# Listas e Estruturas de Repetição em Python

#### **Agenda**

#### **Tópicos:**

- Operações Básicas em Listas
- Fatias de Listas (Slices)
- Operador del
- Laços de Repetição Simples
- Iterando por Itens em Sequências
- Operador in
- Textos São Sequências!
- Procurando Ajuda



#### **Criando listas vazias**

```
>>> lista = []
>>> lista = list()
```

#### **Criando listas preenchidas**



#### Obtendo a quantidade de elementos da lista

```
>>> lista = [8, 9, 1, 15]
>>> len(lista)
4
```

#### Obtendo um elemento da lista

```
>>> lista = [8, 9, 1, 15]
>>> lista[0]
>>> lista[1]
>>> lista[4]
>>> ???
```

#### Obtendo um elemento da lista

```
>>> lista = [8, 9, 1, 15]
>>> lista[0]
>>> lista[1]
>>> lista[4]
>>> ???
         Erro! O primeiro
```

irro! O primeiro índice é 0 e o último é 3

#### Obtendo um elemento da lista

```
>>> lista = [8, 9, 1, 15]
>>> lista[-1]
15
>>> lista[-2]
1
```



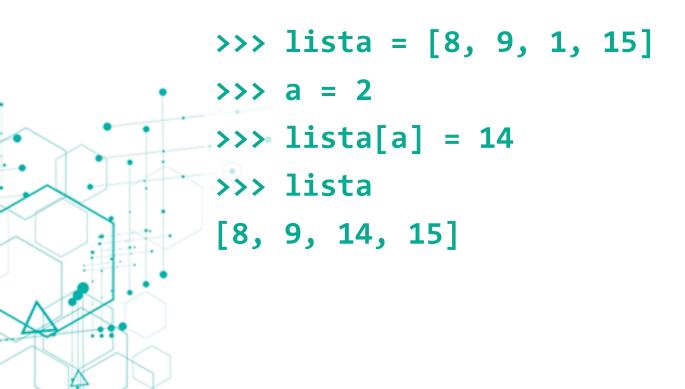
#### Obtendo um elemento da lista indiretamente



#### Alterando um elemento da lista

```
>>> lista = [8, 9, 1, 15]
>>> lista[0] = 14
>>> lista
[14, 9, 1, 15]
```

#### Alterando um elemento da lista indiretamente



#### Acrescentando um novo elemento ao final da lista

```
>>> lista = [8, 9, 1, 15]
>>> lista.append(12)
>>> lista
[8, 9, 1, 15, 12]
```

# Acrescentando vários novos elementos ao final da lista

```
>>> lista = [8, 9, 1, 15]
>>> lista2 = [4, 5, 6]
>>> lista.extend(lista2)
>>> lista
[8, 9, 1, 15, 4, 5, 6]
```

# Acrescentando um novo elemento a uma posição qualquer da lista

```
>>> lista = [8, 9, 1, 15]
>>> lista.insert(1, 12)
>>> lista
[8, 12, 9, 1, 15]
```

#### Obtendo e removendo o último elemento da lista

```
>>> lista = [8, 9, 1, 15]
>>> lista.pop()
>>> lista
[8, 9, 1]
```

#### Obtendo e removendo o último elemento da lista

```
>>> lista = [8, 9, 1, 15]
>>> a = lista.pop()
>>> a
>>> lista
[8, 9, 1]
```

#### Obtendo e removendo um elemento da lista

```
>>> lista = [8, 9, 1, 15]
>>> lista.pop(2)
>>> lista
[8, 9, 15]
```

#### Removendo todos os elementos da lista

```
>>> lista = [8, 9, 1, 15]
>>> lista.clear()
>>> lista
```

#### Contando aparições de elementos na lista

```
>>> lista = [8, 9, 1, 8, 8, 15]
>>> lista.count(8)
>>> lista.count(10)
```

# Ordenando todos os elementos da lista, de forma crescente

```
>>> lista = [8, 9, 1, 15]
>>> lista.sort()
>>> lista
[1, 8, 9, 15]
```

## Ordenando todos os elementos da lista, de forma decrescente



#### Procurando por um elemento na lista

```
>>> lista = [8, 9, 1, 15]
>>> lista.index(15)
3
>>> lista.index(10)
Traceback (most recent call last):
   File "<stdin>", line 1, in <module>
ValueError: 10 is not in list
```

#### Procurando por um elemento na lista

```
>>> lista = [8, 9, 1, 15]
>>> lista.index(15)
3
>>> lista.index(10)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: 10 is not in list
```

Mais para frente veremos como tratar erros!

