

Vetores em Python

- Em computação é comum manipular conjunto de valores como notas, preços e etc. Para manipular um conjunto de dados de um mesmo tipo (inteiro, real, string) há uma estrutura de dados denominada de **vetor**;
- **Vetores** são “agregados homogêneos unidimensionais” que permitem agrupar um conjunto de valores de um mesmo tipo em uma única variável.

Vetores em Python

- Em Python possui uma estrutura similar a vetores denominada **listas**;
- **Lista** é um conjunto ordenado de valores, onde cada valor é identificado por um **índice**;
- Os valores na lista são denominados **elementos**. Podemos denominar uma lista de agregado homogêneo unidimensional se todos os elementos são do mesmo tipo.

Vetores em Python

- Vamos definir um vetor “nota” de tamanho 5 de tipo inteiro, nota = [60, 95, 80, 50, 98]

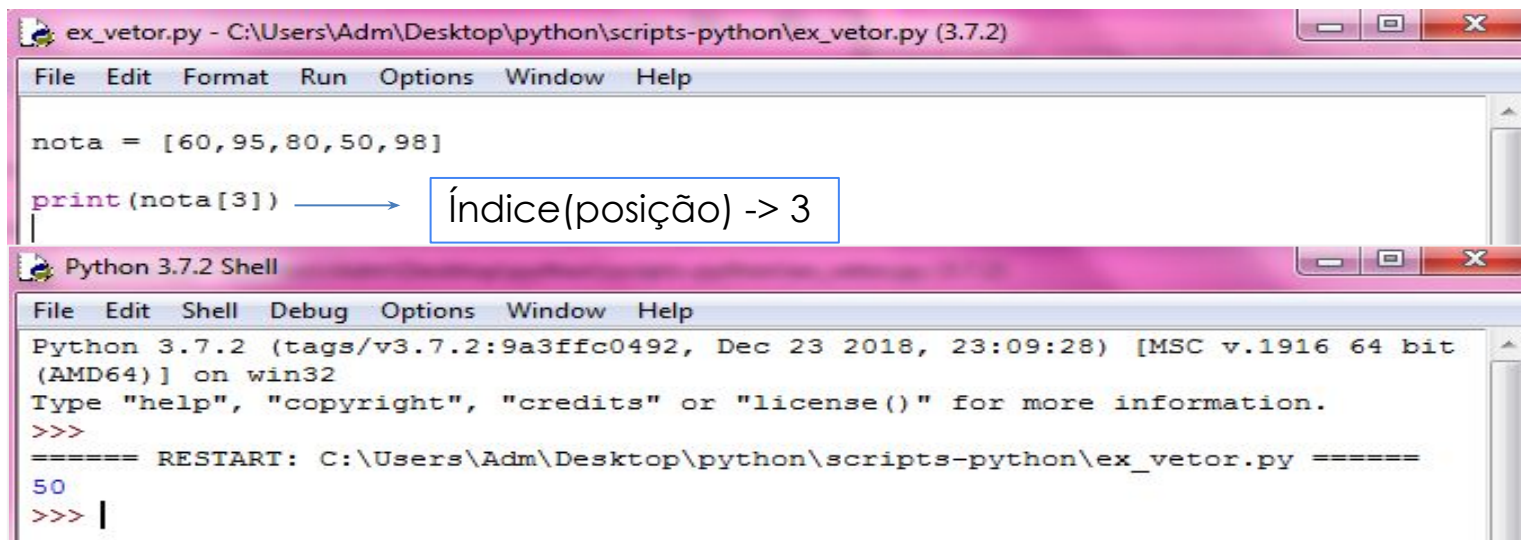


[] é o operador do elemento

Índice	0	1	2	3	4
Valor	60	95	80	50	98
Vetor	Nota[0]	Nota[1]	Nota[2]	Nota[3]	Nota[4]

Vetores em Python

- Agora vamos imprimir a nota 50 do vetor nota:



The image shows a screenshot of a Python IDE with two windows. The top window, titled 'ex_vetor.py - C:\Users\Adm\Desktop\python\scripts-python\ex_vetor.py (3.7.2)', contains the following code:

```
nota = [60, 95, 80, 50, 98]

print(nota[3])
```

An arrow points from the index '3' in the code to a blue-bordered box containing the text 'Índice(posição) -> 3'. The bottom window, titled 'Python 3.7.2 Shell', shows the execution output:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Adm\Desktop\python\scripts-python\ex_vetor.py =====
50
>>> |
```

Listas em Python

- 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, em outras linguagens, os valores precisam ser do mesmo tipo.
- Em Python
 - `lista = ['A', 1, 2, 'Casa', 2.3]`
 - `notas = [10, 5, 6.7, 2, 7.5]`

Listas em Python

- Existe vários tipos de listas criadas para poder trabalhar em Python:

Lista vazia, ou seja, sem elementos:

```
□ Lista_vazia = []
```

Lista com elementos inteiros:

```
□ Lista_inteiros = [2, 4, 6, 8, 10]
```

Lista com elementos reais:

```
□ Lista_reais = [9.0, 10.0, 8.5, 7.8]
```

Listas em Python

- Existe vários tipos de listas criadas para poder trabalhar em Python:

Lista com elementos string:

```
□ Lista_frutas = ["abacaxi", "pera", "uva", "abacate"]
```

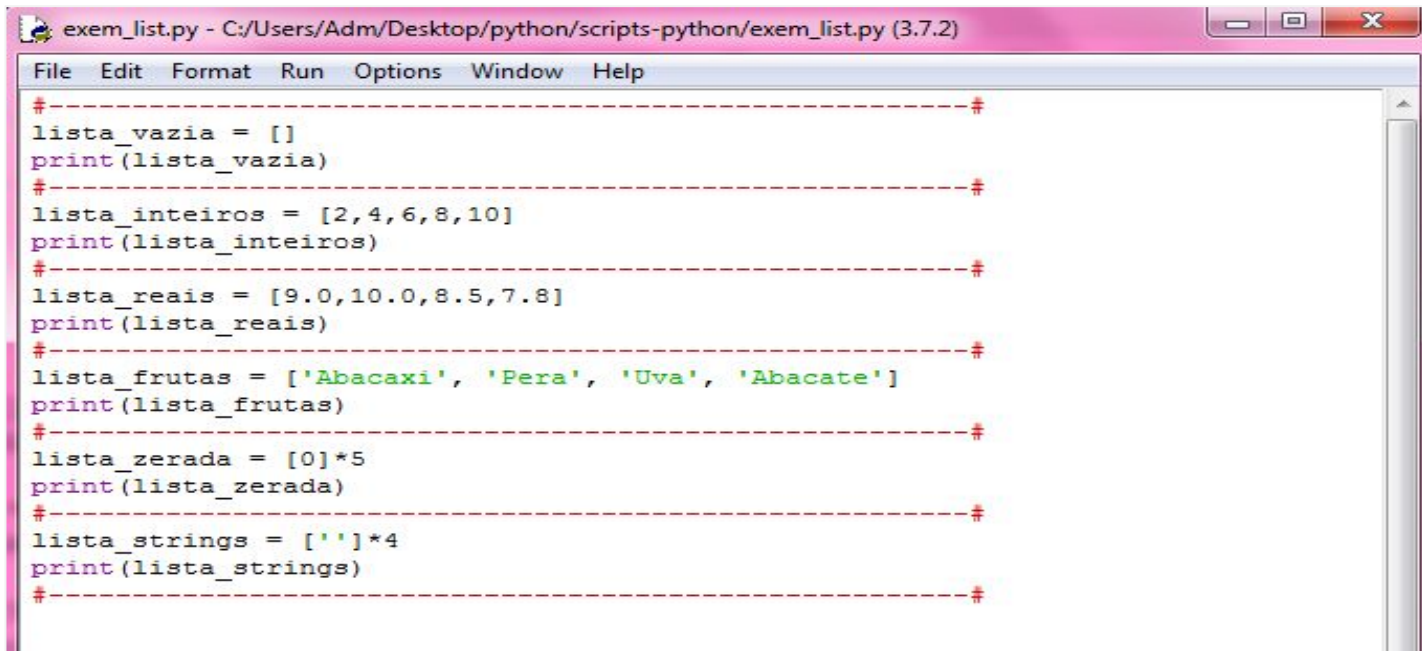
Lista de inteiros com tamanho 5 preenchida por zeros:

