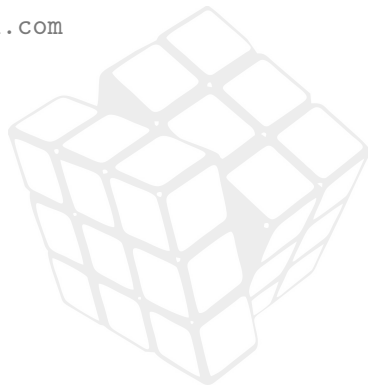


Algoritmos e Programação

Estrutura de Dados - Lista

Gustavo Sávio
gsoprofessor@gmail.com

2017.2



São estruturas utilizadas para manipular dados (armazenar, recuperar, remover...) de modo eficiente

- ▶ Python disponibiliza muitas estruturas de dados de forma nativa

Descrição geral

Uma lista é uma sequência de elementos do **mesmo tipo**, onde cada elemento é armazenado em um índice.

- ▶ Exemplos de listas:
 - ▶ Lista de Jogadores
 - ▶ Lista de *notebooks*
 - ▶ Lista de consoles de vídeo games...
- ▶ Cada item da lista possui uma posição...

lista = $a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_n$

- ▶ Em Python uma lista pode ser representada por colchetes []:

```
1 lista_frutas = ['Morango', 'Banana', 'Limão', 'Maracujá']  
2 lista_impostos = ['IRRF', 'IPVA', 'IPTU' ]
```

- ▶ Os elementos da lista são separados por vírgula

- ▶ Em Python podem existir listas com elementos de tipos diferentes? Sim! Embora não seja algo comum...

```
1 lista_elementos_diferentes = ['IRRF', 127, 9.75, True ]
```

Lista / Acessando elementos

- Como fazemos para acessar o valor de um elemento em uma lista?

```
1 lista_numeros = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
2
3 # Acessando o valor de um elemento
4 lista_numeros[3]
5
6 # Acessando exibindo o valor de um elemento
7 print(lista_numeros[3])
8
9 # Somando valores dos elementos e exibindo-os
10 print(lista_numeros[1] + lista_numeros[9])
```

- Listas são mutáveis, ou seja, seus valores podem ser atualizados

```
1 lista_nomes = ['Maria', 'João', 'José']
2 lista_nomes[2] = 'Josué'
3
4 print(lista_nomes)
```

Lista / Tamanho / Função len(s)

► Saber o seu tamanho

```
1 lista_impostos = ['IRRF', 'IPVA', 'IPTU', 'ISS', 'ICMS']  
2 tamanho = len(lista_impostos)
```

- Obter trechos (*slice notation*) [início, fim, incremento]

```
1 lista_impostos = ['IRRF', 'IPVA', 'IPTU', 'ISS', 'ICMS']
2
3 ultimo_elemento = lista_impostos[-1]
4 elementos_selecionados = lista_impostos[1:3]
5
6 lista_impostos = ['IRRF', 'IPVA', 'IPTU', 'ISS', 'ICMS']
7 lista_impostos[0:5] = ["Sem Impostos"]
8 print(lista_impostos)
```

- ▶ Em Python o if avalia uma lista vazia como falso

```
1 lista = [ ]
2
3 if lista:
4     print('A lista está vazia')
5 else:
6     print('A lista possui elementos')
```

```
1 lista = [ ]
2
3 if not lista:
4     print('A Lista está vazia')
```

- ▶ Como percorrer uma lista?

```
1 pessoas = ['Ana', 'Isabel', 'Mário', 'Goku']  
2  
3 for pessoa in pessoas:  
4     print(pessoa)
```

- ▶ Como podemos obter a posição (índice) do elemento na lista?

```
1 pessoas = ['Ana', 'Isabel', 'Mário', 'Goku']  
2  
3 for indice, pessoa in enumerate(pessoas):  
4     print('Posição: ', indice, ' Pessoa: ', pessoa)
```

```
1 # Concatenando listas
2 pessoas = ['Ana', 'Isabel', 'Mário', 'Goku']
3 pessoas += ['Marcos']
4 pessoas += ['Maria', 'Adriano']
5
6 for indice, pessoa in enumerate(pessoas):
7     print('Posição: ', indice, ' Pessoa: ', pessoa)
```

Lista / Adicionando elementos / Função append(elemento)

- Insere no final da lista

```
1 # Adicionando elementos com a função append(s)
2 pessoas = ['Ana', 'Isabel', 'Mário', 'Goku']
3 pessoas.append('Marcos')
4 pessoas.append('Maria')
5 pessoas.append('Adriano')
6
7 for indice, pessoa in enumerate(pessoas):
8     print('Posição: ', indice, ' Pessoa: ', pessoa)
9
```

Lista / Adicionando elementos / Função insert(posicao, elemento)

- E se eu quiser adicionar em uma posição específica?

```
1 lista = ['João', 'José', 'Gustavo', 'Maria', 'Adriana', 'Rita']
2 lista.insert(4, 'Sávio')
3
4 print(lista)
```

Lista / Removendo elementos / Função pop(indice)

- Por padrão remove o último elemento

```
1 lista = ['João', 'José', 'Gustavo', 'Maria', 'Rita', 'Adriana']
2
3 lista.pop()
4
5 print(lista)
6
7 lista.pop(3)
8
9 print(lista)
```

Lista / Removendo elementos / Função del(s)

```
1 # Adicionando elementos com a função del(s)
2 pessoas = ['Ana', 'Isabel', 'Mário', 'Goku', 'Marcos', 'Maria',
3           ↪ 'Adriano']
4 del(pessoas[0])
5
6 for indice, pessoa in enumerate(pessoas):
7     print('Posição: ', indice, ' Pessoa: ', pessoa)
```

Lista / Removendo elementos / Função del(s)

- Permite que seja especificado um intervalo

```
1 # Removendo elementos com a função del(s) e slice notation
2 pessoas = ['Ana', 'Isabel', 'Mário', 'Goku', 'Marcos', 'Maria',
3           ↪ 'Adriano']
4
5 del(pessoas[0:4])
6
7 for indice, pessoa in enumerate(pessoas):
8     print('Posição: ', indice, ' Pessoa: ', pessoa)
```

Lista / Removendo todos os elementos / Função clear()

```
1 lista = ['João', 'José', 'Gustavo', 'Maria', 'Adriana', 'Rita']
2 lista.clear()
3
4 print(lista)
```

Lista / Ordenando / Função sort()

```
1 lista = ['João', 'José', 'Gustavo', 'Maria', 'Rita', 'Adriana']
2 lista.sort()
3
4 print(lista)
5
6 lista.sort(reverse = True)
7
8 print(lista)
```

Lista / Contando / Funções count(elemento) e index(elemento)

```
1 lista = ['João', 'José', 'Rita', 'Gustavo', 'Rita', 'Maria',  
    ↪     'Rita', 'Adriana']  
2  
3 qtdVezes = lista.count('Rita')  
4  
5 print(qtdVezes)  
6  
7 indice = lista.index('Rita')  
8  
9 print(indice)
```

- São listas que possuem sublistas.

```
1 lista = [['João', 'José', 'Gustavo'], ['Maria', 'Adriana', 'Rita']]
2
3 print(lista)
4
5 print(lista[0][1])
6
7 for pessoas in lista:
8     for nome in pessoas:
9         print(nome)
```

-  Allen B. Downey; Think Python How to Think Like a Computer Scientist.