

# Python - Listas e Estruturas de Repetição



Disciplina: Programação para Internet

Prof. Bruno Gomes



@profbrunogomes



# Aula de Hoje

2

- » **Introdução a Lista**
- » **Estruturas Repetição**
  - » **for**
  - » **while**



# 1.

## Lista

# Lista em Python

4

## Sintaxe

```
[v1, v2, v3]
```

## Exemplos

```
lista = [1,2,53,23,12]
```

```
lista2 = ["nome",34,2.0]
```

# Acessando valor de uma lista

5

## Sintaxe

```
lista[indice]
```

## Exemplos

```
lista = [1,2,53,23,12]  
valor = lista[1]  
print(valor)
```

```
lista2 = ["nome",34,2.0]  
print(lista2[2])
```

# Lista dentro de Lista

6

## Exemplo

```
lista = [1,2,53,[5,3]]
```

## Exemplo

```
lista = [1,2,53,23,12]  
lista2 = ["nome", lista]
```

# Operações

7

Tamanho de uma Lista: len

```
lista = [1,2,53,23,12]  
print(len(lista))
```

Concatenação: +

```
lista = [1,2,53]  
lista2 = [5,3,2]  
  
lista3 = lista + lista2  
#resultado: [1,2,53,5,3,2]  
  
lista4 = lista + [7,3,5.0]  
#resultado: [1,2,53,7,3,5.0]
```



## Função range()

8

```
range( stop )
```

```
range( [start], stop[, step] )
```

- » Retorna uma série numérica no intervalo enviado como argumento
- » Resulta em um objeto iterável





# Função range()

9

## Sintaxe

```
range( stop )  
range( [start], stop[, step] )
```

```
#start: início da sequência  
#stop: último elemento da sequência  
#step: intervalo entre os elementos
```

## Exemplo

```
range(5)  
#retorna: 0, 1, 2, 3, 4
```

```
range(2, 6)  
#retorna: 2, 3, 4, 5
```

```
range(2, 10, 2)  
#retorna: 2, 4, 6, 8
```

# Função range() - Transformando em lista

10

Utiliza o list()

```
list(range(start, stop, step))
```

Exemplo

```
numeros = list(range(2, 10, 2))  
print(numeros)
```

# Dúvidas?



# Atividade

12

- » Desenvolva um Algoritmo em Python que:
  - » Peça para o usuário digitar 3 números inteiros. Logo em seguida, acrescente esses números dentro de uma lista, e imprima a lista.



# Atividade

13

- » **Desenvolva um Algoritmo em Python que:**
  - » **Gere os números de 1 até 20;**
  - » **Gere os números de 20 até 1 (em ordem decrescente)**



# Vamos Praticar!!

Lista de Exercícios





# 2.

## **Estrutura de Repetição - for**



# for

16

## Sintaxe

```
for valor in lista:  
    comando
```

## Exemplo

```
for var in [1,2,5,78]:  
    print(var)
```

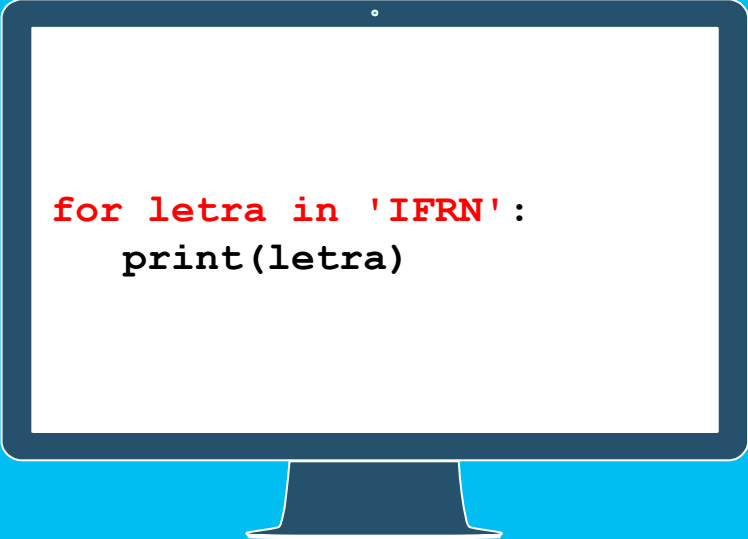
---

```
lista = [1,2,53,23,12]  
for var in lista:  
    print(var)
```

