

**INSTITUTO FEDERAL**

Catarinense

Campus Camboriú

## AULA 8

### (Subalgoritmos - parte II)

**Professora: Lidiane Visintin**

*lidiane.visintin@ifc.edu.br*

**Professor: Rafael de Moura Speroni**

*rafael.speroni@ifc.edu.br*

# Passagem de Parâmetros:

---

O tipo de passagem de parâmetro define se as modificações realizadas nos parâmetros **dentro** da função irão ou não se refletir **fora** da função.

- ❑ Por valor (cópia);  
Não afeta o algoritmo “que fez a chamada”.
- ❑ Por referência(endereço);  
Afeta o dado do algoritmo principal.

# Passagem de Parâmetro por valor

```
Exemplo_1.py > ...  
1  def teste(v1):  
2      #comandos na função  
3      print(v1)  
4      v1 = v1 + 2  
5      print(v1)  
6  
7  v = 10  
8  teste(v)  
9  
10 print(v)
```

10

12

10

# Passagem de Parâmetro por referência

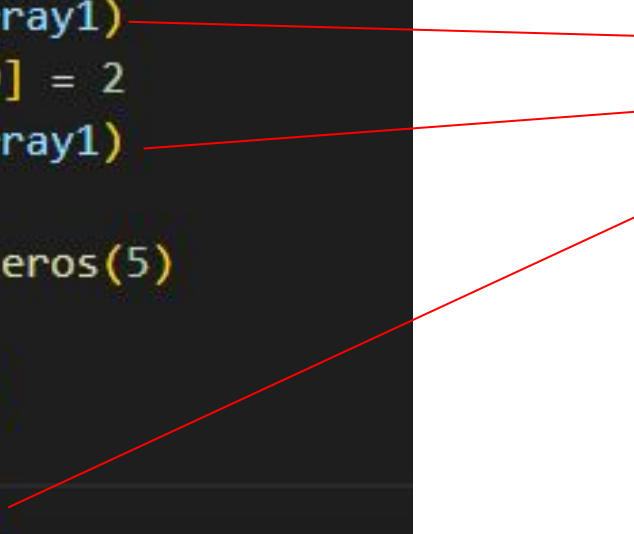
```
Exemplo_2.py > ...  
1  def teste(lista1):  
2      #comandos na função  
3      print(lista1)  
4      lista1.append(2)  
5      print(lista1)  
6  
7  lista = [10,11,12]  
8  |  
9  teste(lista)  
10  
11 print(lista)
```

[10, 11, 12]  
[10, 11, 12, 2]  
[10, 11, 12, 2]

# Passagem de Parâmetro por referência

- O mesmo ocorre com arrays;

```
Exemplo3.py > ...  
1  import numpy as np  
2  
3  def teste(array1):  
4      #comandos na função  
5      print(array1)  
6      array1[0] = 2  
7      print(array1)  
8  
9  array = np.zeros(5)  
10  
11  teste(array)  
12  
13  print(array)
```



[0. 0. 0. 0. 0.]
[2. 0. 0. 0. 0.]
[2. 0. 0. 0. 0.]

# Exercícios

---

1. Faça uma sub-rotina que recebe duas listas A e B com dez elementos inteiros como parâmetro. A sub-rotina deverá determinar e mostrar o vetor C que contém os elementos que estão em A, mas que não estão em B. A lista C deverá ser mostrada no programa principal.
2. Faça uma sub-rotina que receba como parâmetro uma matriz (lista de lista) A[5][5] e retorne a soma de todos os seus elementos.
3. Crie uma sub-rotina que receba como parâmetro um vetor V[25] de números inteiros e substitua todos os valores negativos de A por 0. O vetor deverá ser mostrado no programa principal.

# Referências



## Referências Básicas

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. **Lógica de programação: a construção de algoritmos e estruturas de dados**. 3. ed. Pearson Prentice Hall. 2005

MANZANO, José Augusto N. G; OLIVEIRA, Jayr Figueiredo de. **Algoritmos: lógica para desenvolvimento de programação de computadores**. 27. ed.. Érica. 2014

## Referências Complementares

MENEZES, Nilo Ney de Coutinho. **Introdução a programação com Python**. 3ª Ed. Novatec. 2019.

CORMEN, Thomas H et al. **Algoritmos: teoria e prática**. 2. ed. Elsevier, Campus,. 2002

## Referências na Internet

<https://docs.python.org/3/>

<https://www.w3schools.com/python/default.asp>

<https://panda.ime.usp.br/pensepy/static/pensepy/05-Funcoes/funcoes.html>