

Vetores em Python

Fabio Lubacheski
fabio.lubacheski@mackenzie.br

Vetores

- Um **VETOR** é uma **estrutura de dados linear** que armazena uma **sequência de elementos**, todos do **MESMO TIPO**, em um **TAMANHO FIXO** de posições **consecutivas** na memória.
- A posição de um elemento é dado pelo valor do seu **índice**.
- A **primeira posição** de um vetor possui **índice 0**, a **quarta** possui **índice 3**, e assim por diante. Para um vetor com **10 posições** temos:

0	1	2	3	4	5	6	7	8	9
10	55	2	5	54	24	32	12	6	8

- Note que o último índice é sempre a **quantidade de posições menos 1**.

Vetores X Listas

- Em Python para armazenar uma coleção de elementos temos **Classe List** (LISTA) que possui uma série de funções, métodos e operadores, por exemplo:

append(x) – método adiciona um item ao fim da lista.

insert(i,x) – método insere um elemento x antes da posição i

remove(x) – método remove a primeira ocorrência de x na lista.

pop(i) – método remove o elemento i da lista. Se usado sem valor pop() remove e retorna o último elemento da lista.

len(L) – função retorna o número de elementos da lista L.

index(x) – método retorna a posição(=índice) do elemento cujo valor seja igual ao de x.

count(x) – método retorna o número de vezes que x aparece na lista.

in - operador in permite saber se um elemento pertence a uma lista

Vetores X Listas

- As **LISTAS** em Python podem ter **TAMANHO INDETERMINADO** ou ser **VAZIAS**, e também podem armazenar elementos de **DIFERENTES TIPOS**.
- As **LISTAS** em Python são estruturas de armazenamento em linguagens orientadas a objetos, e como estamos usando uma linguagem imperativa devemos usar **VETORES**.
- Mais informação sobre Listas:
<https://python.ime.usp.br/pensepy/static/pensepy/09-Listas/listas.html>

http://programarcadegames.com/index.php?chapter=introduction_to_lists&lang=pt-br#section_7

Vetores

- Os **VETORES**, em Python, são implementados usando **Listas** (Classe), mas devem ter **TAMANHO FIXO** e todos os elementos devem ser do **MESMO TIPO**.
- Para diferenciar totalmente **VETORES** de **LISTAS** quando tratarmos de **VETORES não usaremos** os métodos, funções e operadores de **LISTAS**, somente utilizaremos a função **len()**.
- O objetivo dessa restrição é para que o aluno aprenda manipular os elementos de uma **estrutura de dados linear**, pois as facilidades oferecidas no Python, não estão presentes em todas as linguagens de programação imperativa, como por exemplo a linguagem C.

Vetores

- Para criar um **VETOR** e inicializa-lo com zero basta digitar o código abaixo, no exemplo é criada uma variável vetor com 10 posições zeradas e de tamanho fixo.

```
>>> vetor = [0]*10  
>>> print(vetor)  
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

- Os elementos de um **VETOR** podem ser referenciados através de um índice, começando com 0 até o último índice, no exemplo acima o último índice é o 9.

```
>>> vetor[10] # o que acontece no código abaixo ?
```

Inserindo elementos em um Vetor

- Uma vez declarada a variável **VETOR**, podemos atribuir qualquer conjunto de 10 valores inteiros à variável, **um valor por vez a cada das posições do vetor**.

```
>>> vetor[0]=10
```

```
>>> vetor[1]=20
```

```
.....
```

```
>>> vetor[9]=90
```

- Como poderíamos ler todos os elementos do **VETOR**.

Exercícios

1) Escreva uma função que recebe um vetor de número inteiros, e verifica se os elementos do vetor estão em ordem crescente, função retorna **True** caso os elementos estejam em ordem decrescente e **False** caso contrário.

2) Escreva uma função que recebe um vetor e devolve o vetor invertido, para o vetor abaixo:

`V = [4, 9, 10, 8, 6]` o vetor `V` invertido é igual a `[6, 8, 10, 9, 4]`

Resolva essa questão **sem utilizar o método `reverse()`** para listas e **sem usar um vetor auxiliar**.

Exercícios

- 3) Escreva uma função que recebe dois vetores inteiros $A[]$ e $B[]$, em seguida, a sua função efetua a INTERSECÇÃO entre os vetores, ou seja, os elementos em comum entre os dois vetores, ao final sua função retorna uma String com a resposta. Os vetores dados não possuem valores duplicados e não estão ordenados.

Exemplo:

$A[] = \{ 7, 2, 5, 8, 4 \}$ e $B[] = \{ 4, 2, 9, 5 \}$ então

$A \cap B = \{ 2, 5, 4 \}$

$A[] = \{ 3, 9, 11 \}$ e $B[] = \{ 2, 6, 1 \}$ então

$A \cap B = \{ \}$

Exercícios

- 4) Repita o exercício anterior, agora deve ser retornado em uma String os elementos que estão em A[] mas não estão em B[], ou seja, a diferença de A – B, por exemplo:

$A[] = \{ 7, 2, 5, 8, 4 \}$ e $B[] = \{ 4, 2, 9, 5 \}$ então

$A - B = \{ 7, 8 \}$

$A[] = \{ 3, 9, 11 \}$ e $B[] = \{ 2, 6, 1 \}$ então

$A - B = \{ 3, 9, 11 \}$

Exercícios

- 5) Escreva uma função que recebe dois vetores inteiros $A[]$ e $B[]$, em seguida, a sua função efetua a UNIÃO entre os vetores, ou seja, os elementos do vetor A mais aos elementos do vetor B , desde que os elementos de B não estejam presentes no vetor A , ao final sua função retorna uma String com a resposta. Os vetores dados não possuem valores duplicados e não estão ordenados.

Exemplo:

$A[] = \{ 7, 2, 5, 8, 4 \}$ e $B[] = \{ 4, 2, 9, 5 \}$ então

$A \cup B = \{ 7, 2, 5, 8, 4, 9 \}$

$A[] = \{ 3, 9, 11 \}$ e $B[] = \{ 2, 6, 1 \}$ então

$A \cup B = \{ 3, 9, 11, 2, 6, 1 \}$

FIM !