

Programação em Python



**Prof. Daniel Santos**

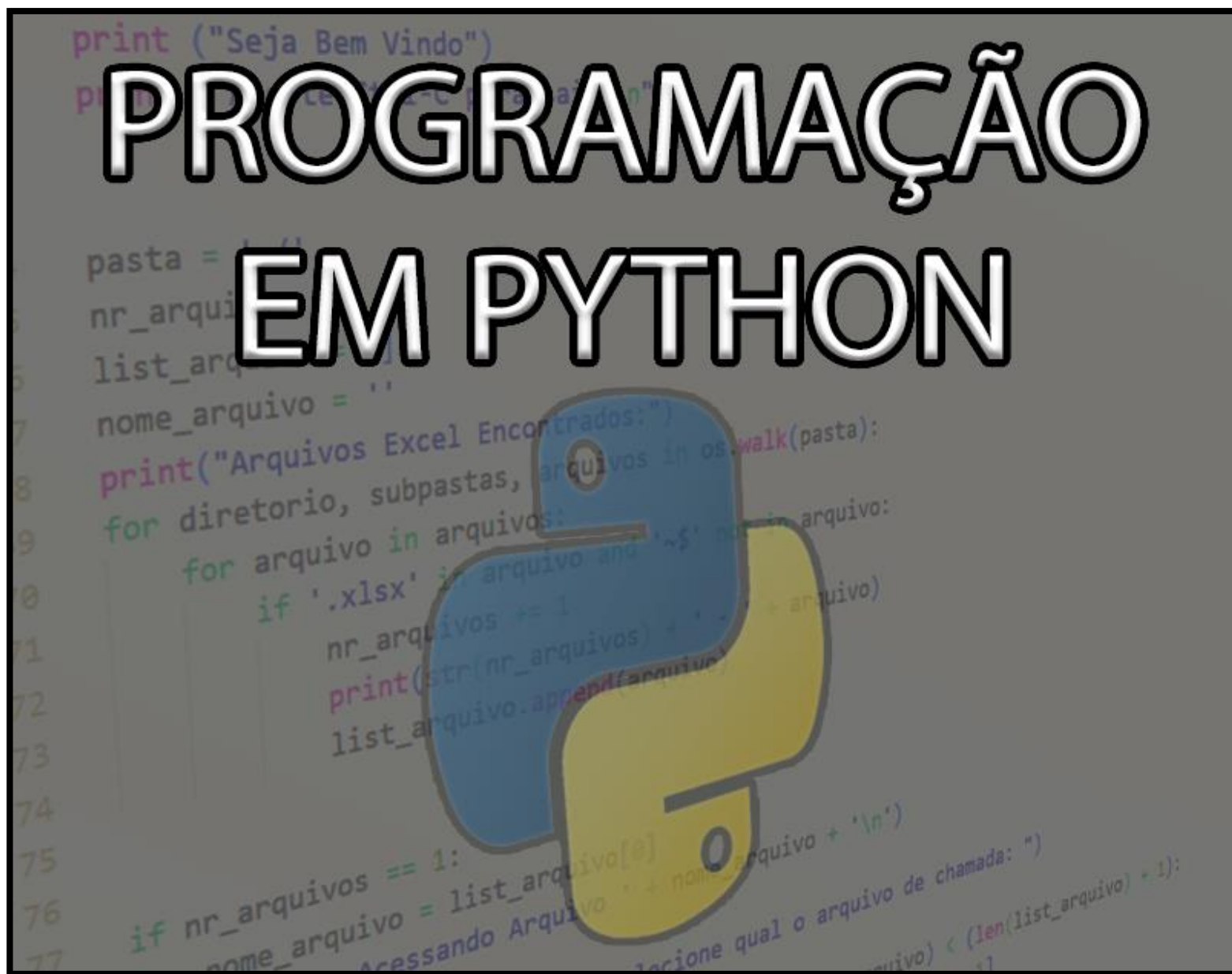
*daniel.sampaio@sp.senai.br*

**Senai Roberto Simonsen**

R. Monsenhor Andrade, 298 - Brás, São Paulo - SP

# PROGRAMAÇÃO

# EM PYTHON



Prof. Daniel Santos

## Estrutura for

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

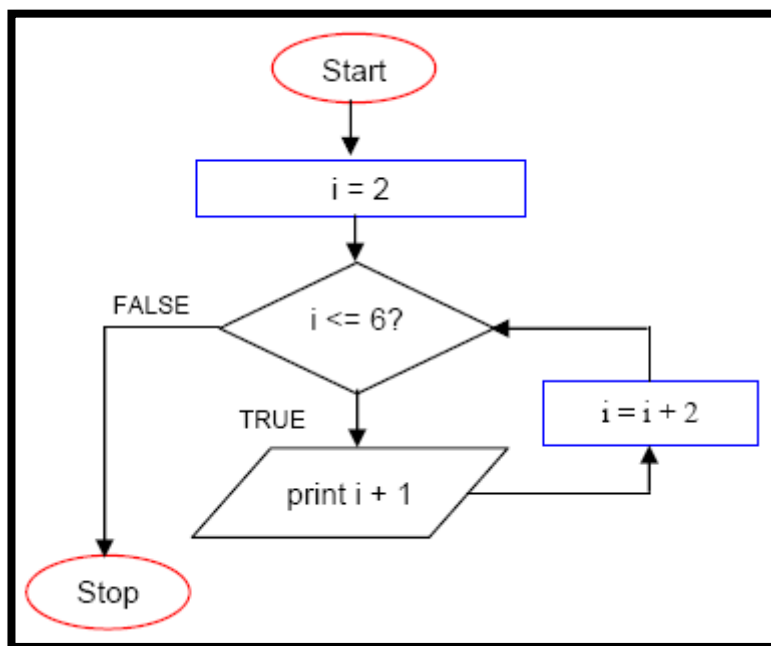


Executa 10 vezes os comandos dentro do laço for  
i vale de 0 a 9, mudando a cada ciclo

```
for i in 'texto':  
    comandos
```



Executa 5 vezes (tamanho do texto) os comandos dentro do laço for  
i vale cada letra da palavra **texto** mudando a cada ciclo



OBS: É possível adicionar **else** após um **for**, que irá executar após o término dos ciclos

## Estrutura for -- Exemplos

```
#Contando de 0 a 9  
for i in range(10):  
    print(i)
```

```
#Texto na Vertical  
for i in 'Daniel':  
    print(i)
```

## Estrutura for -- Exemplos



```
#Jogar um dado 1000 vezes
```

```
import random
```

```
dado1 = 0
```

