

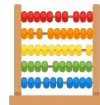
Condicionalis

O que vamos ver hoje?

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

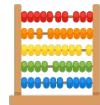
Comparadores

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

Então... 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

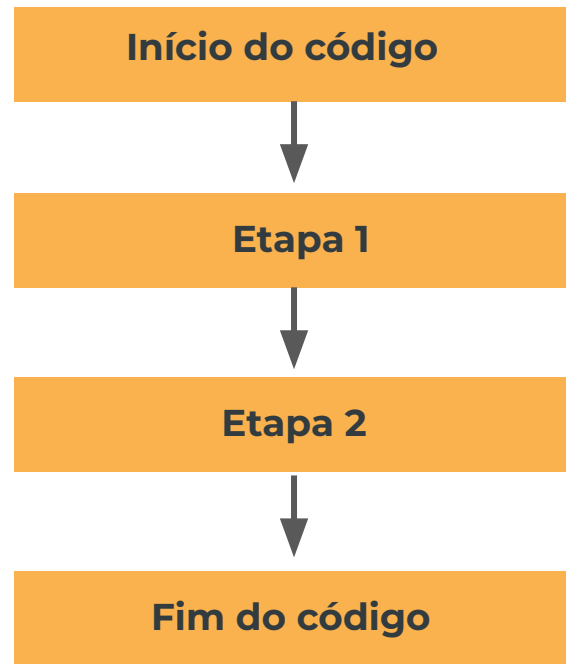
O que são condicionais?

- Condicionais são **estruturas 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

Então... 🤔

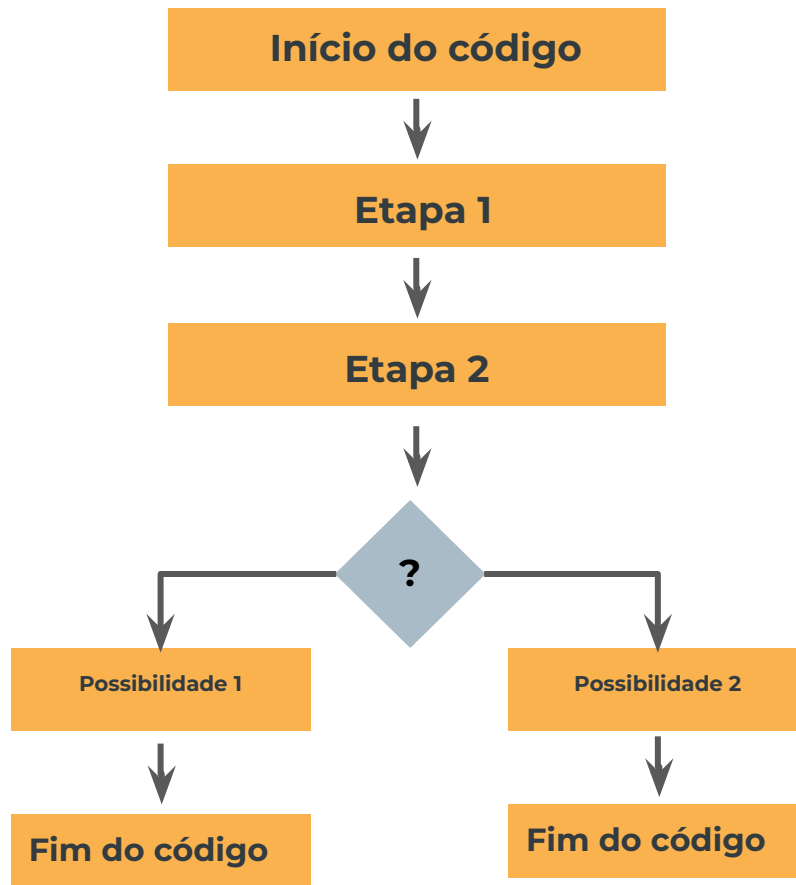
- Até agora vimos que o javascript executa linha por linha de código, de forma síncrona e **sequencial**.

