



# JAVASCRIPT

## VARIAVEIS E CONSTANTES

1. Conceitos Iniciais;
2. Onde ficam localizadas;
3. Variaveis
4. Constantes
5. Tipos de dados
6. Escopo

# 01.

## Conceitos Iniciais

No mundo da programação, **constantes** e **variáveis** são fundamentais para a manipulação de dados. Ambas são utilizadas para armazenar informações, mas têm características distintas que influenciam como podemos usá-las em nossos programas.

**Variáveis** são espaços de armazenamento que podem conter valores que mudam ao longo da execução do programa. Elas permitem que os desenvolvedores ajustem e atualizem dados dinamicamente, o que é essencial para criar aplicativos interativos e responsivos.

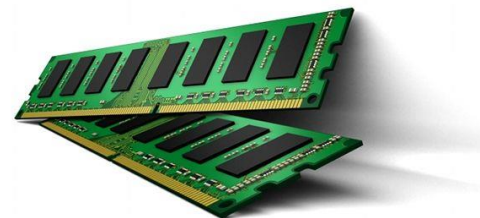
Por outro lado, **constantes** são valores que permanecem inalterados durante a execução do programa. Usar constantes é uma prática recomendada quando você deseja garantir que certos dados, como taxas, limites ou configurações, não sejam acidentalmente modificados, proporcionando maior segurança e clareza ao código.



**ONDE FICAM LOCALIZADAS**

As **variáveis** e **constantes** em um programa não são salvas diretamente no computador em arquivos permanentes, mas sim na **memória RAM** enquanto o programa está em execução.

- Quando um programa é executado, o sistema operacional **aloca** espaço na **memória RAM** para armazenar as **variáveis** e **constantes**.
- As **variáveis podem ser alteradas** durante a execução do programa, enquanto as **constantes têm um valor fixo** que não pode ser modificado.
- É importante notar que, uma vez que o computador é **desligado**, todos **os dados armazenados na memória RAM são perdidos**. Portanto, se houver a necessidade de manter dados de forma persistente, deve-se utilizar métodos de armazenamento, como **arquivos** ou **bancos de dados**.





**VARIÁVEIS**

Uma **variável** é um **espaço na memória** do computador que armazena um valor que pode mudar durante a execução de um programa. Pense nela como uma caixa que você pode usar para guardar informações. Você pode colocar diferentes coisas nessa caixa ao longo do tempo.

Uma variável possui:

- **NOME**
- **TIPO** (em algumas linguagens a tipagem é dinâmica)
- **CONTEÚDO** (inicialmente pode ser nullo ou vazio)

O termo **declarar** no contexto de variavel tem o significado de **criação**.



# ARMAZENAR UM VALOR NA VARIÁVEL



Para atribuir um valor a uma variável usamos o operador = (**igual**) chamado na programação de **recebe**.

```
let A = 10;           // A variável A recebe o valor 10
let texto = "Diego"; // A variável texto recebe o valor "Diego"
```

### Atribuindo valores mais tarde:

Você pode declarar a variável primeiro e atribuir valores depois, como mostrado a seguir:

```
let A;
let texto;

A = 10;           // Agora A recebe o valor 10
texto = "Diego"; // Agora texto recebe o valor "Diego"

console.log(A);   // 10
console.log(texto); // "Diego"
```

# EXEMPLO DE APLICAÇÃO

Em PHP, você utiliza o símbolo **\$** para declarar uma variável. Não é necessário especificar o tipo da variável, pois o PHP é uma linguagem de tipagem dinâmica.

```
<?php
    $nome = "Victor"; // texto (caracter)
    $idade = 33;      // inteiro (int)
    $altura = 1.75;   // decimal (float)
    $data_de_nascimento = "1991-06-29"; // data (date)

    echo "Nome: " . $nome . ", Idade: " . $idade . ", Data de Nascimento: " .
    $data_de_nascimento;
    // Saída: Nome: Victor, Idade: 33, Data de Nascimento: 1991-06-29
?>
```

**Nota:** Neste momento, não se preocupe com os comandos em PHP ou com os tipos de dados, pois abordaremos esses conteúdos em aulas futuras. Uma dica sobre concatenação (juntar tipos diferentes) e que no JavaScript usa o **“+” (mais)** e no PHP usa **“.” (ponto)**.

# JAVASCRIPT (JS)

```
let nome = "Victor"; // let (escopo de bloco) (caractere)
let idade = 33; // let (escopo de bloco) (inteiro)
var altura = 1.75; // (escopo global) (real)
var data_de_nascimento = "1991-06-29"; // (escopo global) (data)

console.log("Nome: " + nome + ", Idade: " + idade + ", Data de Nascimento: " +
data_de_nascimento);
// Saída: Nome: Victor, Idade: 33, Data de Nascimento: 1991-06-29
```

**Nota:** Neste momento, não se preocupe com os comandos em JavaScript, com os tipos de dados ou sobre escopo, pois abordaremos esses conteúdos em aulas futuras.

Variável do tipo var, não está sendo mais recomendada seu uso, por esse motivo iremos priorizar a utilização do **let**.

# PYTHON

```
nome = "Victor"           # texto (character)
idade = 33                # inteiro (int)
altura = 1.75             # decimal (float)
data_de_nascimento = "1991-06-29" # data (date)

print("Nome:", nome, ", Idade:", idade, ", Data de Nascimento:", data_de_nascimento)
# Saída: Nome: Victor , Idade: 33, Data de Nascimento: 1991-06-29
```

**Nota:** O comando em **Python** é apenas um exemplo para ilustrar que o conceito de criação de variáveis é comum entre as diferentes linguagens de programação. O importante é entender a semântica por trás de cada linguagem e fazer os ajustes necessários conforme o contexto. Foque nos conceitos fundamentais, em vez de decorar sintaxes específicas.

# REGRAS

77 anos Senac. #EuFaçoParte **Senac**

- Sempre declare uma variável antes de utilizá-la no código.
- O computador reserva espaço na memória ao declarar uma variável.
- O tamanho que cada tipo de variável ocupa na memória varia entre as linguagens de programação.
- Não declare uma variável se você não for utilizá-la.
- Uma variável deve receber apenas valores do seu tipo declarado, exceto em linguagens de tipagem dinâmica.
- Cada linguagem possui tipos de dados próprios; abordaremos JavaScript e PHP em aulas diferentes.



**CONSTANTES**



Uma **constante** é um espaço na memória do computador que armazena um valor fixo, ou seja, um valor que não pode ser alterado durante a execução de um programa. Pense em uma constante como um recipiente selado que contém uma informação que não deve ser mudada.

**Nota:** Por convenção, ao declarar constantes, utilizamos letras maiúsculas. Essa prática ajuda a indicar que o valor atribuído à não deve ser alterado, tornando o código mais legível e fácil de entender.



**ARMAZENAR UM VALOR NA CONSTANTE**

Para atribuir um valor a uma constante usamos o operador = (**igual**) chamado na programação de **recebe**.

```
const A = 10;           // Agora A é uma constante e recebe o valor 10
const texto = "Diego";  // Agora texto é uma constante e recebe o valor "Diego"
```

## Por que não podemos atribuir valores mais tarde com const?

Ao contrário de **let** ou **var**, quando usamos **const**, não podemos declarar uma constante sem atribuir um valor imediatamente. Se tentarmos declarar e atribuir o valor mais tarde, o JavaScript vai gerar um erro.

# EXEMPLO DE APLICAÇÃO

# PHP

Em **PHP**, você pode definir uma **constante** usando a função **define()** ou a palavra-chave **const**. Constantes em PHP não precisam do símbolo \$.

```
<?php
define("PI", 3.14); // Usando define()
const TAXA_IMPOSTO = 0.15; // Usando const

echo "Valor de PI: " . PI . "\n"; // Saída: Valor de PI: 3.14
echo "Taxa de Imposto: " . TAXA_IMPOSTO . "\n"; // Saída: Taxa de Imposto: 0.15

// Tentar alterar uma constante gerará um erro
// PI = 3.14159; // Erro: Cannot redefine constant PI
?>
```

**Lembrete:** Neste momento, não se preocupe com os comandos em PHP ou com os tipos de dados, pois abordaremos esses conteúdos em aulas futuras.

O “\n” é uma string especial para pular linha no PHP.

# JavaScript

Em **JavaScript**, você pode usar a palavra-chave **const** para declarar uma constante. O valor atribuído não pode ser alterado.

```
const PI = 3.14; // Declaração de constante
const TAXA_IMPOSTO = 0.15; // Outra constante

console.log("Valor de PI:", PI); // Saída: Valor de PI: 3.14
console.log("Taxa de Imposto:", TAXA_IMPOSTO); // Saída: Taxa de Imposto: 0.15

// Tentar alterar uma constante gerará um erro
// PI = 3.14159; // Erro: Assignment to constant variable.
```

**NOTA:** por convenção em constante usamos maiúsculos.

# Python

Em **Python**, **não existe uma palavra-chave** específica para declarar **constantes**. No entanto, a convenção é usar nomes de variáveis em letras maiúsculas para indicar que são constantes.

```
PI = 3.14 # Convenção de constante
TAXA_IMPOSTO = 0.15 # Outra constante

print("Valor de PI:", PI) # Saída: Valor de PI: 3.14
print("Taxa de Imposto:", TAXA_IMPOSTO) # Saída: Taxa de Imposto: 0.15

# Embora não haja um mecanismo para impedir a alteração, é uma prática não modificar
constantes
# PI = 3.14159 # Não recomendado, mas não gerará erro
```



**TIPO DE DADOS (JS)**



Dados Numéricos, **number**:

- Representa números inteiros e decimais (ex: 25, 99.99).

Dados Literais, **(Texto)string**:

- Sequências de caracteres representando texto (ex: "Diego").

Dados Lógicos, **boolean**:

- Pode ser true ou false, para valores lógicos.

Dados Estruturados:

- **object**: Coleção de pares chave-valor (ex: { nome: "Diego", idade: 30 }).
- **array**: Lista ordenada de valores (ex: [1, 2, 3, 4]).

Outros Tipos Especiais:

- **undefined**: Variável declarada sem valor.
- **null**: Ausência intencional de valor.
- **symbol**: Valor único e imutável usado como identificador.
- **bigint**: Números inteiros muito grandes (ex: 1234567890n).

Baseado nas **frases** abaixo, vamos filtrar quais informações são **variáveis**, sugestão de **nome de variável** e qual seu respectivo **tipo**.

## Frases

Olá Victor, você estuda no SENAC ?

**Victor** [**nome**] (**caractere, string**) e **SENAC** [**instituicao**] (**caractere, string**)

Seja bem vindo ao curso de Administração, Matheus.

**Administração** [**curso**] (**caractere, string**) e **Matheus** [**nome**] (**caractere, string**)

Seu peso é: 89,50Kg.

**89,50** [**peso**] (**real, float, number**)

Júlia, você tem 14 anos e mora em Viçosa – MG ?

**Júlia** [**nome**] (**caractere, string**), **14** [**idade**] (**inteiro, int, number**), **Viçosa** [**cidade**] (**caractere, string**) e **MG** [**estado**] (**caractere, string**)

Marcos nasceu em 29/06/1991

**Marcos** [**nome**] (**caractere, string**), **29/06/1991** [**data-nascimento**] (**data, date**)

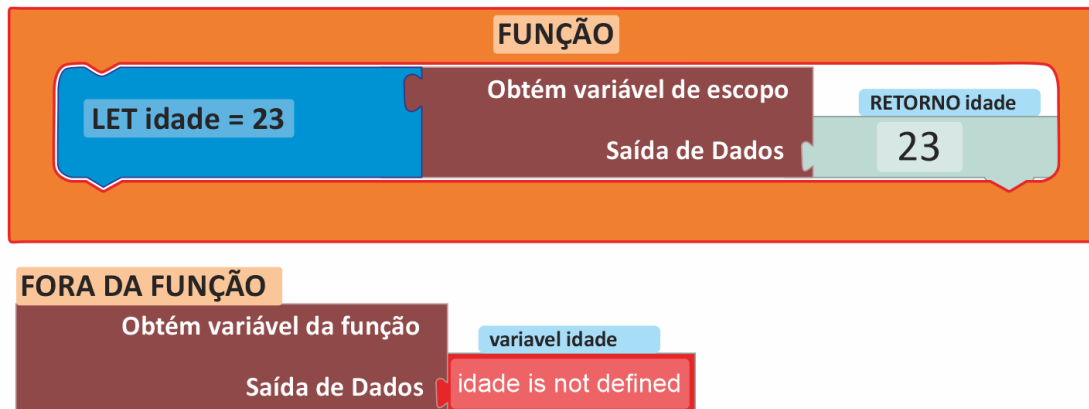


ESCOPO

**Escopo** é o contexto em que as variáveis são declaradas e acessadas no código.

Existem dois tipos principais de escopo:

- **Escopo Global:** Variáveis declaradas fora de qualquer função, acessíveis de qualquer lugar no código.
- **Escopo Local:** Variáveis declaradas dentro de uma função ou bloco, acessíveis apenas dentro desse contexto.



## Escopo Global

Declarado fora de funções/blocos e acessível em qualquer lugar.

```
let nome = "Diego"; // Escopo global
function exibirNome() {
  console.log(nome); // Acessa a variável global
}
```

## Escopo de Função

Variáveis declaradas dentro de uma função só podem ser acessadas nessa função.

```
function mostrarIdade() {
  let idade = 30; // Escopo local (função)
}
console.log(idade); // Erro, fora do escopo da função
```

## Escopo de Bloco

Com **let** e **const**, variáveis e constantes podem ser limitadas a blocos **{ }** como em **if** ou **for**.

```
if (true) {  
    let linguagem = "JavaScript"; // Escopo de bloco  
}  
console.log(linguagem); // Erro, variável está fora do bloco
```

## QUAL É A SAÍDA:

```
if (true) {  
    const LINGUAGEM = "PHP";  
}  
console.log(LINGUAGEM);
```

**LINGUAGEM is not defined**

```
function mostrarIdade() {  
    let idade = 30;  
}  
console.log(idade);
```

**idade is not defined**

```
let idade = 23;  
function mostrarIdade() {  
    let idade = 33;  
}  
console.log(idade);
```

**23**

```
let idade = 23;  
function mostrarIdade() {  
    idade = 33;  
    return idade;  
}  
console.log(mostrarIdade());
```

**33**

```
let idade = Victor;  
console.log(idade);
```

**Victor is not defined**

```
let idade = "Victor";  
console.log(idade);
```

**Victor**



Siga o Senac em Minas nas Redes Sociais:

