



# **INFORMÁTICA PARA ENGENHARIA**

## **ESTRUTURA DE REPETIÇÃO**

Prof. Dr. Daniel Caetano

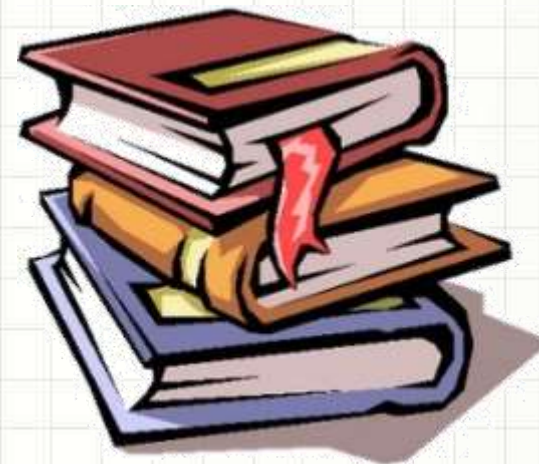
2019 - 1

# Objetivos

- Entender o que é uma estrutura de repetição
- Compreender como implementar as repetições
- Capacitar para a criação de algoritmos que envolvam repetição
- **Atividades Aula 11 – SAVA!**



# Material de Estudo



## Material

## Acesso ao Material

Notas de Aula e  
Apresentação

<http://www.caetano.eng.br/>  
(Informática para Engenharia – Aula 11)

Material Didático

Lógica de Programação, págs 119 a 149.

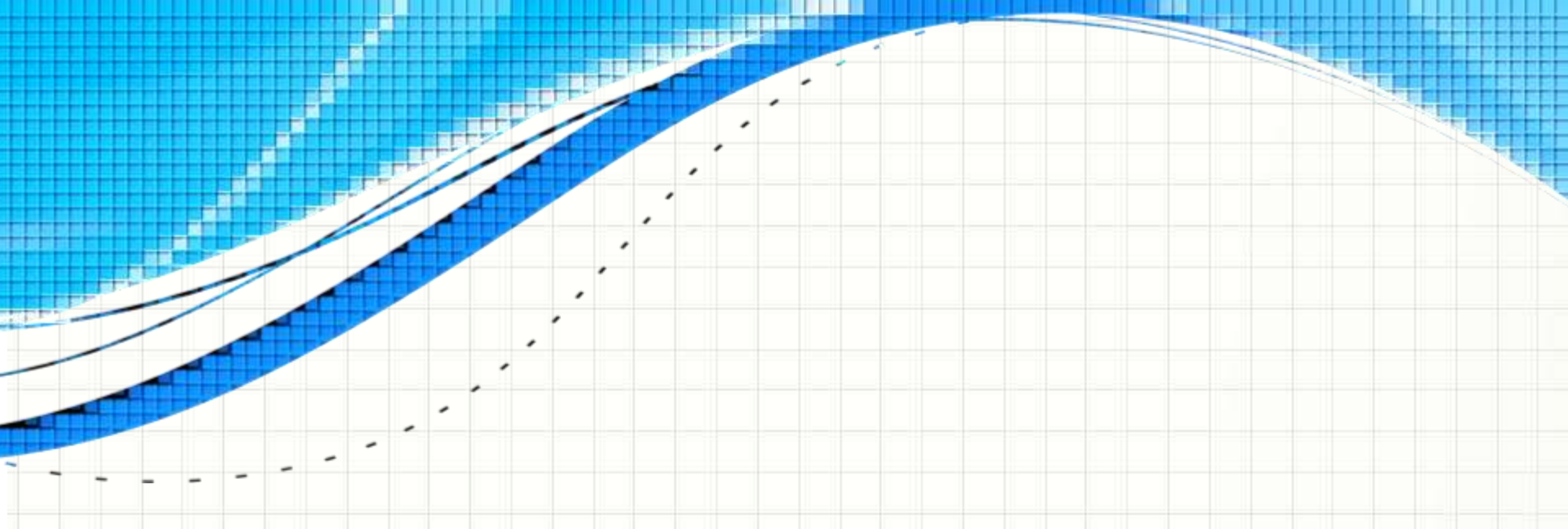
Biblioteca Virtual

“Lógica de Programação – Fundamentos da  
Programação de Computadores”, págs 93 a 144.

**LEMBRETE: CONSULTAR O “DEPOIS” DA AULA 11 NO SAVA!**

**LEMBRETE: CONSULTAR O “ANTES” DA AULA 12 NO SAVA!**





# O QUE É ESTRUTURA DE REPETIÇÃO?

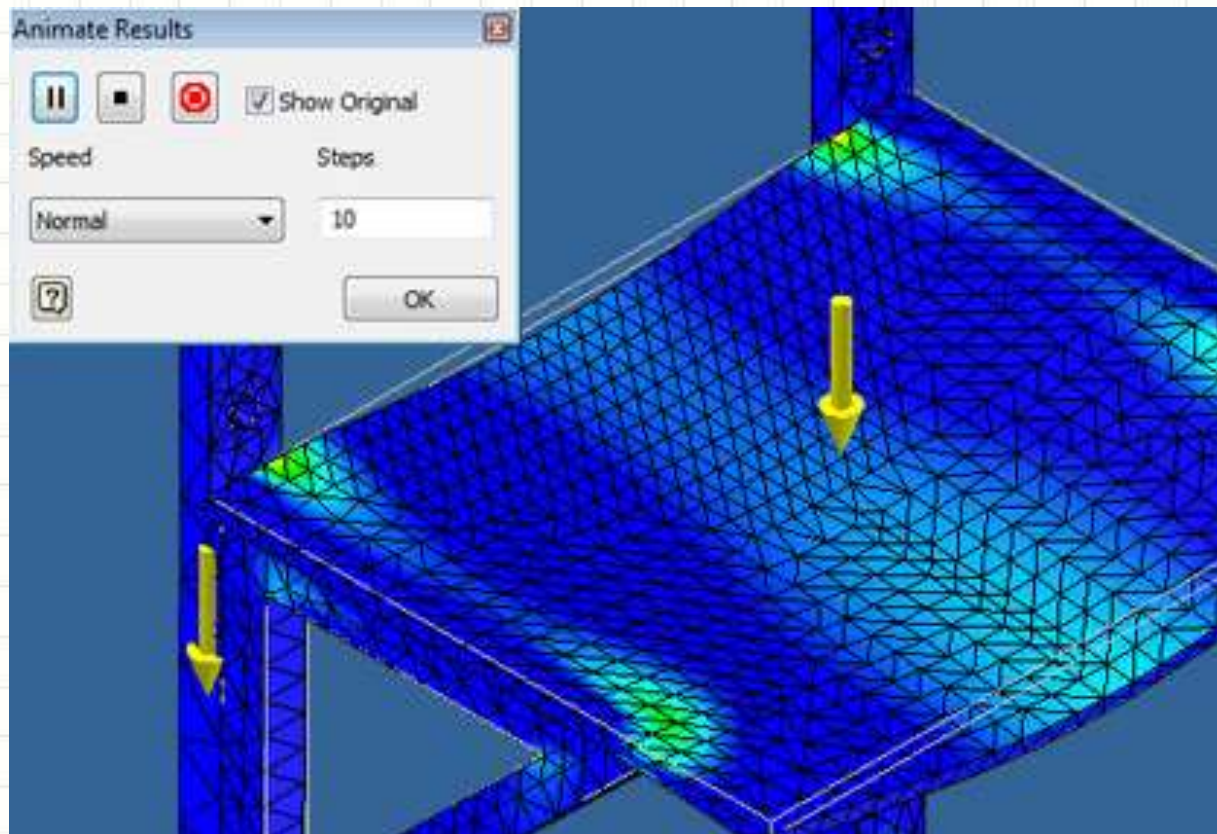
# O que são Estruturas de Repetição?