Como uma escada, que descemos degrau por degrau, sem poder pular nenhum



Então... 🤔

- 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

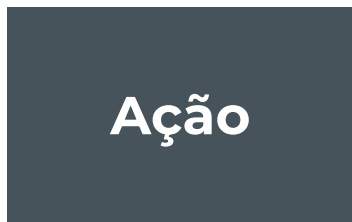
Árvores de Condicionais 🌲

- Uma maneira esquemática de representar condicionais é utilizando fluxogramas:



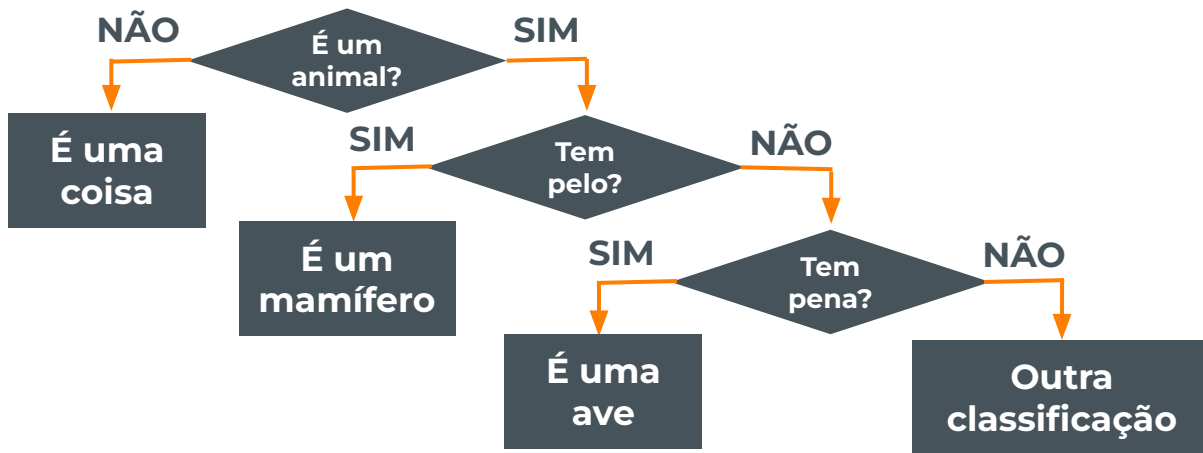
Árvores de Condicionais 🌲

- Nos fluxogramas:
 - Um losango representa uma **condição** para a **tomada de decisão**
 - Um retângulo representa a **ação**



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 :)

Fixação

- Comparadores são operadores usados para **comparar variáveis** (==, !=, >, <)
- **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

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

Primeiro, falaremos do bloco **if** 🤔

- **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  }
```

Vamos ver na prática! 🔬

Exercício 1

Crie uma função que:

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

Agora, adicionamos o **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  }
```

