

8. Revisão

Estruturas de Seleção

Em uma linguagem de programação, uma estrutura de seleção é algo que permite alterar o fluxo de execução do nosso código. Como já vimos, o código é executado em sentido de leitura, da esquerda para a direita, de cima para baixo, podemos utilizar alguma das seguintes estruturas para modificar parte desse fluxo de execução.

`if ... else`

Essa estrutura serve para executar um determinado trecho de código caso uma condição seja verdadeira (ou seja, utiliza uma operação booleana). Podemos utilizar essa estrutura de três formas, somente com o `if`, `if e else` e `if e else` encadeado. Vejamos alguns exemplos:

1. `if` simples

Vamos escrever uma condicional que imprima uma mensagem no console caso o valor da variável `a` seja maior do que 5.

```
const a = 6;

if (a > 5) {
    console.log("O valor de 'a' é maior que 5");
}
```

No exemplo acima, nossa estrutura `if` verifica se a comparação `a > 5` é verdadeira, caso seja, o código dentro do escopo (chaves) do `if` é executado, caso contrário, o código seguirá seu fluxo de execução.

2. `if e else`

Com base no cenário anterior, iremos imprimir uma mensagem caso a comparação seja verdadeira, e outra mensagem caso seja falsa.

```
const a = 5;

if (a > 5) {
    console.log("O valor de 'a' é maior que 5");
} else {
```

```
    console.log("O valor de 'a' é menor que 5");  
}
```

Temos uma estrutura semelhante a anterior, porém agora com um diferencial, caso nossa condição seja falsa, o código dentro do escopo do `else` será executado.

Porém, se observarmos o que está escrito no código, veremos que há um erro, quando comparamos números, existem três cenários possíveis, o valor de `a` pode ser maior, menor ou igual a 5, no caso acima, a mensagem apresentada está incorreta, pois o valor de `a` é igual à 5, para isso, podemos utilizar o terceiro caso.

3. `if` e `else` encadeado

Podemos prever os três casos possíveis utilizando o `if` e `else` encadeado, da seguinte forma:

```
const a = 5;  
  
if (a > 5) {  
    console.log("O valor de 'a' é maior que 5");  
} else if(a == 5) {  
    console.log("O valor de 'a' é igual a 5");  
} else {  
    console.log("O valor de 'a' é menor que 5");  
}
```

Essa estrutura realiza as comparações na ordem em que foram escritas, e para na primeira verdadeira, caso nenhuma seja verdadeira, `else` será executado.

`switch`

Diferente do `if`, `switch` vai comparar um valor com diversas possibilidades, ao invés de fazer uma operação booleana, e ao invés de utilizar um `else` para o caso falso (que não existe, pois não há comparação booleana), utilizamos o `default`, `switch` é escrito da seguinte forma:

```
const fruta = "Maçã";  
  
switch (fruta) {  
    case "Banana":  
        console.log("A fruta escolhida foi a banana");  
        break;  
  
    case "Laranja":
```

```
        console.log("A fruta escolhida foi a laranja");
        break;

    case "Maçã":
        console.log("A fruta escolhida foi a maçã");
        break;

    default:
        console.log("A fruta escolhida não está disponível");
        break;
}
```

Estruturas de Repetição

Em uma linguagem de programação, utilizamos estruturas de repetição para executar um código diversas vezes com base em uma condição.

for

A estrutura `for` realiza diversas execução com base em um contador, sua declaração sempre precisa de um contador, uma comparação e um incremento ou decremento, da seguinte forma:

```
for (let i = 0; i < 5; i++) {
    console.log("Execução atual: " i + 1);
}
```

No código acima, declaramos `i` como 0, dizemos que o `for` deve ser executado enquanto `i` for menor que 5, e após cada execução, incrementamos em 1 (`++`) o valor de `i`.

while

Diferente do `for`, `while` irá realizar repetições enquanto uma determinada condição for verdadeira, observe o seguinte cenário:

```
let condicao = true;
let valor = 0;

while (condicao) {
    if (valor > 3) {
        condicao = false;
    }

    valor = valor + 2;
}
```

```
}  
  
console.log(valor);
```

No código acima, declaramos duas variáveis, a variável `condicao` que recebe o booleano `true`, e a variável `valor` que recebe o número 0, o `while` será executado enquanto `condicao` for `true`, dentro do nosso loop, se `valor` for maior que 3, alteramos `condicao` para `false`, e a cada execução iremos incrementar `valor` em 2.

Execute o código acima, e em seguida execute o seguinte código:

```
let condicao = true;  
let valor = 0;  
  
while (condicao) {  
    valor = valor + 2;  
  
    if (valor > 3) {  
        condicao = false;  
    }  
}  
  
console.log(valor);
```

Os resultados foram iguais? Explique o motivo.

Exercícios

Os exercícios a seguir devem ser realizados de duas maneiras, uma com `for` e outra com `while`.

1. Escreva um código que declara um array com 6 números à sua escolha, passe por todos os itens do array e imprima no console o maior valor dentro do array.
2. Escreva um código que faz uma contagem regressiva de 10 a 0.
3. Escreva um código que faz a soma de todos os valores pares de 0 a 100.
4. Escreva um código que faz a soma de todos os valores de um array.
5. Escreva um código que imprime um array com todos os valores ímpares de 1 a 100.
6. Escreva um código que inverte a `string` "lorem ipsum dolor sit amet".

Os exercícios a seguir podem ser resolvidos com `if` e `else` ou `switch`.

1. Escreva um código que declara uma string, e imprime no console três mensagens, uma caso o tamanho da string seja maior que 10, outra caso seja menor que 10 e uma última caso seja igual a 10.
2. Escreva um código que verifica se um caractere é uma vogal.
3. Escreva um código que indica o desconto de um produto com base na variável `categoria`, se a categoria for "eletrônicos", o desconto é de 20% (0.8), se a categoria for "escritório", o desconto é de 15% (0.15), se não for de outra categoria, não há desconto.
4. Escreva um código que diz se uma pessoa é maior de idade ou não.
5. Escreva um código que imprime a ação tomada por um jogador com base na variável `comando`, os comandos possíveis são "atacar", "defender" e "correr".
6. Escreva um código que imprime a bebida escolhida por um cliente com base na variável `bebidaEscolhida`, as opções disponíveis são "café", "chá" e "suco".