

TECHMIMO

Autor: Rafael Pereira da Silva

Seguem alguns recados para ajudá-los e para contribuir com o curso:

- Fiquem à vontade para me contatar pelo Linkdin, costumo responder por lá também:
<https://www.linkedin.com/in/rafael-pereira-da-silva-23890799/> (<https://www.linkedin.com/in/rafael-pereira-da-silva-23890799/>)
- Fiquem a vontade para compartilharem o certificado do curso no LinkedIn. Eu costumo curtir e comentar para dar mais credibilidade
- Vocês podem usar esses notebooks para resolver os exercícios e desafios
- Não se esqueçam de avaliar o curso e dar feedback, eu costumo criar conteúdos baseado nas demandas de vocês
- Se tiverem gostando do curso, recomendem aos amigos, pois isso também ajuda a impulsionar e a crescer a comunidade
- Bons estudos e grande abraços!

Seção 5 - Funções e módulos

5.1 Funções

Definindo e chamando funções

def minha_funcao(argumentos):

```
    funcao_aqui
    minha_funcao(argumentos) ##para chamar a função
```

Declaração *return*

É usada quando se deseja retornar uma variável específica.

def minha_funcao(argumentos):

```
    resultado = operações
```

```
    **return** resultado
```

In [44]:

```
def minha_funcao():  
    print('meu nome é ' )  
    print('Python')
```

```
meu_nome = 'Rafael'
```

```
minha_funcao()
```

```
meu nome é  
Python
```

In [50]:

```
def soma_quadrados(a,b):  
    resultado = a**2 + b**2  
    return resultado
```

```
n_1 = 3
```

```
n_2 = 2
```

```
a = soma_quadrados(n_1,n_2)  
print(a)
```

```
13
```

In []:

In []:

5.2 Módulos

O que é

Módulos são arquivos Python (extensão `.py`) que contém conjuntos de funções, objetos, variáveis, entre outros. Essas funcionalidades podem ser chamadas das seguintes maneiras:

```
import Nome_do_modulo
```

```
import Nome_do_modulo as abreviação
```

```
from Nome_do_modulo import algum_objeto
```

Referência

Lista de módulos do Python: <https://docs.python.org/3.8/py-modindex.html> (<https://docs.python.org/3.8/py-modindex.html>)

In [53]:

```
import math

a = math.cos(0)
type(a)
```

Out[53]:

float

In [55]:

```
import math as mt

y = mt.sin(0)

y
```

Out[55]:

0.0

In [57]:

```
from math import pi, sin, cos

x = sin(pi/2)

x
```

Out[57]:

1.0

5.3 Criando um módulo

Podemos criar um arquivo `.py` utilizando outra IDE e depois importá-la.

In [59]:

```
import peso

m = 50

p_t = peso.peso_terra(m)
p_t
```

Out[59]:

490.00000000000006

In [63]:

```
from peso import g_terra,g_lua,peso_terra  
  
g_terra  
  
peso_terra(100)
```

Out[63]:

980.0000000000001

In []:

5.4 Exercício 1 -

Crie uma função que receba uma frase e retorne quantas palavras tem a frase. Dica: Utilize o método split().

In []:

In []:

In []:

5.5 Exercício 2 -

Crie uma função que lê um arquivo de extensão .txt e retorne quantas palavras tem esse arquivo. Dica: utilize o método readlines()

In []:

In []:

In []:

5.6 Exercício 3 -

Crie um módulo que receba o peso e a altura e calcula o IMC de uma pessoa.

$$IMC = \frac{peso}{Altura^2}$$

In []:

In []:

In []:

5.7 Preparação para o desafio 1 (matplotlib 3d)

In [97]:

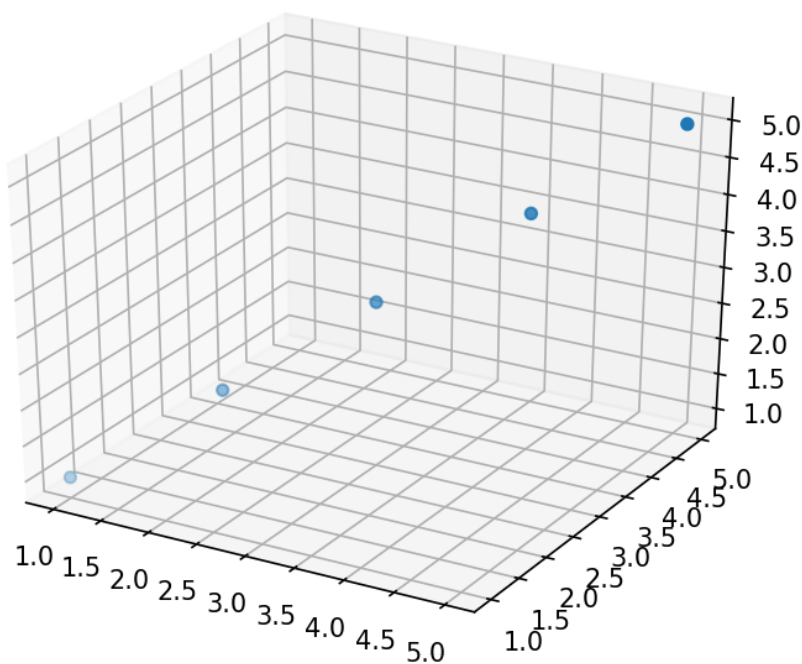
```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
%matplotlib notebook

x = [1,2,3,4,5]
y = [1,2,3,4,5]
z = [1,2,3,4,5]

fig = plt.figure()
ax = fig.add_subplot(111,projection='3d')

ax.scatter(x,y,z)
```

<IPython.core.display.Javascript object>



Out[97]:

<mpl_toolkits.mplot3d.art3d.Path3DCollection at 0x1dc8d60470>

In []:

In []:

In []:

In []:

In []:

5.8 Desafio 1 - Plotando coordenadas

Crie um módulo que leia um arquivo de pontos 3D e faça um gráfico desses pontos.

Arquivo de pontos: PONTOS_ESFERA.txt

Fonte para fazer o gráfico: https://matplotlib.org/mpl_toolkits/mplot3d/tutorial.html
(https://matplotlib.org/mpl_toolkits/mplot3d/tutorial.html)

In []:

In []:

In []: