
html
html
body
JavaScript
h1>Hello</h1>
center>

JAVASCRIPT

LINGUAGEM DE PROGRAMAÇÃO

JAVASCRIPT

LINGUAGEM DE PROGRAMAÇÃO

JavaScript é uma **linguagem de programação de alto nível, interpretada** que permite adicionar **interatividade** às páginas web. Criada em **1995** por **Brendan Eich** na **Netscape**, é uma das **três tecnologias fundamentais da web**, junto com **HTML** e **CSS**.

O **JavaScript** executa no **navegador do usuário (client-side)**, permitindo **manipular elementos da página, responder a eventos** e criar **experiências dinâmicas** sem necessidade de recarregar a página.

JAVASCRIPT

LINGUAGEM DE PROGRAMAÇÃO

O **JavaScript** transformou a web de **páginas estáticas** em **aplicações interativas completas**. É **essencial para o desenvolvimento web moderno**, pois permite criar **interfaces responsivas**, **validar formulários em tempo real**, **carregar conteúdo dinamicamente** e desenvolver **aplicações web complexas**. Com o surgimento do **Node.js**, expandiu seu alcance para o **desenvolvimento backend**, permitindo usar a **mesma linguagem em toda a stack**.

Hoje, o **JavaScript** é a uma das **linguagens mais utilizadas no mundo**, com um **ecossistema vasto de frameworks** como **React, Angular e Vue.js**, sendo **fundamental para qualquer desenvolvedor web**.

JAVASCRIPT

COMO UTILIZAR

O código Javascript deve ser escrito dentro de script **tags**, que podem ser colocadas tanto no **head** quanto no body do documento html.

É mais recomendável, porém, colocá-las ao final do **body**, pois isso melhora o desempenho da página.

```
<body>

    <!-- conteúdo da página  -->

    <script>

        // código javascript

    </script>

</body>
```

JAVASCRIPT

COMO UTILIZAR

Outra maneira, que na verdade é a mais usada, é deixar o código Javascript em arquivos externos (a extensão dos arquivos Javascript é '.js'). Desta forma organizamos melhor o código, separando o que é html e o que é Javascript. Neste caso, a script tag funcionará como um link por meio da propriedade 'src'.

```
<body>

    <!-- html da página  -->

    <script src='js/scripts.js'></script>

</body>
```

JAVASCRIPT

COMO UTILIZAR

Para uma simples utilização de JavaScript, podemos criar um arquivo HTML em nosso VSCode e adicionar a tag script conforme a imagem abaixo.

Temos dois comandos, um aparecerá em uma janela e o outro somente no console do navegador.

```
alert('Mensagem enviada pelo comando alert');

console.log('Mensagem enviada pelo console');
```

JAVASCRIPT

COMANDOS

Comandos (ou Statements) Javascript são instruções que serão executadas pelo navegador quando a página for carregada. Devemos indicar o final de um comando com ponto e vírgula (;). O código Javascript pode ser composto por múltiplos comandos, que serão executados de forma sequencial.

```
console.log('comando 1');
console.log('comando 2');
```

JAVASCRIPT

INTERPRETADORES

Os nossos arquivos Javascript serão compostos por dezenas, centenas ou até milhares de comandos, que juntos formarão toda a lógica de funcionamento da nossa página.

Cada **navegador** possui o seu próprio **interpretador** de **Javascript**, que tratará de executar os comandos que escrevermos. Os interpretadores são também chamados de **Javascript Engines**. Ao lado estão os interpretadores dos navegadores mais populares:

Navegador	Interpretador
Google Chrome	V8
Mozilla Firefox	Gecko, SpiderMonkey e Rhino
Safari	Nitro
Microsoft Edge	Chakra
Opera	Carakan

JAVASCRIPT

COMENTÁRIOS

Comentários são linhas de código que **não são executadas pelo navegador**. Eles servem para explicar o funcionamento do código, ou simplesmente para desativar temporariamente parte do código. A dupla barra (**//**) comenta uma linha de código. Para comentar várias linhas ao mesmo tempo comece com (**/***) e termine com (***/**).

```
// Sou um comentário e não serei executado pelo browser

console.log('Eu serei executado');

// console.log('Não serei executado pois fui desativado');

/*
    Nenhuma destas linhas
    será executada
    pelo browser
    console.log('Não serei executado');
*/
```