```
dado2 = 0
```

```
dado3 = 0
```

```
dado4 = 0
```

```
dado5 = 0
```

```
dado6 = 0
```

```
for i in range(1000):
```

```
    dado = random.randint(1,6)
```

```
    if dado == 1:
```

```
        dado1 += 1
```

```
    elif dado == 2:
```

```
        dado2 += 1
```

```
    elif dado == 3:
```

```
        dado3 += 1
```

```
    elif dado == 4:
```

```
        dado4 += 1
```

```
    elif dado == 5:
```

```
        dado5 += 1
```

```
    elif dado == 6:
```

```
        dado6 += 1
```

```
print('1: ' + str(dado1))
```

```
print('2: ' + str(dado2))
```

```
print('3: ' + str(dado3))
```

```
print('4: ' + str(dado4))
```

```
print('5: ' + str(dado5))
```

```
print('6: ' + str(dado6))
```

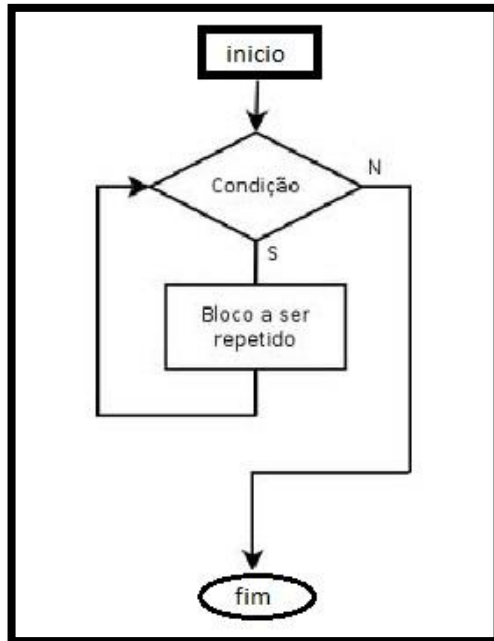
## Estrutura while



**while** condição:  
comandos



Executa os comandos dentro do laço **while** até a condição não ser mais verdadeira



OBS: É possível adicionar **else** após um **while**, que irá executar após o término dos ciclos



