

# Estruturas de controle

## Comandos de seleção

- ✓ Os comandos de seleção da linguagem JavaScript, são fundamentais para especificar as decisões que um computador realizará.
- ✓ Entre os comandos de seleção, temos os comandos if-else (se/senão) e switch (escolha).

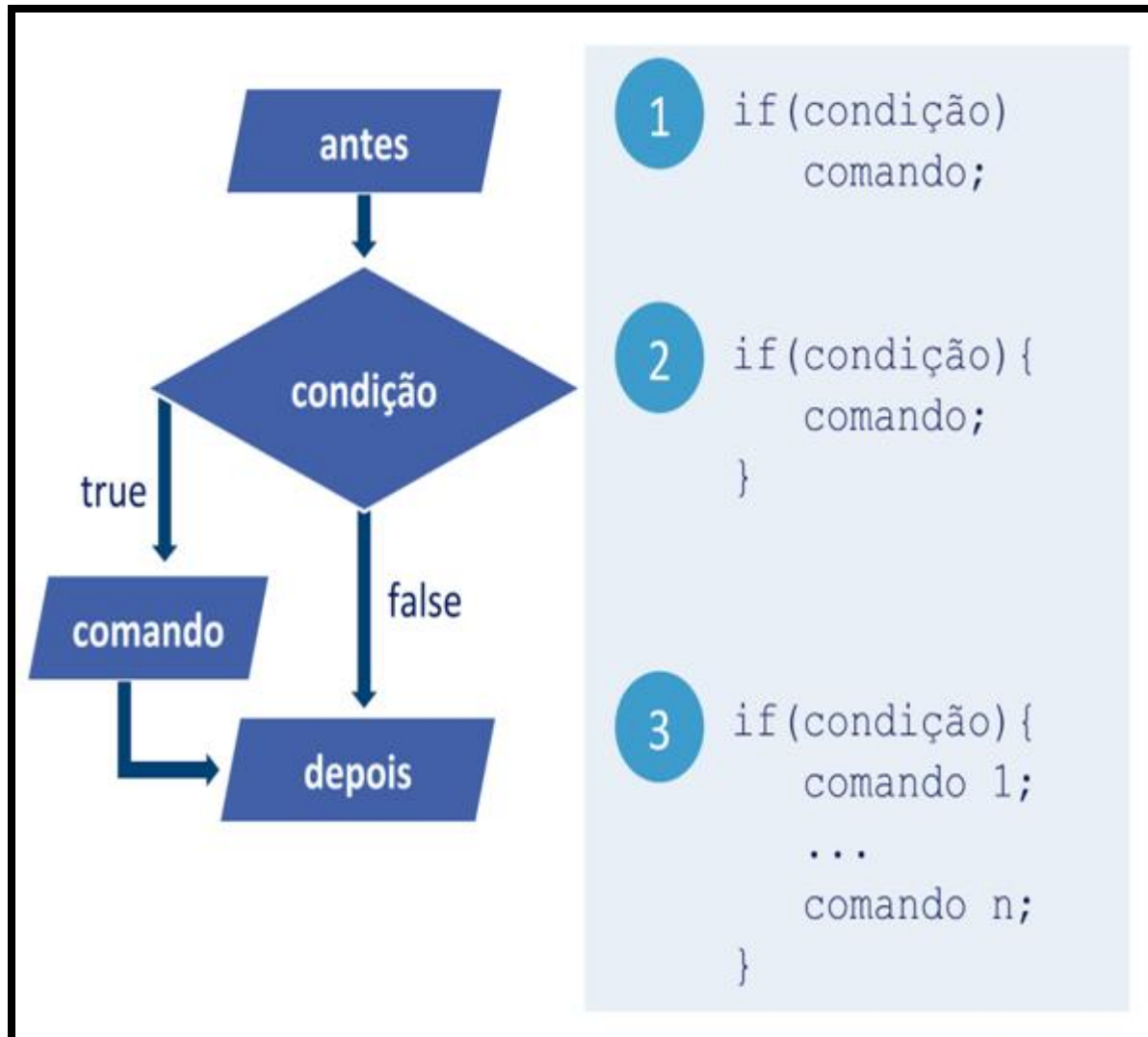
# Estruturas de controle

## Comandos de seleção

### if

O comando if é o mais básico de todos os comandos de seleção e deve ser usado com as duas letras minúsculas.