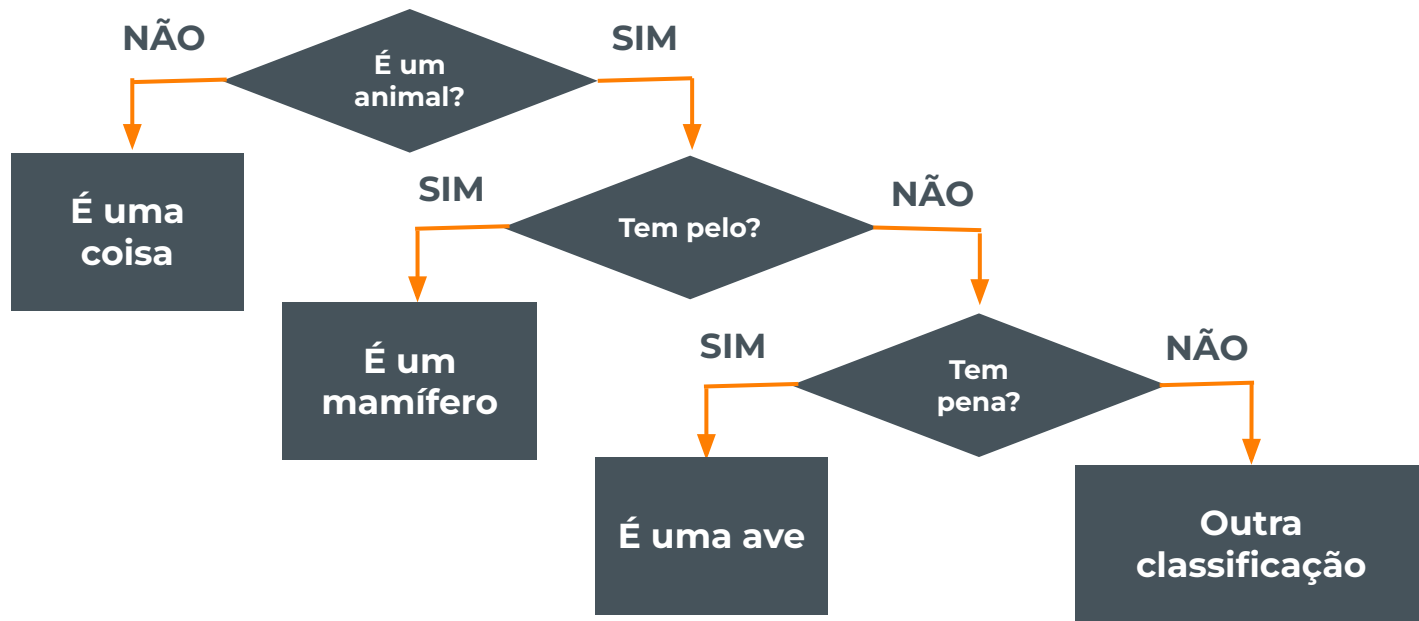
Vamos ver na prática! 🔬

Exercício 2

Escreva uma função que:

- Receba 2 variáveis (chamaremos de **num3** e **num4**)
- **Compare** esses números entre si:
 - Imprimindo no console mensagens dizendo se os números são **iguais** ou **diferentes**

Encadeamento de Condições



Agora, **if + else + if** 🤔

- **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 }
```

[Exemplo](#)

Ufa, dá pra simplificar! **if + else if**

- 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!



if + else if + else? Socorro

- 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! 

[Exemplo](#)

Exercício 3

Escreva uma função que:

- Receba 2 variáveis (chamaremos de **num5** e **num6**)
- **Compare** esses números entre si:
 - Imprime mensagens dizendo se o primeiro número é **{maior/menor/igual}** ao segundo número

Fixação

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

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 }
```

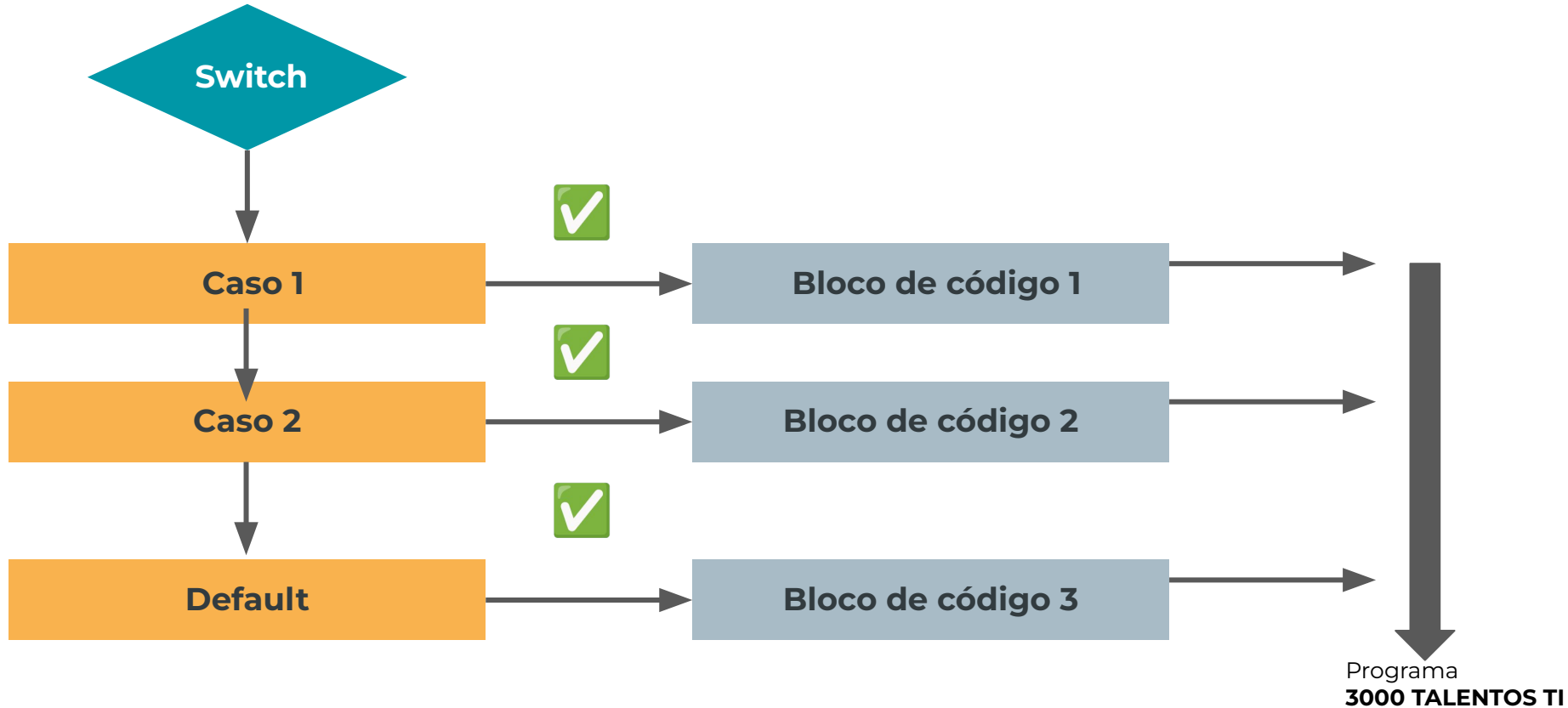
Explicaremos
melhor nos
próximos slides!