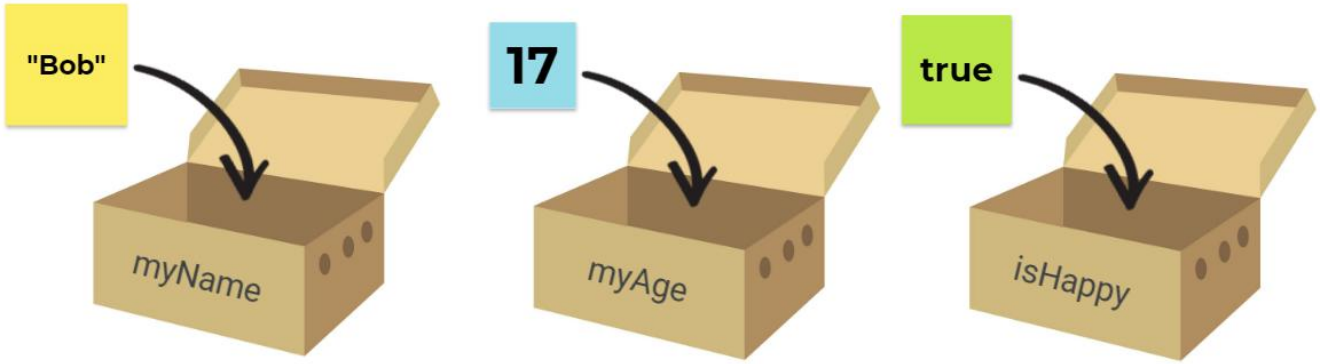
JAVASCRIPT

VARIÁVEIS

Uma variável é uma localização na ***memória RAM*** do computador que é utilizada para ***armazenar temporariamente*** os dados que são utilizados pelo programa.

As variáveis possuem características como:

- Identificação
- Endereço
- Tipo
- Tamanho
- Valor



JAVASCRIPT

VARIÁVEIS

A escolha do estilo de escrita para variáveis, funções e outros elementos do código é mais uma questão de convenção e estilo do que uma questão de funcionalidade direta. No entanto, o estilo de escrita desempenha um papel importante na legibilidade, manutenção e colaboração em projetos de programação. Aqui estão algumas formas de declarar as variáveis:

Exemplos: *Camel case, Pascal case e Snake case*.

JAVASCRIPT

VARIÁVEIS

Camel case: Camel case deve começar com a primeira letra minúscula e a primeira letra de cada nova palavra subsequente maiúscula:

```
valorFinal
```

Pascal case: Também conhecido como “upper camel case” ou “capital case”, pascal case combina palavras colocando todas com a primeira letra maiúscula:

```
ValorFinal
```

JAVASCRIPT

VARIÁVEIS

Snake case: conhecido também como “underscore case”, utilizamos underline no lugar do espaço para separar as palavras.

```
valor_final
```

JAVASCRIPT

VARIÁVEIS

Um nome de variável é uma sequência de letras (a → z, A → Z) e números (0 → 9), que devem sempre começar com uma letra ou $.

Espaços, caracteres especiais como #, @, etc. são proibidos, exceto para o caractere _ (sublinhado/underline) e $ (cifrão).

Pode utilizar palavras acentuadas mas não é recomendado.

JAVASCRIPT

VARIÁVEIS

Além dessa regra é importante também estar atento às palavras reservadas da linguagem.

break	export	super
case	extends	switch
catch	finally	typeof
class	for	this
const	function	throw
continue	if	try
debugger	import	var
default	in	void
delete	instanceof	while
do	new	with
else	return	yield

JAVASCRIPT

VARIÁVEIS

```
a               // ok
Aluno           // ok
$nome           // ok
novo_aluno      // ok
aluno_2         // ok
1_b             // ilegal, pois começa com número
meu-nome        // ilegal, pois contém traço (apenas underscores são permitidos)
meu nome        // ilegal, pois contém espaço
new             // ilegal, pois 'new' é uma keyword, conforme será explicado logo abaixo.
```

JAVASCRIPT

VARIÁVEIS

As variáveis precisam então ter nomes, para que possamos nos referir a elas depois de criá-las. Estes nomes são os identificadores, e devem seguir as regras enumeradas mais abaixo.

Para criar uma variável em Javascript, usamos a keyword 'var', seguida pelo identificador, o sinal de igual (=) e o valor que queremos atribuir a ela:

```
var latitude = 40.87663;
var longitude = -8.08373;
```

JAVASCRIPT

VARIÁVEIS

Além do **var** podemos utilizar o **let** e **const** em variáveis.

var: Possui escopo global ou de função, permite reatribuição e re-declaração, mas ignora blocos {}.

let: Tem escopo de bloco, permite reatribuição, mas não permite re-declaração no mesmo escopo.

const: Também tem escopo de bloco, não permite reatribuição nem re-declaração, mas seus objetos e arrays podem ter valores internos modificados.

