

# Introdução à Javascript

Jesiel Viana  
@jesielviana  
jesiel@ifpi.edu.br

# Roteiro

- Pré-requisitos
- Objetivo
- Introdução
- Histórico
- Primeiros códigos
- Sintaxe básica
- Tipos de dados
- Operadores
- Funções

# Pré-requisitos

- Um entendimento geral da internet e da Web.
- Um bom conhecimento de HTML.
- Alguma experiência em programação.

# O que é Javascript?

JavaScript é uma linguagem de script orientada a objetos, multiplataforma. É uma linguagem leve. Dentro de um ambiente de host (por exemplo, um navegador web) o JavaScript pode ser ligado aos objetos desse ambiente para prover um controle programático sobre eles.

# JavaScript e ECMAScript

O JavaScript é padronizado pela Ecma International para entregar uma linguagem de programação padronizada, baseada em JavaScript.

Para que o Javascript evoluísse obedecendo a determinados padrões e normativas, os criadores do Javascript se associaram ao ECMA (European Computer Manufactures Association) em 1996. Como o nome Javascript já havia sido patenteado pela Sun Microsystems (atual Oracle), optou-se por se definir um novo nome à linguagem, surgindo então o ECMAScript.

# Origem

JavaScript foi originalmente desenvolvido por **Brendan Eich** quando trabalhou na **Netscape** sob o nome de Mocha, posteriormente teve seu nome mudado para **LiveScript** e por fim **JavaScript**. LiveScript foi o nome oficial da linguagem quando foi lançada pela primeira vez na versão beta do navegador Netscape 2.0 em setembro de 1995 (Wikipedia, 2018).



# Características

- Linguagem de scripting
- Interpretado
- Orientada a objetos
- Multiplataforma
- Conversões automáticas
- Client Side e Server Side
- Case

sensitive

# Javascript e Java

JavaScript e Java são similares em algumas coisas, mas são diferentes em outras. O JavaScript assemelha-se ao Java, porém não possui tipagem estática e checagem rígida de tipos como o Java.

JavaScript é uma linguagem mais livre em comparação a Java. Você não tem de declarar todas as variáveis, classes e métodos.



# Quiz 1

Na linguagem Javascript as palavras "nome" e "Nome" são consideradas iguais?

