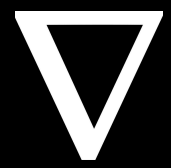


Introdução ao Python 2

Curso de Python para
Data Science

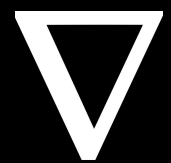


Murilo Leandro
@murilouco_louco

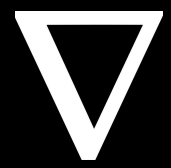
The background of the slide features a large, light blue inverted triangle that overlaps with a slightly smaller, light pink inverted triangle. A blue horizontal line and a pink horizontal line, both with vertical end-caps, frame the title text. The title itself is in a large, bold, black sans-serif font.

Estruturas de dados

<https://docs.python.org/3/tutorial/datastructures.html>

The background features a large, light gray inverted triangle. Overlaid on this are two more inverted triangles: a light blue one and a light pink one, both slightly offset from the center. In the center of these overlapping triangles is a white rectangular box with a thin blue border on the top and left sides, and a thin pink border on the bottom and right sides. The word "Listas" is written in a bold, black, sans-serif font within this box.

Listas



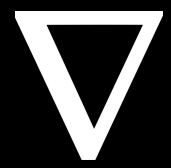
Listas - Slicing

É possível obter “fatias” de uma lista em python

```
main.py
lista = [7, 8, 9, 10, 11, 12, 13]
lista[1:3] # [8, 9]
lista[2:-1] # [9, 10, 11, 12]

string = "Arquivo.txt"
print(string[:-4]) # Arquivo
```

	0	1	2	3	4	5	6
	<code>lista = [7, 8, 9, 10, 11, 12, 13]</code>						
	-7	-6	-5	-4	-3	-2	-1



Listas - Operações

Listas possuem operações associadas que permitem você modificar seus valores:

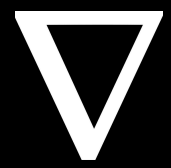
```
main.py  — □ ×

frutas = ["maçã", "banana"]

frutas.append("uva")
print(frutas) # ["maçã", "banana", "uva"]

frutas.extend(["pera", "melão"])
print(frutas) # ["maçã", "banana", "uva", "pera", "melão"]

frutas.pop(1)
print(frutas) # ["maçã", "uva", "pera", "melão"]
```

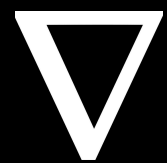


Listas - Comprehention

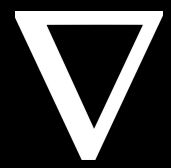
Forma concisa de criar listas.

```
main.py  — □ ×  
  
quadrados = []  
for x in range(5):  
    quadrados.append(x**2)  
print(quadrados)
```

```
main.py  — □ ×  
  
quadrados = [x**2 for x in range(5)]  
print(quadrados)
```

The background features a large, faint graphic of three overlapping inverted triangles in light blue, light purple, and light pink. A blue horizontal line and a pink horizontal line intersect at a right angle, framing the title text.

Dicionários



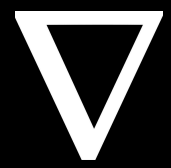
Dicionários

Dicionários são um conjunto de pares “chave”:valor.
O valor é acessado através da chave.

```
main.py

pessoa = {
    "nome": "Murilo Leandro",
    "idade": 19,
    "frutas": ["Uva", "Limão"]
}

print( pessoa["nome"] ) # Murilo Leandro
print( pessoa["idade"] ) # 19
print( pessoa["frutas"] ) # ["Uva", "Limão"]
```

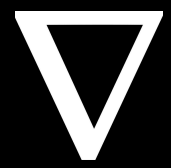
Dicionários

Você pode modificar cada valor associado a uma chave como uma variável.

```
main.py

pessoa = {
    "nome": "Murilo Leandro",
    "idade": 19,
    "frutas": ["Limão", "Uva"]
}

pessoa["idade"] += 1
print(pessoa["idade"]) # 20
```



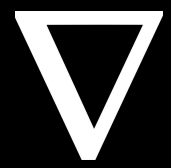
Dicionários

Cheque se uma chave existe:

```
main.py

diccionario = {
    "A": 1,
    "B": 2,
    "C": 3
}

print('A' in diccionario # True
print('D' in diccionario)# False
```

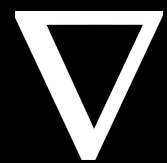


Referências

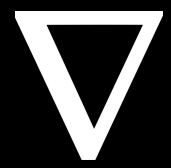
Uma variável contém o 'endereço' de uma lista/dicionário, não seu conteúdo.

```
main.py

lista = [1,2,3]
lista2 = lista
lista2.append(4)
print(lista) # [1, 2, 3, 4]
print(lista2) # [1, 2, 3, 4]
```

The background features a large, light gray inverted triangle. Overlaid on this are two more inverted triangles: a light blue one and a light pink one, both slightly offset from the center. A blue horizontal line and a pink horizontal line intersect at a right angle, forming a frame around the text.

Funções



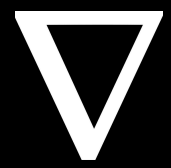
Funções

Quase o mesmo da matemática, com a diferença que não precisa retornar algo.

```
main.py

def printar_n(texto, numero):
    for i in range(numero):
        print(texto, i+1)

printar_n("Olá Mundo!", 5)
```



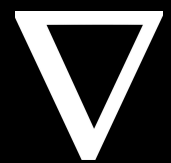
Funções

Mas elas podem retornar:

```
main.py

def bhaskara(a, b, c):
    delta = b**2 - 4*a*c
    x = (-b + delta**0.5) / (2*a)
    return x

print(bhaskara(1, 5, 6)) # -2.0
```



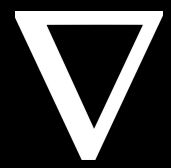
Funções

```
main.py

def derivada(f):
    h = 0.001
    def df(x):
        return (f(x+h) - f(x)) / h
    return df

def f(x):
    return x**2

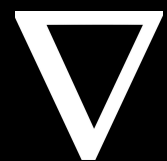
df = derivada(f)
print(df(1), df(2), df(3), df(4))
```



Exercícios

<https://colab.research.google.com/drive/1Jilq2p-PEx54bKocF8UjkWdhnp-Pcna>

<https://tinyurl.com/datapython2>



Presença

