

AULA 8 - CONDICIONAIS

- > Condicionais são **estruturas** de código usadas para fazer escolhas baseadas em alguns critérios
- > Em outras palavras, elas permitem realizar uma determinada **ação** dependendo de uma **condição**
- > Exemplo: baseado na **condição** de estar chovendo eu vou realizar a **ação** de recolher a roupa
- > Até agora vimos que o javascript executa linha por linha de código, de forma **síncrona e sequencial**.
- > **As condicionais são estruturas de código javascript que nos dão o poder de decidir se a próxima linha de código deve ser executada ou não.**

Definição de estrutura

- > Modo como alguma coisa é **construída, organizada ou está disposta**: a estrutura de uma empresa.
- > Aquilo que serve de **base** para algo; **armação ou esqueleto**: a estrutura de um edifício; a estrutura de uma linguagem de programação.

Árvores de Condicionais

- > Uma maneira esquemática de representar condicionais é utilizando **fluxogramas**.

Nos fluxogramas:

- > Um **losango** representa uma condição para a tomada de decisão
- > Um **retângulo** representa a ação
- > Você pode encadear várias condições antes de chegar a uma resposta (por isso chamamos de árvore!)

Dica!!

- > Nós não somos computadores e temos um processo de pensamento diferente
- > O uso de fluxogramas pode te ajudar muito a analisar as possibilidades para fazer decisões!
- > Se ficar confuso com algum exercício que envolve condicionais, tente fazer essa árvore e veja se ajuda :)

Condicionais em JS

Bloco if / else

> if / else é a sintaxe (estrutura) de programação utilizada para condicionais

> Se a condição for verdadeira o código dentro do if é executado

> **if:** Todo código da ação vai entre chaves { } **condição simples**

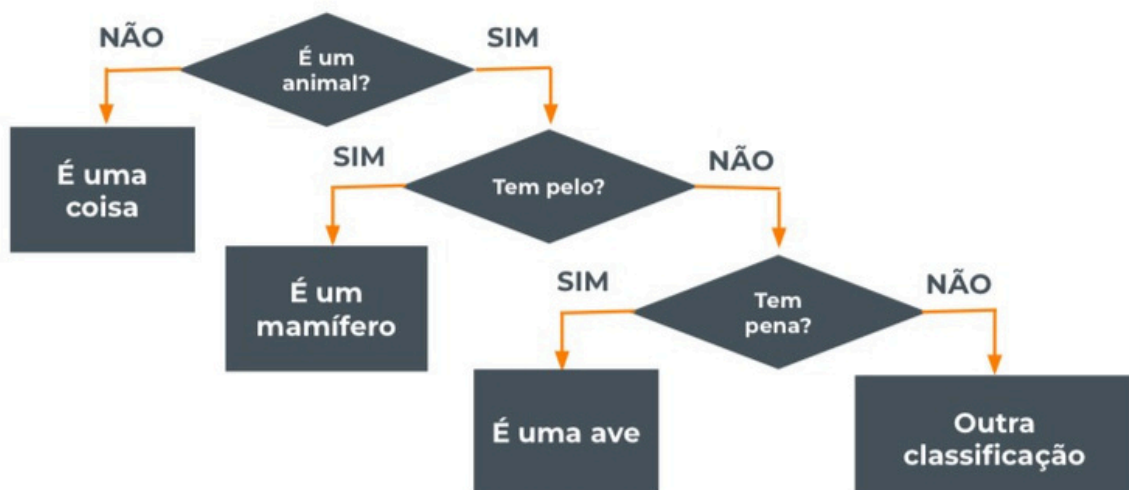
```
1 let condicao1 = true
2
3 if (condicao1){
4     // Como o valor da condição é true,
5     // o código desse bloco é executado
6     console.log('Entrei no if 1!')
7 }
```

```
1 let condicao2 = false
2
3 if (condicao2){
4     // Como o valor da condição é false,
5     // o código desse bloco NÃO é executado
6     console.log('Entrei no if 2!')
7 }
```

> **if + else:** Todo código da ação vai entre chaves { }

```
1 let condicao = false
2
3 if (condicao){
4     console.log('Entrei no if!')
5 } else {
6     // Como o valor da condição é false,
7     // o código do bloco else será executado
8     console.log('Entrei no else!')
9 }
```