1. Sim
2. Não

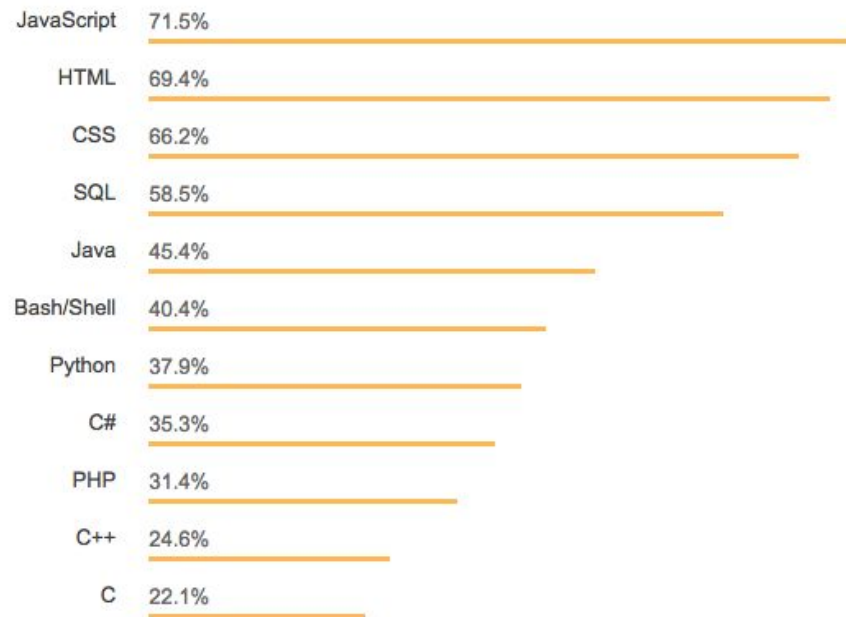


## Most Popular Technologies

### Programming, Scripting, and Markup Languages

All Respondents

Professional Developers



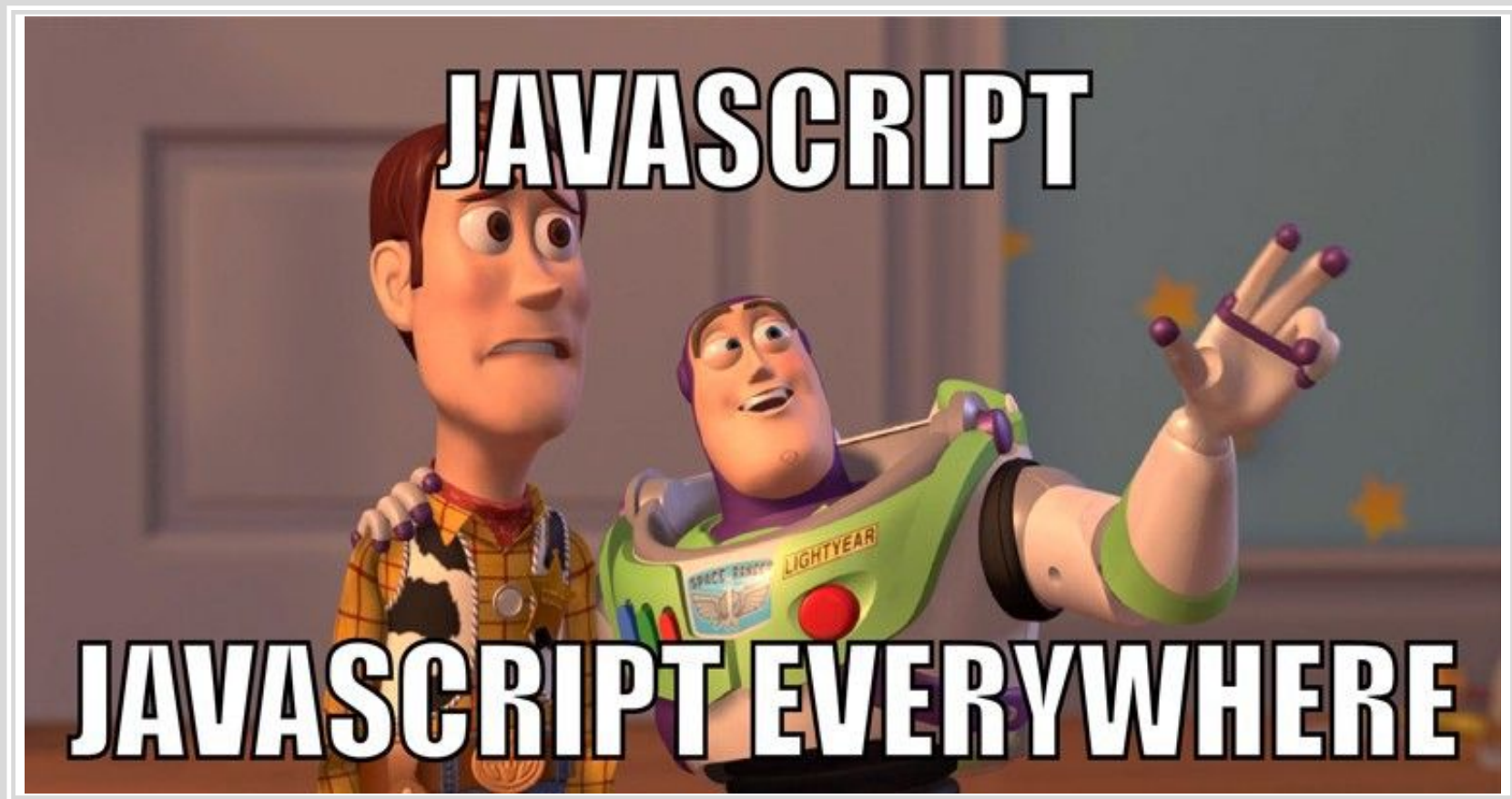
Linguagens  
Mais Populares  
(Stackoverflow, 2018)

