

**INSTITUTO FEDERAL**

Santa Catarina  
Câmpus Lages

# **Desenvolvimento para dispositivos móveis**

## **JAVA SCRIPT**

**ORLANDO OLIVEIRA DOS SANTOS**

# **JAVA SCRIPT(JS)**

Linguagem de programação criada através de uma padronização internacional, por um comitê intitulado TC39;

Órgão responsável: ECMAScript;

Várias versões existentes, e atualmente, está em vigência a ES2025.

# **JAVA SCRIPT(JS)**

Facilita o desenvolvimento de websites por estar integrado ao ECMASCRIPT;

Todo desenvolvimento web é compatível com os navegadores existentes.

# JAVA SCRIPT(JS)

Imagine que:

- A internet é uma reunião internacional.
- Os sites são pessoas dando instruções.
- Os navegadores são tradutores.

# JAVA SCRIPT(JS)

Agora imagine que:

- Um tradutor fala português.
- Outro fala espanhol.
- Outro fala francês.
- Mas não existe um idioma padrão combinado.

# JAVA SCRIPT(JS)

Você diz:

"Abra a porta."

O tradutor A entende como:

👉 "Abrir a janela."

O tradutor B entende como:

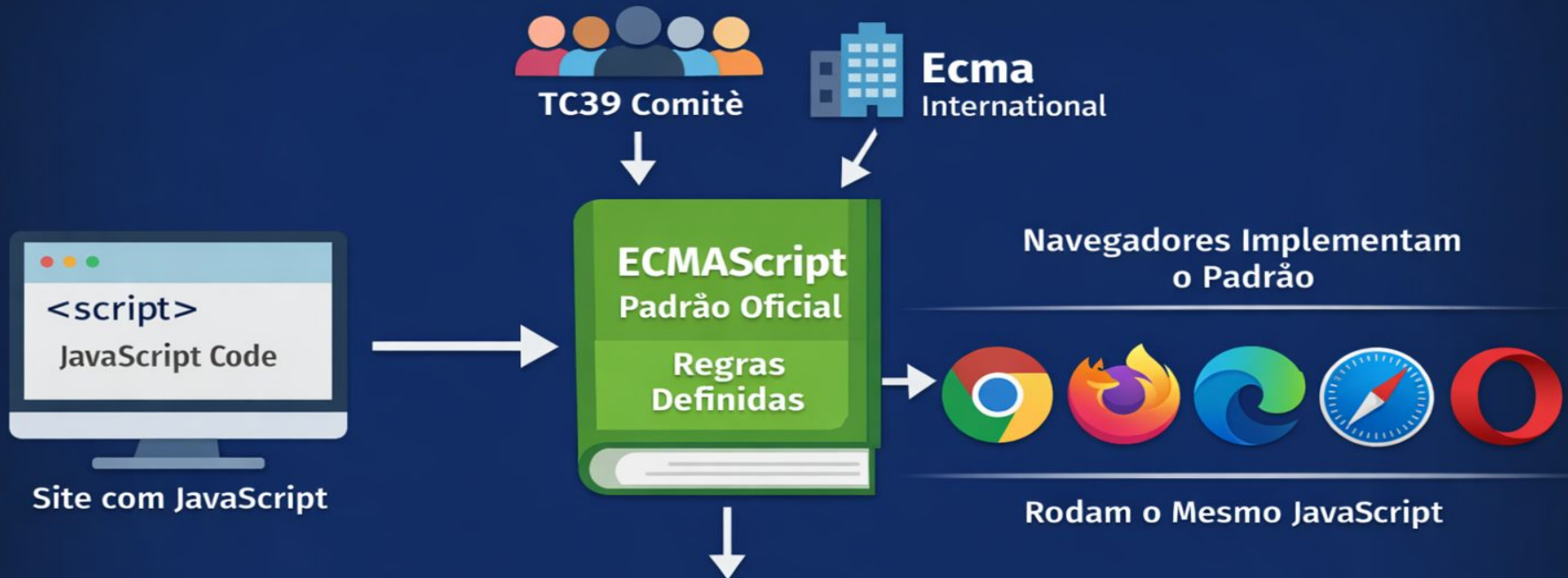
👉 "Fechar a porta."

O tradutor C nem entende o comando.

Confusão total.

# JAVA SCRIPT(JS)

**JavaScript Compatível com Todos os Navegadores**



# JAVA SCRIPT(JS)

**Link ECMAScript:**

**<https://tc39.es/ecma262/#sec-intr>**  
**o**



# **JAVA SCRIPT(JS)**

JavaScript é uma linguagem de programação.

- Ele é usado para:
  - Criar interatividade em sites;
  - Manipular elementos da página;
  - Criar aplicações web;
  - Criar aplicativos móveis (React Native).

# **JAVA SCRIPT(JS)**

- JavaScript roda no navegador (Chrome, Edge, Firefox);
- Rodar no servidor com Node.js.

# JAVA SCRIPT(JS)

JavaScript permite:

- Executar ações quando algo acontece (eventos);
- Criar funções;
- Trabalhar com variáveis;
- Tomar decisões (if/else);
- Repetir ações (loops);

Ele é uma linguagem interpretada=> o navegador lê e executa o código na hora.

# **JAVA SCRIPT - VARIÁVEIS**

Há em formal utilidade 3 variáveis, pelo Javascript:

**var**

**let**

**const**

# JAVA SCRIPT - VARIÁVEIS - VAR

Há em formal utilidade 3 variáveis, pelo Javascript:

## **var:**

- Características:
  - Escopo global ou de função;
  - Pode ser redeclarada;
  - Pode ser reatribuída.

Problema:

Pode causar confusão e erros em projetos grandes.

# JAVA SCRIPT - VARIÁVEIS - VAR

**EXEMPLO:**

```
var x = 10;
```

```
var x = 20;
```

# COMO TESTAR?

1º - Pode-se abrir o navegador, apertar F12 e testar a equação no console, a digitando;

2º - Gerar um arquivo html para poder realizar os testes.  
Teste pressionado o F12.

Nessa segunda opção, inclua os fatores do teste dentro da tag `<script>`, no html.

# COMO TESTAR?

```
<body>

  <script>

    function saudacao() {
      console.log("Olá, mundo!");
    }

    saudacao()

  </script>

</body>
```



# JAVA SCRIPT - VARIÁVEIS - LET

Há em formal utilidade 3 variáveis, pelo Javascript:

**let:**

- Características:
  - Escopo de bloco { };
  - Não pode redeclarar no mesmo escopo;
  - Pode reatribuir valor.

Atualmente, se usa let quando o valor pode mudar.

# JAVA SCRIPT - VARIÁVEIS - LET

## EXEMPLO:

```
let y = 5;
```

```
y = 8;
```

Mas isso NÃO é permitido:

```
let y = 5;
```

```
let y = 10;
```

# JAVASCRIPT - VARIÁVEIS - CONST

Há em formal utilidade 3 variáveis, pelo Javascript:

## **const:**

- Características:
  - Escopo de bloco { };
  - Não pode redeclarar;
  - Não pode reatribuir.

**Por boas práticas se utiliza const quando não se muda uma variável, e let se for atualizá-la.**

# JAVASCRIPT - VARIÁVEIS - CONST

**EXEMPLO:**

```
const z = 3;
```

```
z = 7;
```

# FUNÇÕES

Função é um bloco de código que executa uma tarefa.

Exemplo:

```
function saudacao() {  
  console.log("Olá, mundo!");  
}
```

Aqui está acontecendo:

console.log → Mostra a informação no console do navegador.

function → palavra reservada

saudacao → nome da função

{ } → bloco de código

# FUNÇÕES

Para executar:

```
saudacao();
```

Isso chama a função.

# FUNÇÕES COM PARÂMETROS

Podemos enviar valores para dentro da função.

Exemplo:

```
function soma(a, b) {  
  return a + b;  
}
```

O que acontece aqui:

# FUNÇÕES COM PARÂMETROS

a e b são parâmetros

return devolve o resultado

Executando:

```
let resultado = soma(5, 3);
```

Aqui:

5 substitui "a"

3 substitui "b"

Resultado será 8.



# FUNÇÕES COM PARÂMETROS

```
let idade = prompt("Digite sua idade:");  
  if (idade >= 18) {  
    alert("Maior de idade");  
  } else {  
    alert("Menor de idade");  
  }
```

Prompt: Exibe uma caixa de texto, e interpreta somente texto

Alert: exibe uma janela pop up

# FUNÇÕES COM PARÂMETROS

```
let idade = prompt("Digite sua idade:");  
  if (idade >= 18) {  
    alert("Maior de idade");  
  } else {  
    alert("Menor de idade");  
  }
```

OBS: Caso for preencher com números, configure dessa forma?

```
let idade = +prompt("Digite sua idade: ")
```

# **EXERCÍCIOS**

PDF sugerido pelo professor.