



Enxurrada de Bits

Comandos de decisão e  
Estrutura de repetição

# Operadores

## Operadores matemáticos:

- + soma
- subtração
- / divisão
- \* multiplicação

## Operadores relacionais:

- > maior que
- >= maior ou igual
- < menor que
- <= menor ou igual
- === igual
- !== diferente

## Operadores lógicos:

- ! não
- || ou
- && e

# Comandos de decisão

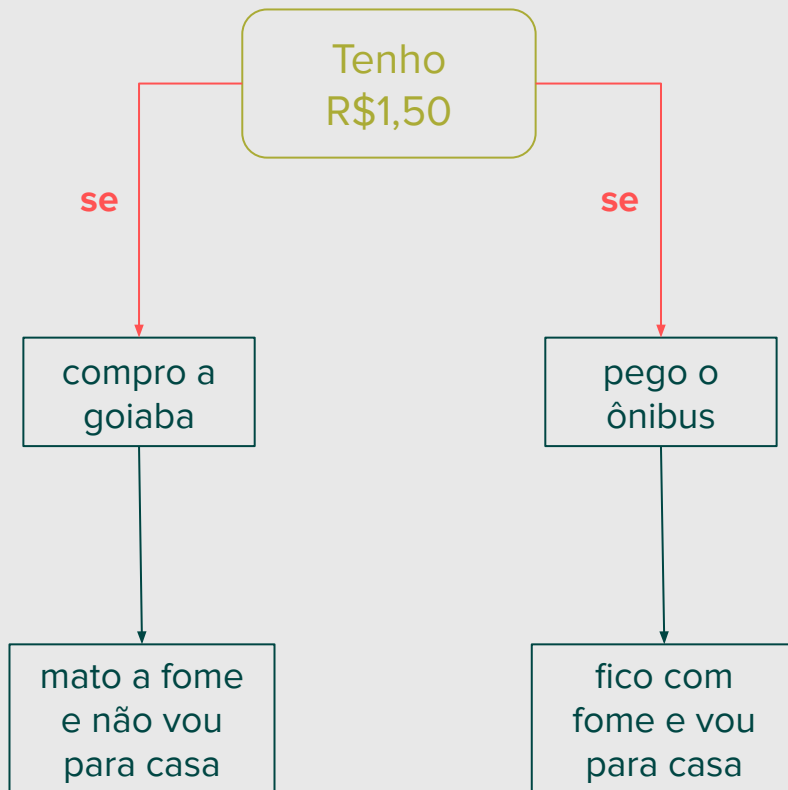
Algoritmos precisam tomar decisões ao longo de sua execução

Para isso existem as estruturas de decisão, que direcionam o caminho que o código irá tomar

# Comandos de decisão



# Comandos de decisão



# Comandos de decisão

Como o código funcionaria:

```
se (compro a goiaba)
    mato a fome e não vou para casa;
senão se (pego o ônibus)
    fico com fome e vou para casa;
senão
    não faço nada
```

```
se (pego o ônibus)
    fico com fome e vou para casa;
senão se (compro a goiaba)
    mato a fome e não vou para casa;
senão
    não faço nada
```

# if, else if, else

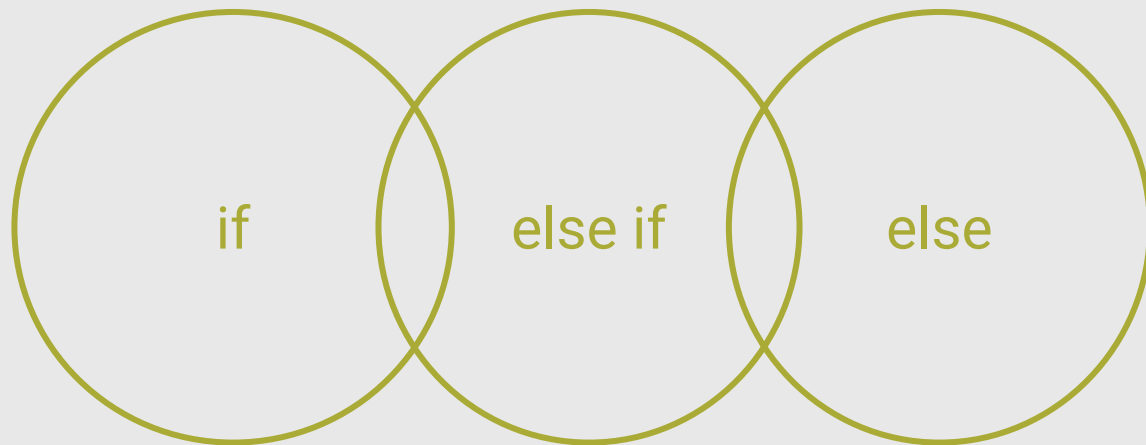
Se determinada condição for verdadeira, determinada instrução é executada

```
if (condição) {  
    instrução  
} else if (condição) {  
    instrução  
} else {  
    instrução  
}
```

Executa determinada instrução se as condições anteriores não forem verdadeiras.