```
□ Lista_zerada = [0]*5
```

Lista de strings com tamanho 4 preenchida com strings vazias:

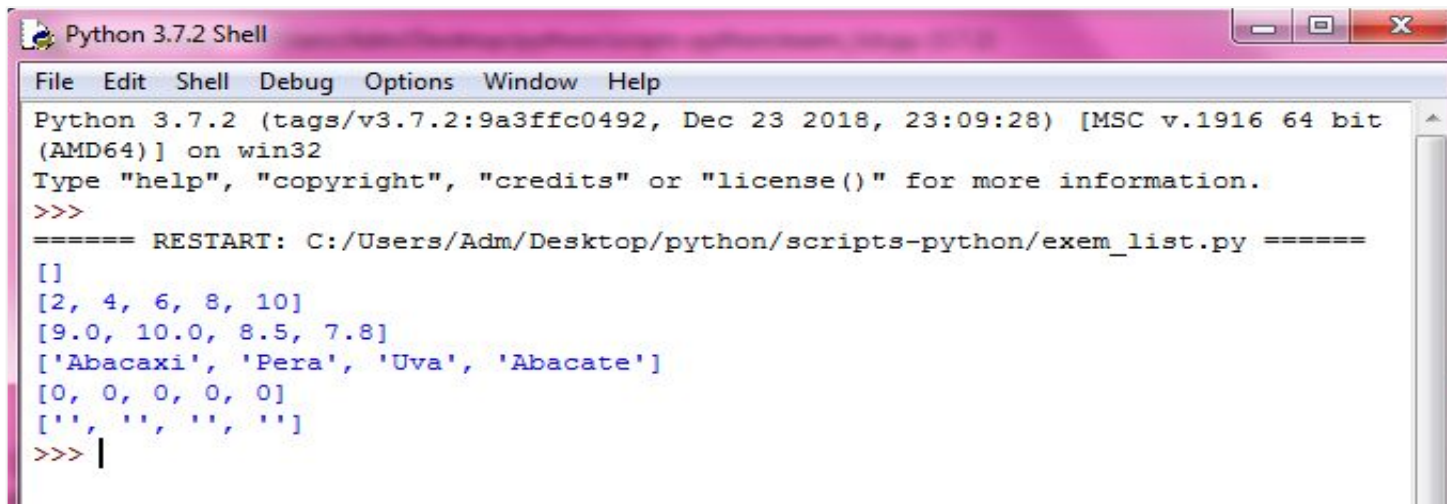
```
□ Lista_strings = [""]*4
```

Listas em Python




```
exem_list.py - C:/Users/Adm/Desktop/python/scripts-python/exem_list.py (3.7.2)
File Edit Format Run Options Window Help
#-----#
lista_vazia = []
print(lista_vazia)
#-----#
lista_inteiros = [2,4,6,8,10]
print(lista_inteiros)
#-----#
lista_reais = [9.0,10.0,8.5,7.8]
print(lista_reais)
#-----#
lista_frutas = ['Abacaxi', 'Pera', 'Uva', 'Abacate']
print(lista_frutas)
#-----#
lista_zerada = [0]*5
print(lista_zerada)
#-----#
lista_strings = ['']*4
print(lista_strings)
#-----#
```


Listas em Python

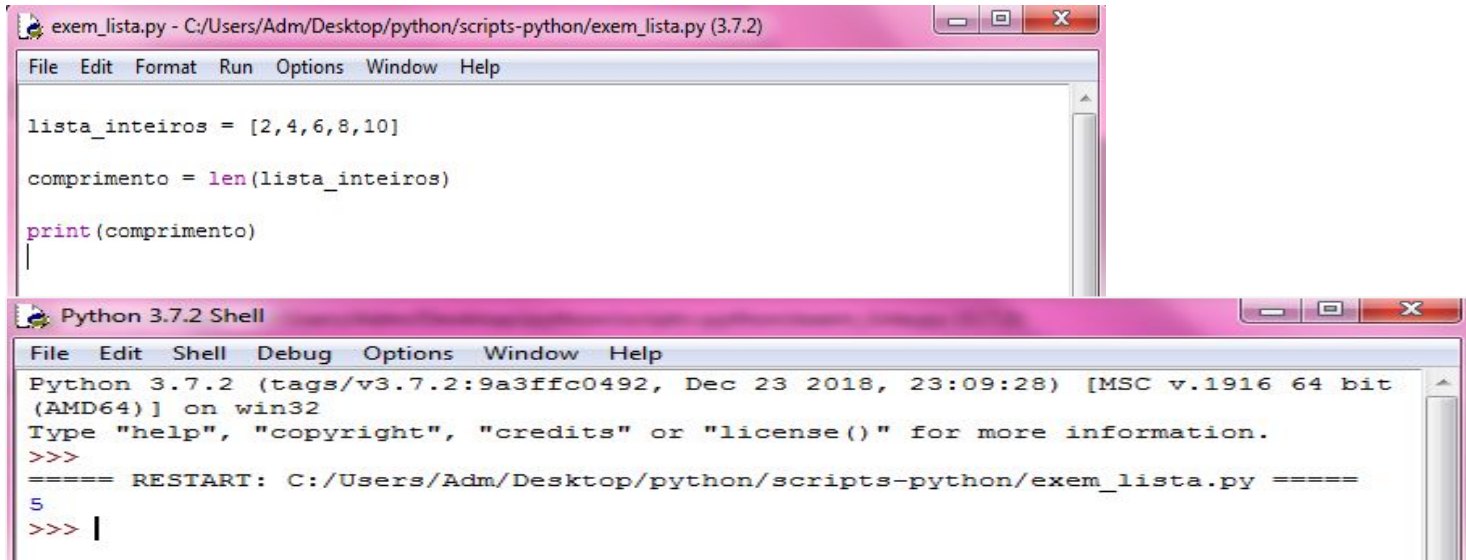


```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exem_list.py =====
[]
[2, 4, 6, 8, 10]
[9.0, 10.0, 8.5, 7.8]
['Abacaxi', 'Pera', 'Uva', 'Abacate']
[0, 0, 0, 0, 0]
['', '', '', '']
>>> |
```

Listas em Python

- O tamanho de uma lista é devolvido pela função `len()`
Comprimento. 

Lenght =



```
exem_lista.py - C:/Users/Adm/Desktop/python/scripts-python/exem_lista.py (3.7.2)
File Edit Format Run Options Window Help

lista_inteiros = [2,4,6,8,10]

