

# Computação I - Python

## Aula 5 - Listas

Apresentado por: Carolina G. Marcelino

Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>



- O que são Listas?

# Listas

- Tipo de dados mais versátil do Python.
- Uma lista é representada como uma sequência de valores entre colchetes e separados por vírgula.
- Os elementos de uma lista podem ser de tipos de dados diferentes.
- Listas são **mutáveis** !!!

## Exemplo

```
1 >>> lista1 = [ 'calculo ', 'fisica ', 'computacao ' ]
2 >>> lista2 = [ 'notas ', 5.4, 'aprovado ' ]
3 >>> lista2[1] = 6
4 >>> lista2
5 [ 'notas ', 6, 'aprovado ' ]
```

- **Atenção:** *Uma lista vazia não contém nenhum elemento*

## Exemplo

```
1 >>> lista3 = [ ]
2 >>> lista3[0]
3 Traceback (most recent call last):
4   File "<pyshell#18>", line 1, in <module>
5     lista3[0]
6 IndexError: list index out of range
```

# Listas

|       |   |     |  |          |
|-------|---|-----|--|----------|
| C[0]  | → | -45 |  |          |
| C[1]  | → | 6   |  | ← C[-13] |
| C[2]  | → | 3   |  | ← C[-12] |
| C[3]  | → | 0   |  | ← C[-11] |
| C[4]  | → | 1   |  | ← C[-10] |
| C[5]  | → | 19  |  | ← C[-9]  |
| C[6]  | → | 32  |  | ← C[-8]  |
| C[7]  | → | -23 |  | ← C[-7]  |
| C[8]  | → | 12  |  | ← C[-6]  |
| C[9]  | → | 5   |  | ← C[-5]  |
| C[10] | → | -3  |  | ← C[-4]  |
| C[11] | → | 8   |  | ← C[-3]  |
| C[12] | → | 2   |  | ← C[-2]  |
|       |   |     |  | ← C[-1]  |

```
1 >>> c=[ -45, 6, 3, 0, 1, 19, 32, -23, 12, 5, -3, 8, 2]
2 >>> c[3]
3 0
4 >>> c[9]==c[-4]
5 True
6 >>> len(c)
7 13
```

# Listas

```
1 >>> [1,2] + [3]
2
3 >>> [1,2] + [[3]]
4
5 >>> [[1,2]] + [[3]]
6
7 >>> [1,2] * 3
```

# Listas

```
1 >>> [1,2] + [3] # Concatenando Listas
2     [1, 2, 3]
3
4 >>> [1,2] + [[3]]
5     [1, 2, [3]]
6
7 >>> [[1,2]] + [[3]]
8     [[1, 2], [3]]
9
10 >>> [1,2] * 3 # Equivale a [1,2]+[1,2]+[1,2]
11     [1, 2, 1, 2, 1, 2]
```

# Listas

```
1 >>> [1,2] * [3]
2
3 >>> [1,2] - [3]
```



# Listas

```
1 >>> [1,2] * [3]
2     Traceback (most recent call last):
3       File "<pyshell#35>", line 1, in <module>
4         [1,2]*[3]
5     TypeError: cant multiply sequence by non-int of type '
    list '
6
7 >>> [1,2]-[3]
8     Traceback (most recent call last):
9       File "<pyshell#37>", line 1, in <module>
10        [1,2]-[2]
11    TypeError: unsupported operand type(s) for -: 'list' and
    'list '
```

Como retirar um elemento de uma lista? Aguarde

# Listas

Faça uma função que receba duas listas como entrada e retorne a concatenação destas listas.

# Listas

Faça uma função que receba duas listas como entrada e retorne a concatenação destas listas.

```
1 def concatenaListas(Lista1, Lista2):  
2  
3     """Funcao que dadas duas listas, retorna a concatenacao das  
4     listas.  
5     O parametros de entrada sao list, list.  
6     O valor de retorno e list."""  
7  
8     return Lista1+Lista2
```

```
1 >>> concatenaListas([1,2,3],[4,5,6])  
2 [1,2,3,4,5,6]
```

Faça uma função que dado um número inteiro como entrada, retorne uma lista com todos os números pares entre 1 e o número dado, inclusive.

# Listas

- A função `range(...)` pode ter 1, 2 ou 3 argumentos:

- `range(numero)`: retorna uma lista contendo uma sequência de valores de 0 a `numero-1`

```
1 >>> list(range(10))
2      [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

- `range(inf,sup)`: retorna uma lista contendo uma sequência de valores de `inf` a `sup-1`

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

- `range(inf, sup, inc)`: retorna uma lista contendo uma sequência de valores de `inf` a `sup-1` com incremento de `inc`

```
1 >>> list(range(3, 8, 2))
2      [3, 5, 7]
```

# Listas

- **ATENÇÃO:** A função `range(...)` começa com **zero**
- São equivalentes:

`range(10)`

`range(0,10)`

`range(0,10,1)`

- **Exemplos**

```
1 >>> list(range(3))
2
3 >>> list(range(2,5,2))
4
5 >>> list(range(5,2,-2))
```

# Listas

- **ATENÇÃO:** A função `range(...)` começa com **zero**
- São equivalentes:

`range(10)`

`range(0,10)`

`range(0,10,1)`

- **Exemplos**

```
1 >>> list(range(3))  
2 [0, 1, 2]  
3 >>> list(range(2, 5, 2))  
4 [2, 4]  
5 >>> list(range(5, 2, -2))  
6 [5, 3]
```

# Listas

Faça uma função que dado um número inteiro como entrada, retorne uma lista com todos os números pares entre 1 e o número dado, inclusive.



# Listas

Faça uma função que dado um número inteiro como entrada, retorne uma lista com todos os números pares entre 1 e o número dado, inclusive.

```
1 def lista(n):  
2  
3     """Funcao que dado um numero inteiro , retorna uma lista com todos  
4         os numeros pares entre 1 e o numero dado, inclusive.  
5         O parametros de entrada e um int.  
6         O valor de retorno e uma lista."""  
7  
8     return list(range(2,n+1,2))
```

```
1 >>> lista(5)  
2     [2,4]  
3  
4 >>> lista(6)  
5     [2,4,6]
```

- Resumo: nesta aula o tipo de dado Lista foi introduzido, bem como operações de manipulação de índice e o uso da função range para preenchimento.

## Autores

- **João C. P. da Silva** ▶ Lattes
- **Carla Delgado** ▶ Lattes
- **Ana Luisa Duboc** ▶ Lattes

## Colaboradores

- **Anamaria Martins Moreira** ▶ Lattes
- **Fabio Mascarenhas** ▶ Lattes
- **Leonardo de Oliveira Carvalho** ▶ Lattes
- **Charles Figueiredo de Barros** ▶ Lattes
- **Fabício Firmino de Faria** ▶ Lattes

# Computação I - Python

## Aula 5 - Listas

Apresentado por: Carolina G. Marcelino

Produção DCC-UFRJ

Metodologia de referência <https://doi.org/10.5753/wei.2016.9683>