- Repetir continuamente um código
  - Solicitação de entradas do usuário



# O que são Estruturas de Repetição?

- Repetir continuamente um código
  - Procedimentos repetitivos





# O que são Estruturas de Repetição?

- Repetir continuamente um código
  - Esperar que alguma coisa ocorra



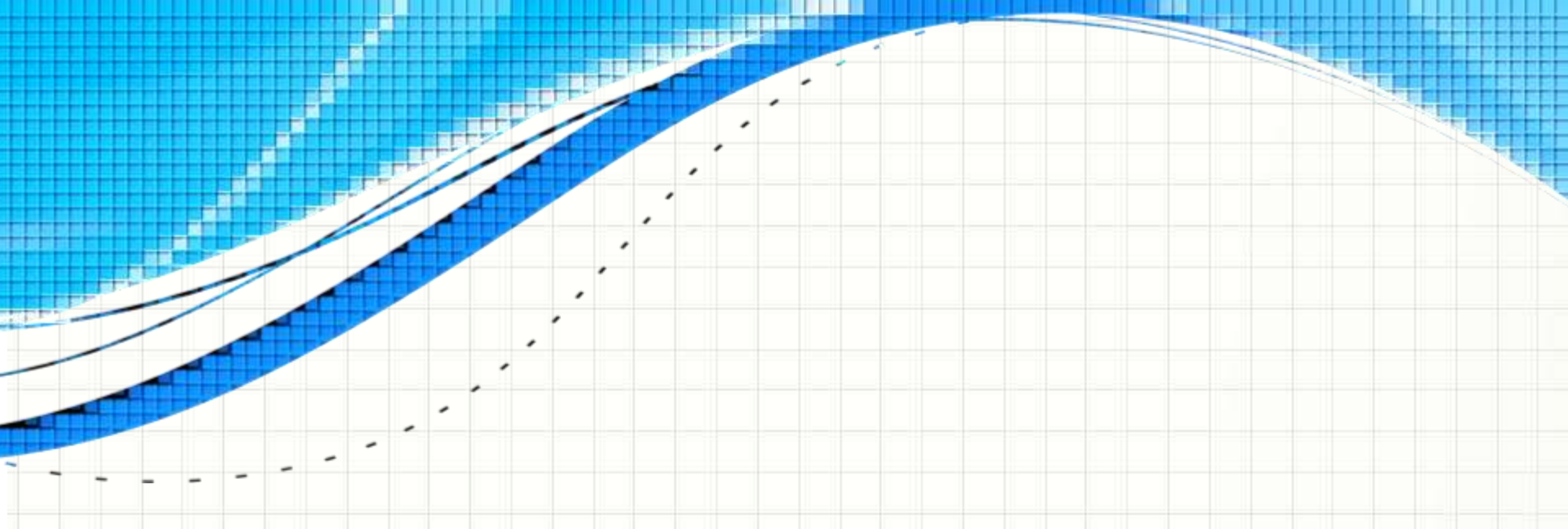
# O que são Estruturas de Repetição?

- Repetir continuamente um código
  - Esperar que alguma coisa ocorra



**MUITO  
USADAS!**





# COMPREENDENDO A REPETIÇÃO

# Compreendendo a Repetição

- Imprimindo números de 1 a 3

aula11ex01.py

```
print(1)
print(2)
print(3)
```

**Nada se repete**

aula11ex01a.py

```
x = 1
print(x) ←
x = 2
print(x) ←
x = 3
print(x) ←
```

**Hm... algo se repete**

# Compreendendo a Repetição

- Imprimindo números de 1 a 3

aula11ex01a.py

```
x = 1  
print(x) ←  
x = 2  
print(x) ←  
x = 3  
print(x) ←
```

**Algo se repete**

aula11ex01b.py

```
x = 1  
print(x)  
x = x + 1  
print(x)  
x = x + 1  
print(x)  
x = x + 1
```

**Algo mais se repete**



# Compreendendo a Repetição

- Imprimindo números de 1 a 3

aula11ex01a.py

```
x = 1  
print(x) ←  
x = 2  
print(x) ←  
x = 3  
print(x) ←
```

**Algo se repete**

aula11ex01b.py

```
x = 1  
print(x)  
x = x + 1  
print(x)  
x = x + 1  
print(x)  
x = x + 1
```

**Algo mais se repete**

# Compreendendo a Repetição

- Imprimindo números de 1 a 3

aula11ex01b.py

```
x = 1  
print(x)  
x = x + 1  
print(x)  
x = x + 1  
print(x)  
x = x + 1
```

**Algo mais se repete**

aula11ex01c.py

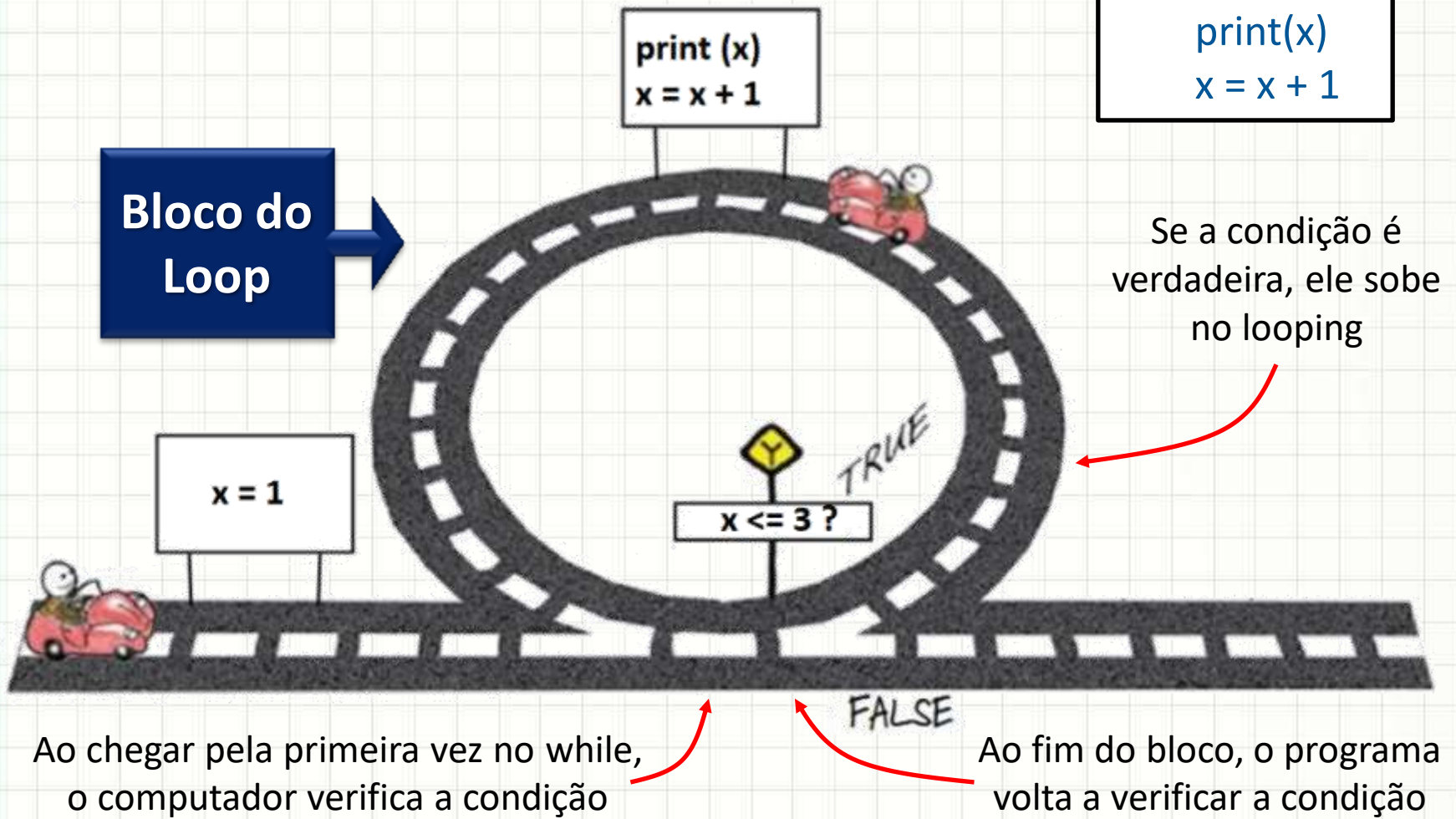
```
x = 1  
while x <= 3:  
    print(x)  
    x = x + 1
```