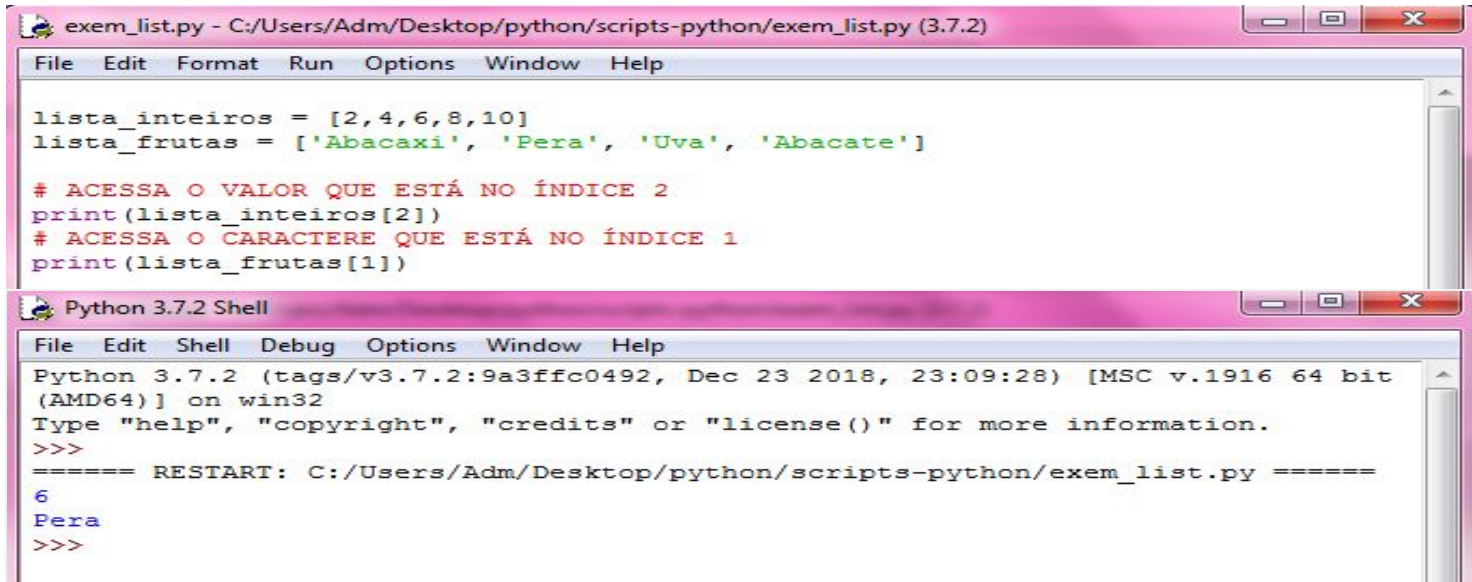
comprimento = len(lista_inteiros)

print(comprimento)

Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exem_lista.py =====
5
>>> |
```

Listas em Python

- Para acessar os elementos da lista é necessário trabalhar com os índices.



The image shows a screenshot of a Python IDE window titled "exem_list.py - C:/Users/Adm/Desktop/python/scripts-python/exem_list.py (3.7.2)". The window contains the following Python code:

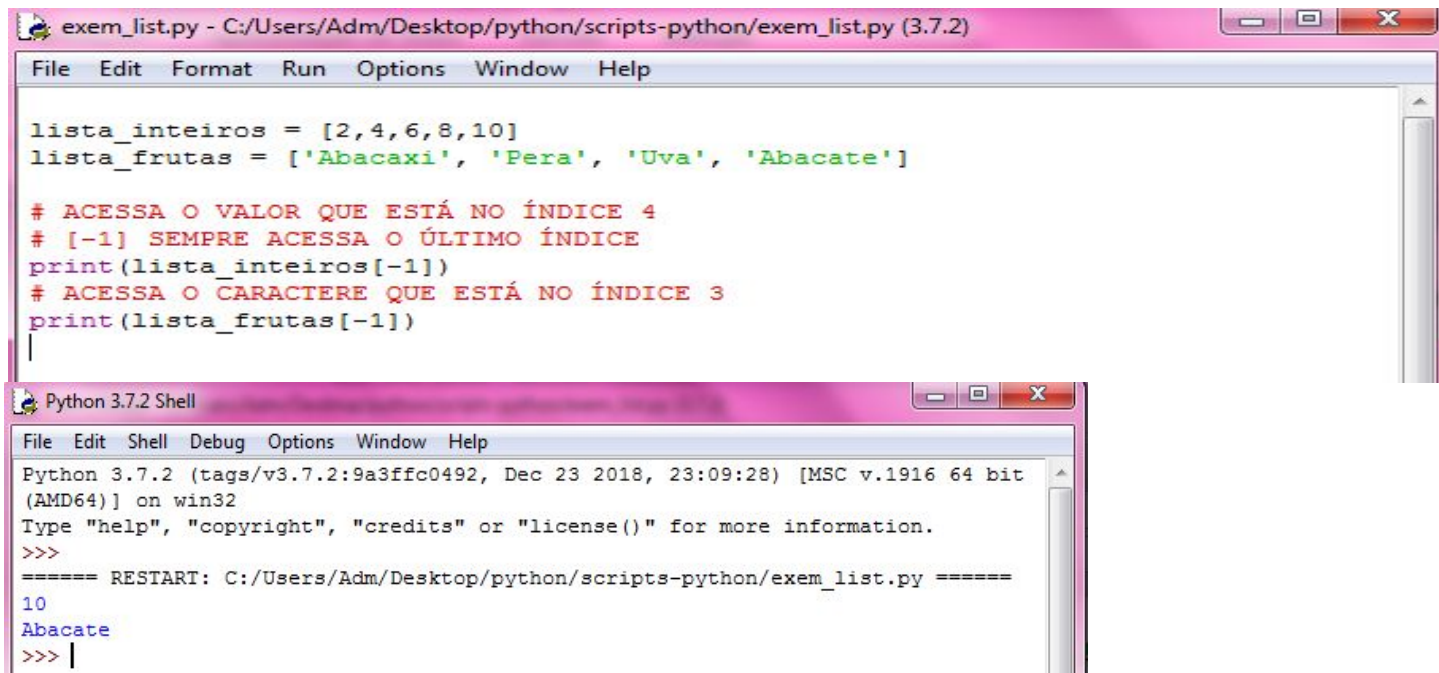
```
lista_inteiros = [2,4,6,8,10]
lista_frutas = ['Abacaxi', 'Pera', 'Uva', 'Abacate']

# ACESSA O VALOR QUE ESTÁ NO ÍNDICE 2
print(lista_inteiros[2])
# ACESSA O CARACTERE QUE ESTÁ NO ÍNDICE 1
print(lista_frutas[1])
```

Below the script editor is a "Python 3.7.2 Shell" window showing the execution output:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exem_list.py =====
6
Pera
>>>
```

Listas em Python



The image shows two windows from a Python IDE. The top window, titled 'exem_list.py - C:/Users/Adm/Desktop/python/scripts-python/exem_list.py (3.7.2)', contains the following Python code:

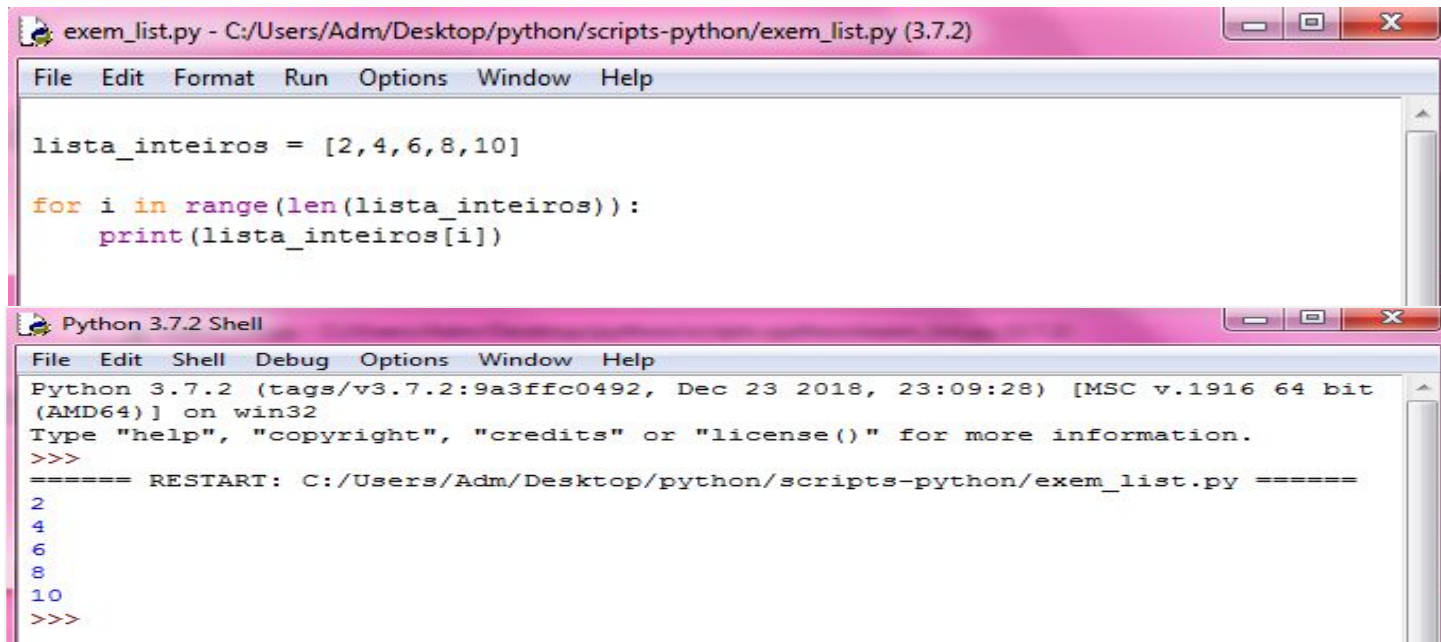
```
lista_inteiros = [2,4,6,8,10]
lista_frutas = ['Abacaxi', 'Pera', 'Uva', 'Abacate']

# ACESSA O VALOR QUE ESTÁ NO ÍNDICE 4
# [-1] SEMPRE ACESSA O ÚLTIMO ÍNDICE
print(lista_inteiros[-1])
# ACESSA O CARACTERE QUE ESTÁ NO ÍNDICE 3
print(lista_frutas[-1])
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution output:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exem_list.py =====
10
Abacate
>>> |
```