A notação de fatia (slice): tem inúmeras aplicações, apesar de parecer um pouco confusa a princípio...

Convém se acostumar com sua leitura, pois essa notação é responsável por parte do poder do Python!





```
Copiando um trecho da lista
             0, 1, 2, 3, 4, 5, 6, 7
>>> lista = [8, 9, 1, 15, 20, 30, 2, 10]
>>> lista2 = lista[3:5]
>>> lista2
[15, 20]
```

```
Copiando um trecho da lista
             0, 1, 2, 3, 4, 5, 6, 7
>>> lista = [8, 9, 1, 15, 20, 30, 2, 10]
>>> lista2 = lista[3:5]
>>> lista2
                        Exclusivo
        Inclusivo
```

```
Copiando um trecho da lista
             0, 1, 2, 3, 4, 5, 6, 7
>>> lista = [8, 9, 1, 15, 20, 30, 2, 10]
>>> lista2 = lista[3:]
>>> lista2
[15, 20, 30, 2, 10]
```

```
Copiando um trecho da lista
             0, 1, 2, 3, 4, 5, 6, 7
>>> lista = [8, 9, 1, 15, 20, 30, 2, 10]
>>> lista2 = lista[:5]
>>> lista2
[8, 9, 1, 15, 20]
```

```
Copiando um trecho da lista
             0, 1, 2, 3, 4, 5, 6, 7
>>> lista = [8, 9, 1, 15, 20, 30, 2, 10]
>>> lista2 = lista[:]
>>> lista2
[8, 9, 1, 15, 20, 30, 2, 10]
```

```
Copiando um trecho da lista
```

```
0, 1, 2, 3, 4, 5, 6, 7
>>> lista = [8, 9, 1, 15, 20, 30, 2, 10]
```

- >>> lista2 = lista[3:-2]
- >>> lista2

[15, 20, 30]



#### Substituindo um trecho da lista

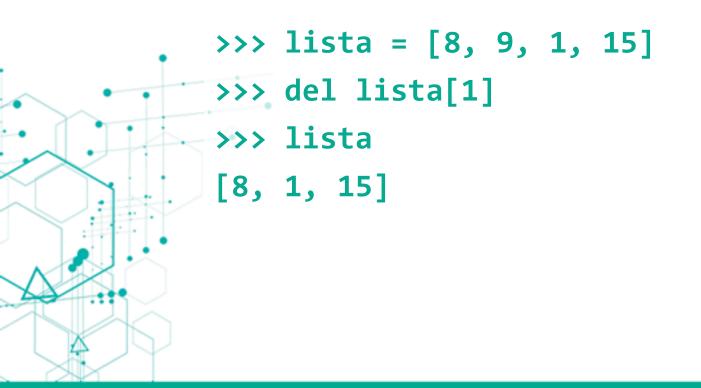
```
0, 1, 2, 3, 4, 5, 6, 7
>>> lista = [8, 9, 1, 15, 20, 30, 2, 10]
>>> lista[3:5] = [90, 91, 92, 93]
>>> lista
[8, 9, 1, 90, 91, 92, 93, 30, 2, 10]
```

#### Substituindo um trecho da lista

```
0, 1, 2, 3, 4, 5, 6, 7
>>> lista = [8, 9, 1, 15, 20, 30, 2, 10]
>>> lista[2:] = []
>>> lista
[8, 9]
```

