

**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**
BAHIA
Campus Salvador

Disciplina : Programação Python

Prof. Domingos Mainart

Email: prof.mainart@gmail.com

Estruturas de Repetição

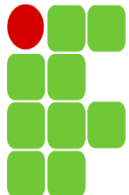
As estruturas de repetição são utilizadas quando queremos que um bloco de código seja executado várias vezes.

Em Python existem duas formas de criar uma estrutura de repetição:

- O **for** é usado quando se quer iterar sobre um bloco de código um número determinado de vezes.
- O **while** é usado quando queremos que o bloco de código seja repetido até que uma condição seja satisfeita.

Ou seja, é necessário que uma expressão booleana dada seja verdadeira. Assim que ela se tornar falsa, o **while** para.

Nota: Na linguagem Python a indentação é obrigatória. assim como nas estruturas de controle, as estruturas de repetição também precisam.



Estruturas de repetição

```
>>> # Aqui repetimos o print 3 vezes
>>> for n in range(0, 3):
...     print(n)
...
0
1
2
```

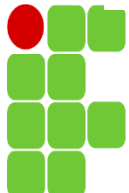
O loop **for** em Python itera sobre os itens de um conjunto, sendo assim, o `range(0, 3)` precisa ser um conjunto de elementos. E na verdade ele é:

Estruturas de repetição

```
>>> list(range(0, 3))  
[0, 1, 2]
```

Para iterar sobre uma lista:

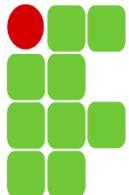
```
>>> lista = [1, 2, 3, 4, 10]  
>>> for numero in lista:  
...     print(numero ** 2)  
...  
1  
4  
9  
16  
100
```



Estruturas de repetição

Isso se aplica para strings também:

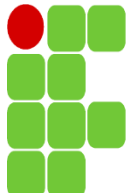
```
>>> # Para cada letra na palavra, imprimir a letra
>>> palavra = "casa"
>>> for letra in palavra:
...     print(letra)
...
c
a
s
a
```



Estruturas de repetição

Em dicionários podemos fazer assim:

```
>>> gatinhos = {"Português": "gato", "Inglês": "cat", "Francês": "chat", "Finlandês":  
    ↳ "Kissa"}  
>>> for chave, valor in gatinhos.items():  
...     print(chave, "->", valor)  
...  
Português -> gato  
Inglês -> cat  
Francês -> chat  
Finlandês -> Kissa
```



Estruturas de repetição

Para auxiliar as estruturas de repetição, existem dois comandos:

- **break**: É usado para sair de um loop, não importando o estado em que se encontra.
- **continue**: Funciona de maneira parecida com a do break, porém no lugar de encerrar o loop, ele faz com que todo o código que esteja abaixo (porém ainda dentro do loop) seja ignorado e avança para a próxima iteração.

Estruturas de repetição

Veja a seguir um exemplo de um código que ilustra o uso desses comandos. Note que há uma string de documentação no começo que explica a funcionalidade.

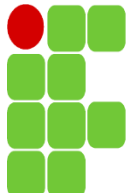
```
"""
Esse código deve rodar até que a palavra "sair" seja digitada.
* Caso uma palavra com 2 ou menos caracteres seja digitada, um aviso
  deve ser exibido e o loop será executado do início (devido ao
  continue), pedindo uma nova palavra ao usuário.
* Caso qualquer outra palavra diferente de "sair" seja digitada, um
  aviso deve ser exibido.
* Por fim, caso a palavra seja "sair", uma mensagem deve ser exibida e o
  loop deve ser encerrado (break).
"""

>>> while True:
...     string_digitada = input("Digite uma palavra: ")
...     if string_digitada.lower() == "sair":
...         print("Fim!")
...         break
...     if len(string_digitada) < 2:
...         print("String muito pequena")
...         continue
...     print("Tente digitar \"sair\"")
...
Digite uma palavra: oi
Tente digitar "sair"
Digite uma palavra: ?
String muito pequena
Digite uma palavra: sair
Fim!
```



Estruturas de repetição

```
>>> # Aqui iniciamos o n em 0, e repetimos o print até que seu valor seja maior ou  
    ↳ igual a 3  
>>> n = 0  
>>> while n < 3:  
...     print(n)  
...     n += 1  
...  
0  
1  
2
```



Estruturas de repetição

FIM