Listas em Python



The image shows two windows from a Python IDE. The top window, titled 'exem_list.py - C:/Users/Adm/Desktop/python/scripts-python/exem_list.py (3.7.2)', contains the following Python code:

```
lista_inteiros = [2,4,6,8,10]

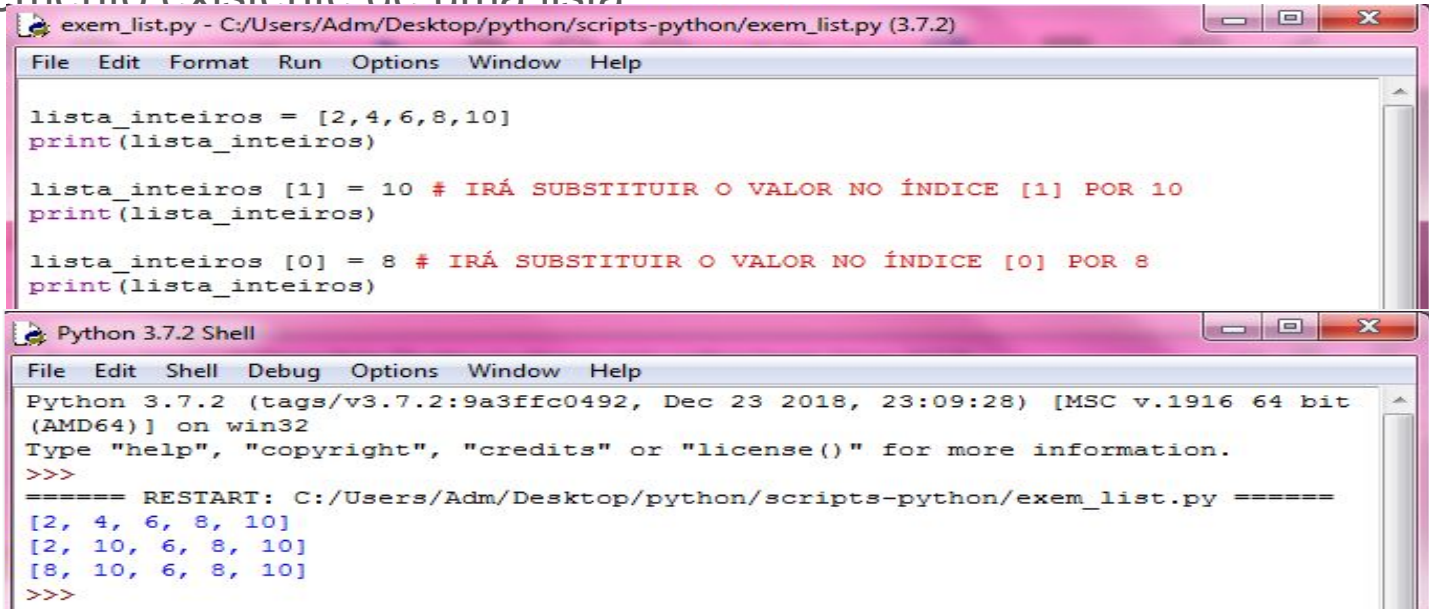
for i in range(len(lista_inteiros)):
    print(lista_inteiros[i])
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution output:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exem_list.py =====
2
4
6
8
10
>>>
```

Listas em Python

- A atribuição em listas permite criar uma nova lista ou modificar um elemento existente de uma lista



The image shows two windows from a Python 3.7.2 IDE. The top window, titled 'exem_list.py', contains the following code:

```
lista_inteiros = [2,4,6,8,10]
print(lista_inteiros)

lista_inteiros [1] = 10 # IRÁ SUBSTITUIR O VALOR NO ÍNDICE [1] POR 10
print(lista_inteiros)

lista_inteiros [0] = 8 # IRÁ SUBSTITUIR O VALOR NO ÍNDICE [0] POR 8
print(lista_inteiros)
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution output:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exem_list.py =====
[2, 4, 6, 8, 10]
[2, 10, 6, 8, 10]
[8, 10, 6, 8, 10]
>>>
```

Listas em Python


- Em Python, a estrutura da lista é dinâmica, ou seja, permite **adicionar e remover** elementos em uma lista existente;
- Ao adicionar ou remover um elemento, o tamanho da lista também é modificado;
- Podemos adicionar um elemento ou vários elementos.

Listas em Python

Append

- O comando `append` é utilizado para adicionar um novo elemento no final da lista.

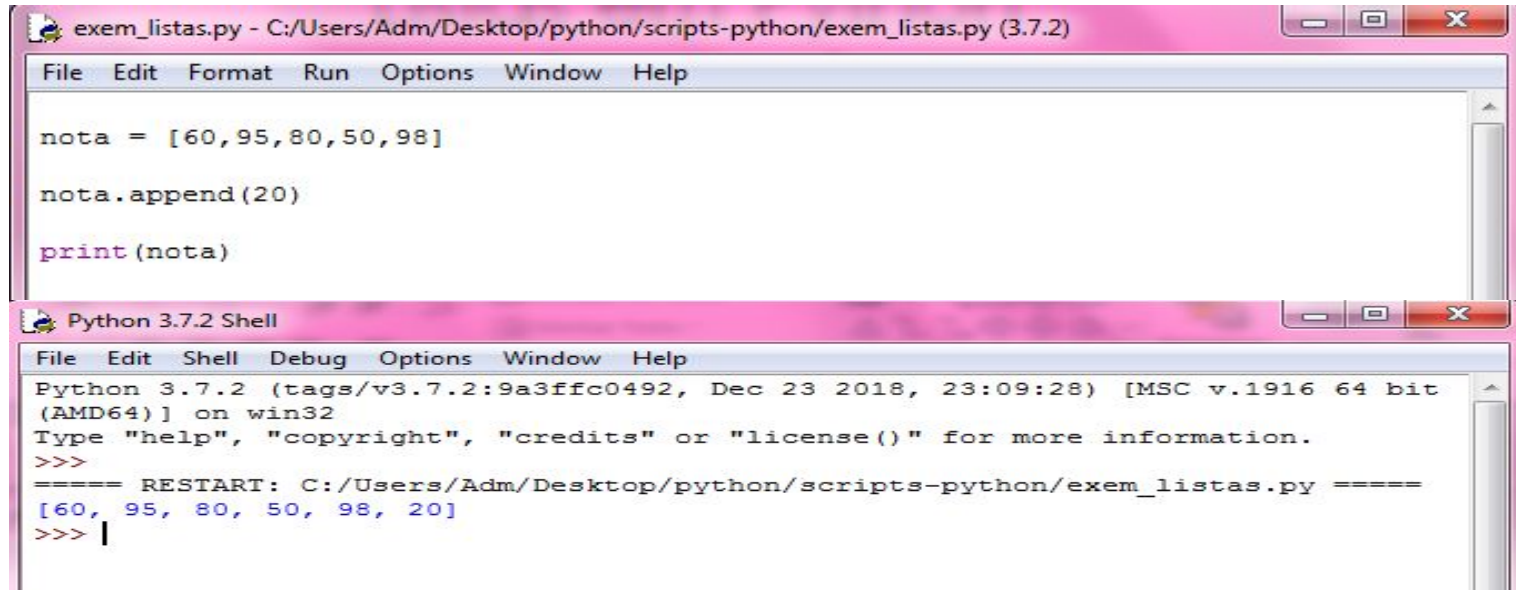
➡ `nota.append(20)`



Índice	0	1	2	3	4	5
Valor	60	95	80	50	98	20
Vetor	Nota[0]	Nota[1]	Nota[2]	Nota[3]	Nota[4]	Nota[5]