Ele determina que o programa deve executar um certo comando ou seção de código apenas se uma certa condição for verdadeira.



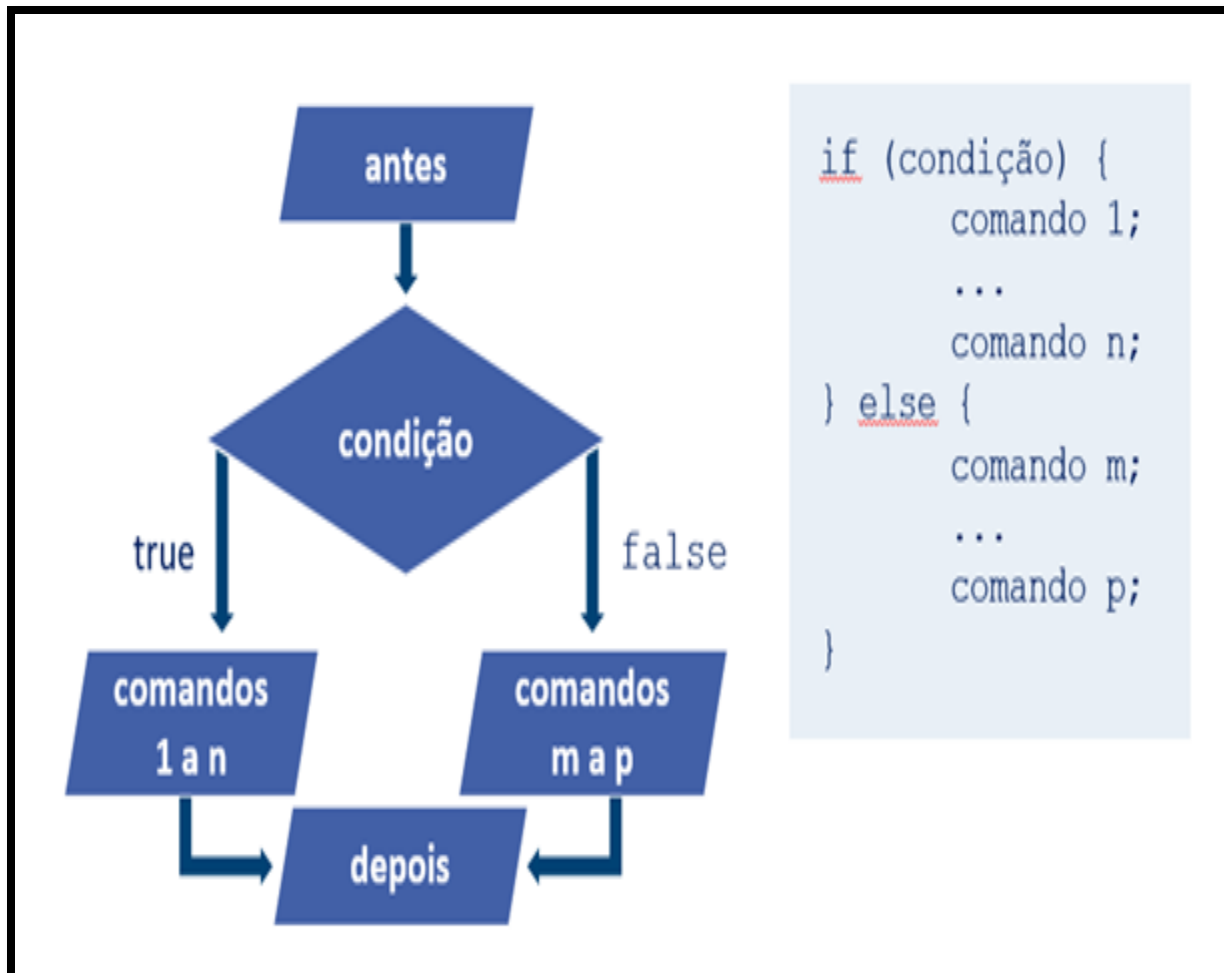
# Estruturas de controle

## Comandos de seleção

### if/else

Nesse comando, temos que, se a condição for verdadeira, apenas os comandos do *if* serão executados.

Senão, ou seja, caso a condição seja falsa, apenas os comandos *else* serão executados.



