é ... .

A subtração dos números ... e ... é ... .

A multiplicação dos números ... e ... é ... .

A divisão dos números ... e ... é ... .

A média dos números ... e ... é ... .



07 – Crie uma aplicação que receba um número inteiro e imprimir seu antecessor e seu sucessor.

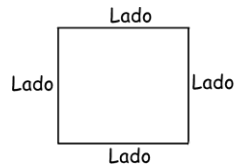
SAÍDA:

O número digitado foi ..., seu antecessor é ... e seu sucessor é ...

08 – Crie uma aplicação para calcular a área e o perímetro de um quadrado.

Área = lado \* lado

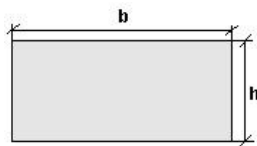
Perímetro = é a soma de todos os lados



09 – Elabore uma aplicação para calcular a área e o perímetro de um retângulo.

Área =  $b \times h$

Perímetro = é a soma de todos os lados



## Operadores reduzidos

`+=, -=, *=, /= e %=`

Exemplo:

```
let valorTexto = prompt("Digite um número")

// convertendo para um número
let numero = parseFloat(valorTexto)
let novoValor = 11;

document.write(`Número Digitado: ${numero} </br>`)
document.write(`novoValor: ${novoValor} </br>`)

// +=
novoValor += numero
document.write(`Número: ${novoValor} </br>`)

// -=
novoValor -= numero
document.write(`Número: ${novoValor} </br>`)

// *=
novoValor *= numero
document.write(`Número: ${novoValor} </br>`)

// /=
novoValor /= numero
document.write(`Número: ${novoValor} </br>`)

// %=
novoValor %= numero
document.write(`Número: ${novoValor} </br>`)
```

## Incremento e decremento

Pré incremento

```
++contador;
```

Pré decremento

```
--contador;
```

Pós incremento

```
contador++;
```

pós decremento

```
contador--;
```

A diferença é que com **++** o cálculo é feito mais rápido que os demais, sendo muito utilizado nos laços (tratado mais adiante).

Exemplo:

```
// suponha que o número digitado seja 10
let valorTexto = prompt("Digite um número")

// convertendo para um número
let numero = parseInt(valorTexto)

document.write(`Número Digitado: ${numero} </br>`) // 10
document.write(`Número Digitado: ${++numero} </br>`) // 11
document.write(`Número Digitado: ${numero} </br>`) // 11
document.write(`Número Digitado: ${numero++} </br>`) // 11
document.write(`Número Digitado: ${numero} </br></br>`) // 12

document.write(`Número Digitado: ${numero} </br>`) // 12
document.write(`Número Digitado: ${--numero} </br>`) // 11
document.write(`Número Digitado: ${numero} </br>`) // 11
document.write(`Número Digitado: ${numero--} </br>`) // 11
document.write(`Número Digitado: ${numero} </br>`) // 10
```

## Funções Matemáticas Básicas

Sugestão:

- Utilize **parseFloat** para não perder o valor das casas decimais.

Potência:

```
// suponha que o número digitado seja 2
let valorTexto = prompt("Digite um número")

// convertendo para um número
let numero = parseFloat(valorTexto)

let potencia_2 = Math.pow(numero,2)
document.write(`Número Digitado: ${potencia_2} </br>`) // 4

let potencia_3 = Math.pow(numero,3)
document.write(`Número Digitado: ${potencia_3} </br>`) // 8
```

## Raiz quadrada:

```
// suponha que o número digitado seja 9
let valorTexto = prompt("Digite um número")

// convertendo para um número
let numero = parseFloat(valorTexto)

let raiz = Math.sqrt(numero)
document.write(`Número Digitado: ${raiz} </br>`) // 3
```

## Raiz Cúbica

```
// suponha que o número digitado seja 27
let valorTexto = prompt("Digite um número")

// convertendo para um número
let numero = parseFloat(valorTexto)

let raiz = Math.cbrt(numero)
document.write(`Número Digitado: ${raiz} </br>`) // 3

Outra maneira de calcular raiz cúbica
raiz = Math.pow(numero, 1/3)
document.write(`Número Digitado: ${raiz} </br>`) // 3
```

## Constante PI:

```
const PI = Math.PI
document.write(PI) // 3.141592653589793
```

## Orientação...

Para qualquer exercício siga os seguintes passos

- Primeiro leia o exercício por inteiro.
- Entenda o exercício, de nada adianta criar um código se você não sabe o que é preciso fazer.
- Defina quais serão as entradas.
- Defina como serão as saídas.
- Execute o código

## Exercícios:

10- Crie uma aplicação que receba um número e imprima sua raiz quadrada.