# for

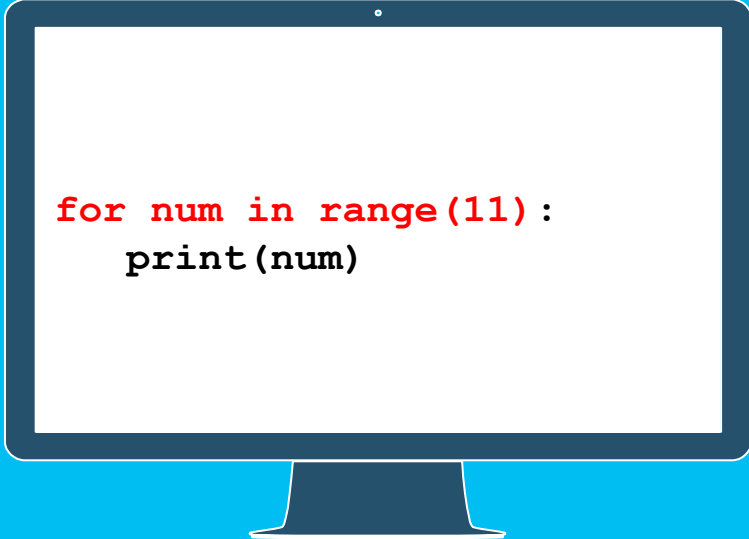
17

## Exemplo com String



```
for letra in 'IFRN':  
    print(letra)
```

## Exemplo com range



```
for num in range(11):  
    print(num)
```

# for...else

18

## Sintaxe

```
for valor in lista:  
    comando  
else:  
    comando
```

## Exemplo

```
for var in [1,2,5,78]:  
    print(var)  
else:  
    print("Fim")
```

## break e continue

19

### break

```
for num in range(11):  
    if num==6:  
        break  
    else:  
        print(num)
```

### Exemplo

```
for num in range(11):  
    if num==6:  
        continue  
    else:  
        print(num)
```

# Dúvidas?



# Atividade

21

- » **Desenvolva um Algoritmo em Python que:**
  - » **Imprima os números:**
    - » **2 4 6 8 10 12**



# Atividade

22

- » **Desenvolva um Algoritmo em Python que:**
  - » Peça para o usuário digitar um número inteiro.
  - » Ao fim, o programa deve imprimir todos os números inteiros a partir do 1 até o número digitado.





- » **Desenvolva um Algoritmo em Python que:**
  - » Peça para o usuário digitar 2 números.
  - » Ao fim, o programa deve imprimir todos os números inteiros no intervalo dos valores armazenados (incluir os números armazenados).



- » **Desenvolva um Algoritmo em Python que:**
  - » Peça para o usuário digitar 2 números.
  - » Ao fim, o programa deve imprimir todos os números inteiros no intervalo dos valores armazenados (não incluir os números armazenados).



# Vamos Praticar!!

Lista de Exercícios



# 3.

## Estrutura de Repetição - while

# while

27

## Sintaxe

```
while(condicao):  
    comando
```

## Exemplo:

```
num = 1  
while(num <= 10):  
    print(num)  
    num += 1
```

# while...else

28

## Sintaxe

```
while (condicao) :  
    comando  
else:  
    comando  
  
#else é acionado quando a  
condição for falsa
```

## Exemplo:

```
num = 1  
while(num <= 10):  
    print(num)  
    num += 1  
else:  
    print("Último valor: ", num)
```



# Dúvidas?





# Atividade

30

- » **Desenvolva um Algoritmo em Python que:**
  - » **Imprima os números:**
    - » **2 4 6 8 10 12**



# Atividade

31

- » **Desenvolva um Algoritmo em Python que:**
  - » Peça para o usuário digitar um número inteiro.
  - » Ao fim, o programa deve imprimir todos os números inteiros a partir do 1 até o número digitado.



- » **Desenvolva um Algoritmo em Python que:**
  - » Peça para o usuário digitar 2 números.
  - » Ao fim, o programa deve imprimir todos os números inteiros no intervalo dos valores armazenados (incluir os números armazenados).



- » **Desenvolva um Algoritmo em Python que:**
  - » Peça para o usuário digitar 2 números.
  - » Ao fim, o programa deve imprimir todos os números inteiros no intervalo dos valores armazenados (não incluir os números armazenados).



# Vamos Praticar!!

Lista de Exercícios