## Estrutura while -- Exemplos

```
print('Quanto é 2 + 2?')
valor = int(input())

while valor != 4:
    print('ERROUUUU!')
    print('Quanto é 2 + 2?')
    valor = int(input())
else:
    print('Certa a resposta')
```

```
# Contando de 0 a 100
i = 0

while i < 100:
    print(i)
    i+=1

print(i)
```

## Estrutura while -- Exemplos



```
print('Digite a senha')
senha = input()

while senha != 'a senha':
    print('Senha incorreta, tente novamente')
    senha = input()
else:
    print('Acesso liberado')
```



## Estrutura while -- Exemplos

```
# Ciclo infinito
import time
print('Este programa vai executar para sempre!')
i = 0

while True:
    print('Número de ciclos: ' + str(i))
    i+=1
    time.sleep(1)
```

## Instruções complementares

Para auxiliar no desenvolvimento de estruturas de controle de fluxo, existem 3 instruções que podem facilitar a resolver algum caso específico:

- **continue**  
Serve para finalizar aquele ciclo e começar o próximo ciclo
- **break**  
Serve para quebrar o ciclo
- **pass**  
Serve apenas para reservar um espaço, não faz nada em específico

## Instrução continue



```
# Contando de 0 a 9 pulando o número 7  
  
for i in range(10):  
    if i == 7:  
        continue  
    print(i)
```

## Instrução break



```
1 # Encontrar a primeira letra 'x' em um texto
2
3 texto = '''Este paragrafo tem muitas palavras, se for fazer uma busca em todo o texto
4         levava mais tempo de processamento, por isso, este programa ira buscar a
5         primeira letra x no texto'''
6
7 for i in texto:
8     print(i, end='')
9     if i == 'x':
10         print('\nLetra x encontrada, finalizando o texto')
11         break
12 print('Texto finalizado')
13
14
15
```

Shell

```
>>> %Run teste3.py
```

```
Este paragrafo tem muitas palavras, se for fazer uma busca em todo o tex
Letra x encontrada, finalizando o texto
Texto finalizado
```

## Instrução pass



```
1 # Programa incompleto
2 import random as rd
3 dado = rd.randint(1,6)
4
5 if dado == 6:
6     #espaço para código futuro
7     pass
8 elif dado == 5:
9     #espaço para código futuro
10    pass
11 elif dado == 4:
12    #espaço para código futuro
13    pass
```

## Exemplo prático – Jogo Par ou Impar



```
1 # Jogo par ou impar
2
3 # Importação das bibliotecas
4 import random as rd
5 import time
6
7 # Criação das variáveis globais
8 num_vit = 0
9 num_der = 0
10
```

## Exemplo prático – Jogo Par ou Impar



```
11 # Loop do Jogo
12 while True:
13     # Escolha inicial Par ou Impar
14     print('Escolha par (P) ou impar (I), para sair digite (Q)')
15     escolha = input()
16
17     # Caso digite Q o jogo é finalizado
18     if escolha == 'Q':
19         print('Jogo Finalizado')
20         break
21
```



## Exemplo prático – Jogo Par ou Impar

```
22     # Número do jogador
23     print('\nEscolha um numero de 0 a 9')
24     jogador_num = int(input())
25     print('\nVocê jogou ' + str(jogador_num))
26     time.sleep(1)
27
28     # Número do Rasp
29     pc_num = rd.randint(0,9)
30     print('\nO Rasp jogou ' + str(pc_num))
31     time.sleep(1)
32
33     # Soma dos números
34     soma = jogador_num + pc_num
35     print('\nA soma é ' + str(soma))
36     time.sleep(2)
37
```

## Exemplo prático – Jogo Par ou Impar



```
38     # Determinação de quem venceu ou perdeu
39     if escolha == 'P':
40         if soma % 2 == 0:
41             print('\nVocê venceu!')
42             num_vit = num_vit + 1
43         else:
44             print('\nVocê perdeu!')
45             num_der = num_der + 1
46     elif escolha == 'I':
47         if soma % 2 == 0:
48             print('\nVocê perdeu!')
49             num_der = num_der + 1
50         else:
51             print('\nVocê venceu!')
52             num_vit = num_vit + 1
53     else:
54         print('\nVocê não escolheu nem par (P) nem impar (I)')
55
```

## Exemplo prático – Jogo Par ou Impar



```
56     # Print do número de vitórias e derrotas atuais
57     time.sleep(1)
58     print('\nNúmero de Vitórias: ' + str(num_vit))
59     print('Número de Derrotas: ' + str(num_der) + '\n')
60     time.sleep(2)
61
62     # Reinicia o ciclo
```