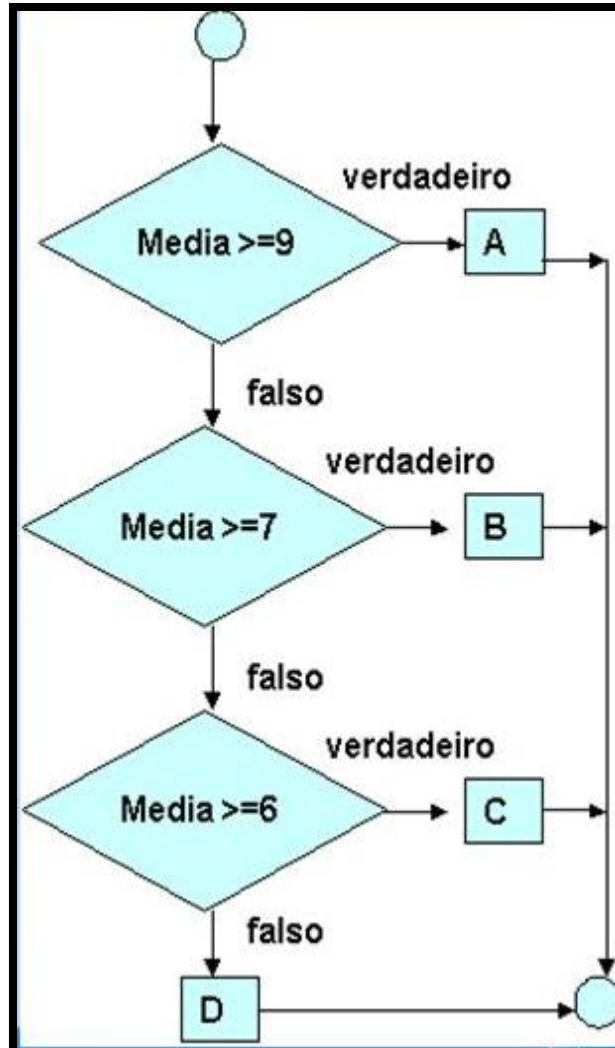
# Estruturas de controle

## Comandos de seleção

### if/else/if (encadeados)

Chamamos de estruturas de decisão encadeadas, quando uma estrutura de decisão está localizada dentro do lado falso da outra.

Este tipo de estrutura também é conhecida como seleção “aninhada” ou seleção “encaixada”.



```
if (condição 1) {  
    comando 1;  
    ...  
    comando n;  
}  
else if (condição 2) {  
    comando m;  
    ...  
    comando p;  
}  
else {  
    comando q;  
    ...  
    comando t;  
}
```

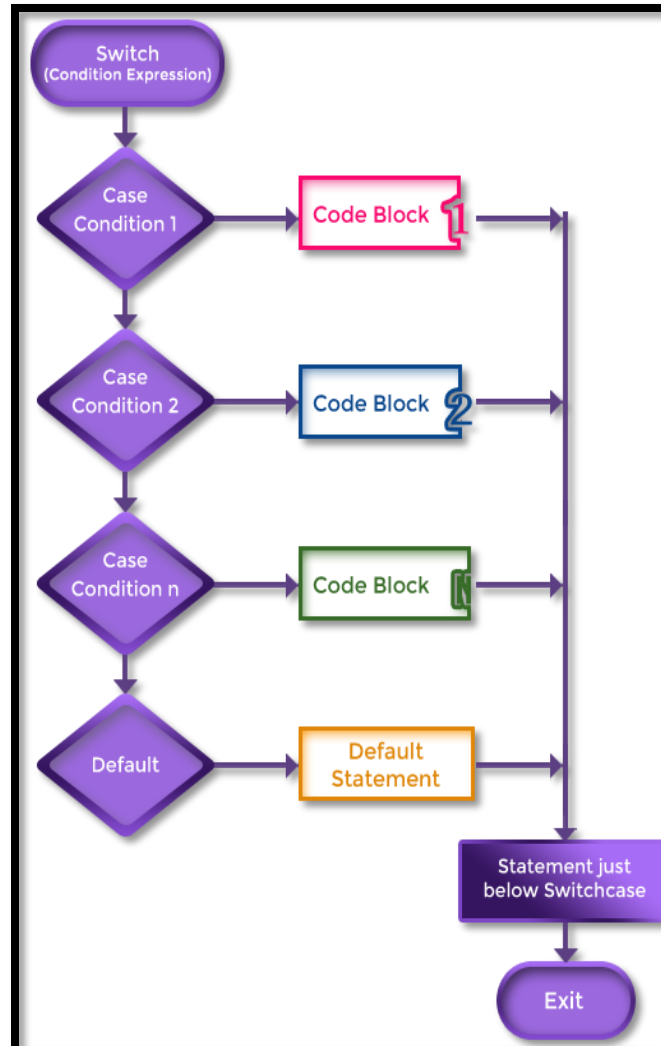
# Estruturas de controle

## Comandos de seleção

### switch

O comando switch facilita a escrita de trechos de programa como este em que a seleção deve ser feita entre várias alternativas.

