



Universidade de Fortaleza - UNIFOR

---

# Sistemas Inteligentes - T951

Centro de Ciências Tecnológicas - CCT  
Universidade de Fortaleza  
Fortaleza, Ceará, Brasil

12 de Agosto, 2021



- 1 Introdução
  - 1.1 Introdução
  - 1.2 Instalação Python
  - 1.3 Visual Studio Code



## Lista de Instalações Necessárias.

- **Link para download no Python** Qual versão deve-se escolher???
- Visual Studio Code;

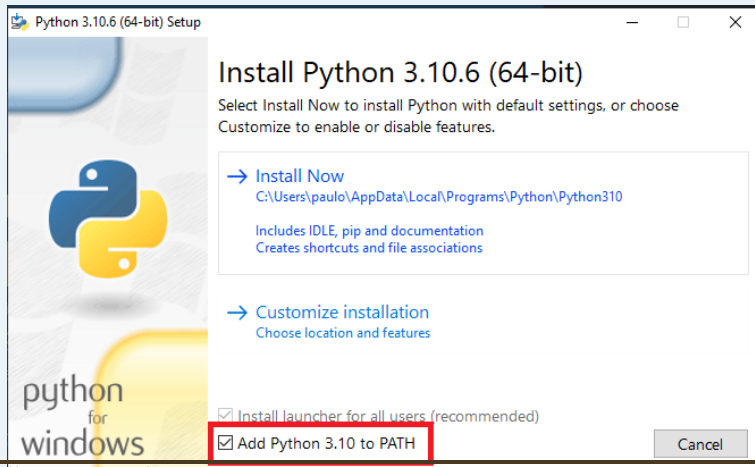
## Configurações Necessárias

- Extensões no VS Code;
- **Matplotlib**
- NumPy - **[Link de Suporte]**
- OpenCV-Python (Caso utilize, será necessário a versão 3.7.6 do Python!)



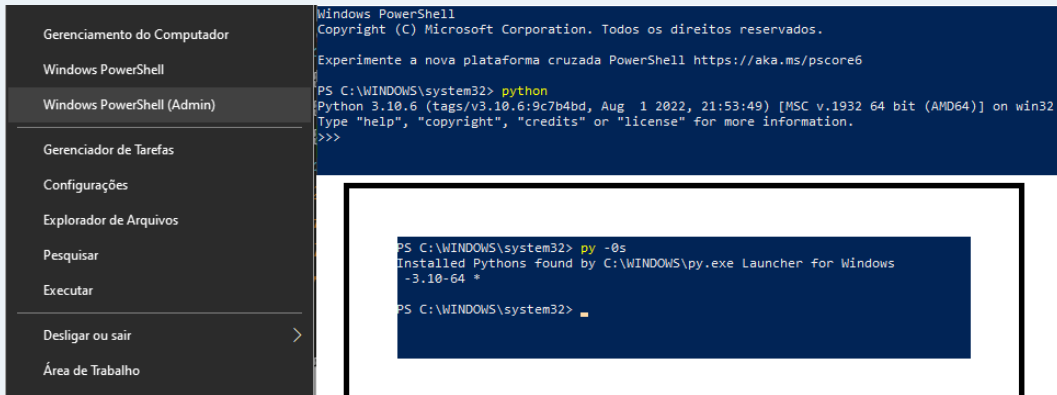
## Após download, inicie a instalação

- Faça a marcação como indicado e clique em **Install Now**



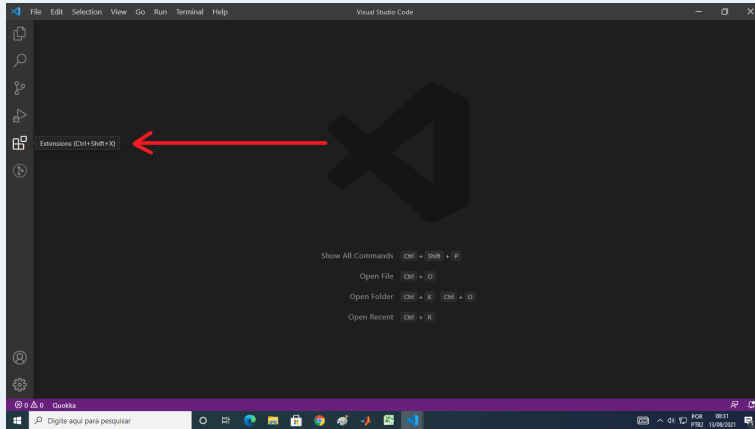


Verifique se foi instalado.





