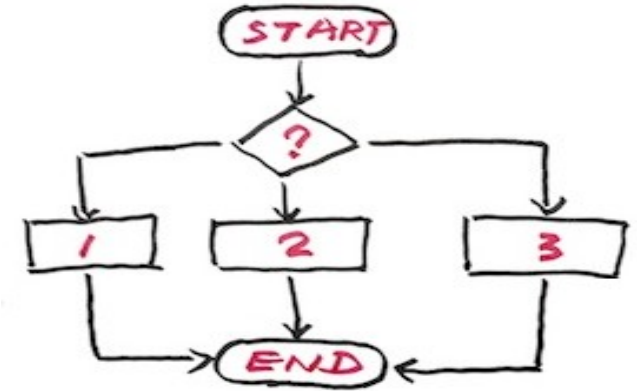


# Introdução à Programação

## Aula 07 *Estruturas de repetição*

*Prof. Dr. Ivan José dos Reis Filho*



# Aula de hoje

- Blocos de instrução
- Estruturas de repetição
  - For: range
  - While

# Blocos de instrução

Conjunto de instruções que estão no mesmo nível.

```
#coding: utf-8

valor = input("Digite um número: ")
num = int(valor) #convertendo de str p/ int

print("O número digitado foi: ", str(num))

if (num > 10):
    print("O número digitado é maior do que 10 ")
    print()
    print()
    print()
    print()
```

# Blocos de instrução

## Vários blocos de instrução

- Bloco 1 de Nível 1
- Bloco 2 de Nível 2
- Bloco 3 de Nível 3
- Bloco 4 de nível 2
- Bloco 1 de nível 1

O bloco 2 e o bloco 4 estão, hierarquicamente, no mesmo nível. Porém, após a execução do bloco 2, todos os nomes utilizados dentro deste serão desalocados da memória e assim, quando o bloco 4 for executado, não será possível enxergar nomes utilizados no bloco 2!

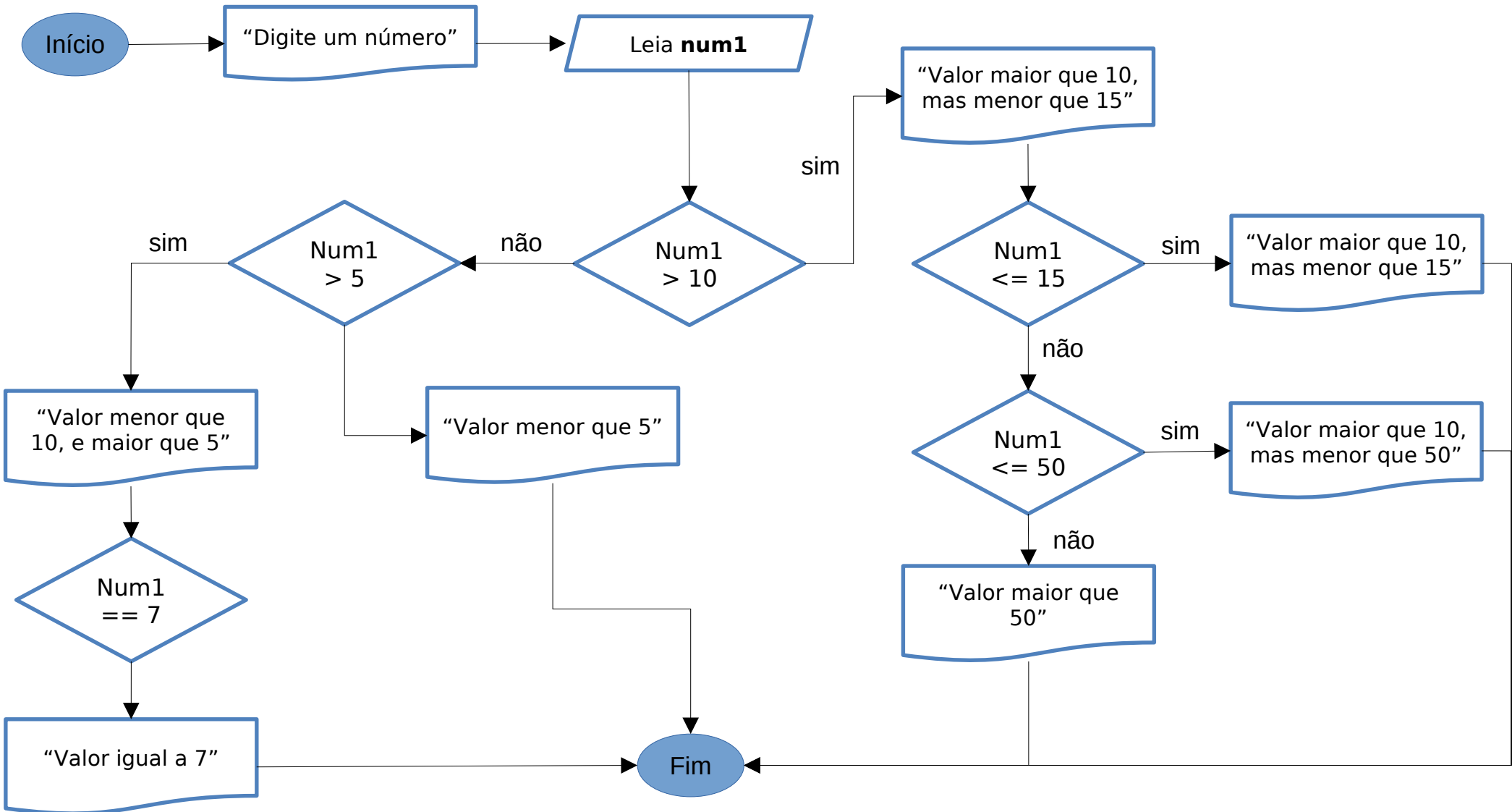
```
#bloco 1
#bloco 1
#bloco 1
    #bloco 2
    #bloco 2
    #bloco 2
        #bloco 3
        #bloco 3
        #bloco 3
        #bloco 3
    #bloco 4
    #bloco 4
    #bloco 4
    #bloco 4
#bloco 1
#bloco 1
#bloco 1
```