SAÍDA:

A raiz quadrada de no número .... é ...

11 – Crie uma aplicação que receba um número e imprima seu valor elevado a 2, elevado a 3, elevado a 4 e elevado a 5.

SAÍDA:

O número digitado foi ... E seu valor elevado a 2 é ..., elevado a 3 é ..., ...

12 – Crie uma aplicação que receba um número e imprima sua raiz quadrada e sua raiz cúbica.

SAÍDA:

O número digitado foi ...

Sua raiz cúbica é ...

Sua raiz quadrada é ...

13 – Crie uma aplicação que receba com quatro números e imprimir a média ponderada, sabendo-se que os pesos são respectivamente 1, 2, 3 e 4.

14- Crie uma aplicação que receba o raio de uma circunferência e calcule a área e o perímetro do círculo correspondente.

A fórmula para se calcular a área da circunferência é :  $A = \pi * \text{raio}^2$

A fórmula para se calcular o perímetro da circunferência é :  $P = 2 * \pi * r$

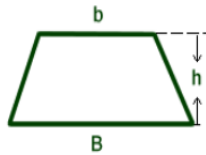
15 – Crie uma aplicação que receba os lados A, B e C de um paralelepípedo.

Calcular e imprimir o volume.

- $\text{Volume} = A * B * C$

16 – Crie uma aplicação para calcular a área de um trapézio qualquer (figura meramente ilustrativa).

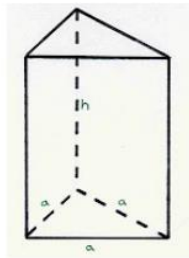
$\text{Área} = ((b + B) * h) / 2$



17 – Crie uma aplicação que possa calcular o volume de um prisma de base triangular (figura meramente ilustrativa).

Para calcular a área de um triângulo utilize a fórmula “(base \* altura)/2”.

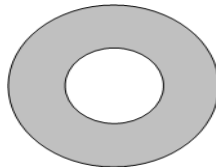
Volume = área da base x altura.



18 – Crie uma aplicação que possa calcular a área de uma coroa de forma circular (figura meramente ilustrativa).

Área da circunferência =  $\pi * \text{raio}^2$

Área = (Área da circunferência Maior) – (Área da circunferência menor)



19 – Crie uma aplicação que possa calcular o volume de um cilindro (figura meramente ilustrativa).

Área da base = área da circunferência

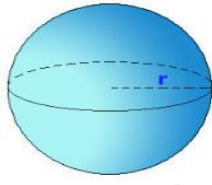
Volume = área da base x altura



20 – Crie uma aplicação para calcular o volume e a área de uma esfera (figura meramente ilustrativa).

$$\text{Área} = 4 * \pi * r^2$$

$$\text{Volume} = (4 * \pi * r^3) / 3$$



## Funções

De modo geral, **função** é um "subprograma" que pode ser chamado por código externo (ou interno no caso de recursão) à função.

Assim como o programa em si, uma função é composta por uma sequência de instruções chamada corpo da função.

Valores podem ser passados para uma função e ela vai retornar um valor.

