



137

## Introdução

- Python oferece uma variedade de tipos de dados compostos, geralmente chamados de sequências.
- Lista é um dos tipos de dados mais frequentemente usados em Python devido a sua versatilidade.

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

138

## Introdução

Se considerarmos a memória como sendo uma rua. Todas as casas desta rua representam um espaço na memória para guardarmos um valor, então elas são as casas.

Deste modo, as estruturas de dados como as listas são como prédios, os quais podem armazenar mais de um andar, ou seja, mais de um valor internamente

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

139

## Criando uma lista

- Na programação Python, uma lista é criada colocando todos os itens (elementos) entre colchetes [], separados por vírgulas.
- A lista pode ter qualquer número de itens e podem ser de tipos diferentes (inteiro, flutuante, string etc.).

```
# lista vazia
minhalista = []

# lista de inteiros
minhalista = [1, 2, 3]

# lista de strings
minhalista = ["Ana", "Beatriz", "Carlos"]

# lista de diferentes tipos
minhalista = [1, "Olá Mundo!", 3.45]
```

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

140

## Acessando valores em uma lista

- Podemos acessar itens em uma lista
  - Índice direto
  - Índice negativo
  - Fatiamento
  - Utilizando métodos

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

141

## Acessando valores em uma lista - Índice Direto

- Podemos usar o operador de índice `[]` para acessar um item em uma lista.
- Em Python, assim como em outras linguagens, os índices começam em 0. Portanto, uma lista com 5 elementos terá um índice de 0 a 4.
- Tentar acessar índices diferentes desses gerará um `IndexError`.
- O índice deve ser um número inteiro. Não podemos usar float ou outros tipos, isso resultará em `TypeError`.

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

142

## Acessando valores em uma lista - Índice Direto

- Exemplo:

```
# lista
minhalista = ['p', 'y', 't', 'h', 'o', 'n']

# Saída: p
print(minhalista[0])

# Saída: t
print(minhalista[2])

# Saída: o
print(minhalista[4])

# Error! Apenas números inteiros podem ser usados
print(minhalista[4.0])
```

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

143

## Acessando valores em uma lista - Índice negativo

- Python permite indexação negativa para suas sequências.
- O índice -1 se refere ao último item, -2 ao penúltimo item e assim por diante:

```
minhalista = ['p','y','t','h','o','n']

# Saída: n
print(minhalista[-1])

# Saída: y
print(minhalista[-5])
```

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

144

## Acessando valores em uma lista

	'p'	'y'	't'	'h'	'o'	'n'
índice	0	1	2	3	4	5
índice negativo	-6	-5	-4	-3	-2	-1

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

145

## Acessando valores em uma lista - Fatiamento

- Podemos acessar um conjunto de elementos em uma lista usando o operador de fatiamento :(dois pontos).

```
minhalista = ['p','y','t','h','o','n']
print(minhalista[2:5])
print(minhalista[:-5])
print(minhalista[5:])
print(minhalista[:])
```

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

146

## Adicionando e alterando elementos

- As listas são **mutáveis**, o que significa que seus elementos podem ser alterados (ao contrário de strings ou tuplas).
- Podemos usar o operador de atribuição `=` para alterar um item ou uma série de itens.

```
# inicializando uma lista
lista = [2, 4, 6, 8]

# alterando o primeiro item
lista[0] = 1

print(lista)

# alterando do segundo ao quarto item
lista[1:4] = [3, 5, 7]

print(lista)
```

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

147

## Adicionando e alterando elementos - `append()` e `extend()`

- Podemos adicionar um item a uma lista usando o método `append()` ou adicionar vários itens usando o método `extend()`.

```
numeros = [1, 3, 5]

numeros.append(7)

print(numeros)

numeros.extend([9, 11, 13])

print(numeros)
```

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

148



## Adicionando e alterando elementos - concatenação

- Podemos usar o operador `+` para combinar duas listas. Isso também é chamado de concatenação.
- O operador `*` repete uma lista pelo número de vezes fornecido.

```
numeros = [1, 3, 5]
print(numeros + [9, 7, 5])
print(["re"] * 3)
```

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

149

## Adicionando e alterando elementos - insert()

- Podemos inserir um elemento em um local desejado usando o método `insert()` ou inserir vários itens comprimindo-o em uma fatia vazia de uma lista.

```
numeros = [1, 9]
numeros.insert(1,3)
print(numeros)
numeros[2:2] = [5, 7]
print(numeros)
```