#### **Operador del**

# O operador del é utilizado para excluir elementos ou fatias de listas



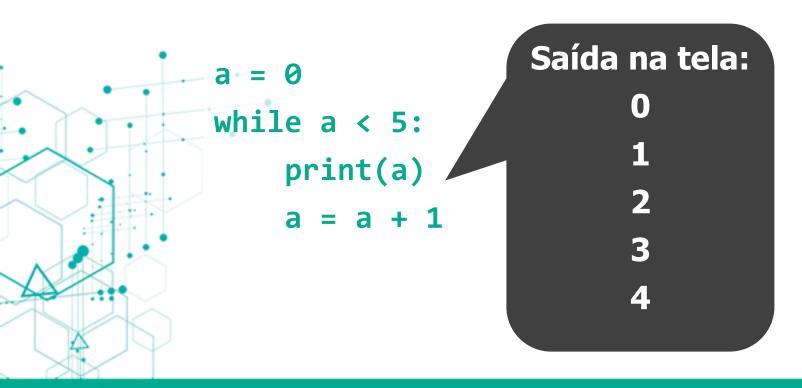
#### **Operador del**

# O operador del é utilizado para excluir elementos ou fatias de listas

```
>>> lista = [8, 9, 1, 15]
>>> del lista[1:3]
>>> lista
[8, 15]
```

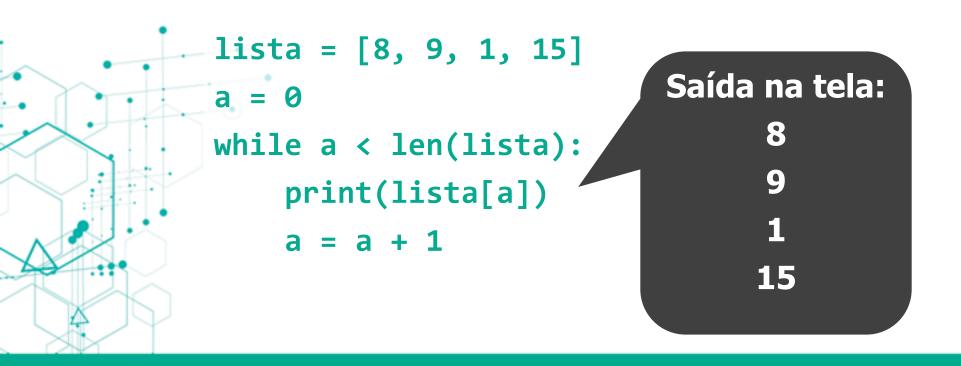
#### Laços de Repetição Simples

Assim como as estruturas de tomada de decisão, laços de repetição precisam da identação correta, e de uma pergunta válida



#### Laços de Repetição Simples

Assim como as estruturas de tomada de decisão, laços de repetição precisam da identação correta, e de uma pergunta válida



# Laços de Repetição Simples

Assim como as estruturas de tomada de decisão,

laços de repetição precisam da identação correta, e

de uma pergunta válida



```
lista = [8, 9, 1, 15]
a = 0
while a < len(lista):
    print(lista[a])
a = a + 1</pre>
```



Apesar de funcionar, essa forma de percorrer uma lista não é a mais utilizada em Python... Veremos outra forma em breve...