Listas em Python

Append



The image shows two overlapping windows from a Python IDE. The top window, titled 'exem_listas.py - C:/Users/Adm/Desktop/python/scripts-python/exem_listas.py (3.7.2)', contains the following Python code:

```
nota = [60,95,80,50,98]

nota.append(20)

print(nota)
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution output:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exem_listas.py =====
[60, 95, 80, 50, 98, 20]
>>> |
```

Listas em Python

Insert

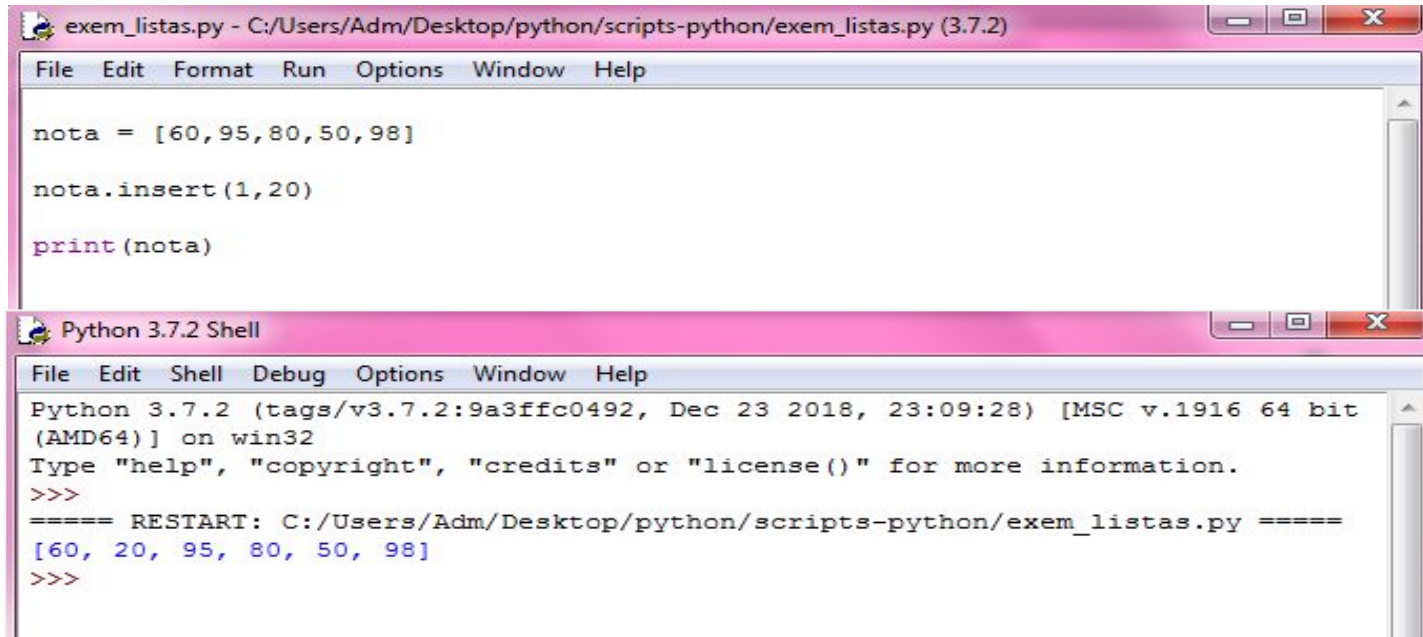
- O comando insert é utilizado para adicionar um elemento contido na lista em uma outra posição.

 nota.insert(1,20)

Índice	0	1	2	3	4	5
Valor	60	20	95	80	50	98
Vetor	Nota[0]	Nota[1]	Nota[2]	Nota[3]	Nota[4]	Nota[5]

Listas em Python

Insert



The image shows two overlapping windows from a Python IDE. The top window, titled 'exem_listas.py - C:/Users/Adm/Desktop/python/scripts-python/exem_listas.py (3.7.2)', contains the following Python code:

```
nota = [60,95,80,50,98]
nota.insert(1,20)
print(nota)
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution output:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exem_listas.py =====
[60, 20, 95, 80, 50, 98]
>>>
```

Listas em Python

Del

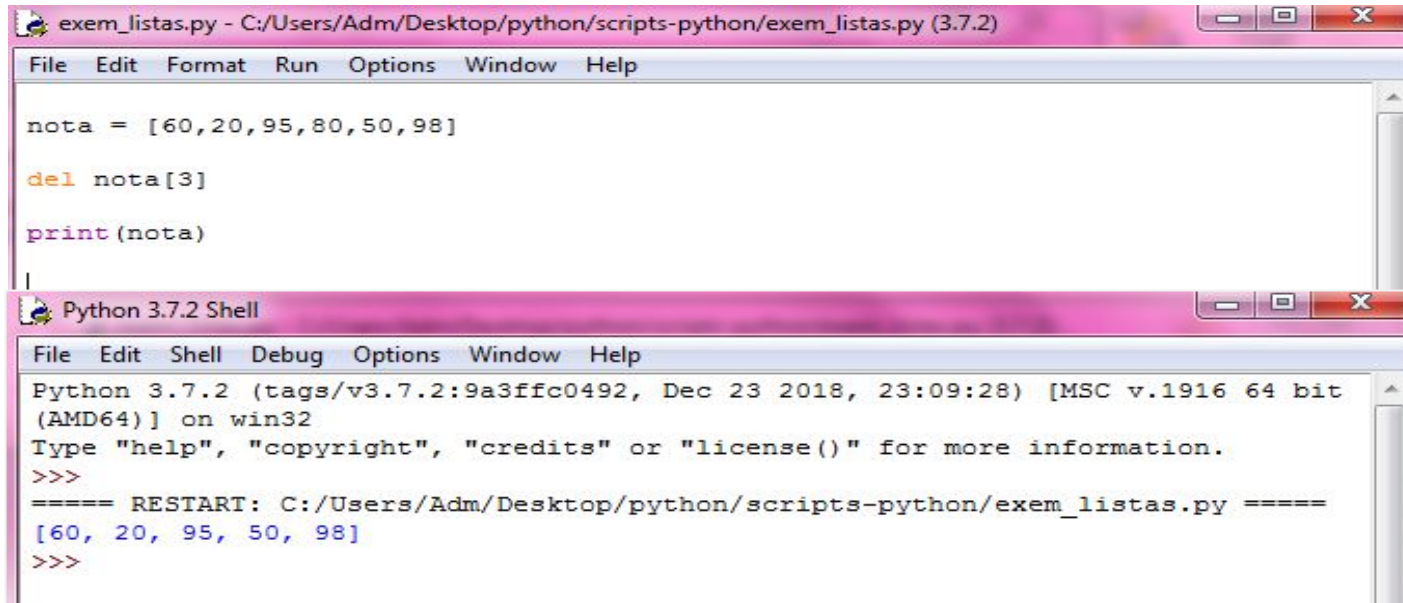
- O comando del é utilizado para remover elementos em uma lista.

