

# SEMANA PYTHON ISTA

*Seja disputado pelo  
mercado e comece  
ganhando 5x mais!*

## **APOSTILA 3**

★ **DALTON PEIXOTO** ★



Nessa aula você irá aprender:

- Tratar erros e exceções no seus programas
- O que são classes e como criar objetos personalizados
- Instalar pacotes externos:
  - Matplotlib: Veremos como usar essa biblioteca gráfica para gerar gráficos
  - OpenPyXL: Veremos como criar um arquivo Excel e editá-lo.
  - Pandas: Veremos como funciona uma das bibliotecas mais famosas da Ciência de Dados.

E no final terão exercícios para praticar tudo o que aprendeu!

---

## Tratamento de Exceções

O **traceback** é a informação que o interpretador Python fornece para nos ajudar a identificar um problema em nosso código. Ele contém algumas informações úteis:

- a sequência de chamadas de funções até o ponto onde ocorreu o problema;
- o número da linha onde o erro foi gerado;
- o tipo de erro que ocorreu, bem como uma pequena mensagem informativa sobre o ocorrido.

```
In [1]: a = 1
```

```
In [2]: b = 0
```

```
In [ ]: try: # tenta executar o código
        print(a / b)
except ZeroDivisionError: # excecao
    print('Não é possível dividir por zero')
except TypeError:
    print('Os valores de "a" e "b" são de tipos diferentes')
except:
    print('Qualquer outro erro de tempo de execução') # b = '0'
else:
    print('Quando não der erro') # b = 1
```

## Runtime Error:

Traceback (most recent call last):

File "", line 2, in  
print(mylist[10])

IndexError: list index out of range

Os quadros coloridos significam:

- **VERDE:** Mostra o tipo do erro.
- **AZUL:** Mostra alguma informação relevante sobre o erro.
- **LARANJA:** Mostra a mensagem padrão de levantamento de uma exceção.
- **VERMELHO:** Mostra a linha e o comando que gerou a exceção no programa.

## Classes

Classes proporcionam uma forma de organizar dados e funcionalidades juntos.

Criar uma nova classe cria um novo “**tipo**” de objeto, permitindo que novas “**instâncias**” desse tipo sejam produzidas.

```
In [26]: class MinhaClasse:
          numero = 12345

          def metodo(self):
              return 'olá mundo'
```

```
In [27]: MinhaClasse.numero
```

```
Out[27]: 12345
```

```
In [30]: MinhaClasse.metodo
```

```
Out[30]: <function __main__.MinhaClasse.metodo>
```

**Cria um objeto (Um objeto é uma instância de uma classe)**

```
In [31]: x = MinhaClasse()
```

```
In [32]: x.numero
```

```
Out[32]: 12345
```

```
In [33]: x.metodo()
```

```
Out[33]: 'olá mundo'
```

Vamos criar uma classe chamada **Cachorro** que vai reunir as principais funcionalidades e características de um cachorro.

```
In [44]: class Cachorro:
          def __init__(self, nome, idade):
              self.nome = nome
              self.idade = idade

          def ir_dormir(self):
              return f'{self.nome} foi dormir!'

          def brincar(self):
              return f'{self.nome} está pronto para brincar.'
```

```
In [45]: pingo = Cachorro('Pingo', 2)
```

```
In [46]: mel = Cachorro('Mel', 7)
```

```
In [47]: pingo.nome, pingo.idade
```

```
Out[47]: ('Pingo', 2)
```

```
In [48]: pingo.brincar()
```

```
Out[48]: 'Pingo está pronto para brincar.'
```

```
In [49]: mel.ir_dormir()
```

```
Out[49]: 'Mel foi dormir!'
```

# Instalando pacotes externos

Podemos instalar pacotes externos utilizando o comando:

```
!pip install nome_do_pacote
```

Ou no terminal executamos o mesmo comando mas sem a exclamação:

```
pip install nome_do_pacote
```

