

# Comandos de repetição

Fundamentos de Lógica e Algoritmos

# Sumário

# Definição

# Comando de repetição

- Wikipedia [https://pt.wikipedia.org/wiki/Estrutura de repetição](https://pt.wikipedia.org/wiki/Estrutura_de_repeti%C3%A7%C3%A3o)
  - É uma instrução de fluxo de controle que permite que o código seja executado repetidamente com base em uma determinada condição booleana
- Sinônimos
  - estrutura de repetição

**Opções**

**Controlar a repetição  
no início**

# Forma geral

**Comando de repetição** **Condição**

**Blocos de Comandos**

**Fim do Comando de repetição**

# Potigol - para

```
# Para decrescente  
# escreve os números de 10 a 1  
para i de 10 até 1 passo -1 faça  
    escreva i  
fim
```

```
# Para com listas  
cores = ["azul", "vermelho", "verde"]  
para cor em cores faça  
    escreva cor  
fim
```

```
# Para com mais de um gerador  
# escreve a tabuada {1..4} x {1..3}  
para i de 1 até 4,  
    j de 1 até 3 faça  
    escreva "{i} * {j} == {i * j}"  
fim
```

# Potigol - enquanto

```
1  # Escreve os números de 1 a 10
2  var i := 0
3  enquanto i <= 10 faça
4      escreva i
5      i := i + 1
6  fim
```

**Controlar a repetição  
no fim**

# Forma geral

**Comando de repetição**

**Blocos de Comandos**

**Condição de Fim do Comando de rep.**

# Exemplo em C

```
4      int i = 10;  
5  
6      do {  
7          printf("%d \n", i);  
8          i = i + 1;  
9      } while (i <= 3);
```

# Exemplo em C

```
4   int i = 10;  
5  
6   do {  
7       printf("%d \n", i);  
8       i = i + 1;  
9   } while (i <= 3);
```

```
[leo:exemplos (master) $ gcc dowhile.c -o dowhile  
[leo:exemplos (master) $ ./dowhile  
10  
leo:exemplos (master) $ █
```

**Exemplos**

# 1º Exemplo

```
1  # escreve os números de 1 a 10
2  para i de 1 até 10 faça
3      escreva i
4  fim
```

# Saída

```
1 # escreve os números de 1 a 10
2 para i de 1 até 10 faça
3     escreva i
4 fim
```

```
[leo:para (master) $ potigol exemplo_1.poti
1
2
3
4
5
6
7
8
9
10
```

# 2º Exemplo

```
# Para decrescente  
# escreve os números de 10 a 1  
para i de 10 até 1 passo -1 faça  
    escreva i  
fim
```

# Saída

```
# Para decrescente  
# escreve os números de 10 a 1  
para i de 10 até 1 passo -1 faça  
  escreva i  
fim
```

```
leo:para (master) $ potigol exemplo4.poti  
10  
9  
8  
7  
6  
5  
4  
3  
2  
1  
leo:para (master) $
```

# 3º Exemplo

```
1  # soma os números de 1 a 10
2  var soma := 0
3  para i de 1 até 10 faça
4      soma := soma + i
5  fim
6  escreva "A soma é {soma}."
```

# Saída

```
1  # soma os números de 1 a 10
2  var soma := 0
3  para i de 1 até 10 faça
4      soma := soma + i
5  fim
6  escreva "A soma é {soma}."
```

```
[leo:para (master) $ potigol exemplo2.poti
A soma é 55.
leo:para (master) $ █
```

# 4º Exemplo

```
# Para com listas  
cores = ["azul", "vermelho", "verde"]  
para cor em cores faça  
    escreva cor  
fim
```

# Saída

```
# Para com listas  
cores = ["azul", "vermelho", "verde"]  
para cor em cores faça  
    escreva cor  
fim
```

```
[leo:para (master) $ potigol exemplo6.poti  
azul  
vermelho  
verde  
leo:para (master) $ █
```

# 5º Exemplo

```
# Para com mais de um gerador  
# escreve a tabuada {1..4} x {1..3}  
para i de 1 até 4,  
    j de 1 até 3 faça  
    escreva "{i} * {j} == {i * j}"  
fim
```