→ del nota[3] } Indica-se o índice que deseja eliminar

Índice	0	1	2	3	4	5
Valor	60	20	95	80	50	98
Vetor	Nota[0]	Nota[1]	Nota[2]	Nota[3]	Nota[4]	Nota[5]

Listas em Python

Del



The image shows two windows from a Python IDE. The top window, titled 'exem_listas.py - C:/Users/Adm/Desktop/python/scripts-python/exem_listas.py (3.7.2)', contains the following Python code:

```
nota = [60,20,95,80,50,98]
del nota[3]
print(nota)
```

The bottom window, titled 'Python 3.7.2 Shell', shows the execution output:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exem_listas.py =====
[60, 20, 95, 50, 98]
>>>
```

Listas em Python

Range em listas

- O comando range é utilizado para criar listas através da estrutura for.

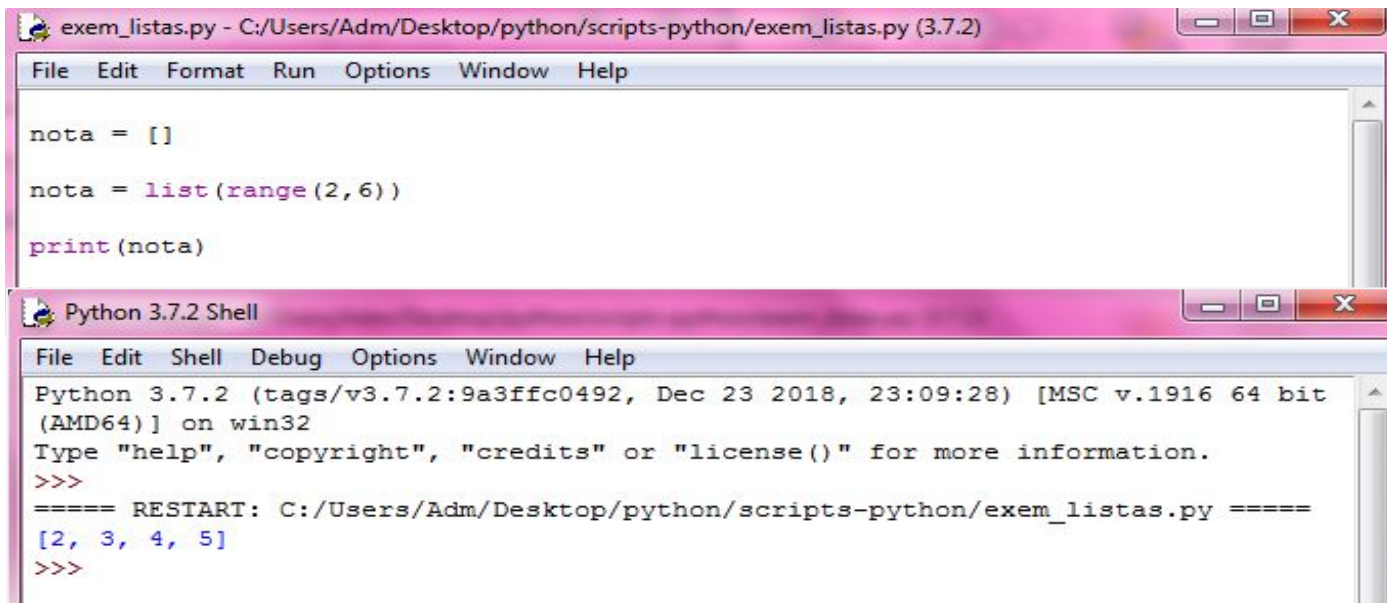
➡ Nota = list(range(2,6))

Índice	0	1	2	3
Valor	2	3	4	5
Vetor	Nota[0]	Nota[1]	Nota[2]	Nota[3]

O comando range já cria uma estrutura ordenada de forma crescente

Listas em Python

Range em listas



The image shows two windows from a Python IDE. The top window, titled 'exem_listas.py - C:/Users/Adm/Desktop/python/scripts-python/exem_listas.py (3.7.2)', contains the following Python code:

```
nota = []  
nota = list(range(2, 6))  
print(nota)
```


The bottom window, titled 'Python 3.7.2 Shell', shows the execution output:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit  
(AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exem_listas.py =====  
[2, 3, 4, 5]  
>>>
```

Listas em Python

Sort

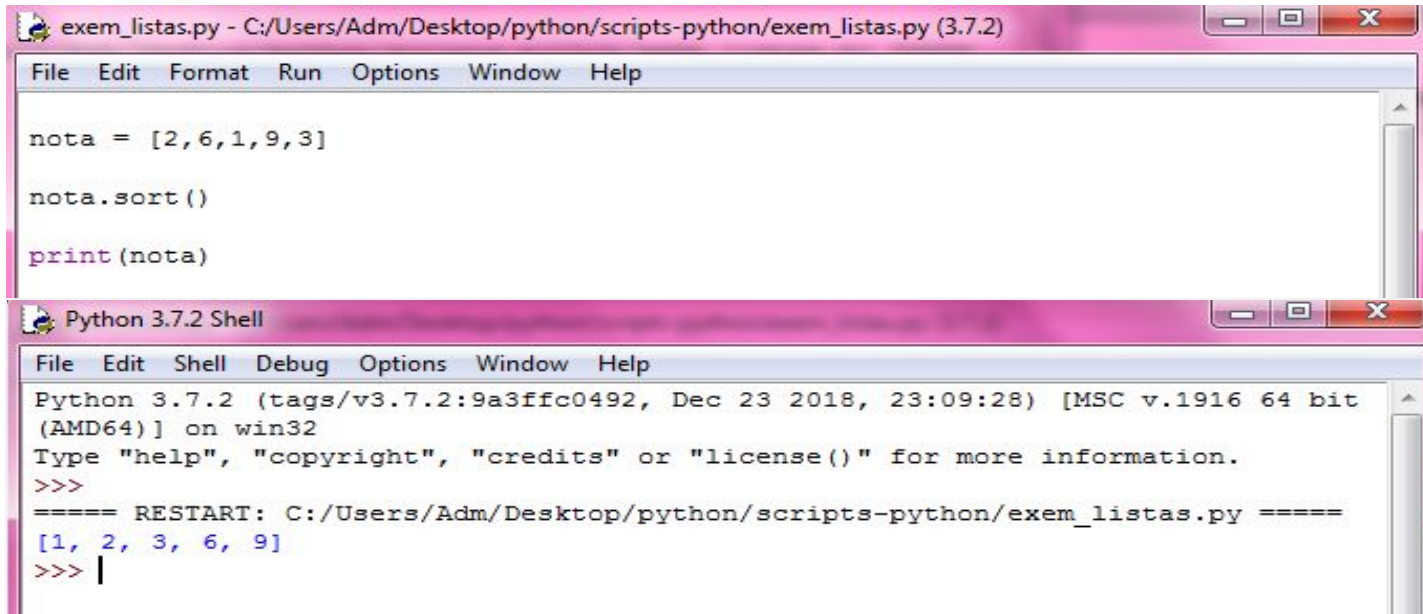
- O comando sort é utilizado para ordenar todos os valores contidos na lista.

 `nota = [2,6,1,9,3]`
`nota.sort()`

Índice	0	1	2	3	4
Valor	1	2	3	6	9
Vetor	Nota[0]	Nota[1]	Nota[2]	Nota[3]	Nota[4]

Listas em Python

Sort



The image shows two windows from a Python IDE. The top window, titled 'exem_listas.py - C:/Users/Adm/Desktop/python/scripts-python/exem_listas.py (3.7.2)', contains the following Python code:

```
nota = [2,6,1,9,3]
nota.sort()
print(nota)
```


The bottom window, titled 'Python 3.7.2 Shell', shows the execution output:

```
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exem_listas.py =====
[1, 2, 3, 6, 9]
>>> |
```