Em JavaScript, funções são objetos de primeira classe, pois elas podem ter propriedades e métodos como qualquer outro objeto.

```
function nomeDaFuncao(){  
    //o que a função faz (corpo da função)  
}
```

## Funções declaradas

O jeito mais básico de definir funções em JavaScript é através da função declarada (function declaration), toda função de declaração começa com a palavra reservada e obrigatória function, seguida pelo nome da função (também obrigatório) e uma lista de parâmetros (opcionais) separados por vírgula e encapsulados em parenteses (obrigatórios), o último passo é definir as chaves (obrigatórias) que será o corpo da função.

```
// criando a função  
function escrever(){  
    let idade = 18  
    let nome = 'fulano'  
    console.log(`Meu nome é ${nome} e eu tenho ${idade} de idade`);  
}  
  
// Chamando a função  
escrever()
```

## Funções declaradas com parâmetros

```
// criando a função  
function escrever(idade, nome){  
    console.log(`Meu nome é ${nome} e eu tenho ${idade} de idade`);  
}  
  
let idade = 18  
let nome = 'fulano'
```



```
// Chamando a função e passando parâmetros
escrever(idade, nome)
```

## Expressões de função

A function expression (expressão de função) é muito parecida com a function declaration, a diferença é que uma função de expressão pode ser lidada como uma qualquer expressão em JavaScript, devendo ser atribuída a uma variável.

```
let nomeDaFuncao = function (){
    //o que a função faz (corpo da função)
}
```

Repare que é bem parecido com as funções de declaração, uma das súteis diferenças é que ela está sendo atribuída para uma variável, onde não definimos o nome da função e sim o nome da variável que irá referenciar a mesma.

```
// criando a função
let escrever = function(){
    let idade = 18
    let nome = 'fulano'
    console.log(`Meu nome é ${nome} e eu tenho ${idade} de idade`);
}

// Chamando a função
escrever()
```

## Expressões de função com parâmetro

```
// criando a função
let escrever = function(nome, idade){
    console.log(`Meu nome é ${nome} e eu tenho ${idade} de idade`);
}

// Chamando a função
let idade = 18
let nome = 'fulano'
escrever(idade, nome)
```

## Arrow Functions

Arrow functions são simplificações para as functions expression..

```
let escrever = () => {  
  let idade = 18  
  let nome = 'fulano'  
  console.log(`Meu nome é ${nome} e eu tenho ${idade} de idade`);  
}  
  
escrever()
```

## Arrow Functions com parâmetros

```
let escrever = (idade, nome) => {  
  console.log(`Meu nome é ${nome} e eu tenho ${idade} de idade`);  
}  
  
let idade = 18  
let nome = 'fulano'  
escrever(idade, nome)
```

## Capturando dados de um formulário

### HTML

Criando o formulário

```
<label for="nome">nome</label>  
<input type="text" id="nome">  
  
<label for="idade">idade</label>  
<input type="text" id="idade">  
  
<button id="btn">Escrever</button>  
  
<div id="app"></div>
```

nome

idade

### JavaScript

Capturando os dados e exibindo os dados na tela.

```
// capturando os dados
let nome = document.getElementById('nome')
let idade = document.getElementById('idade')
let btn = document.getElementById('btn')
let div = document.getElementById('app')

// criando a função para exibir os dados
function escrever(){
  // exibindo os elementos no console
  console.log(nome);
  console.log(idade);
  console.log(btn);
  console.log(app);

  // inserindo os dados formatados no componente DIV
  // para pegar o valor devemos utilizar '.value'
  div.innerHTML = `O nome digitado foi ${nome.value} e a idade digitada foi
  ${idade.value}.`
}

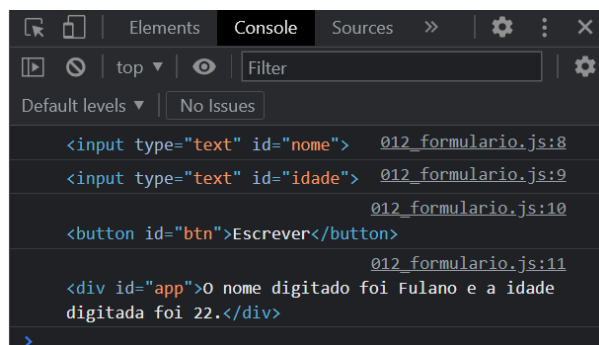
// vinculando o evento de clicar no botão para chamar a função escrever
btn.addEventListener('click', escrever)
```

## Testando a aplicação

nome

idade

O nome digitado foi Fulano e a idade digitada foi 22.



## Exercícios:

Refaça os exercícios de 01 até 20 utilizando formulários conforme demonstrado acima.