

Subprogramação + Vetores

Vanessa Braganholo
vanessa@ic.uff.br

Tipos de passagem de Parâmetro

- ▶ **Por valor:** o valor da variável na chamada é copiado para a variável da função.
 - ▶ Alterações não são refletidas na variável original
- ▶ **Por referência:** é como se o mesmo “escaninho” fosse usado.
 - ▶ Alterações são refletidas na variável original



Tipos de passagem de Parâmetro

- ▶ **Por valor:** o valor da variável na chamada é copiado para a variável da função.
 - ▶ Alterações não são refletidas na variável original

Python usa passagem de parâmetro por valor

Uso de vetores como parâmetros de funções

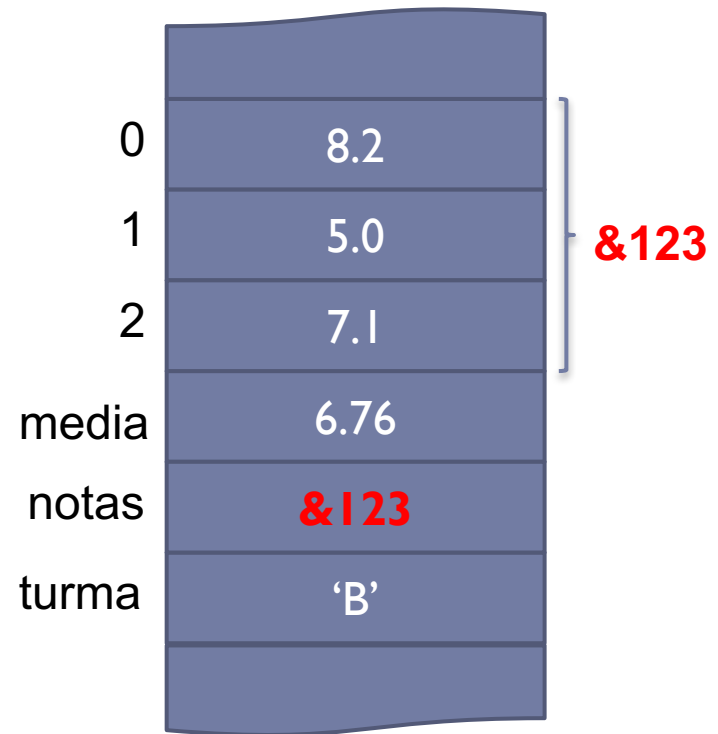
- ▶ Python usa passagem de parâmetro por valor
 - ▶ Faz cópia do valor da variável original para o parâmetro da função
 - ▶ Variável original fica preservada das alterações feitas dentro da função
- ▶ Exceção: **vetores** (ou objetos) funcionam de forma diferente, pois o **valor** de um vetor é seu **endereço**!

Representação de Vetores em Memória

Em Python

```
notas = [8.2, 5.0, 7.1]
turma = 'B'
media = 0
for i in range(len(notas)):
    media = media + notas[i]
media = media/len(notas)
```

Na Memória




Uso de vetores como parâmetros de funções

- ▶ Dessa forma, o que é copiado é o **endereço** do vetor, e portanto qualquer alteração é refletida no programa principal → passagem de parâmetro por referência

Exemplo

```
def maior(vetor):  
    #ordena o vetor usando Bubble Sort  
    if len(vetor) > 1:  
        for j in range(0, len(vetor)):  
            for i in range(0, len(vetor)-1):  
                if vetor[i] > vetor[i+1]:  
                    aux = vetor[i+1]  
                    vetor[i+1] = vetor[i]  
                    vetor[i] = aux  
    #retorna o último elemento do vetor  
    return vetor[len(vetor)-1]
```

```
v = [5, 4, 3, 2, 1]  
print(v)  
m = maior(v)  
print(m)  
print(v)
```



O que será
impresso na tela?

Exercícios (usar funções sempre que possível)

1. Faça um programa que leia N pares de números, e forneça como resultado a multiplicação dos maiores números de cada par. Use função para encontrar os maiores, e para calcular a multiplicação. As funções devem se comportar da seguinte forma:

função **maior**: recebe como parâmetro de entrada uma matriz contendo os pares (cada par é uma linha da matriz), e retorna como resultado um vetor contendo os maiores elementos de cada par.

função **multiplica**: recebe como parâmetro o vetor de maiores elementos, e retorna como resultado um valor que é o resultado da multiplicação dos elementos do vetor.

Exemplo: entrada do programa $n = 3$, $v = [[2,3], [4,5], [6, 7]]$. Função maior recebe v como entrada e retorna um vetor $m = [3, 5, 7]$. Função multiplica recebe como entrada o vetor m e retorna o valor 105.



Exercícios (usar funções sempre que possível)

2. Faça uma função que recebe como parâmetro de entrada um vetor de tamanho qualquer, e retorna como resultado o vetor invertido. Exemplo: ao receber como entrada o vetor $v = [3, 5, 7, 9]$, a função deve retornar o vetor $x = [9, 7, 5, 3]$.

Exercícios (usar funções sempre que possível)

3. Uma sorveteria vende cinco produtos diferentes, cada um com um preço de acordo com a tabela abaixo. Faça um programa que processe diversas vendas, lembrando que cada venda efetuada pode ser composta por diversas unidades de diversos produtos. O programa deverá utilizar:

- (a) uma função que apresente na tela um menu indicando os preços dos produtos. Esse menu deve ser apresentado no início de cada venda.
- (b) uma função que processe cada venda individual e forneça o valor a pagar.
- (c) uma terceira função que emita um relatório no final do dia, informando dados gerais das vendas do dia (número total de itens vendidos de cada produto, total pago para cada produto, total arrecadado e valor médio de cada compra)

Cod	Produto	Preço (R\$)
A	Refrigerante	3,50
B	Casquinha Simples	4,00
C	Casquinha Dupla	5,50
D	Sundae	7,50
E	Banana Split	9,00

Referências

- ▶ Slides baseados nas aulas de Leonardo Murta e Aline Paes

Subprogramação + Vetores

Vanessa Braganholo
vanessa@ic.uff.br