

# Condicionais

Labenu\_



# O que vamos ver hoje?

- Revisão Comparadores
- O que são condicionais
- Sintaxes JS
  - if/else
  - switch case

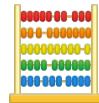


# Comparadores

Labenu\_



# Relembrando Comparadores



- **Comparadores** são operadores que permitem **comparar** duas variáveis entre si
- O resultado destes operadores é sempre um **booleano**
- Quando a comparação for **correta**, o resultado é **true**. Caso **contrário**, **false**



# Relembrando Comparadores



- **Comparadores**

- `===` : valor e tipo iguais
- `!==` : valor ou tipo diferentes
- `>` : maior que
- `>=` : maior ou igual que
- `<` : menor que
- `<=` : menor ou igual que



# Condicionais no dia-a-dia

Labenu\_



# O que são condicionais? 🤔

- Vocês já repararam quantas vezes ao dia temos que tomar uma decisão **dependendo de outros fatores**?
- Nossos cérebros são incríveis e fazem isso de forma automática!
- Vamos tentar pegar alguns exemplos e quebrar o processo de raciocínio em pedacinhos:



# O que são condicionais?

Eu abro as cortinas para ver o dia lá fora...

- **Se** está chovendo:
  - Saio correndo para recolher as roupas do varal
- **Senão:**
  - Fico de boa aproveitando o dia bonito





# O que são condicionais?



Testando um exercício da lista...

- **Se** passa no teste:
  - Fico feliz e vou pro próximo
- **Senão:**
  - Choro e volto para achar o bug



# O que são condicionais? 🍔

Estou ficando com fome e resolvo abrir a geladeira...

- **Se** tem alguma coisa além de gelo e ketchup:
  - Fico feliz por ter sido responsável
  - Preparo um almoço gostoso
- **Senão:**
  - Peço um hambúrguer por aplicativos



# O que são condicionais? ?

- Condicionais são estruturas 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



# Árvores de Condicionais

Labenu\_



# Árvores de Condicionais 🌲

- Uma maneira esquemática de representar condicionais é utilizando fluxogramas:



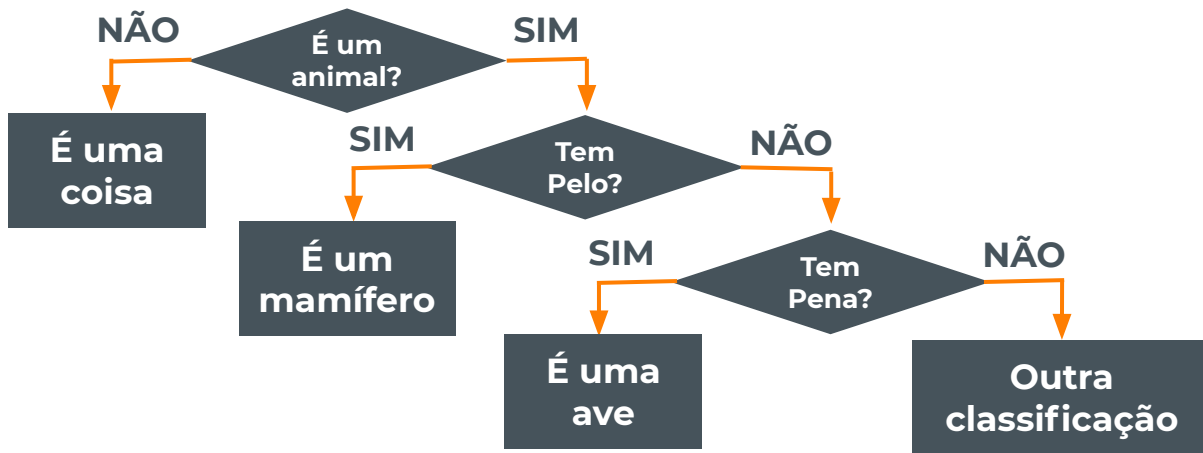
# Árvores de Condicionais 🌲

- Nos fluxogramas:
  - Um retângulo representa um **passo**
  - Um losango representa a **tomada de uma decisão**



# Árvores de Condicionais

- Você pode **encadear** várias condições antes de chegar a uma resposta (por isso chamamos de árvore!)