**Ah!**

# Compreendendo a Repetição

- Imprimindo números de 1 a 3

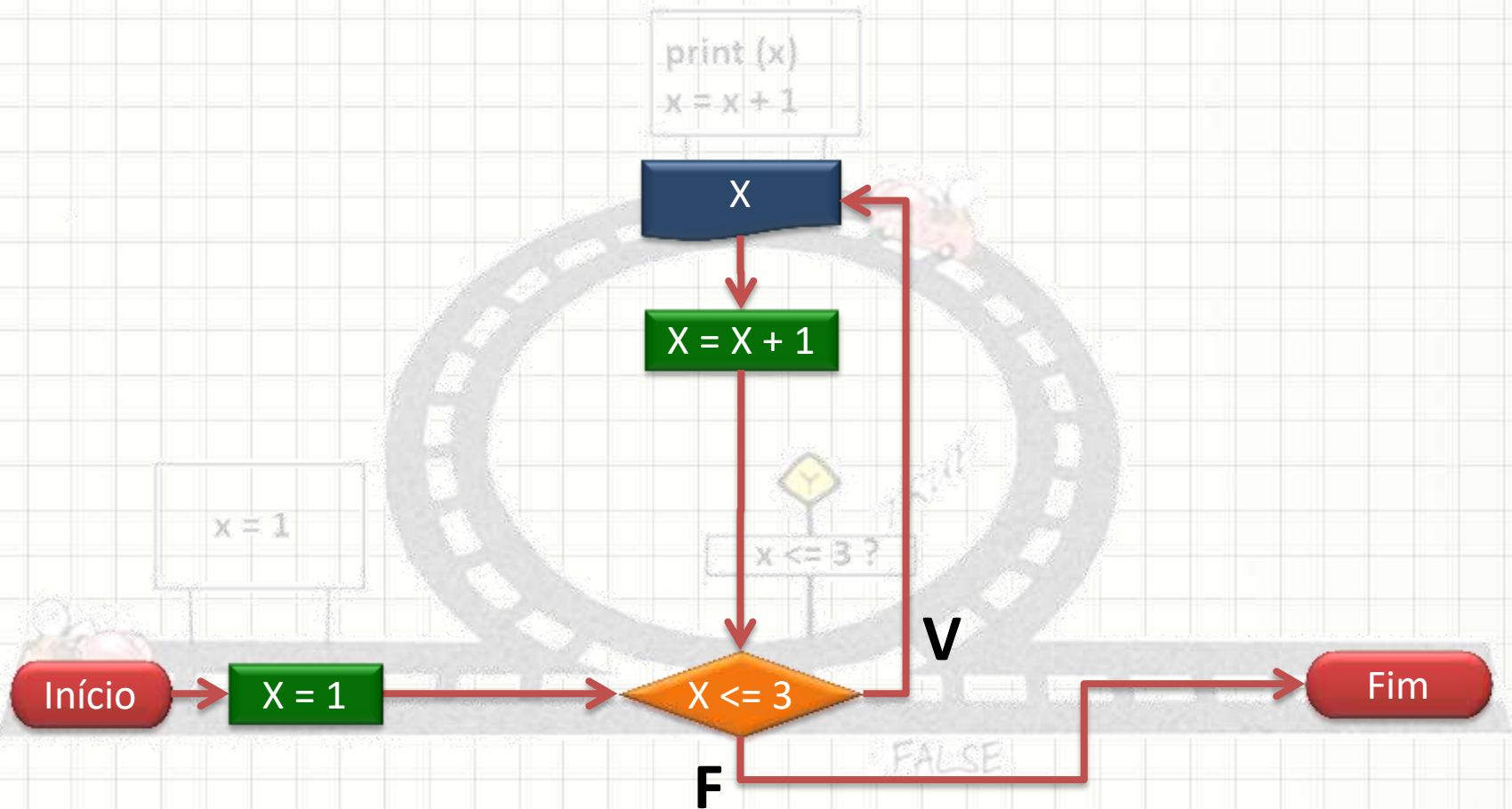
```
x = 1
while x <= 3:
    print(x)
    x = x + 1
```





# Compreendendo a Repetição

- Imprime os números inteiros de 1 a 3

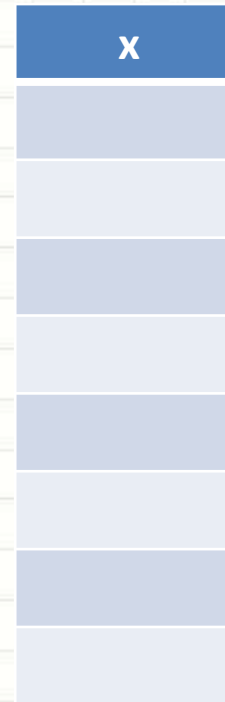


# Compreendendo a Repetição

- Imprimir números de 1 a 3 – Teste de Mesa?

aula11ex01c.py

```
x = 1
while x <= 3:
    print(x)
    x = x + 1
```

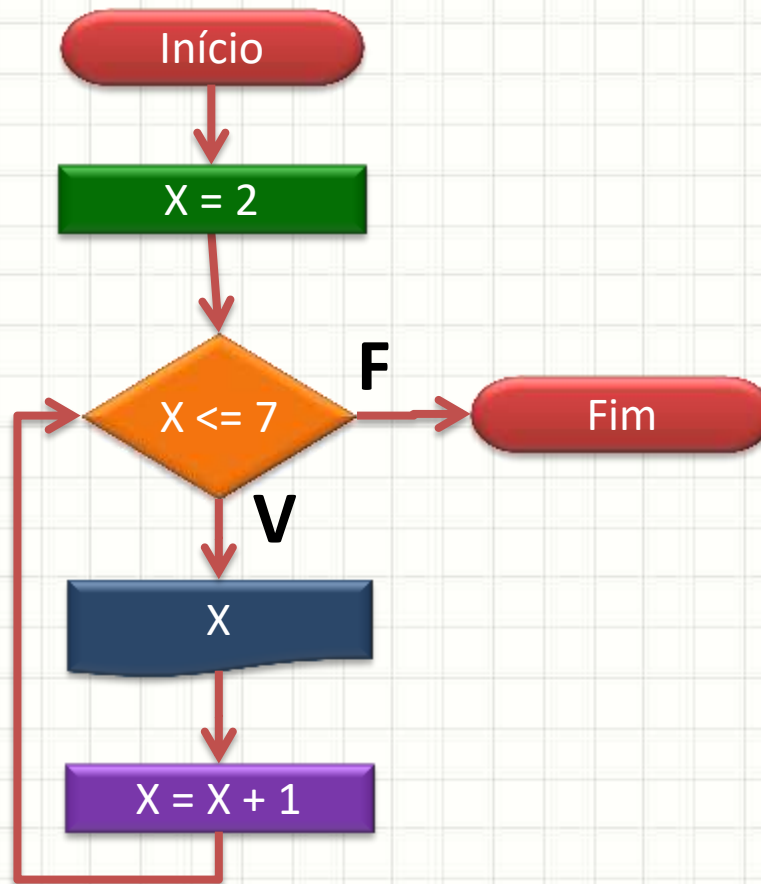


Tela:



# Exemplo

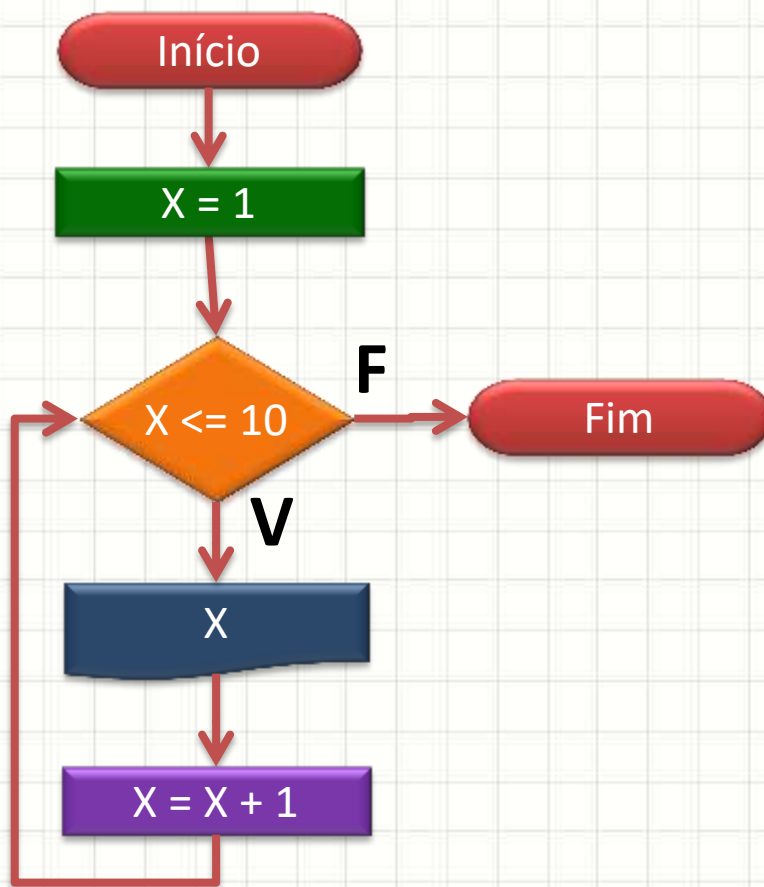
- Crie um algoritmo que imprime os números inteiros de 2 a 7





# Exemplo

- Crie um algoritmo que imprime os números inteiros de 2 a 7



## Algoritmo "Imprime de 1 a 10"

**Início**

inteiro X

$X \leftarrow 1$

enquanto  $X \leq 10$  :

    escreva(X)

$X \leftarrow X + 1$

**Fim**

---

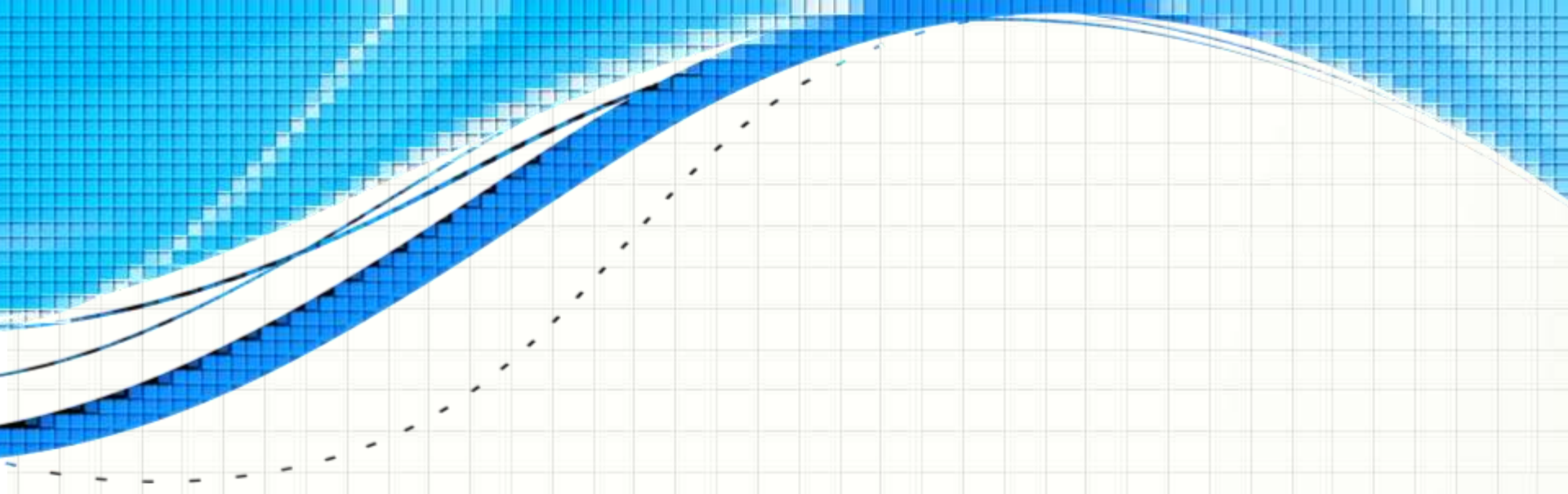
**# Imprime de 1 a 10**

$X = 1$

while  $X \leq 10$  :

    print(X)