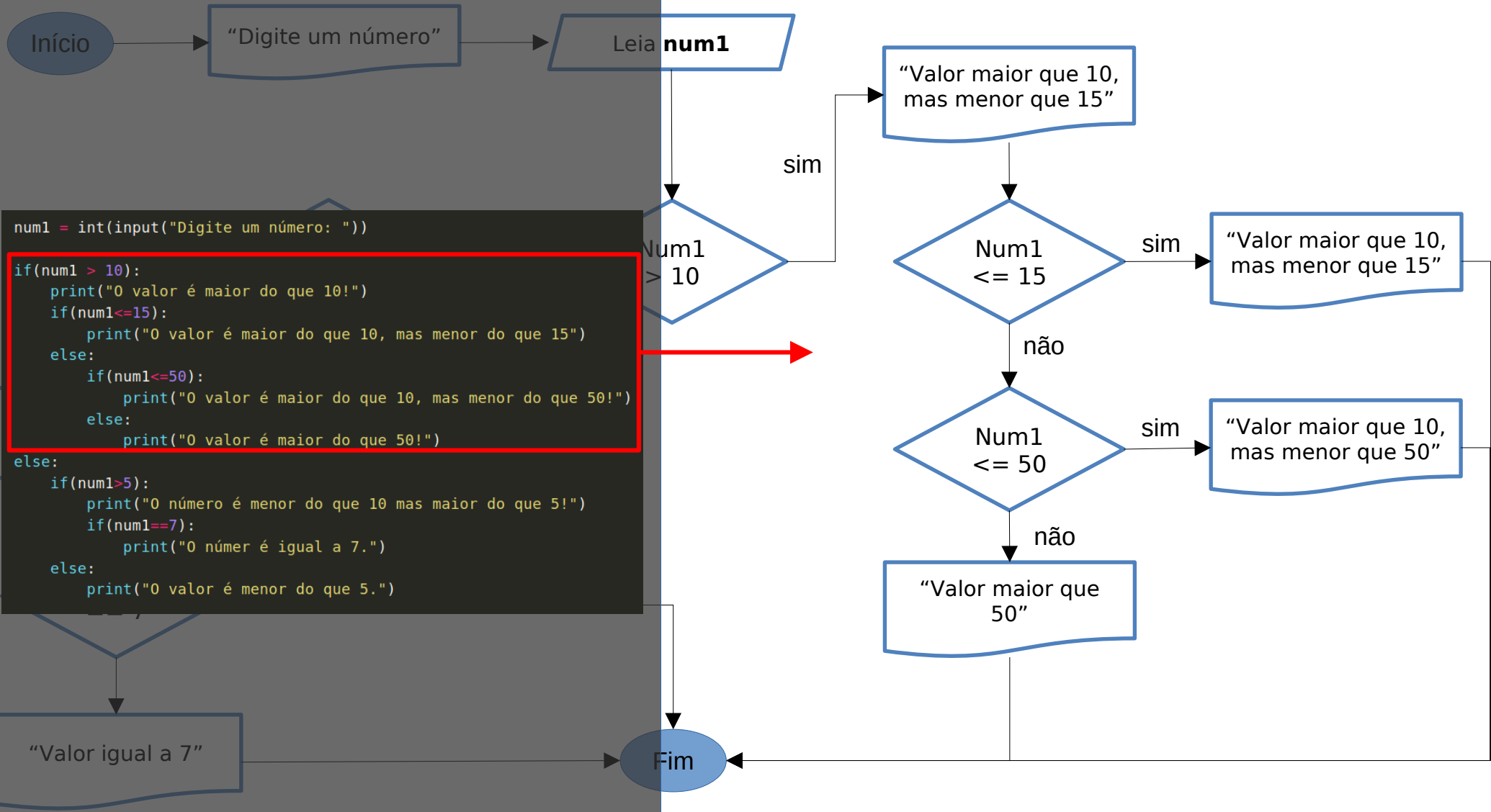
# Blocos de instrução

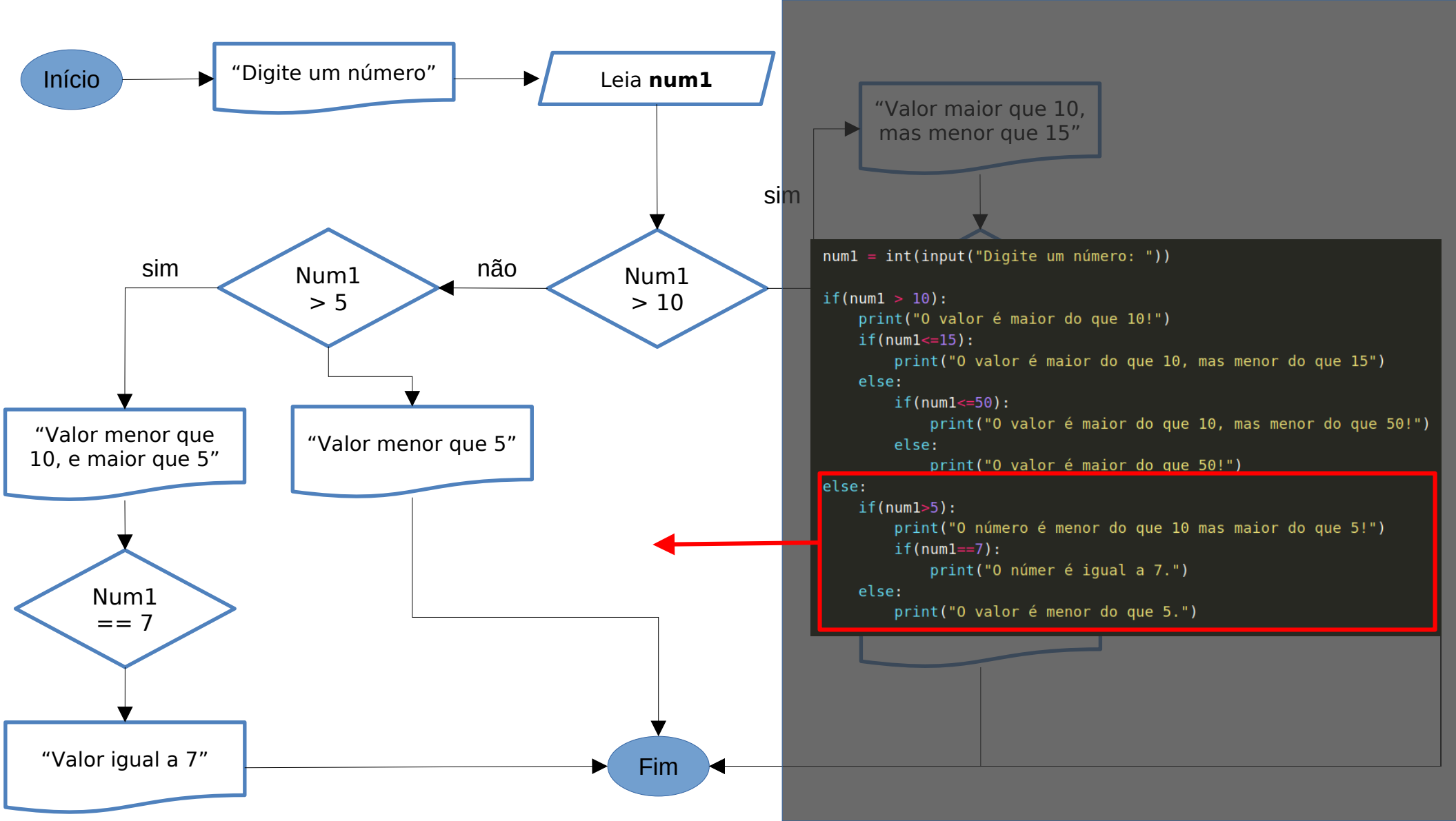
**Atividade:** Monte o fluxograma

```
num1 = int(input("Digite um número: "))

if(num1 > 10):
    print("O valor é maior do que 10!")
    if(num1<=15):
        print("O valor é maior do que 10, mas menor do que 15")
    else:
        if(num1<=50):
            print("O valor é maior do que 10, mas menor do que 50!")
        else:
            print("O valor é maior do que 50!")
else:
    if(num1>5):
        print("O número é menor do que 10 mas maior do que 5!")
        if(num1==7):
            print("O número é igual a 7.")
    else:
        print("O valor é menor do que 5.")
```









# Estrutura de repetição

A blue hexagonal button with a white border and the word 'Repetição' in white text.