# Vagas Javascript

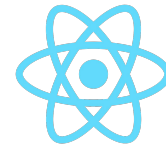
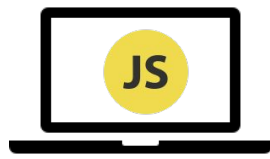
The screenshot shows the LinkedIn search interface. At the top, the search bar contains 'javascript' and the location is set to 'Brazil'. Below the search bar, there are filters for 'Vagas' (Jobs), 'Data do anúncio' (Posting date), and 'Recursos do LinkedIn' (LinkedIn features). The results are sorted by 'Relevância' (Relevance). The first result is 'Javascript | Brazil' with 4,438 results. Below this, there are three job listings:

- Full-stack Developer** by **Blackbook Editora**. Location: Belo Horizonte, MG, BR. Posted 1 week ago. Status: Candidatura simplificada.
- Application Developer - Java & Web Technologies** by **IBM**. Location: Belo Horizonte, BR. Posted 3 weeks ago. Status: 4 conexões trabalham aqui.
- Desenvolvedor Angular** by **SysMap Solutions**. Location: São Paulo e Região, Brasil. Posted 5 hours ago. Status: Candidatura simplificada.

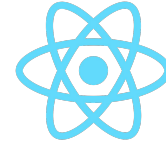
Each listing includes a profile picture of the user who posted it and a toggle switch for 'Alerta de vaga desativado' (Job alert deactivated).



Cliente  
Web



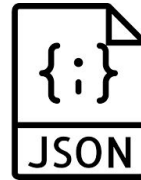
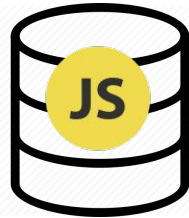
Cliente  
Mobile