Se a condição do comando anterior não for verdadeira e a condição desse comando for verdadeira, então determinada instrução é executada

## if, else if, else



É o primeiro a ser  
usado dentro de uma  
condição

É o último a ser usado  
dentro de uma  
condição



# if, else if, else

*Exemplo:*

```
if (numero > 0) {  
  alert('O número é positivo!');  
} else if (numero < 0) {  
  alert('O número é negativo!');  
} else {  
  alert('O número é igual a zero!');  
}
```

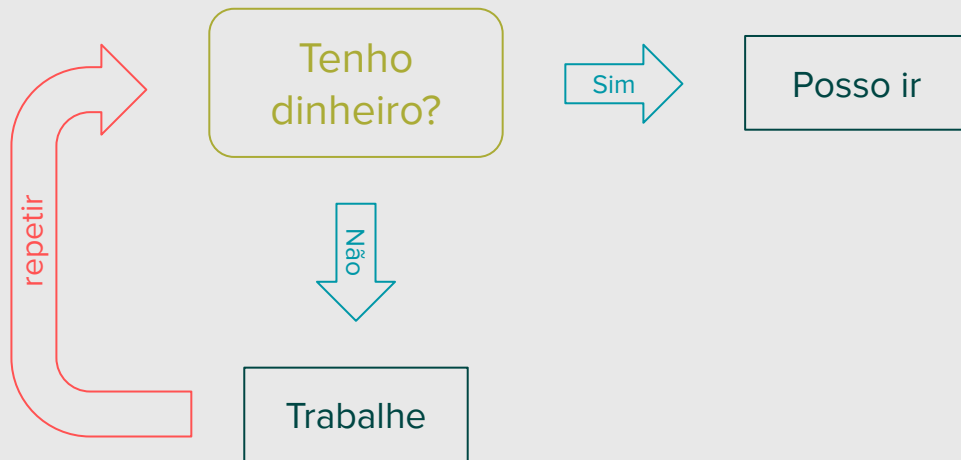
No exemplo, a condicional do primeiro if será verificada. Caso seja verdadeira, sua instrução será realizada e ele não olhará os próximos if's, mas, caso seja falsa ele irá olhar os próximos if's até achar uma condicional verdadeira. Se não achar nenhuma, entrará no último else

# Estrutura de repetição

Algoritmos utilizam das estruturas de repetição para realizar uma instrução repetidamente enquanto determinada condição for satisfeita

# Estrutura de Repetição

Problema: Quero ir para Disney



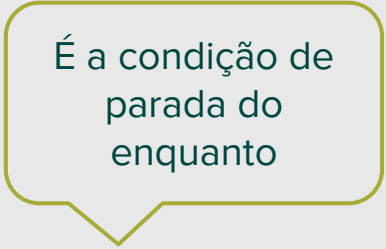
# Comandos de decisão

Como o código funcionaria:

```
enquanto (não tenho dinheiro) {  
    trabalho  
}  
vou para a Disney
```

# while

Este loop executa um bloco de código enquanto uma condição especificada é verdadeira.



É a condição de  
parada do  
enquanto

```
while (condição) {  
    instruções  
}
```

# while

*Exemplo:*

```
var i = 0, texto = ' ';  
while (i < 10) {  
    texto += '0 número é ' + i + '\n';  
    i++;  
}  
  
alert(texto);
```

No exemplo, o código do ciclo será executado, uma e outras vezes, desde que a variável *i* seja inferior a 10

# Referências

Material da disciplina de Programação Web do prof. Flávio Coutinho:

<https://fegemo.github.io/cefet-web/classes/js1/#34>

Documentação da Mozilla Developer Network:

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/if...else>

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/while>

# Leiam

switch case

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/switch>

for

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/for>

do while

<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Statements/do...while>



Dúvidas?

## Colocando em Prática (comando de decisão)

- Crie um programa que simule um sistema bancário. Você deverá ler o saldo inicial do usuário e, em seguida, ler o valor da compra que este deseja realizar. Se o valor da compra for menor ou igual ao saldo inicial da conta, o programa deve informar que a compra ocorreu corretamente e retornar o valor do saldo atual. Se o valor da compra for maior do que o saldo inicial, você deverá retornar a mensagem “saldo insuficiente”.
- Faça uma calculadora. Escreva um programa que receba dois números inteiros. Em seguida, deve ser recebido um caracter (“+”, “-”, “\*” ou “/”) que indicará a operação a ser efetuada com esses dois números. Por último, deve ser mostrado na tela o resultado da operação entre esses dois números.

## Colocando em Prática (estrutura de repetição)

- Faça um programa que leia 4 números digitados pelo usuário e exiba a soma e a média desses valores. Mostre ao usuário o resultado.
- Altere o programa anterior, de tal maneira que o usuário informe a quantidade de números que serão digitados (ou seja, o valor “4” não deve ser fixo no programa).
- Faça um programa que solicite ao usuário dois números inteiros diferentes “num1” e “num2” e calcule a soma de todos os números ímpares dentro do intervalo definido por [num1, ... , num2]. Considere que num1 é sempre menor do que num2. Mostre ao usuário o resultado.