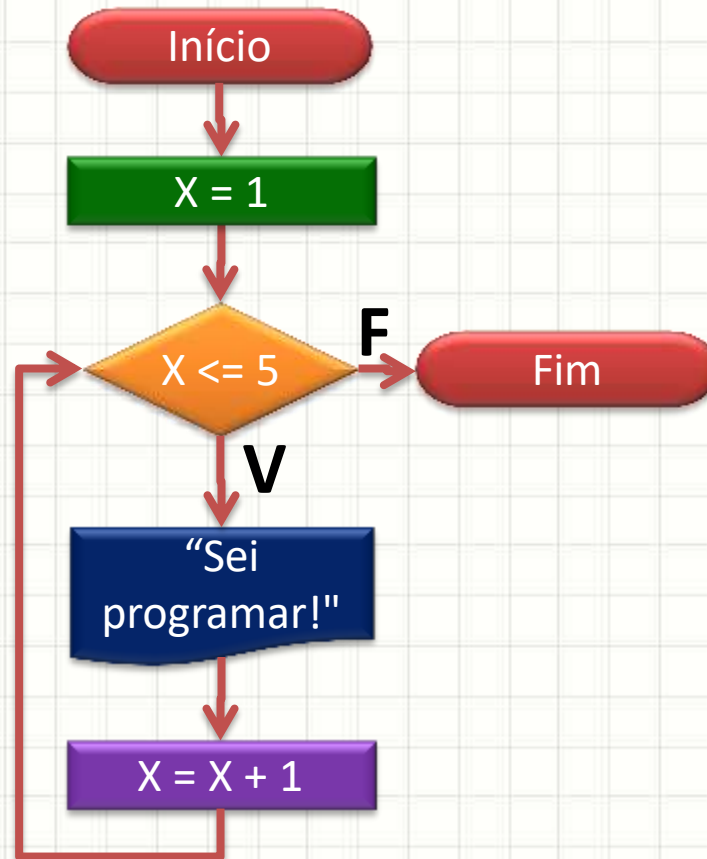
$X = X + 1$



**ATIVIDADE**

# Exercício 1

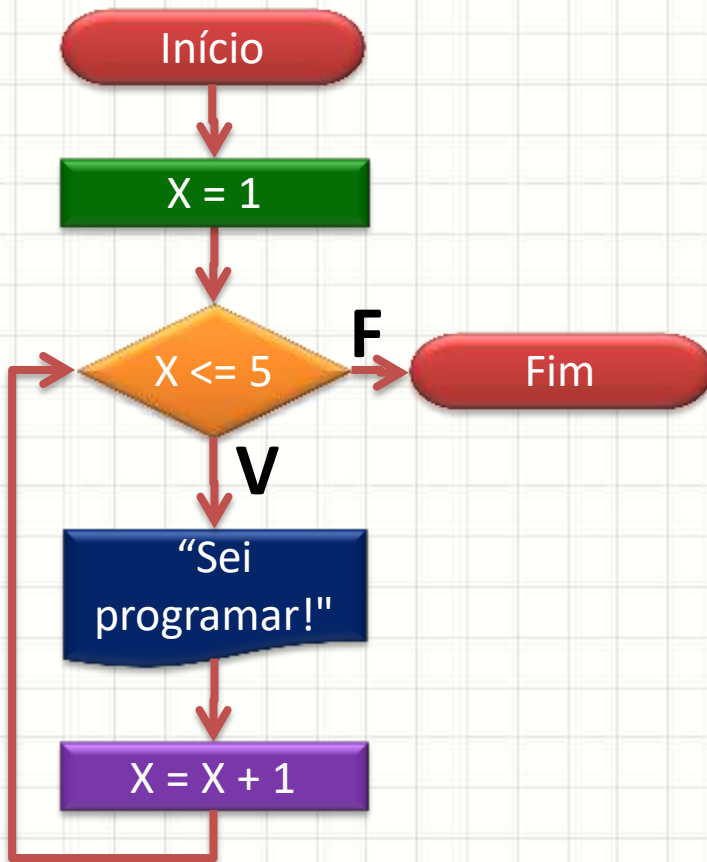
- Faça um programa que imprima 5 vezes a mensagem “Sei programar!”





# Exercício 1

- Faça um programa que imprima 5 vezes a mensagem “Sei programar!”



**# Imprime 5x “Sei Programar”**

X = 1

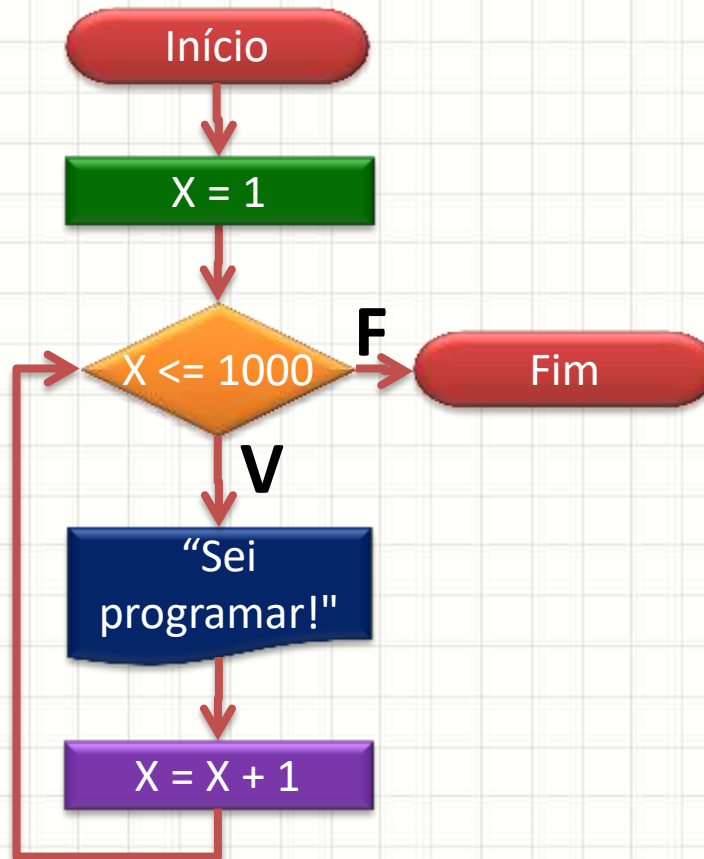
**while** X <= 5 :

**print**(“Sei programar!”)

X = X + 1

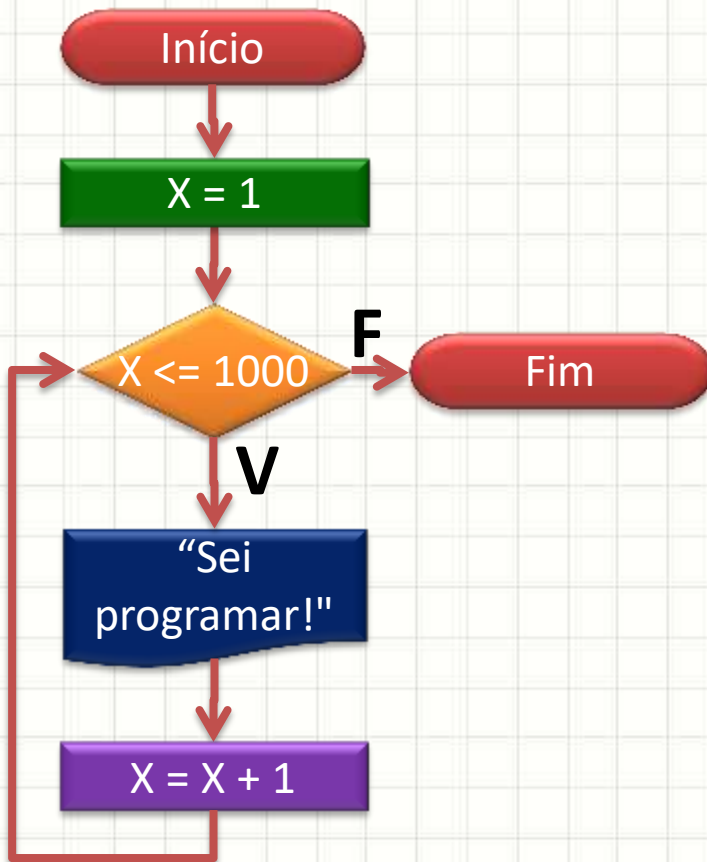
# Exercício 2

- Faça um programa que imprima 1000 vezes a mensagem “Sei programar!”



# Exercício 2

- Faça um programa que imprima 1000 vezes a mensagem “Sei programar!”



# Imprime 1000x“Sei Programar”

X = 1

while X <= 1000 :

print(“Sei programar!”)

X = X + 1

São iguais?

# Imprime 1000x“Sei Programar”

X = 0

while X < 1000 :

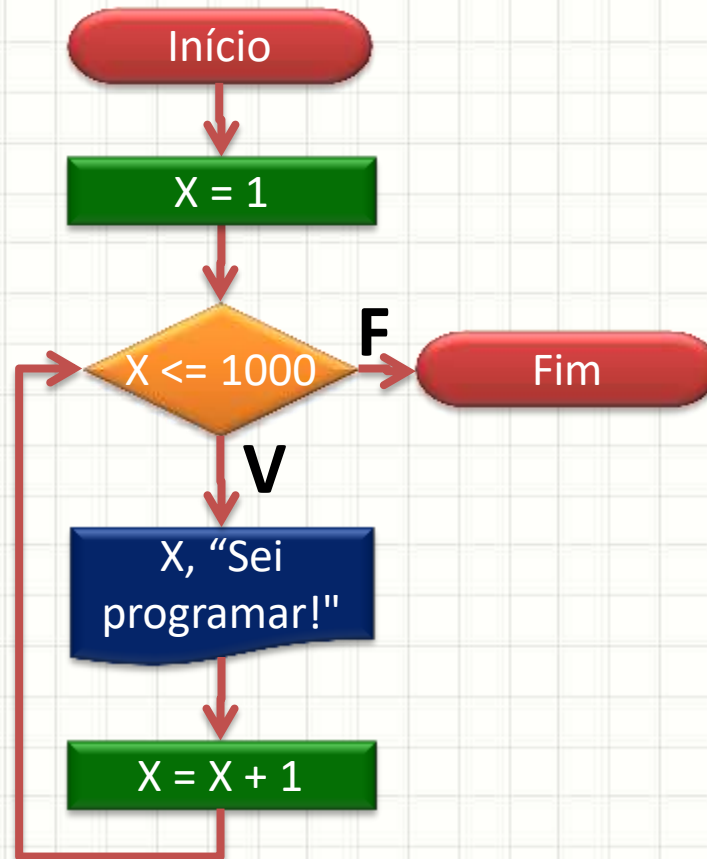
print(“Sei programar!”)