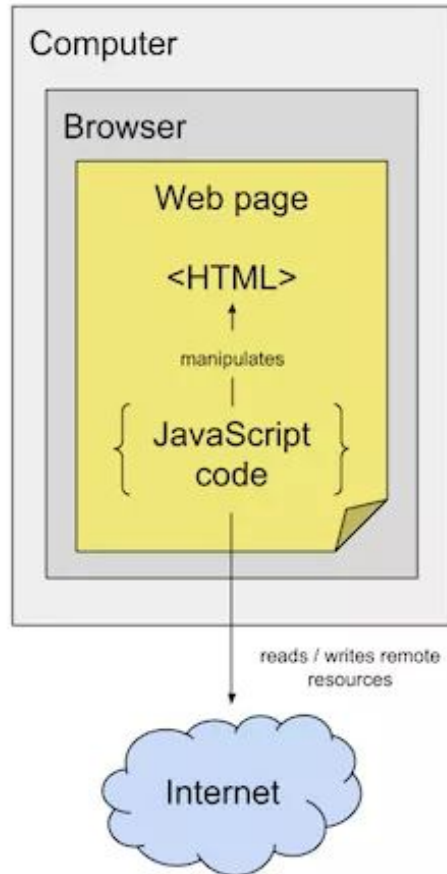
Servidor



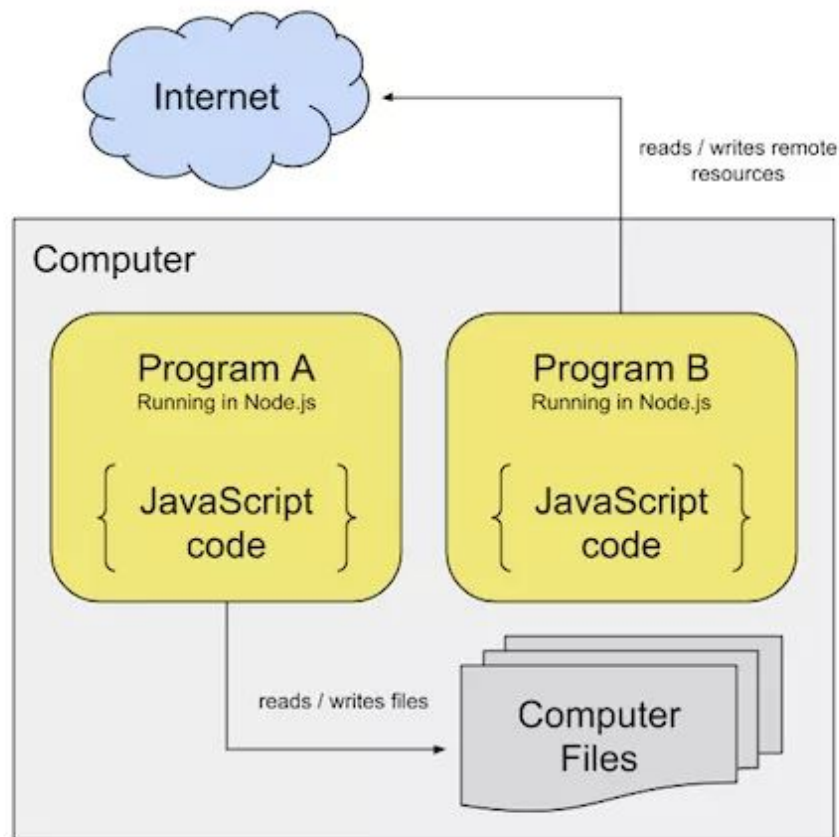
Banco de  
Dados



## Client-side JavaScript



## Server-side JavaScript



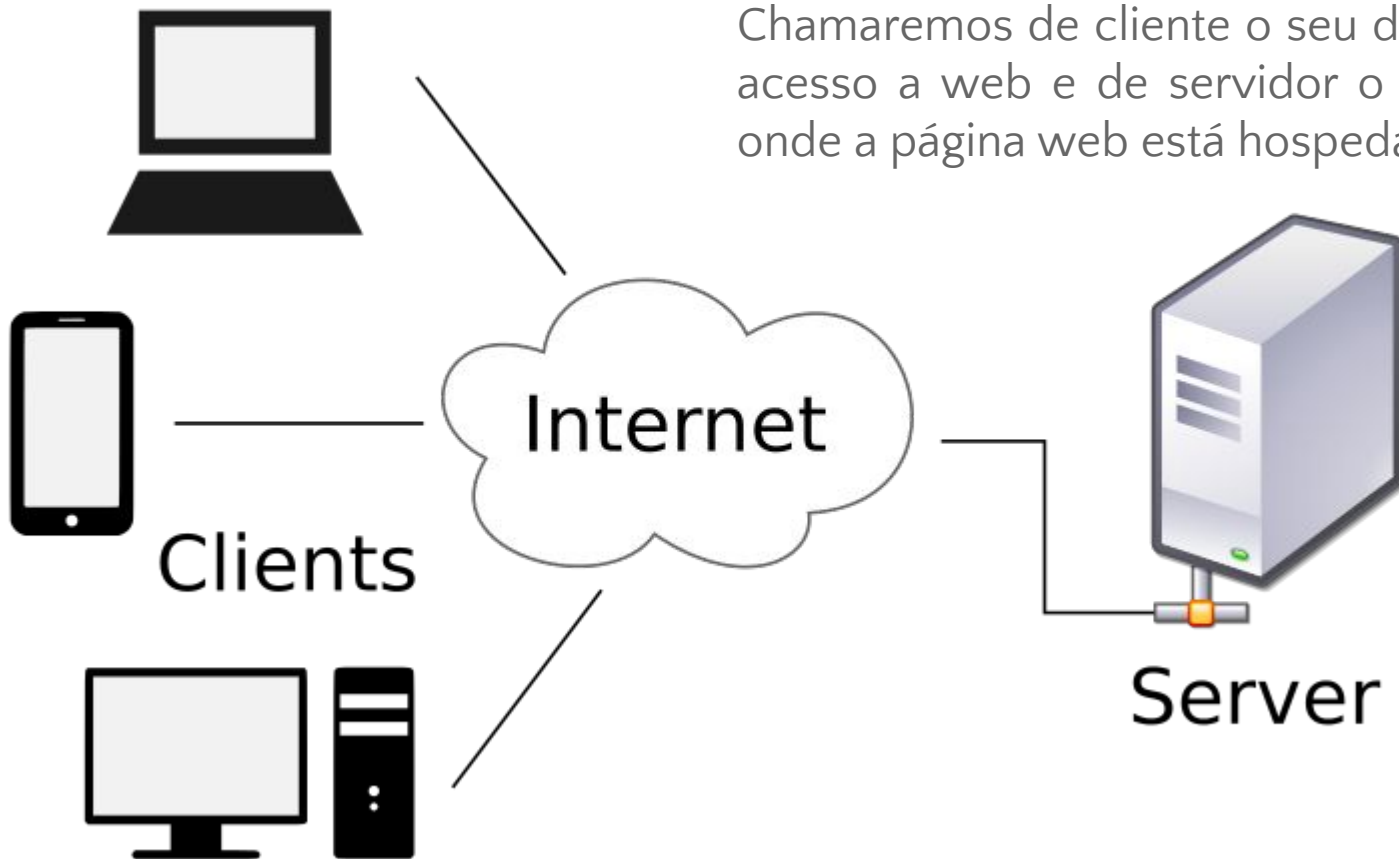
# Atividade 1

Pesquise e encontre pelo menos um exemplo de aplicação:

- web que usa javascript no front end.
- mobile que usa javascript no front end.
- que usa javascript no back end (Nodejs).
- que usa banco de dados(mongodb) com javascript (json).



Chamaremos de cliente o seu dispositivo de acesso a web e de servidor o computador onde a página web está hospedada.





# Javascript do lado cliente

O JavaScript do lado cliente (Javascript Client Site) estende-se do núcleo da linguagem, fornecendo objetos para controlar um navegador web e seu Document Object Model (DOM). Por exemplo, as extensões do lado do cliente permitem que uma aplicação coloque elementos em um formulário HTML e responda a eventos do usuário, como cliques do mouse, entrada de formulário e de navegação da página.

## Quiz 2

Nodejs é um interpretador Javascript que possibilita o uso da linguagem no lado servidor?

- Sim
- Não

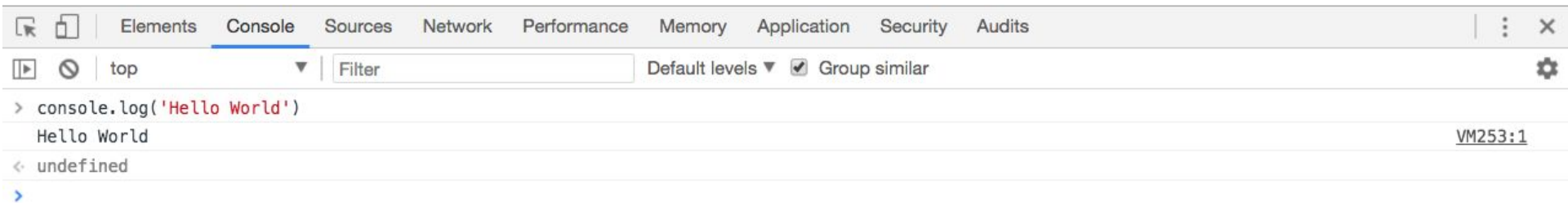
An illustration of a person's hands typing on a laptop keyboard. The laptop screen displays two overlapping windows with code and design elements. To the left is a white coffee cup with a red stripe and steam rising from it. To the right is a small potted plant with green leaves. The background is a solid olive green.

# Vamos Praticar!!!

# Começando com Javascript

Começar a aprender JavaScript é fácil: Tudo o que você precisa é de um navegador web moderno. Abra o **console web**, o atalho geralmente é F12.

O console web mostra informações sobre a página web que está sendo carregada atualmente e também inclui a linha de comando que você pode utilizar para executar códigos JavaScript na página atual.



# Hello World

Para começar a escrever JavaScript, abra o Console Web e escreva seu primeiro código JavaScript "Olá, mundo".

```
1 function primeiroCodigo(nome) {  
2   alert("Olá, " + nome);  
3 }  
4  
5 primeiroCodigo("mundo"); // "Olá, mundo"
```

# Sintax

```
// comentário de uma linha  
/* isto é um comentário longo  
   de múltiplas linhas.  
*/
```

# Declarações

*Existem três tipos de declarações em JavaScript:*

## `var`

Declara uma variável, opcionalmente, inicializando-a com um valor.

## `let`

Declara uma variável local de escopo do bloco, opcionalmente, inicializando-a com um valor.

## `const`

Declara uma constante apenas de leitura.

# Variáveis

Usa-se variáveis como nomes simbólicos para os valores em sua aplicação. O nome das variáveis, chamados de identificadores, obedece determinadas regras.

