



# **PROGRAMAÇÃO FRONT-END**

GIOVANNI RIBEIRO



# For Javascript



Os loops podem executar um bloco de código várias vezes.

Os loops são úteis, se você deseja executar o mesmo código repetidamente, sempre com um valor diferente.

Geralmente, este é o caso ao trabalhar com matrizes:

```
<div id="exemplo"></div>

<script>
  var times = ["Corinthians", "Santos", "Palmeiras"];
  var texto = "";

  for(var i = 0; i < times.length; i++) {
    texto += times[i] + "<br />";
  }

  document.getElementById("exemplo").innerHTML = texto;
</script>
```

# Tipos loops mais usados



JavaScript suporta diferentes tipos de loops:

- **for** - percorre um bloco de código várias vezes
- **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

# For em javascript



```
for (instrucao1; instrucao2; instrucao3) {  
    //codigo executado  
}
```

**A instrução 1** é executada (uma vez) antes da execução do bloco de código.

**A instrução 2** define a condição para executar o bloco de código.

**A instrução 3** é executada (todas as vezes).

```
for ([inicialização]; [condição]; [expressão final])  
    declaração
```

# Exemplo for



## Usando for

A declaração `for` começa declarando a variável `i` e inicializando-a como `0`. Ela verifica se `i` é menor que nove, executa as duas instruções subsequentes e incrementa 1 a variável `i` após cada passagem pelo loop.

```
for(var i = 0; i < 9; i++) {  
    console.log(i);  
}
```

# Exemplo for



## Usando

## for

Vamos verificar o exemplo a seguir (contagem de 1 a 10):

```
<script>
  for(var x = 1; x <= 10; x++) {
    document.write("<br> Contagem: " + x);
  }
</script>
```

# Exemplo for



## Usando

## for

Da mesma maneira a contagem regressiva é bem simples, basta utilizar o decremento **x--** e inverter a lógica de finalização da repetição, veja o exemplo:

```
for(var i = 10; i > 0; i--) {  
    document.write("<br> Contagem regressiva: " + i);  
}
```

## Exemplo for / In



A for/in instrução JavaScript percorre as propriedades de um objeto:

```
var pessoa = {nome: "Pedro", sobrenome: "Da silva", idade: 25};  
var x;  
for(x in pessoa) {  
    console.log(pessoa[x]);  
}
```



# Loop while



O **while** loop percorre um bloco de código desde que uma condição especificada seja verdadeira.

```
while (condicao) {  
  // código executado  
}
```

```
var i = 0;  
  
while (i <= 10) {  
  console.log(i);  
  i++;  
}
```

# Loop while



Tabuada com while

4 \* 1 = 4  
4 \* 2 = 8  
4 \* 3 = 12  
4 \* 4 = 16  
4 \* 5 = 20  
4 \* 6 = 24  
4 \* 7 = 28  
4 \* 8 = 32  
4 \* 9 = 36  
4 \* 10 = 40

```
var mostra = function(frase) {  
    document.write(frase + "<br />");  
}  
  
var multiplicador = 1;  
while(multiplicador <= 10) {  
    mostra(" 4 " + " * " + multiplicador + " = " + 4 * multiplicador);  
    multiplicador = multiplicador + 1;  
}
```

# Loop For



Imagine que queremos mostrar a seguinte sequência de asteriscos na tela. São 3 linhas com 10 asteriscos cada:

```
*****  
*****  
*****
```

```
for(var linha = 0; linha < 3; linha++) {  
    document.write("*****" + "<br />");  
}
```