



JavaScript

Lógica de Programação

Switch (Estrutura Condicional)

Também chamado de estrutura de seleção múltipla (caso).

Assim como o **if-else**, o **switch** também é uma estrutura de seleção.

O **if-else** realiza o teste da condição e considera apenas duas alternativas para a expressão: **verdadeiro** ou **falso**.

Já o switch é próprio para se testar uma variável em relação a diversos valores pré-estabelecidos

Switch (Estrutura Condicional)

A condicional switch avalia uma expressão, combinando o valor da expressão para um cláusula case, e executa as instruções associadas ao case.

```
switch (expressão) {  
    case valor1:  
        //Instruções executadas quando o resultado da expressão for igual á valor1  
        break;  
    case valor2:  
        //Instruções executadas quando o resultado da expressão for igual á valor2  
        break;  
    case valor3:  
        //Instruções executadas quando o resultado da expressão for igual á valor3  
        break;  
    default:  
        //Instruções executadas quando o valor da expressão é diferente de todos os cases  
        break;  
}
```

Laço de Repetição (Loop)

As linguagens de programação são muito úteis para **concluir rapidamente tarefas repetitivas**, desde vários cálculos básicos até praticamente qualquer outra situação em que você tenha muitos itens semelhantes para manipular.

O loop de programação é como fazer a **mesma coisa repetidas vezes** - o que é chamado de **iteração** na linguagem de programação

Estrutura de um Loop

- O **contador**, que é inicializado com um certo valor - este é o ponto inicial do loop.
- A **condição de saída**, que é o critério no qual o loop é interrompido - geralmente o contador atinge um certo valor.
- Um **iterador**, que geralmente incrementa o contador em uma pequena quantidade a cada loop, sucessivamente, até atingir a condição de saída.



For

O você usará, na maior parte do tempo, o loop **for** - ele tem a seguinte sintaxe:

Inicializador— geralmente é uma variável configurada para um número inicial, que é incrementado para contar o número de vezes que o loop foi executado.

Condição-saída — aqui é definido quando o loop deve parar de executar.

Expressão-final — isso sempre é avaliado (ou executado) cada vez que o loop passou por uma iteração completa. Geralmente serve para incrementar ou decrementar a variável contadora

```
<script>

    for (inicializador; condição-saída; expressão-final) {
        // código para executar
    }

</script>
```

Exemplo - For

Imprimir na tela os números de 0 a 50.

```
for (let i = 0; i <= 50; i++) {  
  document.write(i + ',');  
}
```


While

While funciona de maneira muito semelhante ao loop **for**, exceto que a variável inicializador é definida antes do loop, e a expressão final é incluída dentro do loop após o código a ser executado - em vez de esses dois itens serem incluídos dentro dos parênteses.

Inicializador — geralmente é uma variável configurada para um número inicial, que é incrementado para contar o número de vezes que o loop foi executado.

Condição-saída — aqui é definido quando o loop deve parar de executar.

Expressão-final — isso sempre é avaliado (ou executado) cada vez que o loop passou por uma iteração completa. Geralmente serve para incrementar ou decrementar a variável contadora

```
inicializador
while (condição-saída) {
    // código para executar
    expressão-final
}
```


Exemplo - While

Imprimir na tela os números de 0 a 50.

```
let i = 0;
while (i <= 50) {
    document.write(i + ',');

    i++;
}
```

Do...While

Em um loop **do ... while**, o código dentro das chaves **é sempre executado uma vez antes** da verificação ser feita para ver se deve ser executada novamente.

No while e no For, a verificação vem primeiro, então o código pode nunca ser executado.

Inicializador— geralmente é uma variável configurada para um número inicial, que é incrementado para contar o número de vezes que o loop foi executado.

Condição-saída — aqui é definido quando o loop deve parar de executar.

Expressão-final — isso sempre é avaliado (ou executado) cada vez que o loop passou por uma iteração completa. Geralmente serve para incrementar ou decrementar a variável contadora

```
inicializador  
do {  
    // código para executar  
  
    expressão-final  
} while (condição-saída)
```

Exemplo – Do... While

Imprimir na tela os números de 0 a 50.

```
let i = 0;  
do {  
    document.write(i + ',');  
    i++;  
} while (i <= 50);
```



Vamos praticar...



Atividades Proposta

1. Informar todos os números de 1000 a 1999 que quando divididos por 11 obtemos resto = 5
2. Escreva um programa que imprima os números de 1 a 100, sendo que:
 - imprima “fizz” no lugar de números múltiplos de 3
 - imprima “buzz” no lugar de números múltiplos de 5
 - imprima “fizzbuzz” no lugar de números múltiplos de 3 e 5
3. Escreva um programa que pergunte ao usuário quantos alunos tem na sala dele. Em seguida, através de um laço, pede ao usuário para que entre com as notas de todos os alunos da sala, um por vez. Por fim, o programa mostra a média, aritmética, da turma.

Atividades Proposta

4. Criar um programa para identificar o valor a ser pago por um plano de saúde dada a idade do conveniado considerando que todos pagam R\$ 100 mais um adicional conforme a seguinte tabela:
- crianças com menos de 10 anos pagam R\$ 80,00;
 - conveniados com idade entre 10 e 29 anos pagam R\$ 50,00;
 - conveniados com idade entre 30 e 60 anos pagam R\$ 95,00; e
 - conveniados com mais de 60 anos pagam R\$ 130,00.

Senac

Atividades Proposta

5. Um homem decidiu ir à uma revenda comprar um carro. Ele deseja comprar um carro hatch, e a revenda possui, além de carros hatch, sedans, motocicletas e caminhonetes. Utilizando uma estrutura switch/case, caso o comprador queira o hatch, retorne: “Compra efetuada com sucesso”. Nas outras opções, retorne: “Tem certeza que não prefere este modelo?”. Caso seja especificado um modelo que não está disponível, retorne no console: “Não trabalhamos com este tipo de automóvel aqui”