## Matplotlib

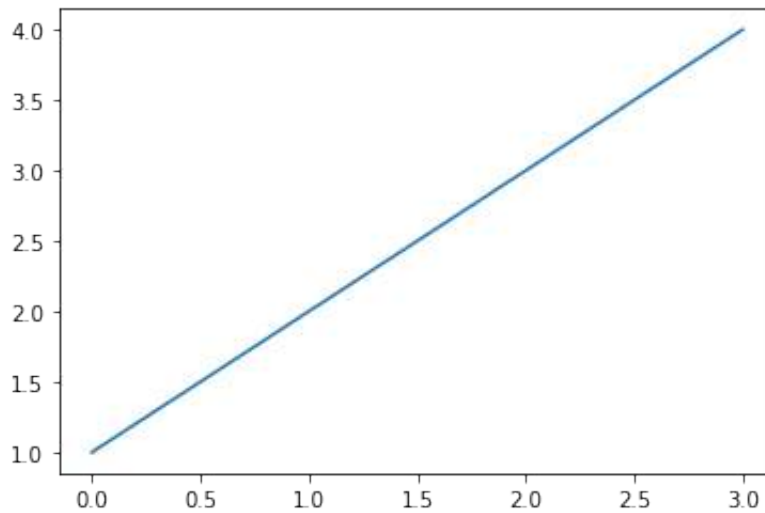
Biblioteca gráfica utilizada para gerar gráficos

```
In [3]: !pip install matplotlib
```

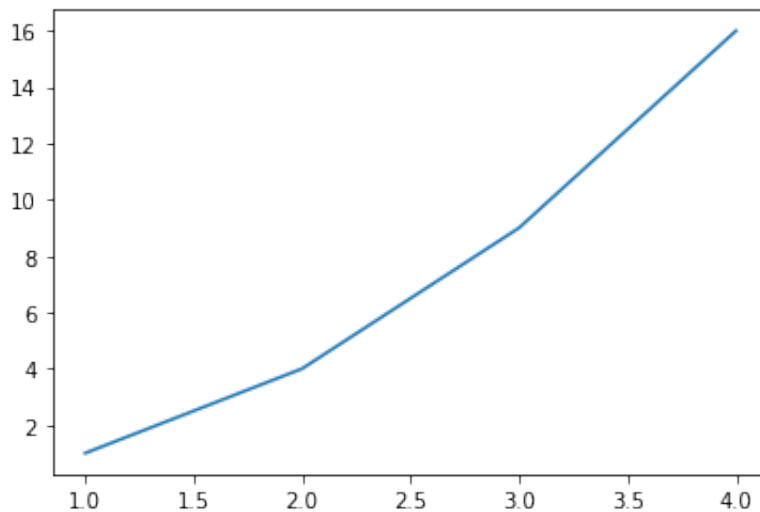
```
Requirement already satisfied: matplotlib in /usr/local/lib/python
3.7/dist-packages (3.2.2)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib
/python3.7/dist-packages (from matplotlib) (1.3.1)
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/pytho
n3.7/dist-packages (from matplotlib) (1.19.5)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/
lib/python3.7/dist-packages (from matplotlib) (2.8.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=
2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib)
(2.4.7)
Requirement already satisfied: cyclor>=0.10 in /usr/local/lib/pyth
on3.7/dist-packages (from matplotlib) (0.10.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.
7/dist-packages (from python-dateutil>=2.1->matplotlib) (1.15.0)
```

```
In [4]: import matplotlib.pyplot as plt
```

```
In [9]: plt.plot([1, 2, 3, 4])  
plt.show()
```



```
In [8]: plt.plot([1, 2, 3, 4], [1, 4, 9, 16])  
plt.show()
```



## OpenPyXL

Biblioteca que cria e manipula arquivos Excel

```
In [10]: !pip install openpyxl
```

```
Requirement already satisfied: openpyxl in /usr/local/lib/python3.7/dist-packages (2.5.9)  
Requirement already satisfied: et-xmlfile in /usr/local/lib/python3.7/dist-packages (from openpyxl) (1.0.1)  
Requirement already satisfied: jdcal in /usr/local/lib/python3.7/dist-packages (from openpyxl) (1.4.1)
```

```
In [11]: from openpyxl import Workbook
```

Criamos um novo Workbook

```
In [12]: wb = Workbook()
```

Selecionamos a primeira planilha ativa

```
In [13]: ws = wb.active
```

Adicionamos o número **42** para a célula **A1**

```
In [14]: ws['A1'] = 42
```

Salvamos como um arquivo Excel

```
In [15]: wb.save("exemplo.xlsx")
```

## Pandas

Biblioteca de manipulação de dados muito usada em Data Science.

```
In [16]: !pip install pandas
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.7/  
dist-packages (1.1.5)  
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/pyth  
on3.7/dist-packages (from pandas) (2018.9)  
Requirement already satisfied: numpy>=1.15.4 in /usr/local/lib/pyt  
hon3.7/dist-packages (from pandas) (1.19.5)  
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/loca  
l/lib/python3.7/dist-packages (from pandas) (2.8.1)  
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.  
7/dist-packages (from python-dateutil>=2.7.3->pandas) (1.15.0)
```

```
In [17]: import pandas as pd
```

