



VETOR E MATRIZ

Em geral são caracterizados por se tratarem de uma única variável de um determinado tamanho que armazena várias informações do mesmo tipo.

Essas informações são gravadas na memória e são referenciadas através de índices.

Chamamos de vetor as variáveis unidimensionais e como matriz as multidimensionais.

VETOR

Possui uma única dimensão. A declaração de vetores em Linguagem C deve obedecer a seguinte sintaxe:

nome_vetor = [];

O tipo será especificado de forma dinâmica, como nos demais exemplos vistos durante o curso.

O primeira posição ou índice de um vetor é 0 (zero).

VETOR: EXEMPLOS

Para entender melhor, este é um exemplo de declaração de um vetor vazio.

Declaração: **vetor_exemplo = []**

Também é possível inicializar o vetor no momento de sua declaração.

Sintaxe: **nome_vetor = [10, 65, 8, 43, 29, 12]**

Todos os elementos devem ser separados por vírgula e serem todos do mesmo tipo.

EXEMPLO: ATRIBUIR E LISTAR UM VETOR

Exemplo inicializando um vetor de inteiros e listando os dados armazenados acessando cada posição ou cada item do vetor.

```
C:\PythonCurso\venv\Scripts\python.exe C:/PythonCurso/exemplo25.py
```

```
5
```

```
64
```

```
87     vetor = [5, 64, 87, 26, 44, 31]
26     for indice in range(len(vetor)):
44         print(vetor[indice])
```

```
31
```

```
vetor = [5, 64, 87, 26, 44, 31]
for indice in range(6):
    print(vetor[indice])
```

```
vetor = [5, 64, 87, 26, 44, 31]
for item in vetor:
    print(item)
```

Posição do vetor

Item do vetor

LISTAS SÃO VETORES

Em outras linguagens de programação, listas são chamadas de vetores e possuem restrições, que Python não impõe:

Em Python, os valores de uma lista podem ser de qualquer tipo, inclusive contendo tipos diferentes de dados em uma mesma variável.

Exemplos: `lista1 = ['X', 5, 9.5, 'curso', True]`

`lista2 = [10.99, 3, 50.5, -3.5, 15.0]`

EXEMPLO: LISTA – ENTRADA E EXIBIÇÃO

Deverá ser armazenado o nome e nota de 3 aluno, digitados pelo usuário. Ao final deverá ser exibido o nome e nota de cada aluno. Usar uma lista para nomes e outra para notas.

A função **append** deve ser usada para adicionar valores em uma lista.

```
1  alunos = []
2  notas = []
3
4  for i in range(3):
5      alunos.append(input('Nome do(a) ' + str(i+1) + 'º aluno(a): '))
6      notas.append(eval(input('Digite a nota do(a) ' + alunos[i] + ': ')))
7
8  for i in range(3):
9      print('\n Aluno(a):', alunos[i], ': nota: ', notas[i])
```

RESULTADO: LISTA – ENTRADA E EXIBIÇÃO

```
C:\PythonCurso\venv\Scripts\python.exe C:/PythonCurso/exemplo26.py
Nome do(a) 1º aluno(a): Leonardo
Digite a nota do(a) Leonardo: 7.9
Nome do(a) 2º aluno(a): Ana Clara
Digite a nota do(a) Ana Clara: 7.1
Nome do(a) 3º aluno(a): Luana
Digite a nota do(a) Luana: 9.3

Aluno(a): Leonardo : nota: 7.9

Aluno(a): Ana Clara : nota: 7.1

Aluno(a): Luana : nota: 9.3
```


EXEMPLO: LISTA DE FUNCIONÁRIOS

Armazenar o nome e o salário de 3 funcionários. Após o cadastro destas informações, deverá ser digitado o índice de reajuste salarial para os funcionários. O programa deverá calcular o novo salário e exibir os dados na tela.

```
1  import locale
2
3  func = []
4  sal = []
5
6  for i in range(3):
7      func.append(input('Nome do(a) funcionário(a): '))
8      sal.append(eval(input('Digite o salário de ' + func[i] + ': ')))
9
10     reajuste = float(input('Digite o índice para cálculo do reajuste salarial: '))
11
12     for i in range(3):
13         sal[i] = sal[i] + (sal[i] * reajuste) / 100
14         print('\n Aluno(a):', func[i], ' Salário: ', locale.format_string("%.2f", sal[i], 0))
```

RESULTADO: LISTA DE FUNCIONÁRIOS

```
C:\PythonCurso\venv\Scripts\python.exe C:/PythonCurso/exemplo27.py
Nome do(a) funcionário(a): Claudio
Digite o salário de Claudio: 2999.65
Nome do(a) funcionário(a): Adilson
Digite o salário de Adilson: 3856.45
Nome do(a) funcionário(a): Cleber
Digite o salário de Cleber: 3331.99
Digite o índice para cálculo do reajuste salarial: 7.15

Aluno(a): Claudio Salário: 3214.12

Aluno(a): Adilson Salário: 4132.19

Aluno(a): Cleber Salário: 3570.23
```

EXEMPLO: LISTA - STRING

Cadastrar uma frase, e exibir a quantidade de palavras, vogais, consoantes e símbolos da frase.

```
1  vogais = consoantes = simbolos = espaco = 0
2
3  texto = input('Digite uma frase: ')
4
5  for caracter in texto.lower():
6      if caracter in 'aeiou':
7          vogais = vogais + 1
8      elif caracter in ' ':
9          espaco = espaco + 1
10     elif caracter.isalpha():
11         consoantes = consoantes + 1
12     else:
13         simbolos = simbolos + 1
14
15     print('Quantidade de palavras: ', espaco+1)
16     print('Quantidade de vogais: ', vogais)
17     print('Quantidade de consoantes: ', consoantes)
18     print('Quantidade de símbolos: ', simbolos)
```

RESULTADO: LISTA STRING

```
C:\PythonCurso\venv\Scripts\python.exe C:/PythonCurso/exemplo28.py
Digite uma frase: Estamos estudando: Python
Quantidade de palavras: 3
Quantidade de vogais: 8
Quantidade de consoantes: 14
Quantidade de símbolos: 1
```

Analizando melhor o código, após digitar a frase, convertamos primeiramente para letras minúsculas, antes de iniciar a comparação os caracteres com vogais.