## Instalação de Extensões





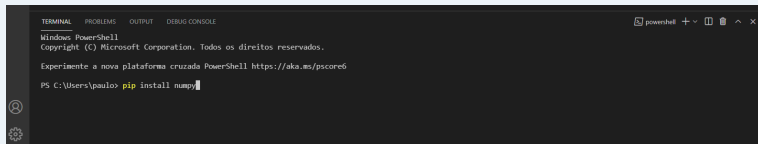
## Lista de Extensões

- **Python**
- Python Indent (não necessária)
- Pylance (não necessária)
- Jupyter (não necessária)
- Bracket Pair Colorizer (não necessária)
- Kite Autocomplete (não necessária)
- Path Intellisense (não necessária)



## Instalação de bibliotecas

- Abra o terminal do VS Code ( *ctrl + `* )
- A Instalação é realizada digitando os comandos:
  - 1 `python -m pip install --upgrade pip`
  - 2 `pip install numpy`
  - 3 `pip install matplotlib`
  - 4 `pip install seaborn`



```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos os direitos reservados.

Experimente a nova plataforma cruzada PowerShell https://aka.ms/pscore6

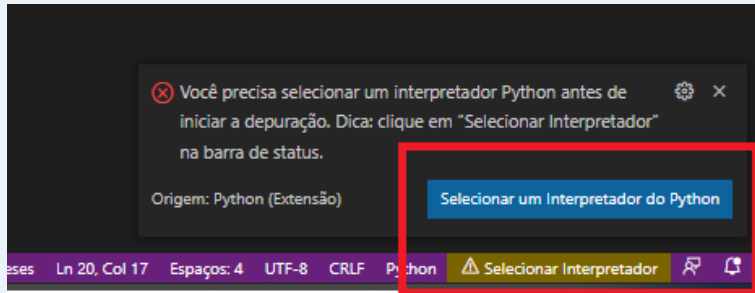
PS C:\Users\paulo> pip install numpy
```





## Instalação de bibliotecas

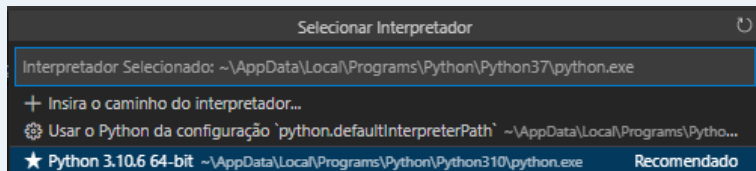
- Para criar um novo arquivo em **Python**, basta criar um novo arquivo de texto (CTRL+N) e em seguida salvá-lo com a extensão **.py**
- Para executar o primeiro algoritmo desenvolvido, basta aperta F5 no teclado. Uma janela no canto inferior direito surgirá para que seja selecionado o interpretador.





## Instalação de bibliotecas

- Para criar um novo arquivo em **Python**, basta criar um novo arquivo de texto (CTRL+N) e em seguida salvá-lo com a extensão **.py**
- Para executar o primeiro algoritmo desenvolvido, basta aperta F5 no teclado. Uma janela no canto inferior direito surgirá para que seja selecionado o interpretador.





## Declarações de variáveis e tipos de variáveis

- Como declarar uma variável em Python?
- Como imprimir alguma mensagem no console?
- Quais os tipos de variáveis em Python (ou pelo menos os suficientes para a disciplina).
- Como realizar a conversão dos tipos?
  - 1 Conversão para string: `str()`
  - 2 Conversão para inteiro: `int()`
  - 3 Conversão para real: `float()`
- Como imprimir os valores das variáveis concatenados com uma string no console?

```
1 x = 1
2 y = 2
3 z = 3.9999
4 print("X = " + str(x) + " \nY = " + str(y) + "\nZ = "+str(z))
```