Encadeamento de Condições

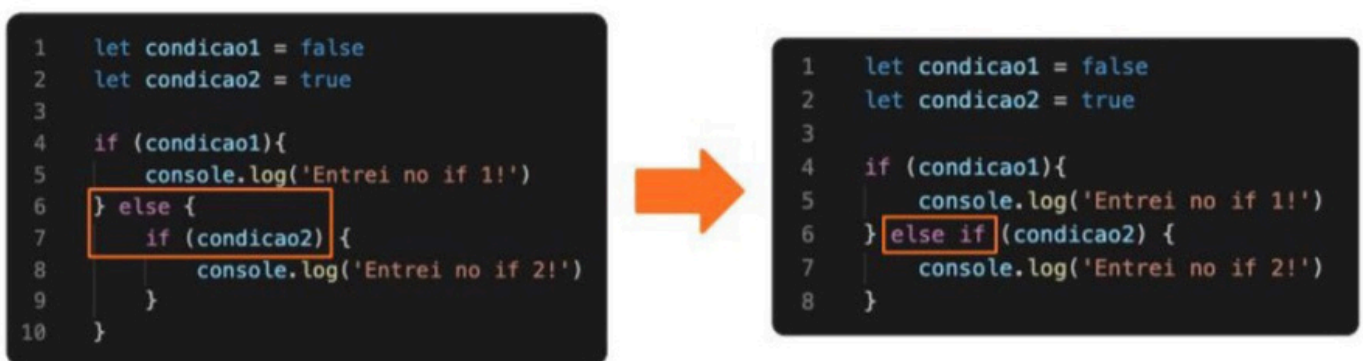


> **if + else + if**: Todo código da ação vai entre chaves { }

```
1  let condicao1 = false
2  let condicao2 = true
3
4  if (condicao1){
5      console.log('Entrei no if 1!')
6  } else {
7      // Como o valor da condicao1 é false,
8      // o código do else será executado
9      if (condicao2) {
10         // Como o valor da condicao2 é true,
11         // executaremos esse código!
12         console.log('Entrei no if 2!')
13     }
14 }
```

if + else if

> Tem um jeito mais simples de escrever!



The diagram illustrates a code transformation. On the left, a code block shows an `if` statement followed by an `else` block containing another `if` statement. The `else` and the inner `if` are highlighted with an orange box. A large orange arrow points to the right, where the transformed code is shown. In the transformed code, the `else if` construct is used, and the `else if` and its inner block are highlighted with an orange box.

```
1  let condicao1 = false
2  let condicao2 = true
3
4  if (condicao1){
5      console.log('Entrei no if 1!')
6  } else {
7      if (condicao2) {
8          console.log('Entrei no if 2!')
9      }
10 }
```

```
1  let condicao1 = false
2  let condicao2 = true
3
4  if (condicao1){
5      console.log('Entrei no if 1!')
6  } else if (condicao2) {
7      console.log('Entrei no if 2!')
8  }
```

● if + else if + else

```

1  let condicao1 = false
2  let condicao2 = false
3
4  if (condicao1){
5      console.log('Entrei no if 1!')
6  } else if (condicao2) {
7      console.log('Entrei no if 2!')
8  } else {
9      // Como tanto o valor da condicao1 e
10     // da condicao2 são false, executa
11     // os comandos do bloco else
12     console.log('Entrei no else!')
13 }

```

Switch Case

```

1  let paisDeOrigem
2  if (paisDeOrigem === 'Brasil'){
3      console.log('brasileiro')
4  } else if (paisDeOrigem === 'EUA'){
5      console.log('norte americano')
6  } else if (paisDeOrigem === 'Inglaterra'){
7      console.log('inglês')
8  } else if (paisDeOrigem === 'França'){
9      console.log('francês')
10 } else if (paisDeOrigem === 'Itália'){
11     console.log('italiano')
12 } else if (paisDeOrigem === 'Canadá'){
13     console.log('canadense')
14 } else {
15     console.log('nacionalidade não encontrada')
16 }

```

```

1  let paisDeOrigem
2  switch (paisDeOrigem){
3      case 'Brasil':
4          console.log('brasileiro')
5          break
6      case 'EUA':
7          console.log('norte americano')
8          break
9      case 'Inglaterra':
10         console.log('inglês')
11         break
12     default:
13         console.log('nacionalidade não encontrada')
14         break
15 }

```

>Os cases indicam as condições.

>Se o valor da variável for igual ao que está no case, o código de dentro será executado.

>Conseguimos colocar um caso padrão chamado default

>O código dentro dele será executado se o valor da variável **não bater** com as opções dos cases.

>**break** é a palavra que faz com que a execução do código saia do bloco em questão. Caso não exista o break, o código continuará executando.

Condicionais e Operadores Lógicos

> Como os operadores lógicos retornam **booleanos**, conseguimos usá-los diretamente na

condição dos blocos **if/else**.

```
1  let condicao1 = true
2  let condicao2 = false
3
4  if (condicao1 && condicao2){
5      // Entra aqui se ambas forem true
6  }
7
8  if (!condicao1){
9      // Entra aqui se condicao1 for false
10 }
```