Repetição

# Fluxograma

## Símbolos

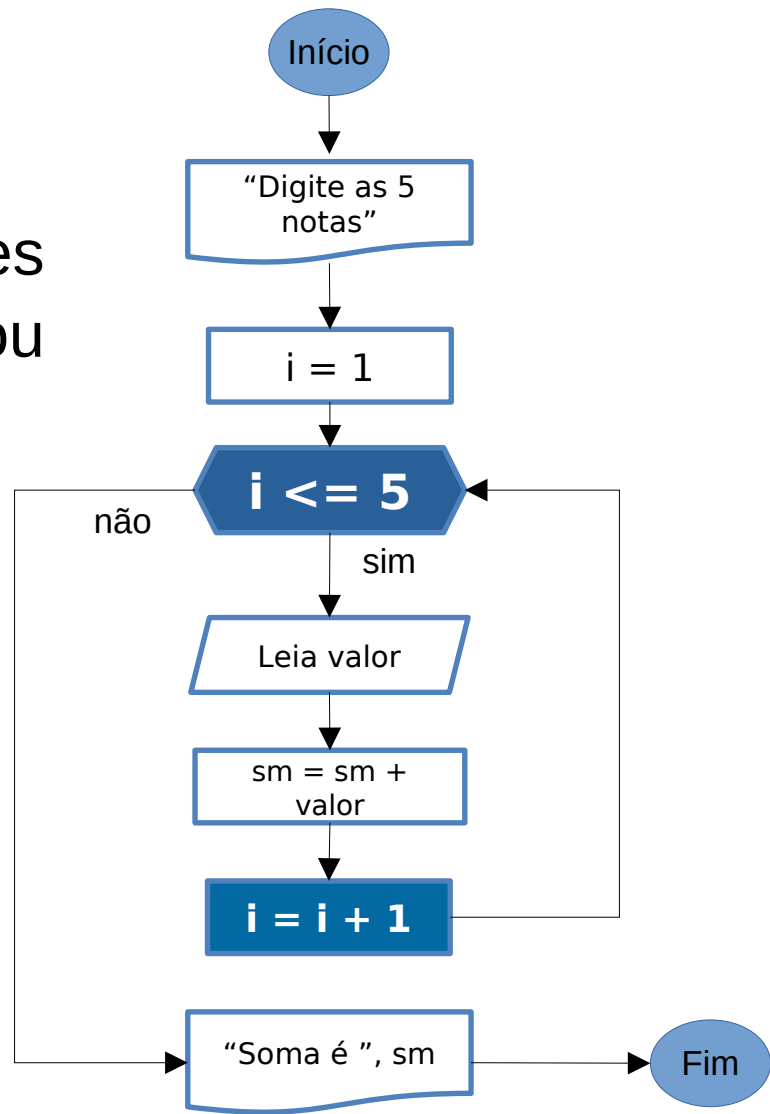


# Estrutura de repetição

Repetição de um conjunto de instruções por uma quantidade **finita** de vezes ou enquanto uma condição seja aceita.

Também conhecida como:

- Iteração;
- Looping;
- Laço de repetição;
- Laços condicionais;
- Repetição.



# Estrutura de repetição

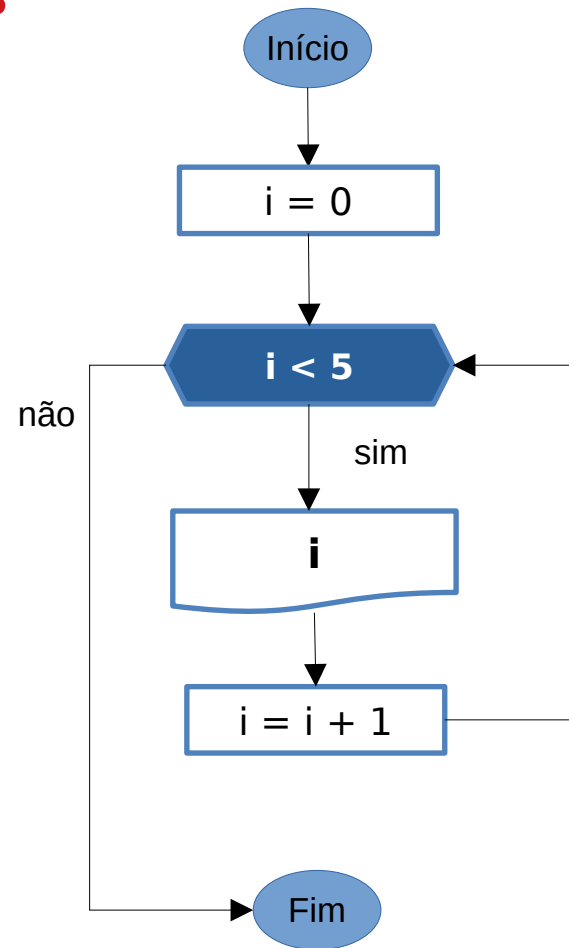
## Tipos

### For

- Define **explicitamente** a quantidade de vezes [ciclos] que será executado o bloco de instruções;

### While

- Repete um bloco de instrução **enquanto** a condição definida no seu cabeçalho for verdadeira.



# Estrutura de repetição

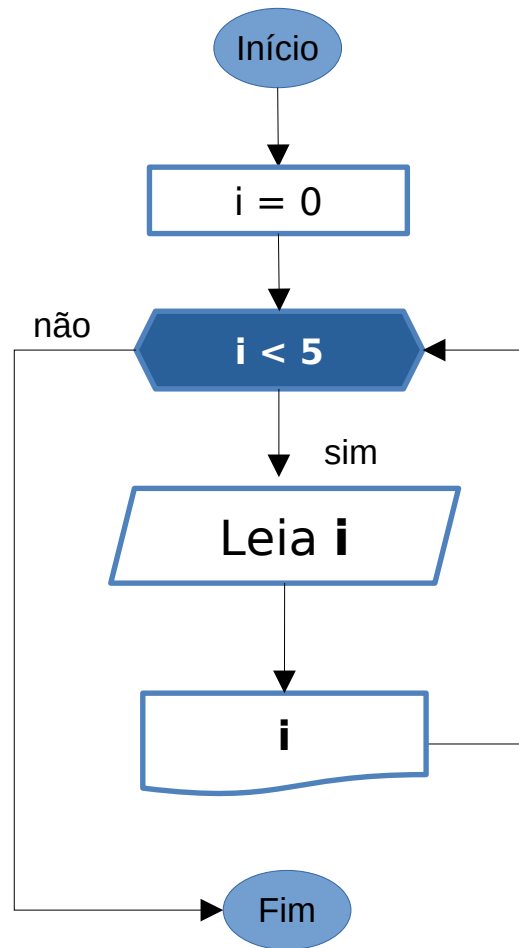
## Tipos

### For

- Define **explicitamente** a quantidade de vezes [ciclos] que será executado o bloco de instruções;

### While

- Repete um bloco de instrução **enquanto** a condição definida no seu cabeçalho for verdadeira.



# Estrutura de repetição

## Diferenças

### For

- Algoritmo que **recebe cinco números inteiros** e exibe a soma dos números informados

### While

- Algoritmo que recebe **números positivos** e faça a soma dos número informados. **Enquanto números positivos for informados**, o algoritmo deve realizar a soma, senão o programa deve encerrar sua execução.

# Estrutura de repetição

## Diferenças

### For

- Algoritmo que faz a **iteração** de uma variável (**x**) **cinco vezes**.

Sintaxe e algoritmo:

```
PARA x FAÇA ATÉ valor  
INÍCIO  
    // instruções  
  
    x = x + 1  
FIM
```

```
1 for x in range(5):  
2 |     print(x)  
3
```

# Estrutura de repetição

## Atividade 01

Escreva um programa que exiba os números de 0 a 9 utilizando um laço de repetição.

Exemplo saída na tela



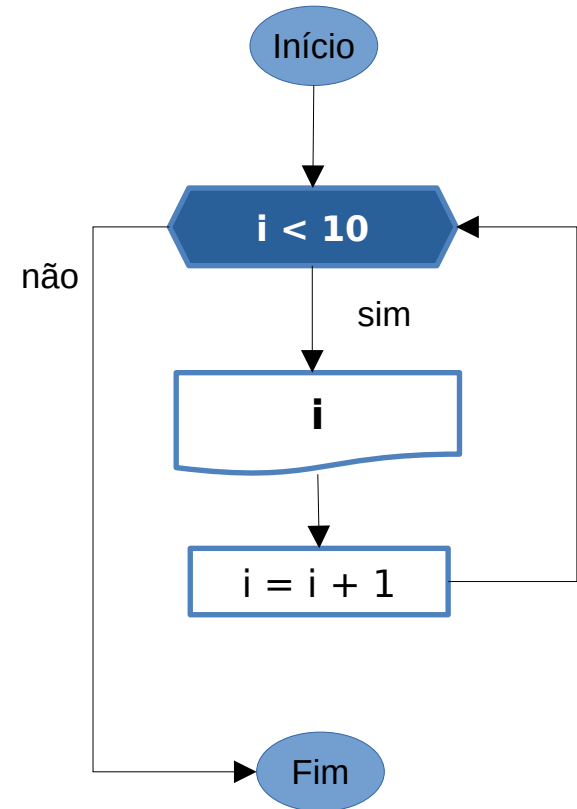
0  
1  
2  
3  
4  
5  
6  
7  
8  
9



# Estrutura de repetição

## Atividade 01: Solução

Escreva um programa que exiba os números de 0 a 9 utilizando um laço de repetição.