Na sequência temos uma verificação se for espaço em branco e depois outra verificando se é consoante. E por fim, caso nenhuma das anteriores considera-se como sendo símbolo.

CONCATENAÇÃO DE LISTAS

Para que possamos juntar duas listas em uma única lista, devemos usar o sinal de adição. Importante lembrar que as listas originais não sofrem alterações.

```
lista1 = [5, 64, 87, 26, 44, 31]
lista2 = [1, 9, 19]

lista3 = lista1 + lista2

print('Lista 1: ', lista1)
print('Lista 2: ', lista2)
print('Lista 3: ', lista3)
```

```
C:\PythonCurso\venv\Scripts\python.exe C:/PythonCurso/exemplo29.py
Lista 1:  [5, 64, 87, 26, 44, 31]
Lista 2:  [1, 9, 19]
Lista 3:  [5, 64, 87, 26, 44, 31, 1, 9, 19]
```

INSERIR EM POSIÇÃO ESPECÍFICA NA LISTA

Podemos inserir um item, em uma posição específica da lista, indicando a posição/índice que deverá ser inserido o item.

```
lista = [1, 9, 19]
print(lista)

# inserindo item na posição 0
lista.insert(0, 52)
print(lista)

# inserindo item na posição 3
lista.insert(3, 74)
print(lista)
```

```
C:\PythonCurso\venv\Scripts\python.exe C:/PythonCurso/exemplo30.py
[1, 9, 19]
[52, 1, 9, 19]
[52, 1, 9, 74, 19]
```


APAGAR ITEM DE UMA LISTA

Para excluir um item da lista podemos usar a função **del** informando a posição que será removida ou a função **remove** informando o item à ser excluído.

```
lista = [59, 132, 845, 12, 45, 94]
print(lista)

# removendo item da posição 2
del lista[2]
print(lista)

# removendo item 59 da lista
lista.remove(59)
print(lista)
```

```
C:\PythonCurso\venv\Scripts\python.exe C:/PythonCurso/exemplo31.py
[59, 132, 845, 12, 45, 94]
[59, 132, 12, 45, 94]
[132, 12, 45, 94]
```

FATIAS DE LISTAS

Podemos usar a notação lista[*início:fim*] para acessar parte da lista, retornando os valores entre os índices início e fim-1.

Dada a lista: num = [2, 3, 5, 7, 11]

num[1:2]

retorna [3]

num[2:4]

retorna os números [5, 7]

num[:3]

Sem indicar a posição inicial, retorna desde o primeiro valor [2, 3, 5]

num[3:]

Sem indicar a posição final, retorna até o último valor [7, 11]

num[:]

Sem ambos, retorna a lista toda [2, 3, 5, 7, 11]

CLONES DE LISTA

Devemos prestar muita atenção ao igualar uma lista com outra, pois poderá perder toda lista original.

```
num = [2, 3, 5, 7, 11]
```

```
# Aqui p é uma referência a num
```

```
p = num
```

```
# Substitui o valor de p na posição [2] para 6
```

```
p[2] = 6
```

```
# imprime 6, ou seja, foi alterado o valor de num[2]
```

```
print(p[2])
```

Ou seja, o que alterar em **p**, altera também em **num**.

CLONES DE LISTA

Vamos fazer realmente um clone de `num = [2, 3, 5, 7, 11]`

Agora **p** é um clone de **num**

```
p = num[:]
```

Alterando o valor de **p[2]** para **6**

```
p[2] = 6
```

Imprime 5, não mudou o valor

```
print(primos[2])
```

Isso é especialmente importante quando for passar uma lista para uma função

EXEMPLO: CADASTRO E CONSULTA DE ESPORTES

O usuário deverá digitar 5 nomes de esportes. Na sequência deverá digitar um esporte para consultar se consta na lista. Uma mensagem deverá ser exibida se encontrou ou não o esporte na lista.

```
1  esportes = []
2  quantidade = 5
3
4  for i in range(quantidade):
5      esportes.append(input('Digite o nome de um esporte: '))
6
7  busca = input('Digite o nome de um esporte para consulta: ')
8
9  if busca in esportes:
10     print(busca, ' consta na relação de esportes cadastrados')
11 else:
12     print(busca, ' não consta na relação de esportes cadastrados')
```

RESULTADO: CADASTRO E CONSULTA DE ESPORTES

Podemos notar que não houve a necessidade de percorrer cada posição da lista para verificar a existência do elemento (texto digitado) para consulta.

```
C:\PythonCurso\venv\Scripts\python.exe C:/PythonCurso/exemplo34.py
Digite o nome de um esporte: futebol
Digite o nome de um esporte: natacao
Digite o nome de um esporte: volei
Digite o nome de um esporte: basquete
Digite o nome de um esporte: tenis de mesa
Digite o nome de um esporte para consulta: volei
volei consta na relação de esportes cadastrados
```