O switch é chamado de comando interno de seleção múltipla, ele compara o valor de uma expressão com uma lista de valores.



```
switch(expressão) {  
    case valor1:  
        // bloco de código  
        break;  
    case valor2:  
        // bloco de código  
        break;  
    ...  
    default:  
        // bloco de código  
}  
  
// resto do programa
```

# Estruturas de controle

## Comandos de repetição(iteração)

- ✓ Existem situações nas quais você precisará repetir a execução de um mesmo bloco de código.
- ✓ Os comandos utilizados em tais situações, são chamados de comandos de iteração.

***São eles:***

***for***: percorre um bloco de código várias vezes;

***for/in***: percorre as propriedades de um objeto;

***for/of***: percorre os valores de um objeto iterável;

***while***: percorre um bloco de código enquanto uma condição especificada é verdadeira;

***do/while***: também percorre um bloco de código enquanto uma condição especificada é verdadeira, mas executa esse bloco de código pelo menos uma vez.

# Estruturas de controle

## Comandos de repetição(iteração)

for

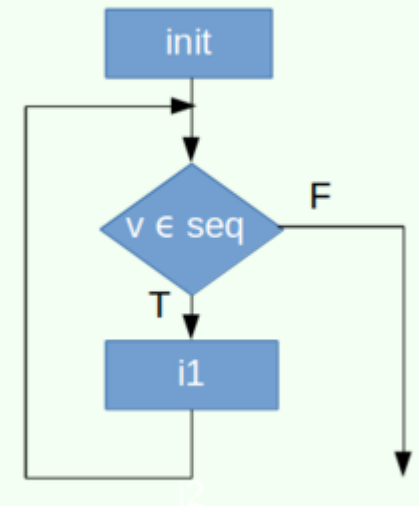
A instrução for cria um loop que consiste em três expressões opcionais, dentro de parênteses e separadas por ponto e vírgula, seguidas por uma declaração ou uma sequência de declarações executadas em sequência.

```
for (Inicialização; Condição; Atualização) {  
    // Lista de comandos  
}
```



```
for (i = 1; i <= n; i++) {  
    texto = texto + i + "<br>";  
}
```

For loop



# Estruturas de controle

## Comandos de repetição(iteração)

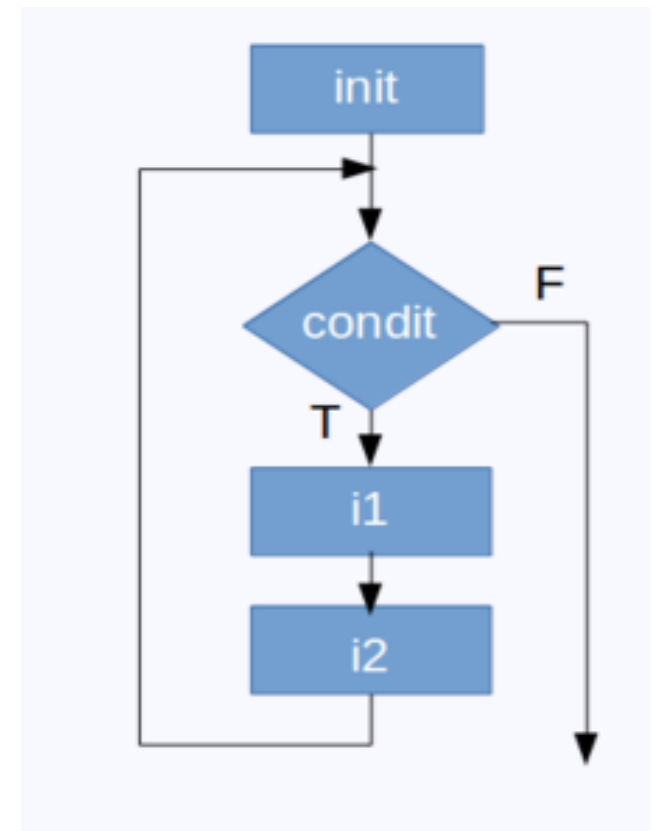
### while

Assim como o laço *for*, o laço *while* é utilizado para repetir a execução de um bloco de comandos.

```
while (condição) {  
    // Lista de Comandos  
}
```



```
var i = 1;  
while (i <= n) {  
    texto = texto + i + "<br>";  
    i++;  
}
```