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

150

## Excluir ou remover elementos - del

- Podemos excluir um ou mais itens de uma lista usando a palavra-chave `del`.
- Ela pode até mesmo excluir totalmente uma lista.

```
letras = ['p', 'r', 'o', 'g', 'r', 'a', 'm', 'a']
del letras[2]
print(letras)

del letras[1:5]
print(letras)

del letras
print(letras)
```

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

151

## Excluir ou remover elementos - remove(), pop() e clear()

- Podemos usar método `remove()` para remover o item fornecido ou o método `pop()` para remover um item no índice fornecido.
- O `pop()` remove e retorna o último item se o índice não for fornecido. Isso nos ajuda a implementar listas como pilhas (estrutura de dados primeiro a entrar, último a sair).
- Também podemos usar o método `clear()` para esvaziar uma lista.

```
letras = ['p', 'r', 'o', 'g', 'r', 'a', 'm', 'a']
letras.remove('p')
print(letras)

print(letras.pop(1))
print(letras)

print(letras.pop())
print(letras)

letras.clear()
print(letras)
```

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

152



## Excluir ou remover elementos

- Podemos excluir itens de uma lista atribuindo uma lista vazia a uma parte dos elementos.

```
letras = ['p', 'r', 'o', 'g', 'r', 'a', 'm', 'a']  
letras[2:3] = []  
print(letras)
```

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

153

## Métodos de lista em Python

- Os métodos que estão disponíveis com objetos de lista na programação Python são tabulados a seguir.
- Eles são acessados como `list.method()`. Alguns dos métodos já foram usados acima.

Professor Ariel da Silva Dias - [www.arieldias.com](http://www.arieldias.com) - Obra Gratuita, proibida reprodução e venda

154

## Métodos de lista em Python

Métodos	Definição
append()	Adiciona um elemento ao final da lista
extend()	Adiciona elementos de uma lista em outra
insert()	Adiciona um elemento no índice definido
remove()	Remove um elemento da lista
pop()	Remove e retorna um elemento da lista
clear()	Remove todos os elementos da lista
index()	Retorna o índice do primeiro elemento
count()	Retorna a contagem de número de elementos passados como argumento
sort()	Ordena a lista em ordem crescente
reverse()	Inverte a ordem dos elementos da lista
copy	Retorna uma cópia da lista

Professor Ariel da Silva Dias - [www.arieldias.com](http://www.arieldias.com) - Obra Gratuita, proibida reprodução e venda

155

## Métodos de lista em Python - Exemplos

```
# lista
lista = [3, 8, 1, 6, 0, 8, 4]

# Saída: 1
print(lista.index(8))

# Saída: 2
print(lista.count(8))

lista.sort()

# Saída: [0, 1, 3, 4, 6, 8, 8]
print(lista)

lista.reverse()

# Saída: [8, 8, 6, 4, 3, 1, 0]
print(lista)
```

156

## Localizar um elemento em uma lista

- Podemos testar se um elemento existe ou não em uma lista, usando a palavra-chave `in`

```
lista = ['p', 'r', 'o', 'g', 'r', 'a', 'm', 'a']  
  
# Saída: True  
print('p' in lista)  
  
# Saída: False  
print('i' in lista)  
  
# Saída: True  
print('c' not in lista)
```

Professor Ariel da Silva Dias - [www.arieldias.com](http://www.arieldias.com) - Obra Gratuita, proibida reprodução e venda

157

## Percorrendo uma lista

- Podemos usar um laço de repetição para iterar cada elemento de uma lista

```
for fruta in ['maçã', 'banana', 'manga']:  
    print("Eu gosto de", fruta)
```

- Também é possível percorrer uma lista utilizando laço `while`

Professor Ariel da Silva Dias - [www.arieldias.com](http://www.arieldias.com) - Obra Gratuita, proibida reprodução e venda

158

## Por hoje é só!

### Orientações

- Os códigos de exemplo desta aula
- Pratique os exercícios desta aula
- Somente com a prática constante você será fluente em Python!

### Qualquer dúvida:

- [www.arieldias.com](http://www.arieldias.com)
- [contato@arieldias.com](mailto:contato@arieldias.com)

Bons estudos! Até a próxima!

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

159

## Referências para esta aula

- GURU, 99. Python. Disponível em: <https://www.guru99.com/python-tutorials.html>. Acesso em: 02 fev. 2021.
- PROGRAMIZ. Python. Disponível em: <https://www.programiz.com/python-programming>. Acesso em: 02 fev. 2021.

Professor Ariel Dias - [www.arieldias.com](http://www.arieldias.com) - Obra licenciada, proibida reprodução e uso sem autorização

160