Um identificador JavaScript deve começar com uma letra, underline (\_), ou cifrão (\$); os caracteres subsequentes podem também ser números (0-9). Devido JavaScript ser case-sensitive, letras incluem caracteres de "A" a "Z" (maiúsculos) e caracteres de "a" a "z" (minúsculos).

Nomes válidos: *Idade, temp99, e \_nome*



# Declarando variáveis

Você pode declarar uma variável de três formas:

- Com a palavra chave **var**. Por exemplo, `var x = 42`. Esta sintaxe pode ser usada para declarar tanto variáveis locais como variáveis globais.
- Por simples adição de valor. Por exemplo, `x = 42`. Isso declara uma variável global. Você não deve usar essa variante.
- Com a palavra chave **let**. Por exemplo, `let y = 13`. Essa sintaxe pode ser usada para declarar uma variável local de escopo de bloco.

# Variáveis

- Uma variável declarada usando a declaração **var** ou **let** sem especificar o valor inicial tem o valor **undefined**.
- Uma tentativa de acessar uma variável não declarada resultará no lançamento de uma exceção **ReferenceError**:

```
let a;  
console.log("O valor de a é " + a); // saída "O valor de a é undefined"  
console.log("O valor de b é " + b); // executa um erro de referência (ReferenceError)  
let c = 10;  
console.log("O valor de c é " + c); // saída "O valor de c é 10"
```

# Escopo de variáveis

Quando você declara uma **variável fora de qualquer função**, ela é chamada de **variável global**, porque está disponível para qualquer outro código no documento atual. Quando você declara uma **variável dentro de uma função**, é chamada de **variável local**, pois ela está disponível somente dentro dessa função.

## Quiz 3

Os dois exemplos acima (A e B) executam sem problemas e apresentarão na tela o mesmo resultado?

- Sim
- Não

A)

```
let x = 0;  
function executa(){  
    x = 20;  
}  
executa();  
console.log(x);
```

B)

```
function executa(){  
    let y = 20;  
}  
executa();  
console.log(y);
```

# Escopo de variáveis

Execute:

```
if (true) {  
    var x = 5;  
}  
console.log(x); // 5
```

```
if (true) {  
    let y = 5;  
}  
console.log(y); // ReferenceError: y não está definido
```

# Constantes

Constante é criada por meio da palavra-chave `const` e deve ser inicializada com um valor, após criada é apenas de leitura. A sintaxe de um identificador de uma constante é semelhante ao identificador de uma variável.

```
const DDD = '89';
```

- Uma constante não pode alterar seu valor por meio de uma atribuição ou ao ser declarada novamente enquanto o script é executado.
- As regras de escopo para as constantes são as mesmas para as variáveis let de escopo de bloco.

# Tipos de dados

O mais recente padrão ECMAScript define sete tipos de dados:

- Seis tipos de dados são os chamados primitivos:
  - Boolean. *true e false.*
  - null. *Uma palavra-chave que indica valor nulo.*
  - undefined. *Uma propriedade superior cujo valor é indefinido.*
  - Number. *29 ou 5.75.*
  - String. *"Howdy"*
  - Symbol. *Um tipo de dado cuja as instâncias são únicas e imutáveis.*
- e Object

Objetos e funções são outros elementos fundamentais na linguagem. Você pode pensar em objetos como recipientes para os valores, e funções como métodos que suas aplicações podem executar.



# Conversão de tipos de dados

JavaScript é uma linguagem dinamicamente tipada. Isso significa que você não precisa especificar o tipo de dado de uma variável quando declará-la, e tipos de dados são convertidos automaticamente conforme a necessidade durante a execução do script.

Então, por exemplo, você pode definir uma variável da seguinte forma:

```
let answer = 42;
```

E depois, você pode atribuir uma string para a mesma variável, por exemplo:

```
answer = "Obrigado pelo convite.";
```

# Conversão de tipos de dados

Em expressões envolvendo valores numérico e string com o operador +, JavaScript converte valores numérico para strings. Por exemplo, considere a seguinte declaração:

```
x = "A resposta é " + 42 // "A resposta é 42"
```

```
y = 42 + " é a resposta" // "42 é a resposta"
```

# Convertendo strings para números

No caso de um valor que representa um número está armazenado na memória como uma string, existem métodos para a conversão.