X = X + 1



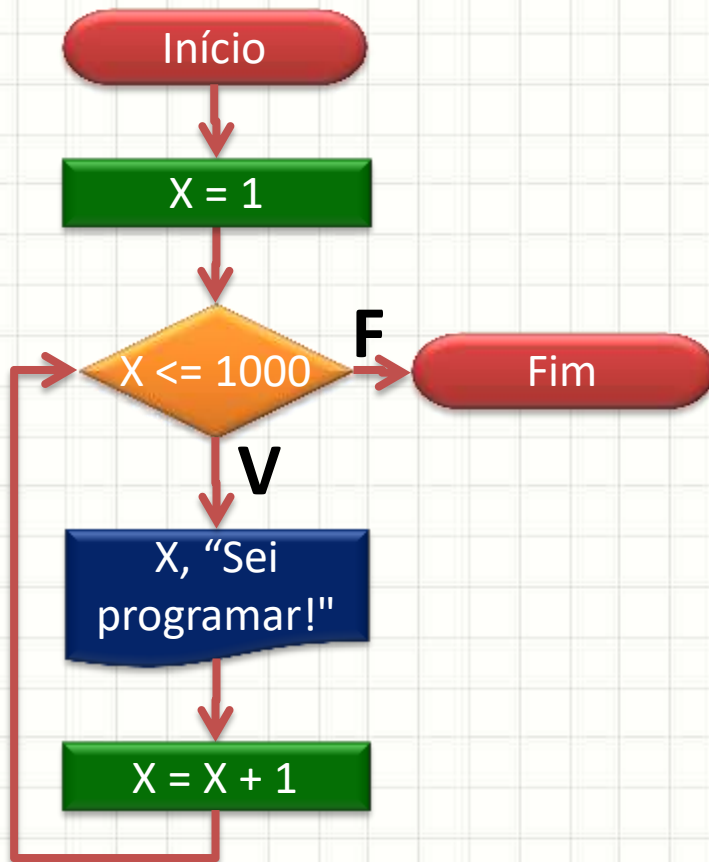
# Exercício 3

- Modifique o programa para imprimir o **contador** antes de imprimir o texto “Sei programar!”



# Exercício 3

- Modifique o programa para imprimir o contador antes de imprimir o texto “Sei programar!”



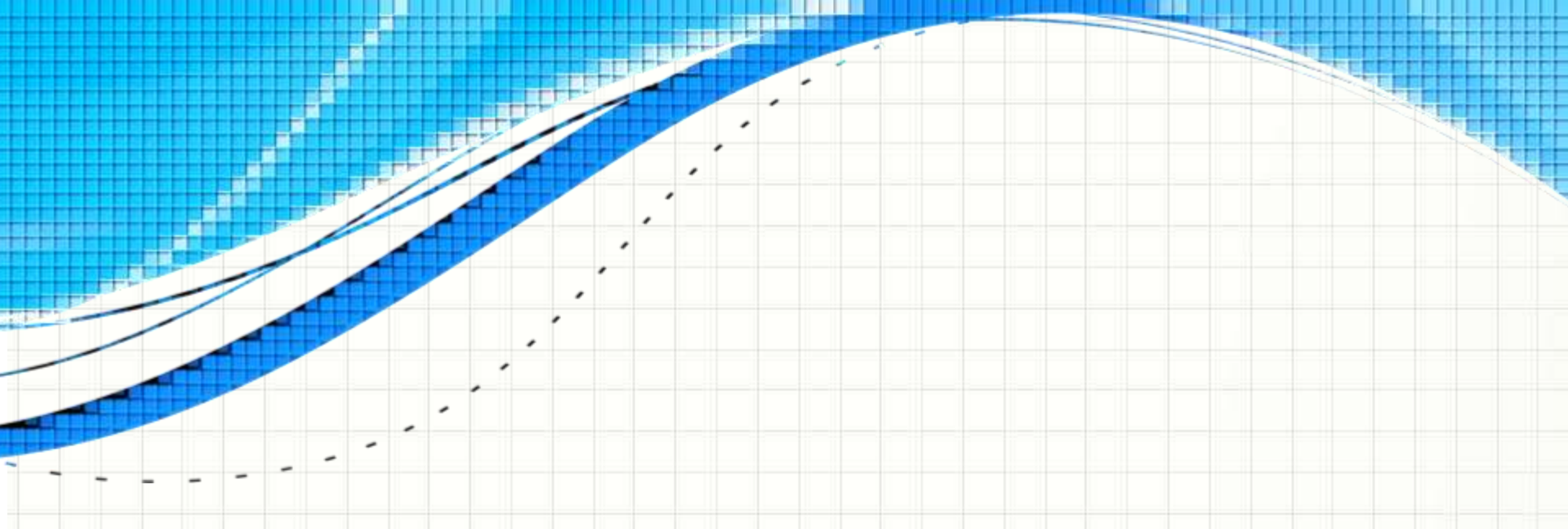
**# Imprime 1000x “Sei Programar”**

$X = 1$

**while**  $X \leq 1000$  :

**print**( $X$ , “Sei programar!”)

$X = X + 1$

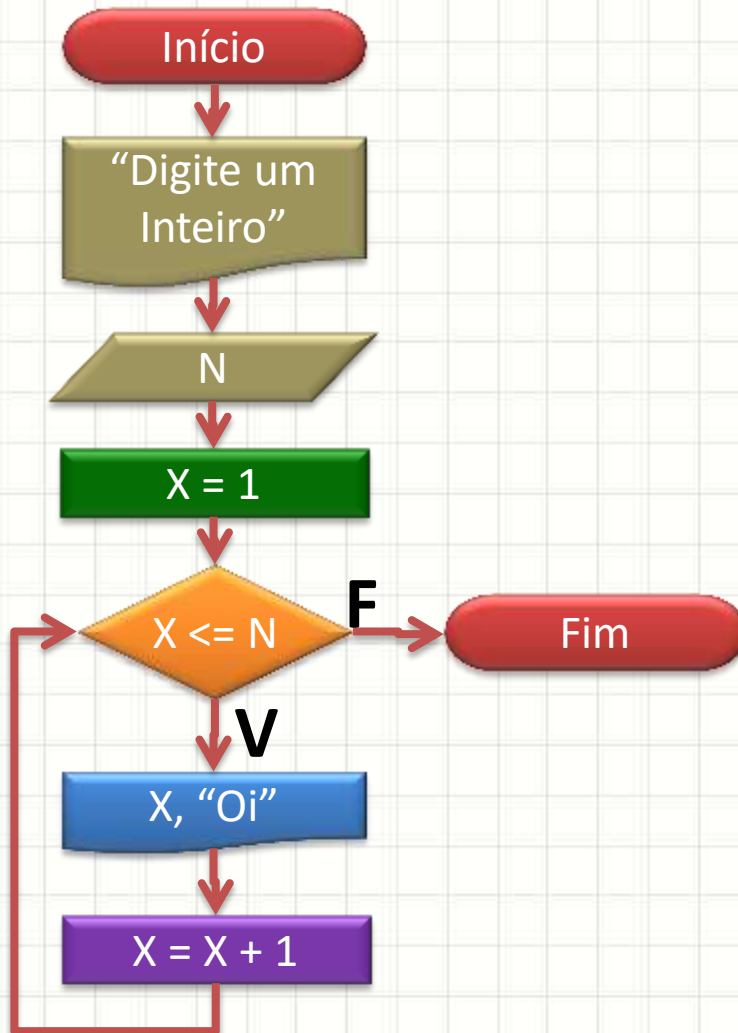


**REPETINDO  
CÓDIGO N VEZES**



# Repetindo Código N Vezes

- Observe:



**# Imprime Nx "Oi"**

```
N = int(input("Digite um inteiro:"))
```

```
X = 1
```

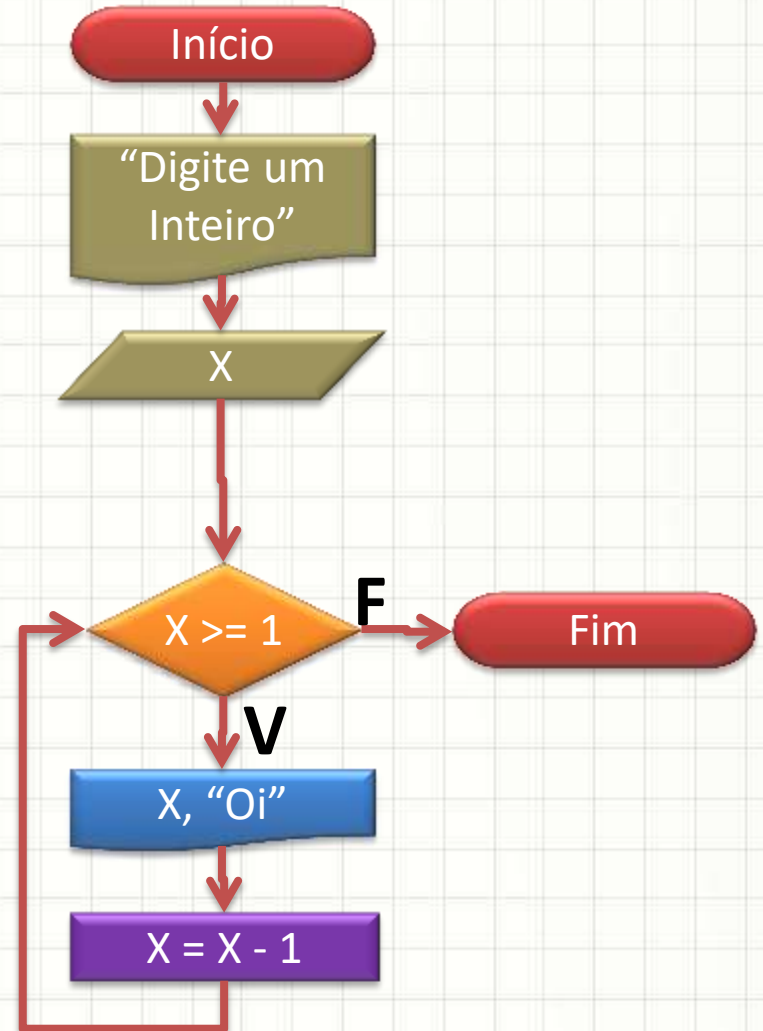
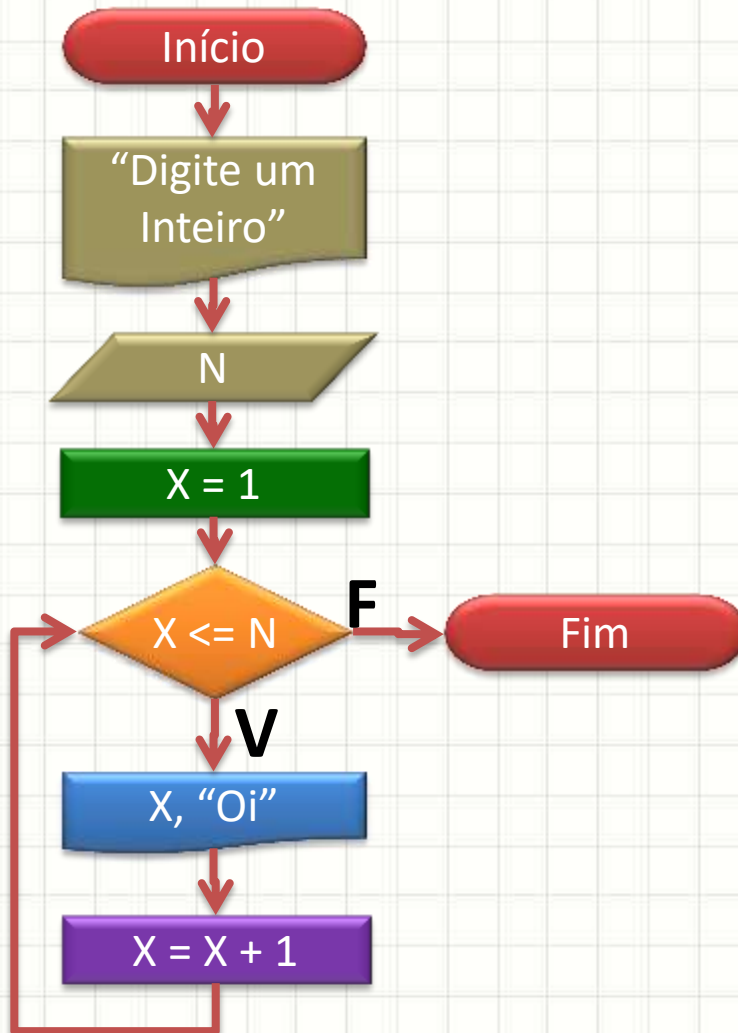
```
while X <= N :
```

```
    print(X, "Oi")
```

```
    X = X + 1
```

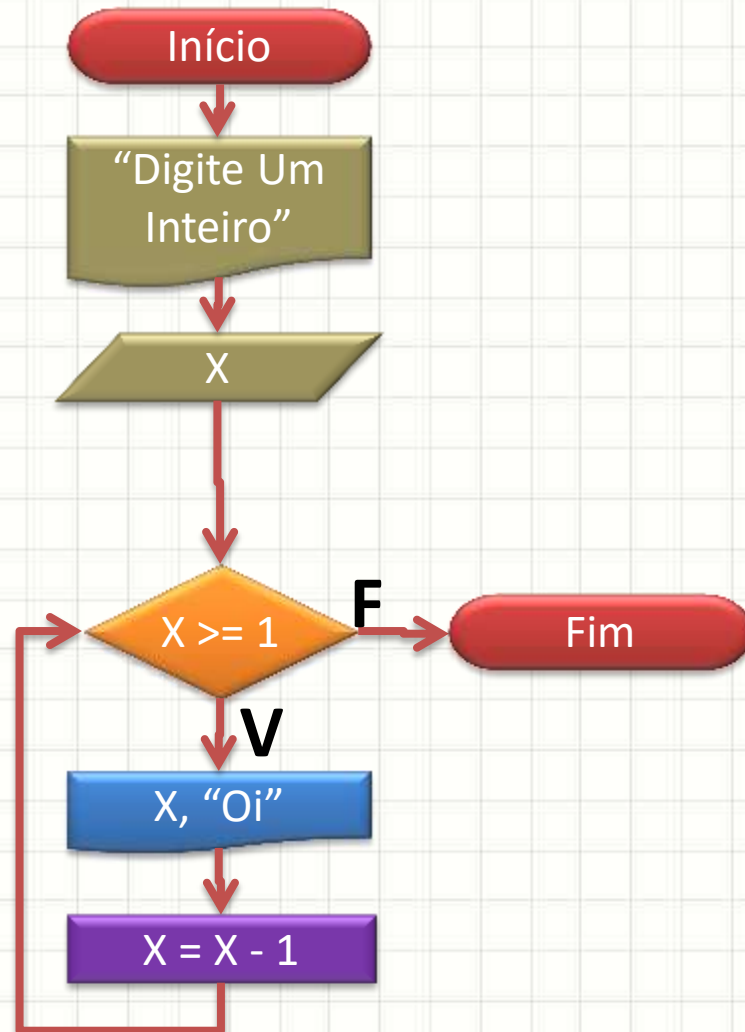
# Repetindo Código N Vezes

- O que muda de um caso para outro?



