

The background features a collection of white, realistic water droplets of various sizes scattered across the page. On the left side, there is a large, intricate geometric pattern composed of many small triangles in shades of brown, tan, and beige, some of which are semi-transparent. The title text is positioned on the right side of the image.

# PT LAÇOS DE REPETIÇÃO

Prof. Anderson Vanin

# LAÇOS DE REPETIÇÃO

Laços oferecem um jeito fácil e rápido de executar uma ação repetidas vezes. Existem várias formas diferentes de laços, mas eles essencialmente fazem a mesma coisa: repetir uma ação múltiplas vezes ( inclusive você poderá repetir 0 vezes). Os vários mecanismos diferentes de laços oferecem diferentes formas de determinar quando este irá começar ou terminar. Há várias situações em que é mais fácil resolver um problema utilizando um determinado tipo de laço do que outros.

# TIPOS DE LAÇOS DE REPETIÇÃO

- for
- do...while
- while
- for...in

# FOR

Um laço for é repetido até que a condição especificada seja falsa. O laço for no JavaScript é similar ao Java e C. Uma declaração for é feita da seguinte maneira:

```
for ([expressaoInicial]; [condicao]; [incremento])  
    declaracao
```

# FOR – EXEMPLO 01

JS exemplo01.js > ...

```
1  var passo;  
2  for (passo = 0; passo < 5; passo++) {  
3      // Executa 5 vezes, com os valores de passos de 0 a 4.  
4      console.log(`Passo nº ${passo} ==> Passo`)  
5  }
```

PROBLEMAS

SAÍDA

CONSOLE DE DEPURAÇÃO

TERMINAL

Info: Start process (11:12:40)

Passo nº 0 ==> Passo

Passo nº 1 ==> Passo

Passo nº 2 ==> Passo

Passo nº 3 ==> Passo

Passo nº 4 ==> Passo

Info: End process (11:12:40)

# DO..WHILE

A instrução do...while repetirá até que a condição especificada seja falsa.

```
do  
    declaracao  
while (condicao);
```

# DO..WHILE – EXEMPLO 02

A instrução do...while repetirá até que a condição especificada seja falsa.

JS exemplo02.js > ...

```
1  var i = 0
2  do {
3      i += 1
4      console.log(i)
5  } while (i < 5)
```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

Info: Start process (11:18:03)

1  
2  
3  
4  
5

Info: End process (11:18:03)

# WHILE

Uma declaração while executa suas instruções, desde que uma condição especificada seja avaliada como verdadeira. Segue uma declaração while:

```
while (condicao)  
    declaracao
```



# WHILE – EXEMPLO 03

Uma declaração while executa suas instruções, desde que uma condição especificada seja avaliada como verdadeira. Segue uma declaração while:

JS exemplo03.js

```
1  n = 0;
2  x = 0;
3  while (n < 3) {
4      n++;
5      console.log(`Valor de n após incremento ==> ${n}`)
6      x += n;
7      console.log(`Valor de x ==> ${x}`)
8  }
```

PROBLEMAS SAÍDA CONSOLE DE DEPURAÇÃO TERMINAL

```
Info: Start process (11:22:01)
Valor de n após incremento ==> 1
Valor de x ==> 1
Valor de n após incremento ==> 2
Valor de x ==> 3
Valor de n após incremento ==> 3
Valor de x ==> 6
Info: End process (11:22:02)
```

# FOR..IN

A declaração for...in executa iterações a partir de uma variável específica, percorrendo todas as propriedades de um objeto.

Para cada propriedade distinta, o JavaScript executará uma iteração. Segue a sintaxe:

```
for (variavel in objeto) {  
    declaracoes  
}
```

# FOR..IN – EXEMPLO 04

JS exemplo04.js > ...

```
1  var pessoas = {fname:"John", lname:"Doe", age:25};
2
3  var text = "";
4  for (var x in pessoas) {
5      console.log(`Valor do x ==> ${x}`)
6      text += pessoas[x];
7      console.log(text)
8  }
```

PROBLEMAS

SAÍDA

CONSOLE DE DEPURAÇÃO

TERMINAL

Info: Start process (11:29:06)

Valor do x ==> fname

John

Valor do x ==> lname

JohnDoe

Valor do x ==> age

JohnDoe25

Info: End process (11:29:06)

# EXERCÍCIOS

1. Faça um programa que peça uma nota, entre zero e dez. Mostre uma mensagem caso o valor seja inválido e continue pedindo até que o usuário informe um valor válido.
2. Supondo que a população de um país A seja da ordem de 80000 habitantes com uma taxa anual de crescimento de 3% e que a população de B seja 200000 habitantes com uma taxa de crescimento de 1.5%. Faça um programa que calcule e escreva o número de anos necessários para que a população do país A ultrapasse ou iguale a população do país B, mantidas as taxas de crescimento.
3. Altere o programa anterior permitindo ao usuário informar as populações e as taxas de crescimento iniciais. Valide a entrada e permita repetir a operação.
4. Faça um programa que imprima na tela apenas os números ímpares entre 1 e 50.

# EXERCÍCIOS

5. Desenvolva um gerador de tabuada, capaz de gerar a tabuada de qualquer número inteiro entre 1 a 10. O usuário deve informar de qual numero ele deseja ver a tabuada. A saída deve ser conforme o exemplo abaixo:

```
Tabuada de 5:  
5 X 1 = 5  
5 X 2 = 10  
...  
5 X 10 = 50
```

6. Faça um programa que peça dois números, base e expoente, calcule e mostre o primeiro número elevado ao segundo número. Não utilize a função de potência da linguagem.
7. Faça um programa que peça 10 números inteiros, calcule e mostre a quantidade de números pares e a quantidade de números ímpares.

# ENVIAR EXERCÍCIOS PELO SEU GITHUB!

- Crie uma pasta em seu computador nomeada como:  
EXERCICIOS\_REPETIÇÃO\_JS
- Dentro desta pasta, crie uma pasta para cada exercício.  
Exemplo: EXERCICIO\_01, EXERCICIO\_02, ...
- No seu Github das aulas de IW-II, envie a pasta criada em seu computador com todos os exercícios propostos.