# Saída

```
# Para com mais de um gerador  
# escreve a tabuada {1..4} x {1..3}  
para i de 1 até 4,  
    j de 1 até 3 faça  
    escreva "{i} * {j} == {i * j}"  
fim
```

```
[leo:para (master) $ potigol exemplo5.poti  
1 * 1 == 1  
1 * 2 == 2  
1 * 3 == 3  
2 * 1 == 2  
2 * 2 == 4  
2 * 3 == 6  
3 * 1 == 3  
3 * 2 == 6  
3 * 3 == 9  
4 * 1 == 4  
4 * 2 == 8  
4 * 3 == 12  
leo:para (master) $
```

# 6º Exemplo

```
1  # Escreve os números de 1 a 10  
2  var i := 0  
3  enquanto i <= 10 faça  
4      escreva i  
5      i := i + 1  
6  fim
```

# Saída

```
1  # Escreve os números de 1 a 10
2  var i := 0
3  enquanto i <= 10 faça
4      escreva i
5      i := i + 1
6  fim
```

```
[leo:enquanto (master) $ potigol exemplo_1.poti
0
1
2
3
4
5
6
7
8
9
10
leo:enquanto (master) $ █
```

# 7º Exemplo

```
1  # soma os números de 1 a 10
2  var soma := 0
3  var i := 0
4  enquanto i <= 10 faça
5      soma := soma + i
6      i := i + 1
7  fim
8  escreva "A soma é {soma}."
```

# Saída

```
1  # soma os números de 1 a 10
2  var soma := 0
3  var i := 0
4  enquanto i <= 10 faça
5      soma := soma + i
6      i := i + 1
7  fim
8  escreva "A soma é {soma}."
```

```
[leo:enquanto (master) $ potigol exemplo_2.poti
A soma é 55.
leo:enquanto (master) $ █
```

# 8º Exemplo

```
1   var i := 0
2   enquanto i <= 10 faça
3       escreva i
4       i := i + 2
5   fim
```

# Saída

```
1  var i := 0
2  enquanto i <= 10 faça
3      escreva i
4      i := i + 2
5  fim
```

```
[leo:enquanto (master) $ potigol exemplo_3.poti
0
2
4
6
8
10
leo:enquanto (master) $ █
```

# 9º Exemplo

```
1  # Para com mais de um gerador
2  # escreve a tabuada {1..4} x {1..3}
3  var i := 1
4  enquanto i <= 4 faça
5      var j := 1
6      enquanto j <= 3 faça
7          escreva "{i} * {j} == {i * j}"
8          j := j + 1
9      fim
10     i := i + 1
11 fim
```

# Saída

```
1  # Para com mais de um gerador
2  # escreve a tabuada {1..4} x {1..3}
3  var i := 1
4  enquanto i <= 4 faça
5      var j := 1
6      enquanto j <= 3 faça
7          escreva "{i} * {j} == {i * j}"
8          j := j + 1
9      fim
10     i := i + 1
11 fim
```

```
[leo:enquanto (master) $ potigol exemplo_5.poti
1 * 1 == 1
1 * 2 == 2
1 * 3 == 3
2 * 1 == 2
2 * 2 == 4
2 * 3 == 6
3 * 1 == 3
3 * 2 == 6
3 * 3 == 9
4 * 1 == 4
4 * 2 == 8
4 * 3 == 12
leo:enquanto (master) $
```

# 10º Exemplo

```
1  # Para com listas
2  var i := 1
3  cores = ["azul", "vermelho", "verde"]
4  enquanto i <= cores.tamanho faça
5      cor = cores[i]
6      escreva cor
7      i := i + 1
8  fim
```

# Saída

```
1  # Para com listas
2  var i := 1
3  cores = ["azul", "vermelho", "verde"]
4  enquanto i <= cores.tamanho faça
5      cor = cores[i]
6      escreva cor
7      i := i + 1
8  fim
```

```
[leo:enquanto (master) $ potigol exemplo_6.poti
azul
vermelho
verde
```

**Desafios**

# Qual a saída?

```
1  # Para gerando uma lista
2  números = para i de 1 até 5 gere i fim
3
4  pares = para i de 1 até 10 se i mod 2 == 0 gere i
5
6  escreva "{números}"
7  escreva "{pares}"
```