JAVASCRIPT

CASE SENSITIVE

Javascript é uma linguagem "**case sensitive**", o que significa que ela é "sensível à caixa" ou, num português mais compreensível, sensível a letras maiúsculas e minúsculas.

Portanto há de se tomar um certo cuidado ao nomear variáveis e funções.

Por exemplo, se criarmos uma variável com o nome "**matricula**" e depois chamarmos esta variável por "**MATRICULA**", ela não será encontrada e o comando gerará um erro.

Isto também significa que apesar de existir uma keyword chamada "**function**", teoricamente podemos usar os nomes "**Function**" ou "**FUNCTION**" para outras coisas, mas não é aconselhável.

JAVASCRIPT

TIPOS DE DADOS - STRINGS

São ***caracteres*** delimitados pelo símbolo aspas (**" "**), eles são representados por letras (de A até Z) números (de 0 até 9), símbolos (exemplo, todos os símbolos imprimíveis existentes em um teclado) ou palavras contendo esses símbolos.

O tipo ***caractere*** também é conhecido como ***alfanumérico***, ***string*** (em inglês, cordão, colar).

```
var nome = "João";  // podem ser usadas aspas duplas
var sobrenome = 'Gomes';  // ou aspas simples
var cpf = '111.111.111-11';
var telefone = "998887655";
var ddd = '21';
var email = 'joao@gmail.com';
```

JAVASCRIPT

TIPOS DE DADOS - NUMBERS

Números (ou numbers) podem ser positivos, negativos, inteiros ou decimais (também chamados de floats).

```
var ano_nascimento = 1986;
var preco = 16.99;  // floats devem ser usadas com ponto, nunca com vírgula.
var temperatura = -5;
```

JAVASCRIPT

TIPOS DE DADOS - BOOLEANS

Booleanos (ou booleans) são os valores verdadeiro (true) e falso (false).

Os booleanos surgem quando usamos operadores de comparação:

Operador	Descrição
>	Maior que
>=	Maior que ou igual a
<	Menor que
<=	Menor que ou igual a
==	Igual a
===	Igual tipo e valor
!=	Não é igual a
!==	Não é igual o tipo OU não é igual o valor

```
var teste1 = 20 > 2; // true

var teste2 = 2 >= 2; // true

var teste3 = 5 < 5; // false

var teste4 = 5 <= 10; // true

var teste5 = 20 == 20; // true

var teste6 = 'ivan' == 'joão' // false
```

JAVASCRIPT

TIPOS DE DADOS – NULL E UNDEFINED

Nem sempre variáveis possuem valores e os tipos de dados null e undefined indicam justamente a ausência de valor de uma variável.

Quando criamos uma variável e não lhe atribuímos valor, ela automaticamente recebe o valor **undefined** e este é também o seu tipo de dados.

Quando temos uma variável com algum valor e queremos "apagar" este valor, podemos atribuir-lhe o valor de **null**.

```
var undf;
console.log(undf);
```

```
var temperatura = 35;
console.log(temperatura);

temperatura = null;
console.log(temperatura); // O console mostrará null
```

JAVASCRIPT

CONVERSÕES

Operador	Descrição	Exemplo	Resultado
String para Number	Converte uma string para um número. Retorna NaN para strings.	Number("123"); Number("123teste");	123 NaN
String para Float	Converte uma string para um número decimal (float). Tenta extrair números.	parseFloat("123.45"); parseFloat("42.5teste")	123.45 42.5
String para Inteiro	Converte uma string para um número inteiro. Tenta extrair números.	parseInt("123.45"); parseInt("42.5teste")	123 42
Number para String	Converte um número para uma string.	String(123);	“123”
Number para String (toString)	Converte um número para uma string usando o método toString().	(123).toString();	“123”

JAVASCRIPT

OPERADORES ARITMÉTICOS

Operador	Descrição	Exemplo	Resultado
+	Adição	5 + 3	8
-	Subtração	5 - 3	2
*	Multiplicação	5 * 3	15
/	Divisão	6 / 3	2
%	Resto da divisão (módulo)	5 % 3	2
++	Incremento (aumenta em 1)	x = 5; x++	6 (valor de x)
--	Decremento (diminui em 1)	x = 5; x--	4 (valor de x)
**	Exponenciação	2 ** 3	8

JAVASCRIPT

INPUT - PROMPT

O prompt() é um método em JavaScript utilizado para exibir uma caixa de diálogo que solicita ao usuário que insira um valor. Essa caixa de diálogo geralmente apresenta um campo de texto e um botão para confirmar a entrada. O valor inserido pelo usuário é retornado como uma string.

```
var valor = prompt("Mensagem de solicitação");

var nome = prompt("Qual é o seu nome?");
console.log("Olá, " + nome);
```


LET'S
GO!!!