# Árvores de Condicionais

- Depois de percorrer várias condições, você chega à uma conclusão
- E, a partir da conclusão, você pode realizar alguma ação





# 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 :)



# Pausa para relaxar 🤔

5 min



- Variáveis, usando os comparadores (==, !=, >, <)
- Condicional é uma estrutura que permite escolher uma ação dependendo de uma condição
- Árvores de condicionais são uma maneira de se representar esquematicamente os condicionais



# Condicionais em JS

Labenu\_



# Bloco if /else 🤔

- **if/else** é a **sintaxe** de programação utilizada para **condicionais**
- Se a **condição for verdadeira** o código **dentro do if** é executado



# Bloco if / else 🤔

- **if:** Todo código da ação vai entre chaves `{ }`

```
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  }
```

Vamos ver na prática! 🔬





# Exercício 1

Crie uma função que:

- Recebe 2 números (chamaremos de **num1** e **num2**)
- **Compara** esses números entre si:
  - Se os números **forem iguais**, retorna uma mensagem de sucesso

Depois, chame a função com números que foram inseridos pelo usuário através do prompt



# Bloco if / else 🤔

- **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  }
```

Vamos ver na prática! 🔬





## Exercício 2

Crie uma função que:

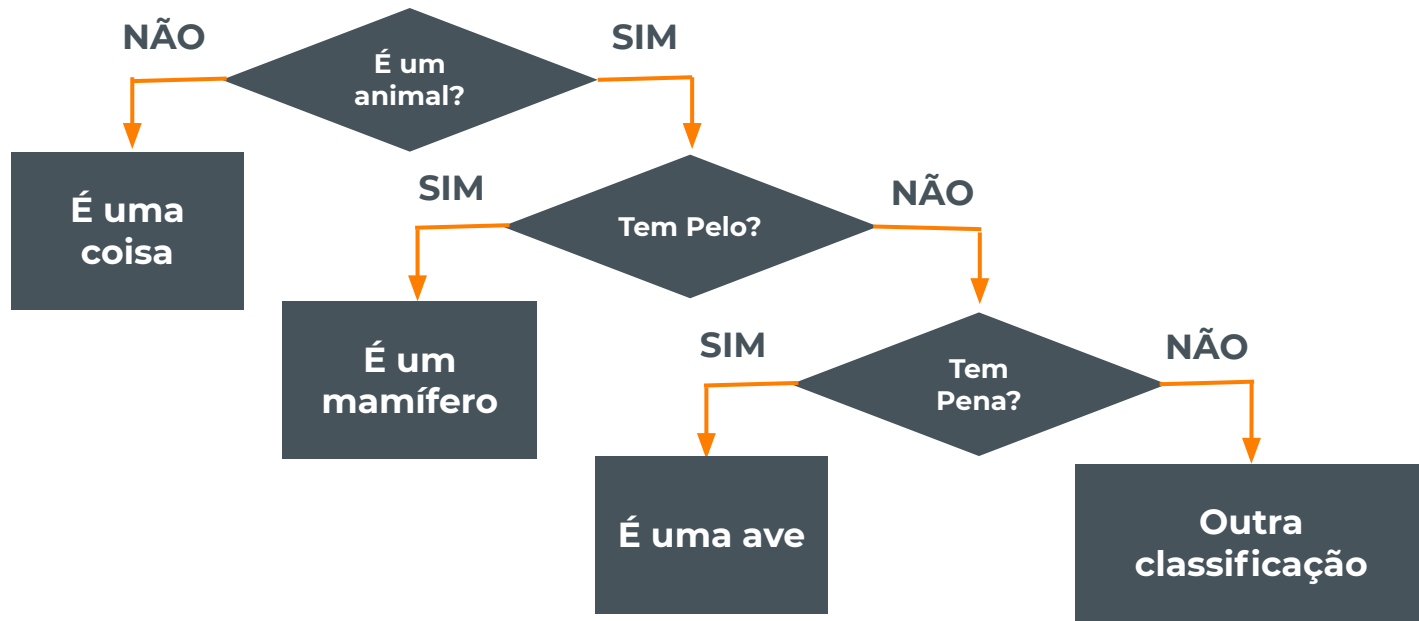
- Recebe 2 números (chamaremos de **num1** e **num2**)
- **Compara** esses números entre si:
  - Retorna mensagens dizendo se os números são **iguais** ou **diferentes**

Depois, chame a função com números que foram inseridos pelo usuário através do prompt





# Encadeamento de Condições



# Bloco if /else 🤔

- **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 }
```



# Bloco if /else 🤔

- Tem um jeito mais simples de escrever!

```
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  }
```