# Estrutura de repetição

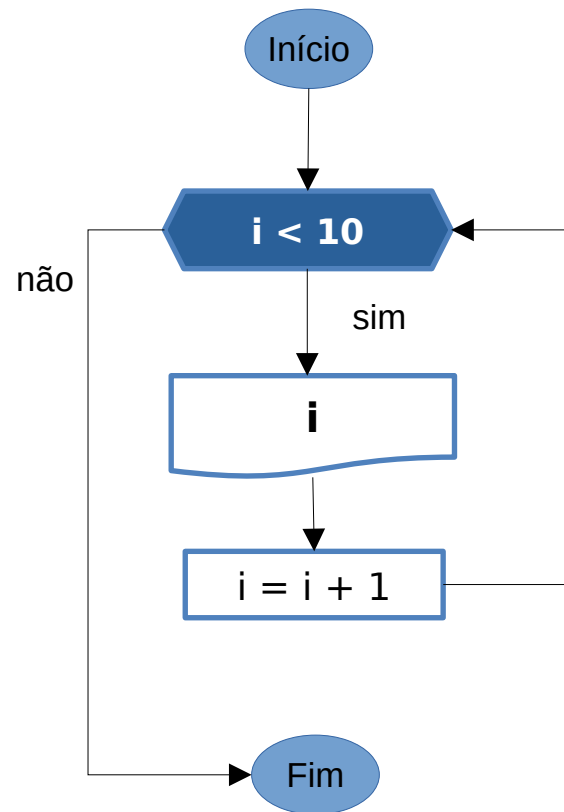
## Atividade 01: Solução

Escreva um programa que exiba os números de 0 a 9 utilizando um laço de repetição.

```
1 for x in range(10):  
2 |     print(x)  
3
```



0  
1  
2  
3  
4  
5  
6  
7  
8  
9



# Estrutura de repetição

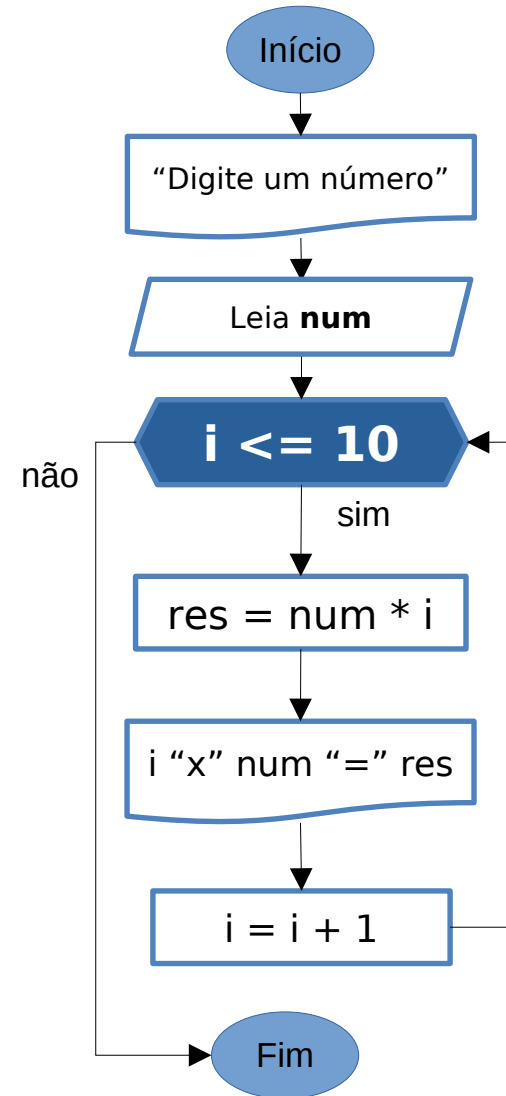
## Atividade 02

Escreva um programa que solicite ao usuário um número e exiba a tabuada desse número de 1 a 10.

# Estrutura de repetição

## Atividade 02

Escreva um programa que solicite ao usuário um número e exiba a tabuada desse número de 1 a 10.



