



**Prof. Dr. Leandro Luque**  
Faculdade de Tecnologia de Mogi das Cruzes

# JAVASCRIPT

---

Visão geral (cont.)

## HISTÓRICO E VERSÕES

Do Mocha aos dias atuais

01

## O CENÁRIO ATUAL

Problemas resolvidos e novos  
problemas

02

## PRINCIPAIS CARACTERÍSTICA S

Paradigmas, tipagem, virtudes e  
fraquezas

03

# CONTEÚD O

04

## VARIÁVEIS E TIPOS DE DADOS

var, let, const e hoisting

05

## ESCOPO

Global e local

06

## ESTRUTURAS DE CONTROLE

Diferentes estruturas  
condicionais e de repetição

## 6. MAIS DO MESMO, MAS...

- if... else... else if...
- switch... case...
- ? :
  - (condicao) ? <se true> : <se false>;
  - let teste = "" ? "verdadeiro" : "falso";  
console.log(teste); // falso.
- mas ... (o que é falso?) - falsy

false	The keyword <a href="#">false</a>
0	The number <a href="#">zero</a>
-0	The number negative <a href="#">zero</a>
0n	<a href="#">BigInt</a> , when used as a boolean, follows the same rule as a Number. 0n is <i>falsy</i> .
""	Empty <a href="#">string</a> value
null	<a href="#">null</a> - the absence of any value
undefined	<a href="#">undefined</a> - the primitive value
NaN	<a href="#">NaN</a> - not a number

FONTES:  
<https://developer.mozilla.org/en-US/docs/Glossary/Falsy>

# ATRIBUINDO VALORES PADRÕES E EXECUTANDO CÓDIGO CONDICIONALMENTE

- `const var2 = outraVar ?? 0; // coalescência nula`  
`const var1 = outraVar || 0;`

- `const var3 = true;`
- `if(var3) {`
- `console.log('Var3 é true');`
- `}`
- `var3 && console.log('Var3 é true');`

# ESTRUTURAS DE REPETIÇÃO

- for;
  - do... while
  - while... do
- 
- for... in (chaves de objetos)
  - for... of (iterables)
  - .forEach



# ARRAYS

- São objetos em Javascript;
- Declaração:

```
var alunos = ["Maria", "Joao", "Jose"]; // empty-slot na posição 2 (undefined)
console.log(alunos);
var alunos1 = new Array("Maria", "Joao", "Jose"); // ,, não funciona
console.log(alunos1);
```

- Principais propriedades, métodos e características:
  - sort;
  - push & pop;
  - unshift & shift (remove o primeiro elemento);
  - splice(inicio, quantosARemover, aAdicionar1, aAdicionar2, ...);
  - reverse;
  - slice(inicio, fim) (cria nova array com pedaço da primeira); // ver -1, -2
  - ... (spread - ver Math.min e Math.max)
  - filter
  - map
  - reduce



# USAR PONTO E VÍRGULA?

- Efeitos indesejados de não usar:

```
function getObjeto() {  
  return  
  {  
    atributo: "valor"  
  }  
}
```

Ele interpretou  
assim

```
function getObjeto() {  
  return  
  {  
    atributo: "valor"  
  }  
}
```

```
const a = 1  
const b = 2  
const c = a + b  
(a + b).toString(); // TypeError: b is not a function
```

Ele interpretou  
assim

```
const a = 1  
const b = 2  
const c = a + b(a + b).toString(); // TypeError: b is not a function
```