Vamos ver na prática! 🔬



# Bloco if /else 🤔

- 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 }
```

Vamos ver na prática! 🔬





## Exercício 3

Crie uma função que:

- Recebe 2 números (chamaremos de **num1** e **num2**)
- **Compara** esses números entre si:
  - Retorna mensagens dizendo se o primeiro número é **{maior/menor/igual}** ao segundo número

Depois, chame a função com números que foram inseridos pelo usuário através do prompt

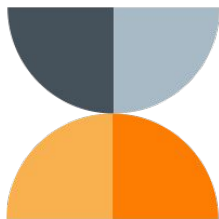


# Pausa para relaxar 🤪

10 min

Vimos como a gente pode pedir para o computador realizar condições usando o JavaScript:

- Condicional simples (**if**)
- Condicional composta (**if/else**)
- Condicional aninhada (**if + else if + else**)



# Switch Case

Labenu\_



# Switch case 🤔

- Vamos dar uma olhada no código abaixo:

```
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 }
```





# Switch case 🤔

- Há uma maneira de **simplificar**: usando **switch case**

```
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 }
```

A palavra **switch** começa o bloco.

Entre os parênteses () vai a variável a ser comparada



# Switch case 🤔

- Há uma maneira de **simplificar**: usando **switch case**

```
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 a variável for igual ao que está no case, o código de dentro será executado



# Switch case 🤔

- Há uma maneira de **simplificar**: usando **switch case**

```
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 a variável for igual ao que está no case, o código de dentro será executado



# Switch case 🤔

- Há uma maneira de **simplificar**: usando **switch case**

```
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 }
```

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**



# Switch case 🤔

- Há uma maneira de **simplificar**: usando **switch case**

```
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 }
```

**break** é a palavra que faz com que a execução do código saia do bloco em questão

Vamos ver na prática! 🔬

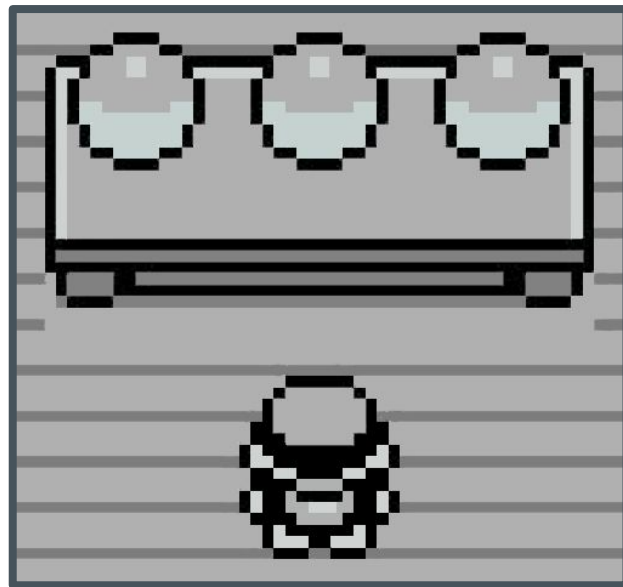




## Exercício 4

Escreva um programa que receba o Pokémon inicial escolhido pela pessoa e retorna o seu tipo:

- Bulbasaur (Planta e Veneno)
- Charmander (Fogo)
- Squirtle (Água)



# Condicionais e Operadores Lógicos

Labenu\_



# Operadores Lógicos e Condicionais 🧠

- 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 }
```







## Exercício 5

Uma pessoa pode estudar em uma faculdade se:

- Tiver concluído o ensino médio
- Tiver 18 anos ou mais
- Não estiver cursando outra faculdade

Escreva uma função que receba estes parâmetros e devolva se a pessoa pode ou não estudar nesta faculdade. Receba os dados do usuário pelo prompt e execute a função



# Resumo

Labenu\_



# Resumo

- **Operadores de comparação** são usados para se compararem valores de mais de uma variável. Eles sempre **retornam** um valor **booleano**
  - **===**: valor e tipo iguais
  - **!==**: valor ou tipo diferentes
  - **>**: maior que
  - **>=**: maior ou igual que
  - **<**: menor que
  - **<=**: menor ou igual que



# Resumo

- **Condicionais** são estruturas que simbolizam decisões tomadas dependendo de certas **condições**
- **Árvore condicional** é uma estrutura **esquemática** que pode ser usada para **facilitar** nossa análise e construção de condicionais



# Resumo

- **if/else** são blocos que permitem fazer uma **condicional**. Eles recebem uma condição e o código
  - do **if** é executado se a **condição** for **true**
  - do **else** é executado se a **condição** for **false**
- **switch case** são blocos que permitem simplificar **if/else** apenas no caso de **comparador de igualdade**
  - **lembram-se de escrever o break**



# Dúvidas? 🧐

Labenu\_





Obrigado(a)!