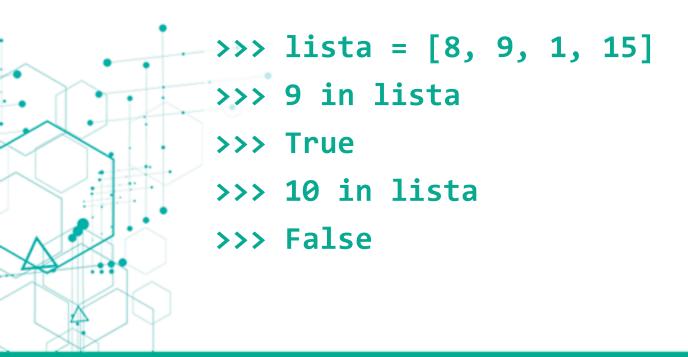
# Laços de Repetição Simples

#### A instrução break aborta o laço de repetição

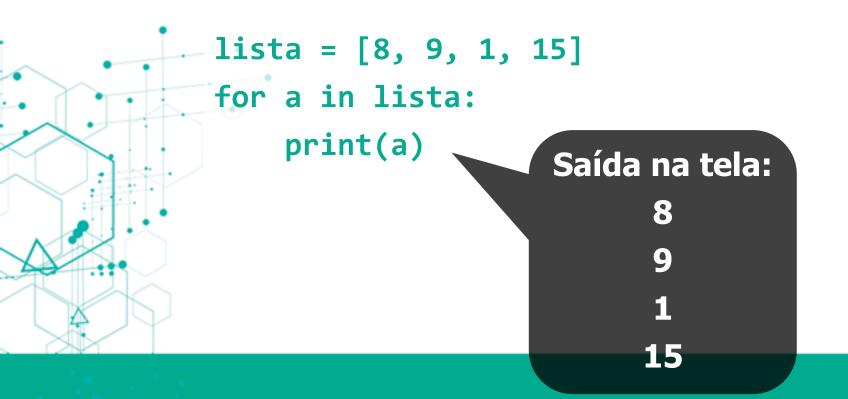
```
bomba = int(input('Digite o número bomba: '))
lista = [8, 9, 1, 15]
while a < len(lista):</pre>
    if bomba == lista[a]:
        print('BUUUMMMMM!')
        break
    print(lista[a])
```

#### **Operador in**

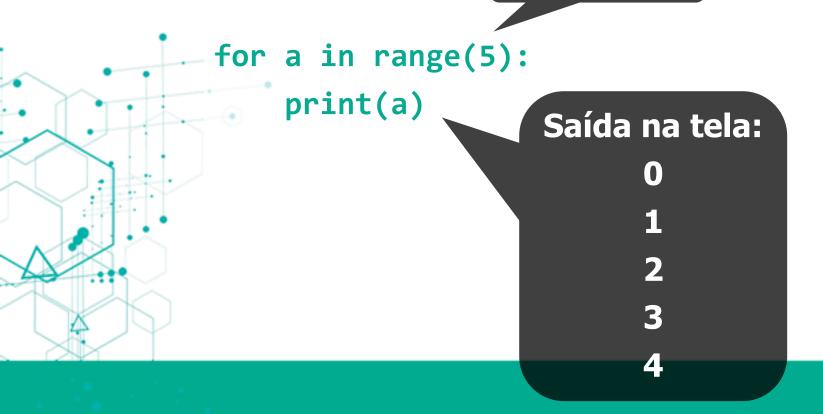
O operador in é utilizado para trabalhar com o conteúdo de listas e sequências (como veremos em breve)



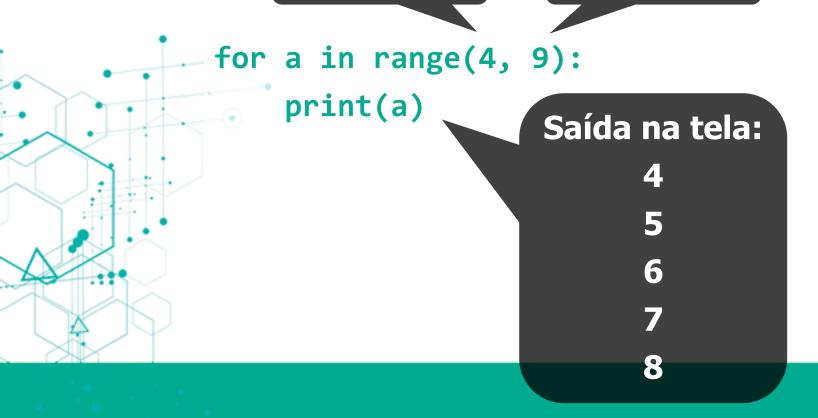
A estrutura for das outras linguagens é parecida com o while. Mas em Python, o for serve apenas para iterar sobre itens de sequências



A estrutura for das outras linguagens é parecida com o while. Mas em Python, o for serve apenas para iterar so exclusivo de sequências