# Estrutura de repetição

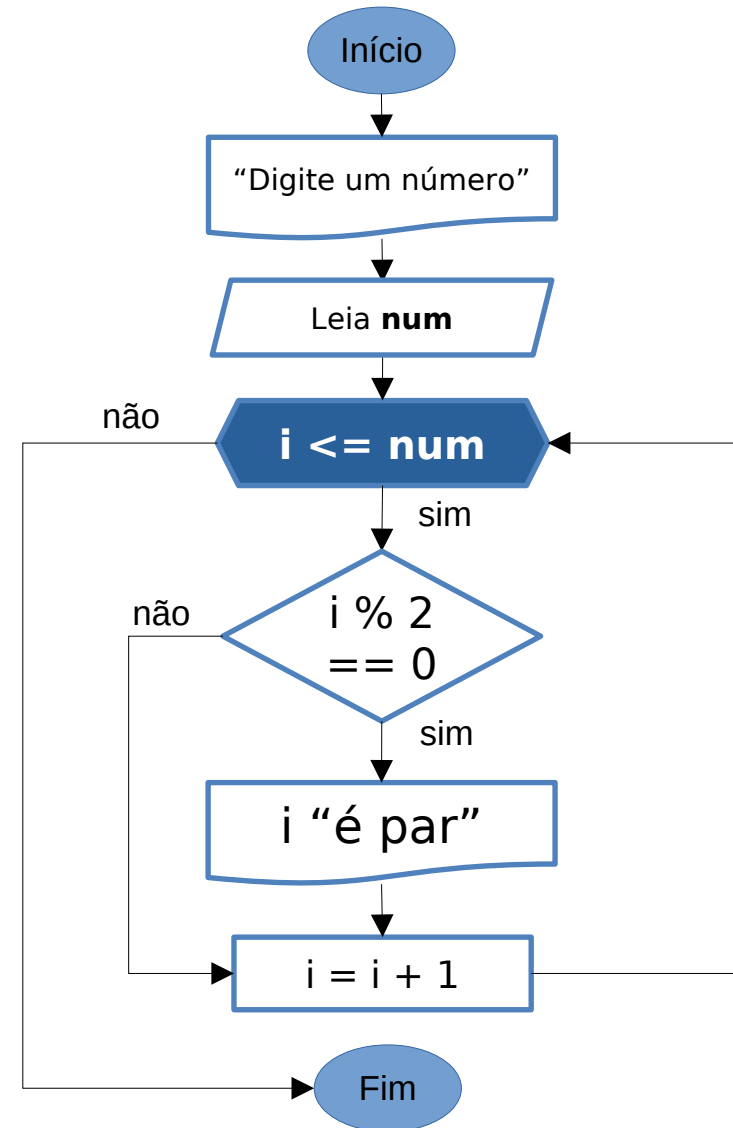
## Atividade 03

Escreva um programa que solicite ao usuário um número e exiba todos os números pares de 0 até esse número.

# Estrutura de repetição

## Atividade 03: Solução

Escreva um programa que solicite ao usuário um número e exiba todos os números pares de 0 até esse número.