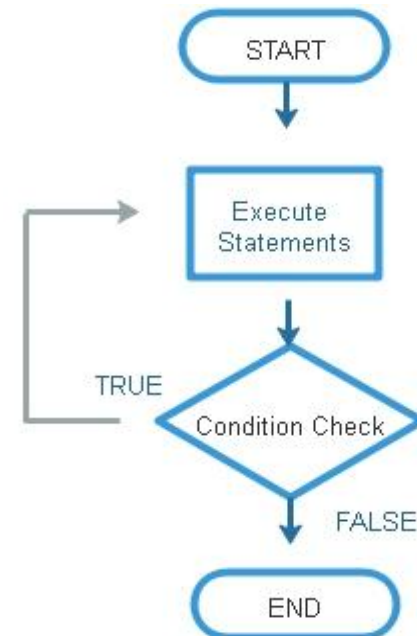
# Estruturas de controle

## Comandos de repetição(iteração)

### do/while

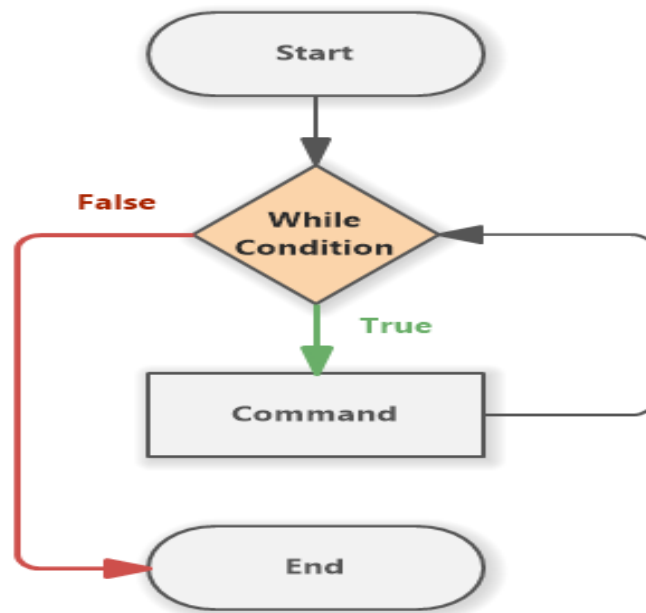
Cria-se um laço que executa uma declaração até que o teste da condição for falsa (false). A condição é avaliada depois que o bloco de código é executado, resultando que uma declaração seja executada pelo menos uma vez.

```
do{  
    try();  
}while(failure);
```

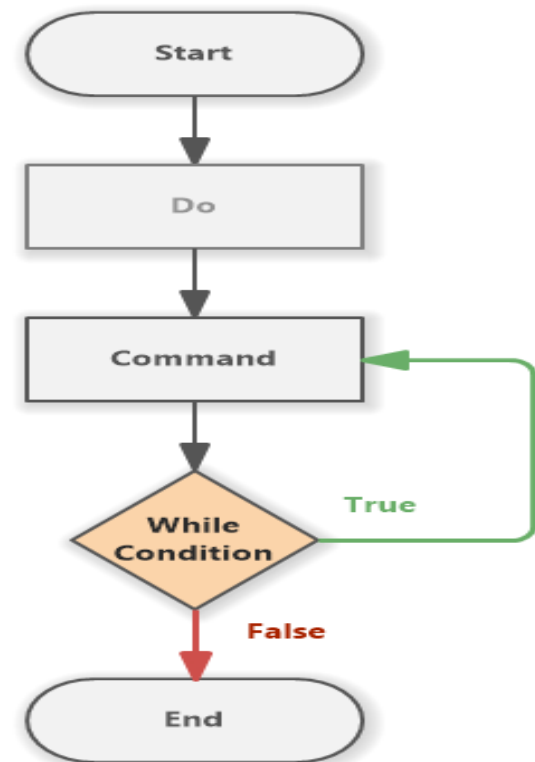


# Estrutura

## WHILE



## DO-WHILE



# Exemplo