Carrega o DataFrame para a memória do programa

```
In [20]: df = pd.read_csv("https://raw.githubusercontent.com/cs109/2014_data  
/master/countries.csv")
```

Exibe as 5 primeiras linhas do DataSet

```
In [21]: df.head()
```

```
Out[21]:
```

	Country	Region
0	Algeria	AFRICA
1	Angola	AFRICA
2	Benin	AFRICA
3	Botswana	AFRICA
4	Burkina	AFRICA

Agrupa o DataSet por região

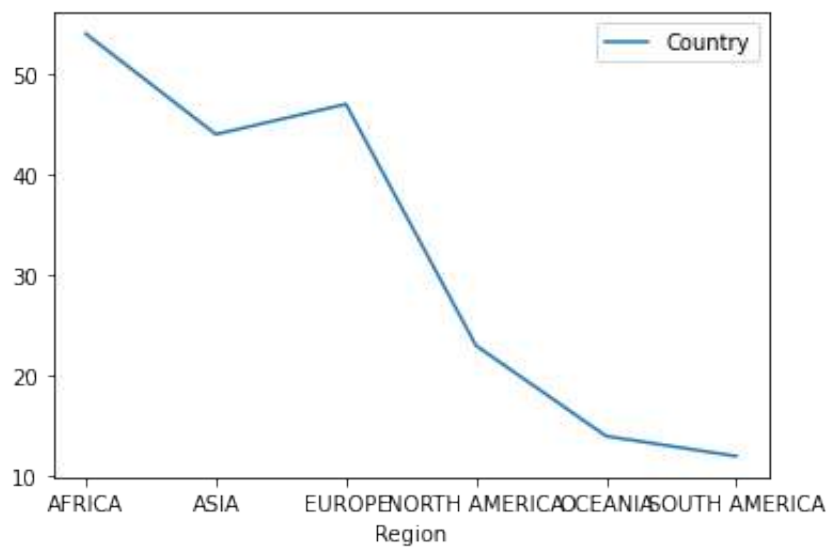
```
In [22]: df.groupby('Region').count()
```

```
Out[22]:
```

	Country
Region	
AFRICA	54
ASIA	44
EUROPE	47
NORTH AMERICA	23
OCEANIA	14
SOUTH AMERICA	12

```
In [25]: df.groupby('Region').count().plot()
```

```
Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9e51ffc490>
```





## Exercícios

1. Escreva uma classe Python chamada **Retangulo** construída dos atributos **comprimento** e **largura** e um método que irá computar a **área** de um retângulo. Outro método deve receber as dimensões dos 4 lados do triângulo e dizer se esse retângulo é um quadrado.

### Nota:

- Na geometria plana euclidiana, um retângulo é um quadrilátero com quatro ângulos retos. Para encontrar a área de um retângulo, multiplique o comprimento pela largura.
- Um retângulo com quatro lados de igual comprimento é um quadrado.
- A imagem a seguir representa a área de um retângulo.

# Área do Retângulo

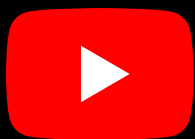


$$A = b \cdot h$$

1. Escreva uma classe chamada **Circulo** construída por um **raio** e dois métodos que irão calcular a **área** e o **perímetro** de um círculo.

# SIGA-NOS NAS REDES SOCIAIS

*Clique no ícone para ser redirecionado:*



## **YouTube**

Aqui eu compartilho vídeos com muito conteúdo para ajudar você a conquistar seus objetivos o mais rápido possível.



## **Blog**

Este é o portal de conteúdo, você vai encontrar diversos artigos passo a passo com estratégias para seu desenvolvimento pessoal na programação sobre diversas linguagens.



## **Instagram**

Aqui eu compartilho imagens dos meus bastidores, do meu dia a dia, pequenos vídeos com dicas etc.



## **Facebook**

Aqui você verá sempre a atualização das postagens do blog, comentários e engajamento dos meus alunos, fotos e dicas rápidas de programação.