# Estrutura de repetição

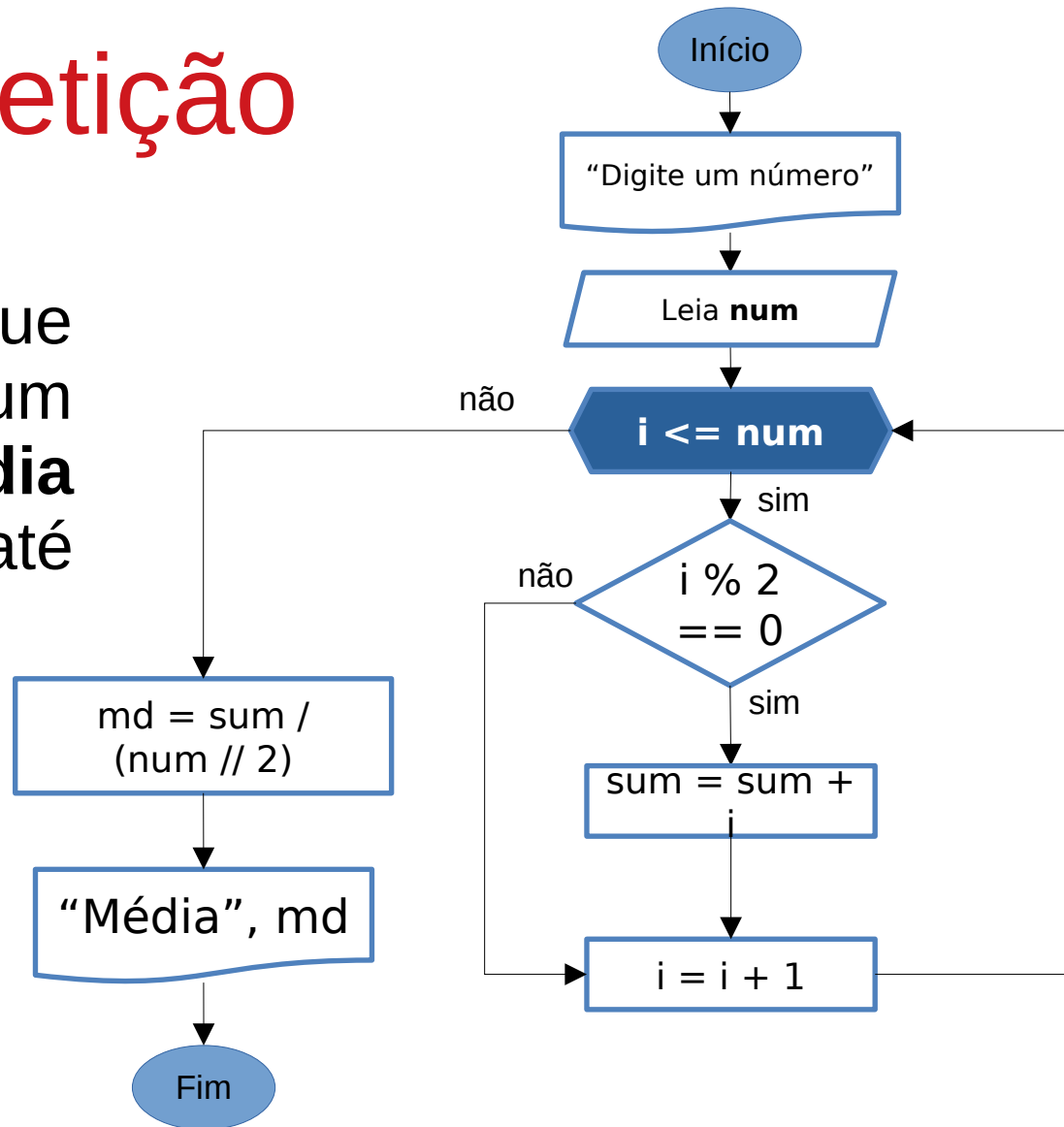
## Atividade 04

Escreva um programa que solicite ao usuário um número e calcule a média dos números pares de 1 até esse número.

# Estrutura de repetição

## Atividade 04: Solução

Escreva um programa que solicite ao usuário um número e calcule a **média dos números pares** de 1 até esse número.





# Estrutura de repetição

## Atividade 05

Escreva um programa que solicite ao usuário um número e verifique se ele é um número primo. Um número primo é aquele que é divisível apenas por 1 e por ele mesmo.

Dica: Número primo não pode ser divisível por 2 ou 3 ou 5.

# Estrutura de repetição

## função range()

# A função range()

Retorna uma série numérica no intervalo enviado como argumento

- A série retornada é um objeto iterável tipo range e os elementos contidos serão gerados sob demanda.
  - Start: início da sequência
  - Stop: último elemento da sequência
  - Step: intervalo entre os elementos.

```
#coding: utf-8
```

```
range( stop ) #primeira definição ou definição simplificada
```

```
range( [start], stop[, step] ) #segunda definição ou definição completa
```

# Estrutura de repetição

for .. in range()

## Sintaxe

PARA variável INCREMENTE ATÉ (valor):  
exibir(variável)

```
for i in range(6):  
    print(i)
```

# Estrutura de repetição

for .. in range()

Exemplo

```
val = 5
```

```
for i in range(10):  
    res = i * val  
    print(res)
```

```
0  
5  
10  
15  
20  
25  
30  
35  
40  
45
```

# Estrutura de repetição

range()

## Sintaxe da função range

```
range (start, end, step)
```

Obs: valor padrão do **start** é 0, e **step** é 1.

# Estrutura de repetição

range()

Exemplo: range(start, end, step)

```
for i in range(10):  
    print(i)
```

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

Obs: valor padrão do **start** é 0, e **step** é 1.

# Estrutura de repetição

range()

Exemplo: range(start, end, step)

```
for i in range(10, 20):  
    print(i)
```

10  
11  
12  
13  
14  
15  
16  
17  
18  
19

Obs: valor padrão do step é 1



# Estrutura de repetição

range()

Exemplo: range(start, end, step)

```
for i in range(5, 30, 3):  
    print(i)
```

5  
8  
11  
14  
17  
20  
23  
26  
29

# Estrutura de repetição

range()

Exemplo: range(start, end, step)

```
for i in range(-30, 50, 10):  
    print(i)
```

-30

-20

-10

0

10

20

30

40

# Estrutura de repetição

## Atividade 06

Faça um algoritmo que exiba todos os números pares de 2 até 100, utilizando o for com range(início, fim, passo)

# Estrutura de repetição

## Atividade 06

Crie um algoritmo que mostre uma contagem regressiva de 10 até 1, utilizando range com valores decrescentes.

# Estrutura de repetição

## Atividade 07

Faça um algoritmo que calcule e exiba a soma de todos os múltiplos de 3 entre 1 e 50.

# Estrutura de repetição

## Atividade 09

Escreva um algoritmo que exiba todos os números ímpares de 99 até 1, utilizando for com contagem regressiva.

# Estrutura de repetição

## Atividade 10

Faça um algoritmo que leia um número base e um número limite, e exiba as potências da base começando de 1 até o limite.

Entrada:

Base: 2

Limite: 5

Saída:

$$2^1 = 2$$

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^5 = 32$$