# Repetindo Código N Vezes

- Contando de trás pra frente:



```
# Imprime Nx "Oi"
```

```
X = int(input("Digite um inteiro:"))
```

```
while X >= 1 :
```

```
    print(X, "Oi")
```

```
    X = X - 1
```



# Repetindo Código N Vezes

```
# Imprime Nx "Oi"
```

```
X = int(input("Digite um inteiro:"))
```

```
while X >= 1 :
```

```
    print(X, "Oi")
```

```
    X = X - 1
```

1. Digite Este Programa
2. Experimente executá-lo com diferentes valores. Exemplo: 5, 1, 0, -10
3. Experimente modificar o **while** para que a condição seja **X >= 0** .
4. O que aconteceu / mudou em cada caso?

# Repetindo Código N Vezes

```
# Imprime Nx "Oi"
```

```
X = int(input("Digite um inteiro:"))
```

```
while X >= 1 :
```

```
    print(X, "Oi")
```

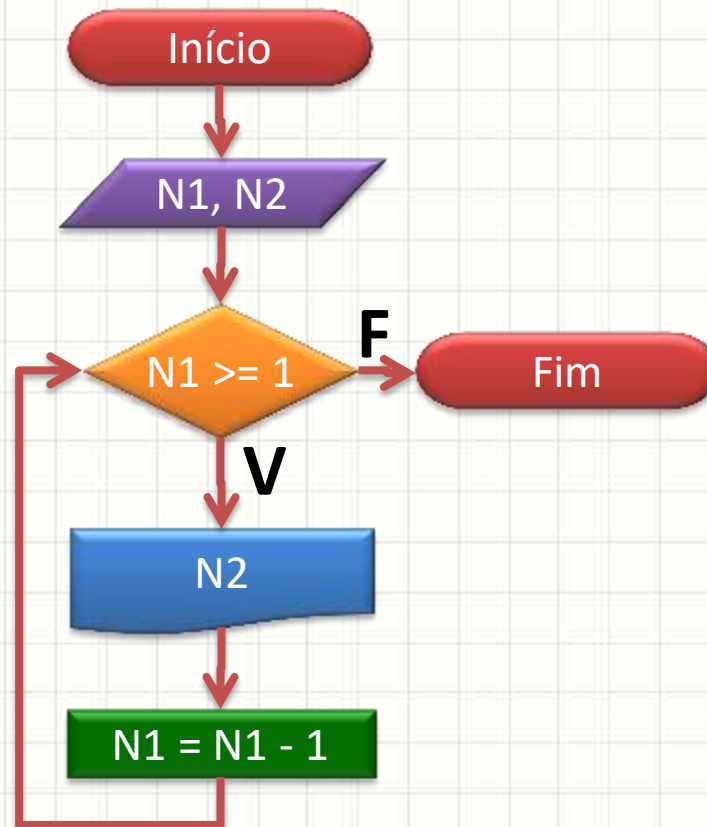
```
    X = X - 1
```

## ATENÇÃO

1. Cuidado ao construir as proposições de repetição!
2. É responsabilidade do programador garantir que a condição de finalização seja atendida!
  - 2.1. Experimente modificar a atualização para  **$X = X + 1$**

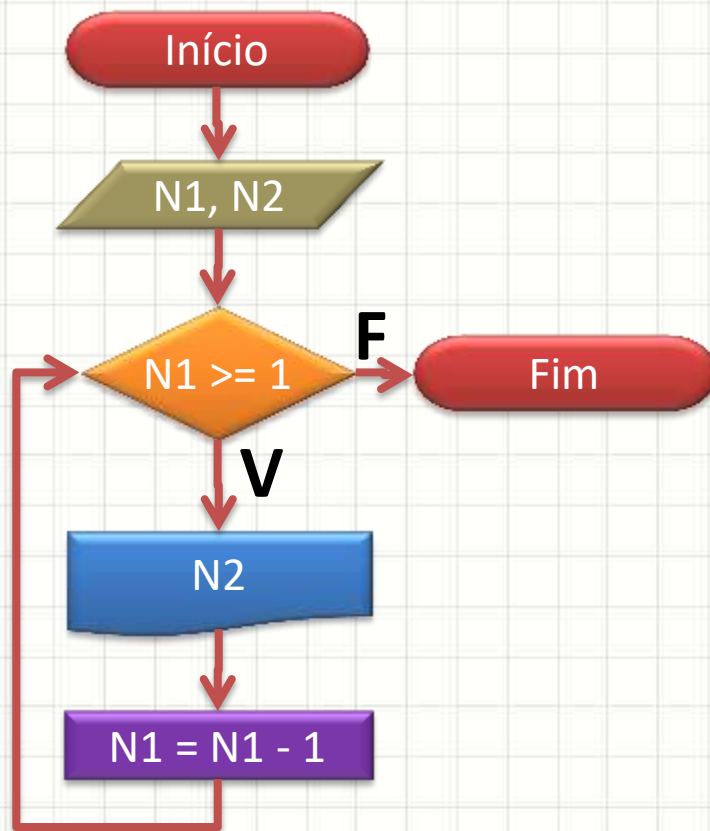
# Exercício 4

- Faça um programa que
  - a) Leia dois números N1 e N2
  - b) Imprima, N1 vezes, o valor de N2.



# Exercício 4

- Faça um programa que
  - a) Leia dois números N1 e N2
  - b) Imprima N1 vezes o valor de N2.



**# Mostra N1x o valor de N2**

```
N1 = int(input("Quantas vezes? "))
```

```
N2 = int(input("Qual valor? "))
```

```
while N1 >= 1 :
```

```
    print(N2)
```

```
    N1 = N1 - 1
```



# Exercício 4 - Variação

- Experimente mudar a linha do print por essa:

**# O que esse programa faz?**

```
N1 = int(input("Quantas vezes? "))
```

```
N2 = int(input("Qual valor? "))
```

```
while N1 >= 1 :
```

```
     print(N1, "*", N2, "=", N1*N2 )
```

```
    N1 = N1 - 1
```

**O que acontece?**



# A ESTRUTURA DE REPETIÇÃO FOR

# O que é a estrutura **for**

- Forma simples de repetir operações
  - Para uma **lista** de valores

aula11ex05.py

```
# Repetição em um conjunto de valores  
for X in [1, 2, 3]:  
    print(X)
```

**Falaremos mais sobre listas no futuro!**

# O que é a estrutura **for**

- Não precisam ser
  - Valores contínuos
  - Estar em ordem

```
# Repetição em um conjunto de valores
```

```
for X in [1, 3, 10, 25, -8]:
```

```
    print(X)
```

- Pode ser usado para letras de um texto

```
# Repetição nas letras de um texto
```

```
for X in "Abacaxi":
```

```
    print(X)
```



# O que é a estrutura **for**

- Se for em uma faixa contínua: **range**

# Repetição em uma faixa de 0 a 9

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

Número de Elementos

# Repetição em uma faixa de 1 a 9

```
for X in range(1, 10):  
    print(X)
```

Inicial (inclusive) e final (não incluído)

# Repetição de 1 a 9, de 2 em 2

```
for X in range(1, 10, 2):  
    print(X)
```

Inicial (inclusive) e final (não incluído), passo



**CONCLUSÕES**

# Resumo

- Repetição: Amplia a utilidade do computador
- Decisão: Verifica “se continua repetindo”.
- Não deixe de praticar!
- **TAREFA: Atividade Aula 11!**

**SAVA11/12!**

- Aprofundando repetição e o comando **break**
  - Revendo/Aprofundando: construindo funções
- 
- Exercitar a repetição!



**PERGUNTAS?**