Listas em Python

Reverse

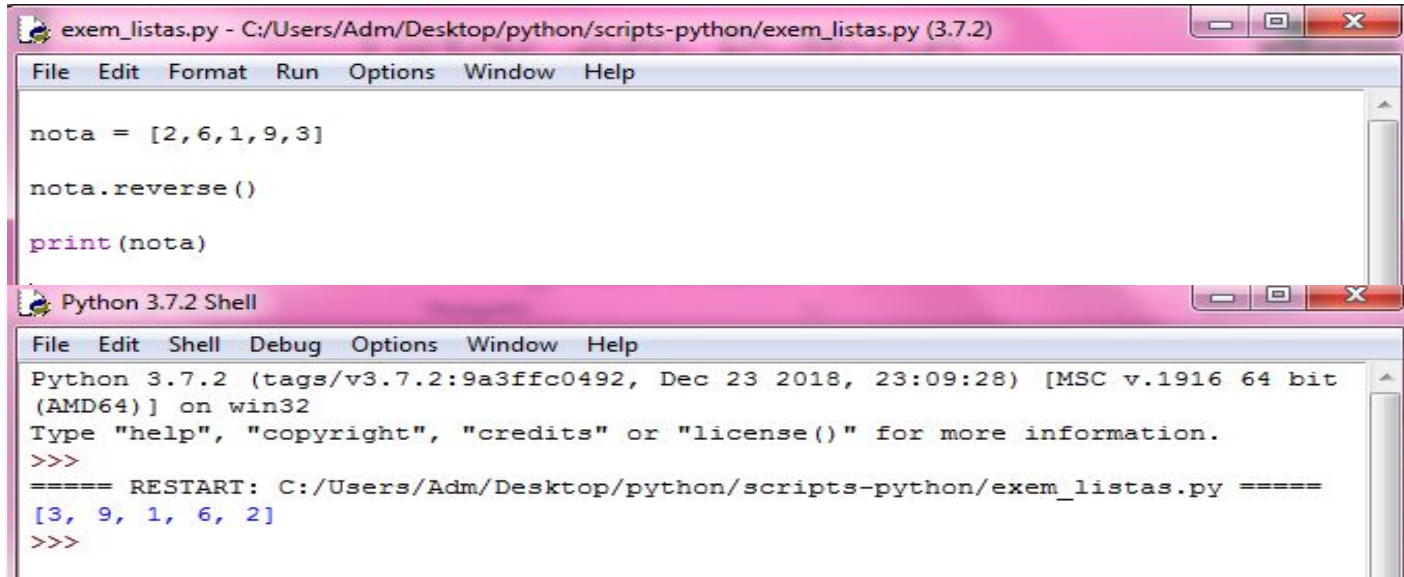
- O comando reverse é utilizado para ordenar na forma inversa todos os valores contidos na lista.

 `nota = [2,6,1,9,3]`
`nota.reverse()`

Índice	0	1	2	3	4
Valor	3	9	1	6	2
Vetor	Nota[0]	Nota[1]	Nota[2]	Nota[3]	Nota[4]

Listas em Python

Reverse



The image shows a screenshot of a Python IDE window titled "exem_listas.py - C:/Users/Adm/Desktop/python/scripts-python/exem_listas.py (3.7.2)". The window contains a Python script with the following code:

```
nota = [2,6,1,9,3]
nota.reverse()
print(nota)
```

Below the script editor is a "Python 3.7.2 Shell" window. The shell displays the following output:

```
Python 3.7.2 (tags/v3.7.2:9a3fffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exem_listas.py =====
[3, 9, 1, 6, 2]
>>>
```

Matrizes em Python

- **Matrizes** é equivalente a um vetor, contudo permite a utilização de diversas dimensões acessadas via diferentes índices;
- Pode ser pensada como um vetor onde cada célula é outro vetor, recursivamente;
- Em diversas situações matrizes são necessárias para correlacionar informações.

Matrizes em Python

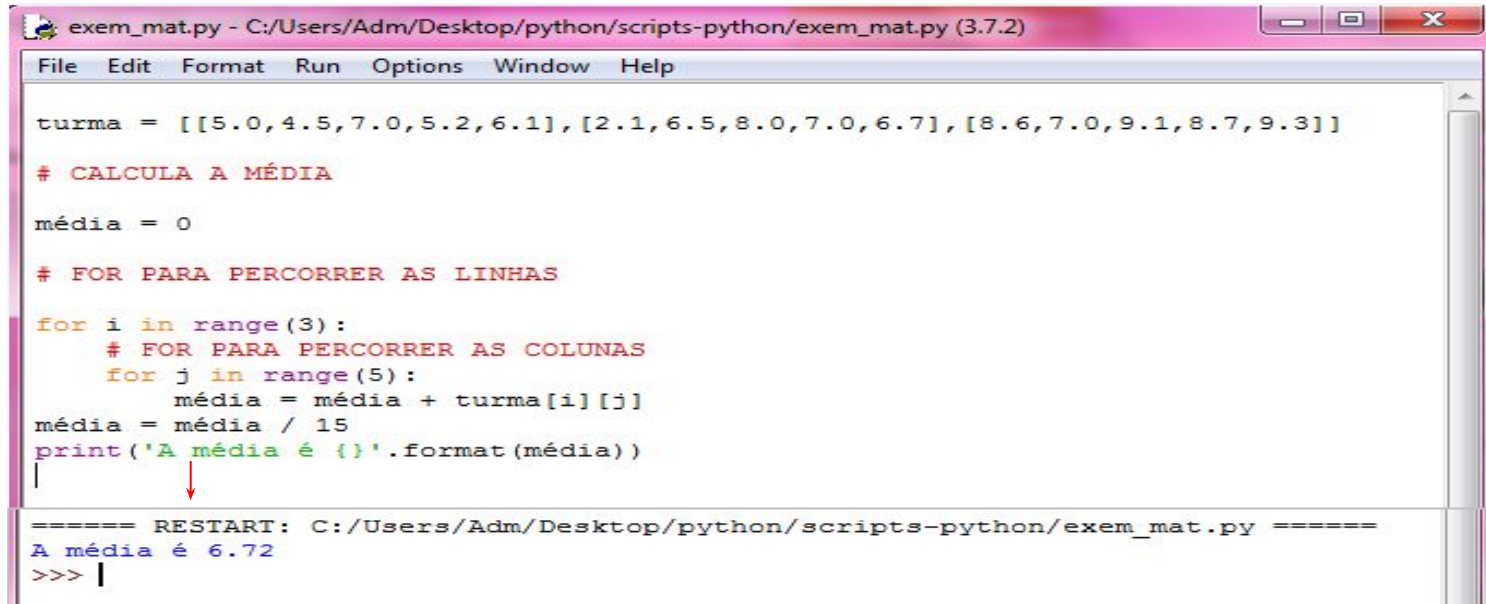
- Assumindo que **uma turma tem três alunos**, seria necessária uma matriz bidimensional para guardar as notas de todos os alunos de uma turma.

		Notas				
		0	1	2	3	4
Alunos	0	5.0	4.5	7.0	5.2	6.1
	1	2.1	6.5	8.0	7.0	6.7
	2	8.6	7.0	9.1	8.7	9.3

```
turma = [[5.0, 4.5, 7.0, 5.2, 6.1], [2.1, 6.5, 8.0, 7.0, 6.7],[8.6, 7.0, 9.1, 8.7, 9.3]]
```