- *parseInt()*
- *parseFloat()*

```
let valorString = "50";
```

```
typeof(valorString);
```

```
let valorInteiro = parseInt(valorString);
```

```
typeof(valorInteiro);
```

# Literais

Você usa literais para representar valores em JavaScript. Estes são valores fixados, não variáveis, que você literalmente insere em seu script.

O tipo Boolean tem dois valores literal: *true* e *false*.

Uma string literal são zero ou mais caracteres dispostos em aspas duplas (") ou aspas simples ('). Uma sequência de caracteres deve ser delimitada por aspas do mesmo tipo; A seguir um exemplo de strings literais.

```
"foo"
```

```
'bar'
```

```
"1234"
```

# Array literal

Um array literal é uma lista de zero ou mais expressões, onde cada uma delas representam um elemento do array, inseridas entre colchetes (`[]`). Quando você cria um array usando um array literal, ele é inicializado com os valores especificados como seus elementos, e seu comprimento é definido com o número de elementos especificados.

O exemplo a seguir cria um array `coffees` com três elementos e um comprimento de três:

```
let coffees = ["French Roast", "Colombian", "Kona"];
```

# Atividade 2



1. Explique e dê exemplos:
  - a. Dos tipos de declarações (variáveis, constantes e funções) em Javascript.
  - b. Dos tipos de dados em Javascript.
2. Declare um array com 5 elementos do tipo inteiro.

# Controle de Fluxo e Manipulação de Erro

# Declaração em bloco

Uma declaração em bloco é utilizada para agrupar declarações. O bloco é delimitado por um par de chaves:

```
{  
    declaracao_1;  
    declaracao_2;  
    .  
    .  
    declaracao_n;  
}
```

Declarações em bloco são utilizadas geralmente com declarações de fluxo de controle: (ex. **if**, **for**, **while**).



# Declarações condicionais

Uma declaração condicional é um conjunto de comandos que são executados caso uma condição especificada seja verdadeira. O JavaScript suporta duas declarações condicionais: `if...else` e `switch`.

# Declaração if...else

Você pode também combinar declarações utilizando else if para obter várias condições testadas em sequência, como o seguinte:

```
if (condicao_1) {  
    declaracao_1;  
} else if (condicao_2) {  
    declaracao_2;  
} else {  
    declaracao_2;  
}
```

# Declaração if...else

Use a declaração if para executar alguma declaração caso a condição lógica for verdadeira. Use a cláusula opcional else para executar alguma declaração caso a condição lógica for falsa.

Uma declaração if é declarada da seguinte maneira:

```
if (condicao) {  
    declaracao_1;  
} else {  
    declaracao_2;  
}
```

# Quiz

O que será exibido ao executar o seguinte código?

```
let media = 7;  
if (media >= 7) {  
  console.log('Aprovado')  
} else {  
  console.log('Reprovado')  
}
```

# Atividade 3

Escreva um programa que recebe 3 notas digitadas em três campos de input HTML, calcula a média simples e exibe em um campo de texto HTML: **"Aprovado"** se a média for igual ou maior que 7; **"Recuperação"** se a média for maior ou igual a 4 e menor que 7; e **"Reprovado"** se a média for menor que 4.

# Valores avaliados como falsos

- `false`
- `Undefined`
- `Null`
- `0`
- `NaN`
- `string vazia ("" )`

# Vamos Ver Exemplos

[https://www.w3schools.com/js/js\\_examples.asp](https://www.w3schools.com/js/js_examples.asp)

<https://gamejs--jesielviana.repl.co/>

# Código Fonte do Jogo

[https://www.w3schools.com/graphics/game\\_intro.asp](https://www.w3schools.com/graphics/game_intro.asp)



# Vamos praticar, vamos usar um editor online.



# Bibliografia

- [https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/](https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide)
- <https://pt.wikipedia.org/wiki/JavaScript>
- <https://insights.stackoverflow.com/survey/2018/#technology>

# Obrigado!

Thanks!

jesiel@ifpi.edu.br



@jesielviana



<https://github.com/jesielviana>



na