## Declarações de variáveis e tipos de variáveis

- Como declarar uma variável em Python?
- Como imprimir alguma mensagem no console?
- Quais os tipos de variáveis em Python (ou pelo menos os suficientes para a disciplina).
- Como realizar a conversão dos tipos?
  - 1 Conversão para string: `str()`
  - 2 Conversão para inteiro: `int()`
  - 3 Conversão para real: `float()`
- Como imprimir os valores das variáveis concatenados com uma string no console?

```
1 x = 1
2 y = 2
3 z = 3.9999
4 print("X = ",str(x),"\nY = ",str(y),"\nZ = "+str(z))
```



## Declarações de variáveis e tipos de variáveis

- Como declarar uma variável em Python?
- Como imprimir alguma mensagem no console?
- Quais os tipos de variáveis em Python (ou pelo menos os suficientes para a disciplina).
- Como realizar a conversão dos tipos?
  - 1 Conversão para string: `str()`
  - 2 Conversão para inteiro: `int()`
  - 3 Conversão para real: `float()`
- Como imprimir os valores das variáveis concatenados com uma string no console?

```
1 x = 1
2 y = 2
3 z = 3.9999
4 print("X = {0} \nY = {1} \nZ = {2}".format(x,y,z))
```



## Declarações de variáveis e tipos de variáveis

- Como declarar uma variável em Python?
- Como imprimir alguma mensagem no console?
- Quais os tipos de variáveis em Python (ou pelo menos os suficientes para a disciplina).
- Como realizar a conversão dos tipos?
  - 1 Conversão para string: `str()`
  - 2 Conversão para inteiro: `int()`
  - 3 Conversão para real: `float()`
- Como imprimir os valores das variáveis concatenados com uma string no console?

```
1 x = 1
2 y = 2
3 z = 3.9999
4 print(f"X = {x} \nY = {y} \nZ = {z}")
```



## Declarações de variáveis e tipos de variáveis

- Como declarar uma variável em Python?
- Como imprimir alguma mensagem no console?
- Quais os tipos de variáveis em Python (ou pelo menos os suficientes para a disciplina).
- Como realizar a conversão dos tipos?
  - 1 Conversão para string: `str()`
  - 2 Conversão para inteiro: `int()`
  - 3 Conversão para real: `float()`
- Como imprimir os valores das variáveis concatenados com uma string no console?

```
1 x = 1
2 y = 2
3 z = 3.9999
4 print("X = {0} \nY = {1} \nZ = {2:0.2f}".format(x,y,z))
```



## Declarações de variáveis e tipos de variáveis

- Como declarar uma variável em Python?
- Como imprimir alguma mensagem no console?
- Quais os tipos de variáveis em Python (ou pelo menos os suficientes para a disciplina).
- Como realizar a conversão dos tipos?
  - 1 Conversão para string: `str()`
  - 2 Conversão para inteiro: `int()`
  - 3 Conversão para real: `float()`
- Como imprimir os valores das variáveis concatenados com uma string no console?

```
1 x = 1
2 y = 2
3 z = 3.9999
4 print(f"X = {x} \nY = {y} \nZ = {z:.5f}")
```





## Operadores Aritméticos e Relacionais

- $+$   $\rightarrow$  Soma.
- $-$   $\rightarrow$  Subtração.
- $*$   $\rightarrow$  Multiplicação.
- $/$   $\rightarrow$  Divisão.
- $\%$   $\rightarrow$  Resto da Divisão.
- $**$   $\rightarrow$  Potenciação.
- $//$   $\rightarrow$  Divisão inteira.
- $>$   $\rightarrow$  Maior.
- $<$   $\rightarrow$  Menor.
- $>=$   $\rightarrow$  Maior ou igual.
- $<=$   $\rightarrow$  Menor ou igual.
- $==$   $\rightarrow$  Igual.
- $!=$   $\rightarrow$  Diferente.



