



# **Desenvolvimento Web Completo 2022**


Por Jorge Sant Ana, Programador  
Jamilton Damasceno, Analista de Sistema e Professor





# Seção 4: JavaScript



# Aula 27: Estrutura de Repetição

- 
- O que são arrays estruturas de repetição?
  - Na programação de forma geral, as estruturas de repetição, também chamadas de laços ou loops, são estruturas que permitem com que a gente faça a repetição de um comando ou de um bloco de códigos até atender a uma determinada condição, ou seja, até atingir um determinado ponto em que falamos que não existe mais a necessidade de repetição do código e onde utilizamos as estruturas de repetição.

- 
- E onde utilizamos as estruturas de repetição?
  - Em geral, para percorrer arrays e, em JavaScript, em especial, para percorrer objetos literais.
  - As estruturas de repetição são tão importantes. Podemos, através das estruturas de repetição, criar um único bloco de códigos e executar esse bloco de códigos modificando, apenas, os valores dinâmicos tantas vezes quanto necessárias.

- 
- De forma geral, os laços esperam o critério de parada, ou seja, uma instrução que indica que a repetição do comando, ou do bloco de códigos, deve ser interrompida.
  - A ausência de um critério de parada implica no que chamamos de loop infinito, resultando no travamento da aplicação.

# Loop While

- A sintaxe do laço de repetição While constitui-se na palavra reservada “while”, abre e fecha parênteses, onde será estabelecida a condição de parada da interação e, na sequência, abre e fecha chaves, onde serão incluídos os códigos a serem repetidos.


# Exemplo

```
//while
```

```
while(condicao) {
```

```
}
```



- 
- E o que é importante para nós?
  - Primeira coisa: a condição. Aqui estabeleceremos operações de comparação e operações lógicas para encontrar verdadeiro ou falso e, com isso, decidir se vamos ou não manter o nosso laço em execução.


# Exemplo

```
//While
var x = 1;

while(x <= 10) {
    //Códigos
    document.write(x + '<br>');
    x++;
}
```

## Estrutura de Repetição -- While

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

- 
- Além do critério de parada, é necessário ver se as variáveis utilizadas nesse critério estão sendo devidamente trabalhadas dentro do loop, para evitar, assim, o loop infinito.
  - Outro detalhe é que podemos fazer iterações, não apenas uma unidade, mas sim, de quantidade.


# Exemplo

```
//While
var x = 1;

while(x <= 10) {
    //Códigos
    document.write(x + '<br>');
    x += 3;
}
```

## Estrutura de Repetição -- While

1  
4  
7  
10

- 
- Não importa a forma como as nossas variáveis serão implementadas ou controladas, é importante, realmente, que a lógica de incremento atenda a necessidade da condição em algum momento.




# Loop Do While

- Diferente do comando “while”, no “do while”, a sintaxe começa pela palavra reservada “do”, na sequência, temos o bloco de códigos e, apenas no término do bloco de códigos, que é feito a instrução “while” com a condição.

# Exemplo

```
//Do While  
✓ do {  
    //Comandos  
} while(condicao)
```

- 
- No comando “do while”, a condição é feita após a primeira execução do bloco. Isso significa que, independente da condição estabelecida para o “do while”, o conteúdo contido dentro das chaves do “do while” será executado pelo menos uma vez. Isso porque o teste é feito após a primeira interpretação do bloco.



# Exemplo

```
//Do While  
var teste = 11;  
  
do {  
    document.write(teste + '<br>');  
} while(x <= 10)
```


## Estrutura de Repetição -- Do While

11



# For

- A sintaxe do comando “for” consiste na palavra reservada “for”, seguida de abre e fecha parênteses, sendo que dentro dos parênteses, passaremos, além da condição, outros 2 (dois) parâmetros, que são separados por ponto e vírgula.
- Passaremos a variável, que vai ser responsável por estabelecer uma lógica, uma condição na parada do laço, também passaremos a condição propriamente dita e o controle, ou seja, a interação sobre essa variável que será utilizada na condição.

- 
- Então, a ordem dos parâmetros no for são importantíssimos e juntas, estabelecem todo o controle necessário do laço em uma única linha, ou seja, o for se torna uma estrutura de repetição muito mais fácil de ser entendida e controlada, porque todas as instruções referentes à condição de parada estão lado a lado.

# Exemplo


```
//For
for(vaiavel; condicao; controle) {
    //Código
}
```

```
//For
for(var y = 10; y >= 0; y--) {
    document.write(y + '<br>');
}
```



## Estrutura de Repetição -- For

10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
0

- 
- Basicamente falando, toda a lógica responsável pelo controle do laço está lado a lado, facilitando a manutenção ou, até mesmo, o entendimento de qual é o critério de parada do laço.

# Loop foreach

- O “foreach”, em linguagens como PHP, por exemplo, é um comando de repetição, ou seja, é um laço, um loop, como propriamente dito.
- O foreach, no contexto do JavaScript, é aplicado em arrays e funciona como uma função. Ele não é bem uma estrutura de repetição, porém, ele serve para esse propósito.

# Exemplo

```
//Foreach
var lista = ['Vivian', 'Lorena', 'Yara', 'Julia', 'Gabrielle'];

lista.forEach(function(valor, indice, array){
    //Lógica
    document.write(lista + '<br>');
})
```