Matrizes em Python

- Vamos calcular a média do problema anterior em Python:



```
exem_mat.py - C:/Users/Adm/Desktop/python/scripts-python/exem_mat.py (3.7.2)
File Edit Format Run Options Window Help

turma = [[5.0,4.5,7.0,5.2,6.1],[2.1,6.5,8.0,7.0,6.7],[8.6,7.0,9.1,8.7,9.3]]

# CALCULA A MÉDIA

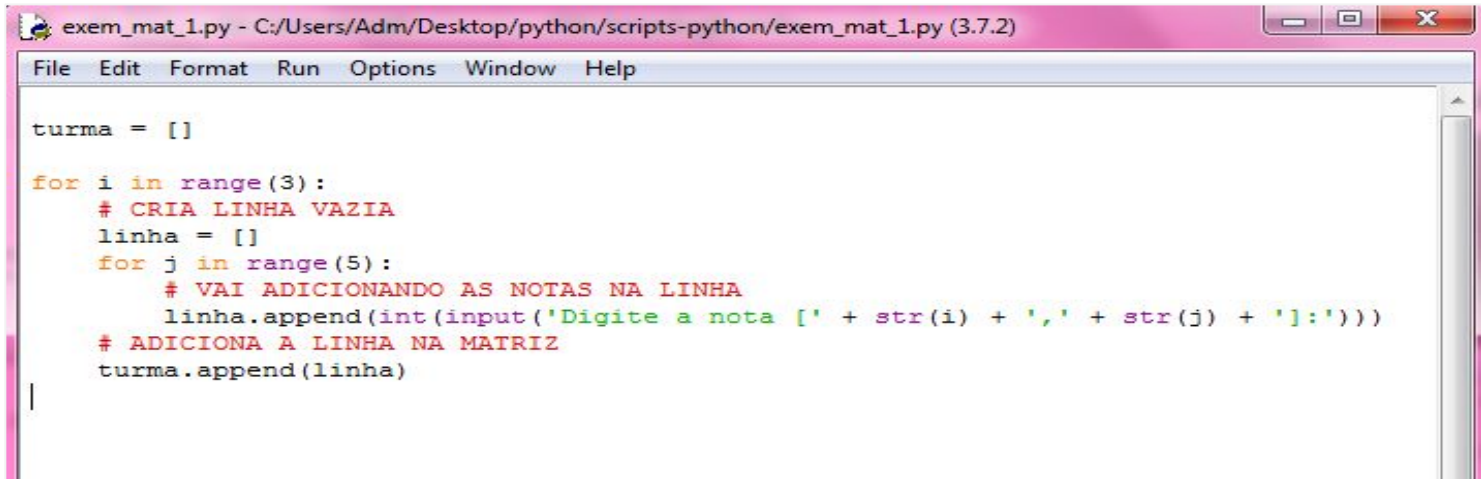
média = 0

# FOR PARA PERCORRER AS LINHAS

for i in range(3):
    # FOR PARA PERCORRER AS COLUNAS
    for j in range(5):
        média = média + turma[i][j]
média = média / 15
print('A média é {}'.format(média))
|
↓
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exem_mat.py =====
A média é 6.72
>>> |
```

Matrizes em Python

- Vamos calcular a média por preenchimento da matriz por leitura em Python:



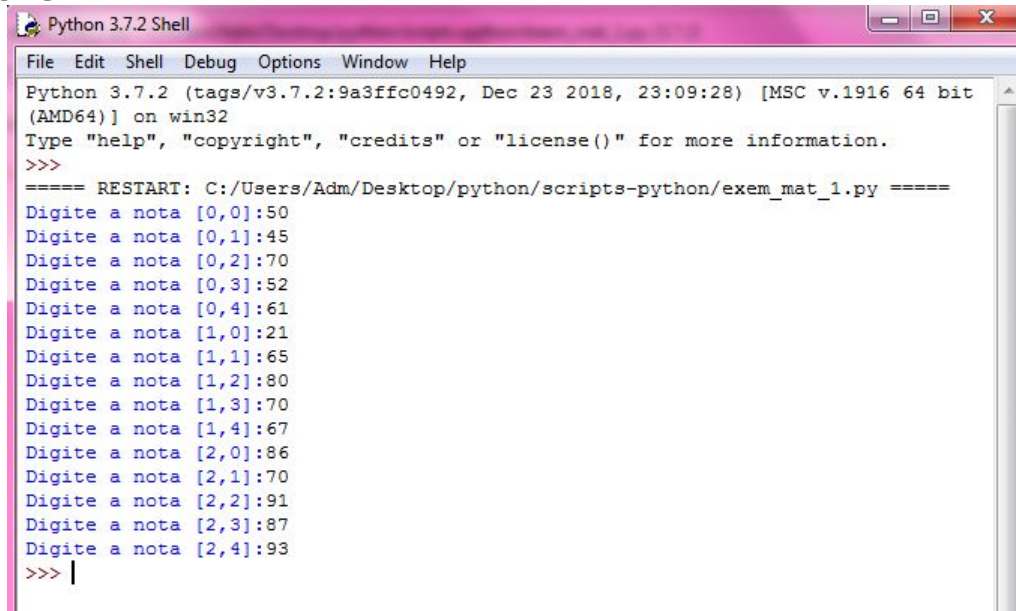
```
exem_mat_1.py - C:/Users/Adm/Desktop/python/scripts-python/exem_mat_1.py (3.7.2)
File Edit Format Run Options Window Help

turma = []

for i in range(3):
    # CRIA LINHA VAZIA
    linha = []
    for j in range(5):
        # VAI ADICIONANDO AS NOTAS NA LINHA
        linha.append(int(input('Digite a nota [' + str(i) + ',' + str(j) + ']:')))
    # ADICIONA A LINHA NA MATRIZ
    turma.append(linha)
```

Matrizes em Python

- Vamos calcular a média por preenchimento da matriz por leitura em Python:



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Adm/Desktop/python/scripts-python/exem_mat_1.py =====
Digite a nota [0,0]:50
Digite a nota [0,1]:45
Digite a nota [0,2]:70
Digite a nota [0,3]:52
Digite a nota [0,4]:61
Digite a nota [1,0]:21
Digite a nota [1,1]:65
Digite a nota [1,2]:80
Digite a nota [1,3]:70
Digite a nota [1,4]:67
Digite a nota [2,0]:86
Digite a nota [2,1]:70
Digite a nota [2,2]:91
Digite a nota [2,3]:87
Digite a nota [2,4]:93
>>> |
```


Exercícios Práticos

- Exercício 1:
- ✓ Faça um Programa que leia um vetor de 5 números inteiros e mostre-os.

Exercícios Práticos

- Exercício 2:
- ✓ Faça um Programa que leia um vetor de 10 números reais e mostre-os na ordem inversa.

Exercícios Práticos

- Exercício 3:
- ✓ Faça um Programa que leia um vetor de 10 caracteres, e diga quantas consoantes foram lidas. Imprima as consoantes.

Exercícios Práticos

- Exercício 4:
- ✓ Faça um programa que leia 15 números inteiros e armazene-os num vetor. Armazene os números pares no vetor PAR e os números IMPARES no vetor impar. Imprima os três vetores.

Exercícios Práticos

- Exercício 5:
- ✓ Faça um Programa que peça as quatro notas de 3 alunos, calcule e armazene num vetor a média de cada aluno, imprima o número de alunos com média maior ou igual a 6.0.

Exercícios Práticos

- Exercício 6:
- ✓ Faça um Programa que peça a idade e a altura de 5 pessoas, armazene cada informação no seu respectivo vetor. Imprima a idade e a altura na ordem inversa a ordem lida.

Exercícios Práticos

- Exercício 7:
- ✓ Faça um Programa que leia dois vetores com 10 elementos cada. Gere um terceiro vetor de 20 elementos, cujos valores deverão ser compostos pelos elementos intercalados dos dois outros vetores.

Exercícios Práticos

- Exercício 8:
- ✓ Faça um programa que leia 5 valores numéricos e guarde-os em uma lista. No final, mostre qual foi o maior e o menor valor digitado e as suas respectivas posições na lista.

Exercícios Práticos

- Exercício 9:
- ✓ Utilizando listas faça um programa que faça 5 perguntas para uma pessoa sobre um crime. As perguntas são:
 - "Telefonou para a vítima?"
 - "Esteve no local do crime?"
 - "Mora perto da vítima?"
 - "Devia para a vítima?"
 - "Já trabalhou com a vítima?"
- O programa deve no final emitir uma classificação sobre a participação da pessoa no crime. Se a pessoa responder positivamente a 2 questões ela deve ser classificada como "Suspeita", entre 3 e 4 como "Cúmplice" e 5 como "Assassino". Caso contrário, ele será classificado como "Inocente".

Exercícios Práticos

- Exercício 10:
- ✓ Crie um programa que declare uma matriz de dimensão 3x3 e preencha com valores lidos pelo teclado. No final, mostre a matriz na tela, com a formatação correta.

Exercícios Práticos

- Exercício 11:
- ✓ Aprimore o exercício 10, mostrando no final:
 - Soma de todos os valores pares digitados;
 - A soma dos valores da terceira coluna;
 - O maior valor da segunda linha.