## Operadores Lógicos.

- E  $\leftarrow$  and
- Ou  $\leftarrow$  or
- Negado  $\leftarrow$  not

## Estrutura de seleção.

**if**    condição:  
      executa os comandos presentes nesta seção

**else:**  
      executa os comandos presentes nesta seção



## Estrutura de seleção.

**if**    condição:  
    executa os comandos presentes nesta seção

**elif**    condição:  
    executa os comandos presentes nesta seção

**else:**  
    executa os comandos presentes nesta seção

## Estrutura de seleção (Operador Ternário).

*determinado retorno* **if**    condição    **else**    *determinado retorno*



## Estrutura de repetição.

**while**    condição:

    executa os comandos presentes nesta seção

else:

    executa os comandos presentes nesta seção

- **break e continue**
- E o **for**?



## Funções (Criando funções)

```
def NomeFunção(parâmetros):    determinado processo
```

## Funções (Utilização)

```
Retorno = NomeFunção(parâmetros)
```

## Observações

- 1 Quando um parâmetro estiver acompanhado de um asterisco (\*parametro), significa que o número de argumentos é desconhecido. A função trata-os como vetores.
- 2 É possível enviar os argumentos em qualquer ordem se na utilização da função tiver explicito a atribuição ao parâmetro (param3=1, param2=1, param1='a').
- 3 É possível definir parâmetros padrão para funções.



## Observações

- 1 PODE-SE consumir funções implementadas em outros arquivos?
- 2 Uma linha de comando que vocês verão com frequência (nas pesquisas que vocês fizerem) é: `if __name__ == '__main__':`

## Vetores e Matrizes

- Há certas variações de como criar um vetor em Python, porém, o tipo de variável é categorizado como *list*.
- Em Python uma string é uma *list*?
- Como criar um vetor/matriz vazia com ordem predefinida?
- Apesar destes conceitos interessantes, a maioria das implementações que envolvam vetores e matrizes na disciplina, serão auxiliadas pelo uso da biblioteca **numpy** ([[Link de Suporte](#)] )



## Primeira Utilização do numpy e matplotlib

- Constantemente ao longo da disciplina, dados serão fornecidos.
- Como exemplo inicial, faça o download do arquivo disponibilizado no AVA, com o bloco de nome "Datasets".
- Este arquivo está no formato .csv e contém dados gerados em um processo controlado!
- Você deve levar em consideração que há pares de informações (coordenadas em x e y) para cada dado. Ou seja, este dado possui **dois** preditores.
- É possível perceber também que nos dados disponíveis, não há informação alguma sobre os rótulos, ou seja, cada AMOSTRA está sem informação alguma associada!
- Pergunta: Com essas informações, imagine que será utilizado um modelo que treinará a partir desses dados. Qual será o tipo desse aprendizado?? Bom, a priori um **não supervisionado**.



## Primeira Utilização do numpy e matplotlib

- Contudo, passo a seguinte informação: Os dados são compostos por dois tipos diferentes de classes, ou seja, tem dois rótulos diferentes.
- O primeiro rótulo está associado aos 1000 primeiros dados disponibilizados no CSV.
- O segundo rótulo está associado aos demais dados.
- Como realizar a primeira análise dos dados? Bom, inicialmente é interessante construir a estrutura:

$$\mathbf{X} \in \mathbb{R}^{N \times p}$$

$$\mathbf{y} \in \mathbb{R}^{N \times 1}$$





## Primeira Utilização do numpy e matplotlib

- Para isso, na biblioteca numpy, há um método que faz o acesso ao arquivo csv e resgata os dados: `loadtxt(fname, delimiter)`
- Como as amostras não possuem rótulos, você deve construí-las. Existem diversas abordagens para tal tarefa. Contudo, como um primeiro contato, considere que são classes binárias!
- Após a construção da matriz **X** e do vetor **y**, faça a primeira visualização dos dados com a função **scatter** do matplotlib.



## Primeira Utilização do numpy e matplotlib

```
1 import numpy as np
2 import matplotlib.pyplot as plt      paulo, há 4 meses
3
4 X = np.loadtxt(fname = "SpiralData.csv", delimiter=',')
5 Y = np.ones((1000,1))
6 Y = np.concatenate((Y, -np.ones((1000,1))), axis=0)
7 plt.scatter(X[0:1000,0], X[0:1000,1])
8 plt.scatter(X[1000:,:0], X[1000:,:1])
9 plt.show()
```