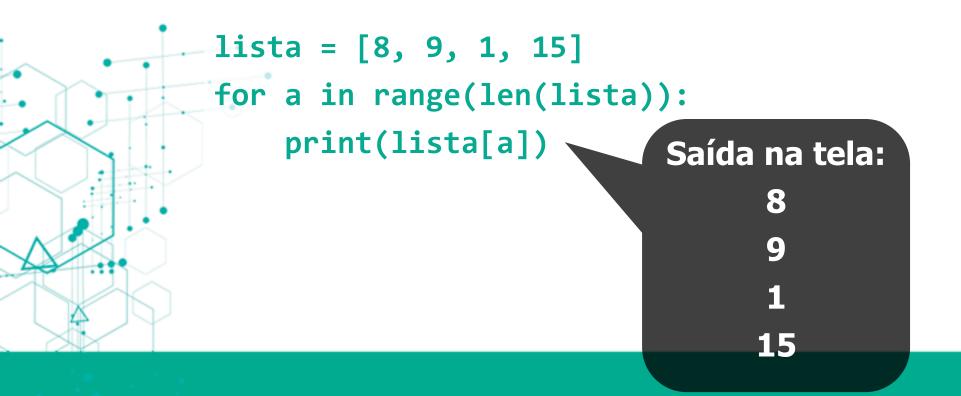


A estrutura for das outras linguagens é parecida com o while. Mas em Python, o for serve apenas para Inclusivo sobre Exclusivo equências



A estrutura for das outras linguagens é parecida com o while. Mas em Python, o for serve apenas para Inclusivo sobre Exclusivo equências for a in range(4, 9, 2): print(a) Saída na tela:

A estrutura for das outras linguagens é parecida com o while. Mas em Python, o for serve apenas para iterar sobre itens de sequências



```
>>> lista = list(range(8))
>>> lista
[0, 1, 2, 3, 4, 5, 6, 7]
```

```
>>> lista = list(range(4, 8))
>>> lista
[4, 5, 6, 7]
```

```
>>> lista = list(range(4, 8))
>>> lista
[4, 5, 6, 7]

>>> lista = list(2 * x for x in range(4, 8))
>>> lista
[8, 10, 12, 14]
```

```
>>> lista = list(0 for x in range(4))
>>> lista
[0, 0, 0, 0]

>>> lista = list(1 for x in range(3))
>>> lista
[1, 1, 1]
```





Vimos alguns exemplos com listas, mas algumas operações de listas também podem ser utilizadas com textos...

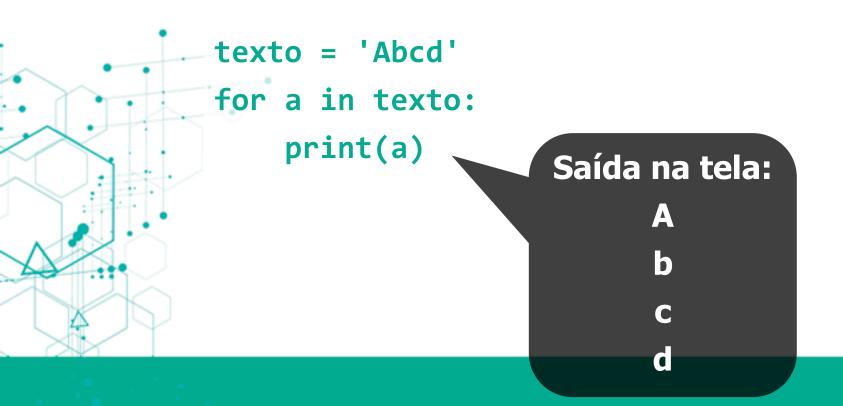
```
>>> texto = 'Abcd'
>>> texto[2]
'c'
```

Vimos alguns exemplos com listas, mas algumas operações de listas também podem ser utilizadas com textos...

```
>>> texto = 'Abcd'
>>> texto[1:]
'bcd'
```

```
Vimos alguns exemplos com listas, mas algumas
 operações de listas também rodem ser utilizadas
                   com te
                           listas, textos
                               são
                           imutáveis!!!
>>> texto = 'Abcd'
>>> texto[1] = 'X'
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object does not support item
assignment
```

Vimos alguns exemplos com listas, mas algumas operações de listas também podem ser utilizadas com textos...



#### **Procurando Ajuda**

Para mais informações sobre listas, sequências, tuplas, conjuntos e dicionários:

docs.python.org/3/tutorial/datastructures.html

#### **Procurando Ajuda**

www.python.org/doc docs.python.org/3/reference docs.python.org/release/3.7.3/reference docs.python.org/3/library docs.python.org/release/3.7.3/library docs.python.org/3/tutorial/introduction.html stackoverflow.com pt.stackoverflow.com