Antes, vamos comparar

```
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 }
```

Switch case a estrutura 🤔



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 o valor da 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  }
```

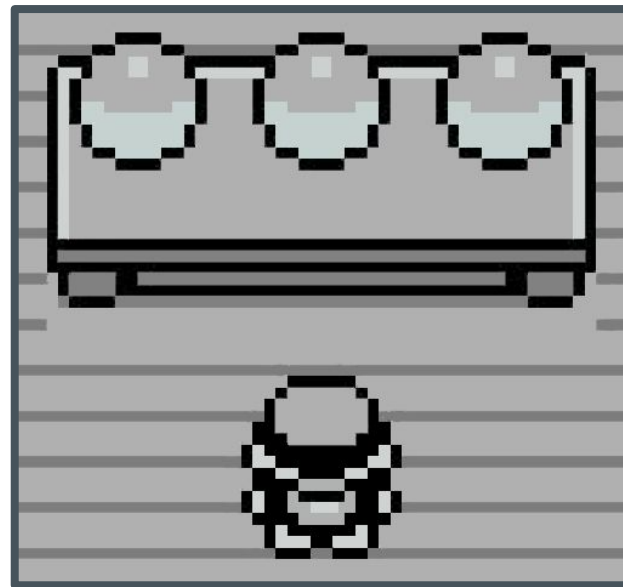
break é a palavra que faz com que a execução do código saia do bloco em questão. Caso não exista o break, o código continuará executando

Vamos ver na prática! 

Exercício 4

Escreva uma função que receba o Pokémon inicial escolhido por você e imprima no console o seu tipo:

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



Fixação

Podemos realizar condições usando no JavaScript o **if + else**. Além disso, podemos usar o **switch** case.

Programa
3000 TALENTOS TI

- Switch case **evita códigos verbosos**
- Só serve para **verificar casos de igualdade**
- Permite executar bloco de código diferente baseado em **cada opção**
- **Break** = Para a execução do código quando a **condição é atendida**

Condicionais e Operadores Lógicos

Operadores lógicos

Relembrando Operadores Lógicos



- **&&** : *AND* - verdadeiro se ambos os operandos forem verdadeiros
- **||** : *OR* lógico: verdadeiro se um ou ambos os operandos forem verdadeiros
- **!** : *NOT* negação - sua função é simplesmente inverter os valores. Ou seja, se o valor de entrada for VERDADEIRO, o resultado será FALSO e se o valor de entrada for FALSO, o resultado será VERDADEIRO

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 estas três variáveis (ensinoMedioConcluido, idade e cursandoOutraFaculdade) com valores booleanos e imprima na tela se esta pessoa pode ou não estudar nesta faculdade.

Resumo

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**
 - **lembrem-se de escrever o break**

Dúvidas?



Programa
3000 TALENTOS TI
Obrigado(a)!