



 Neste guião irás **aprender a importar** e utilizar algumas das mais **importantes bibliotecas** ao teu dispor em **PYTHON**.

 Cria um **novo programa** de nome **8 importmath.py**

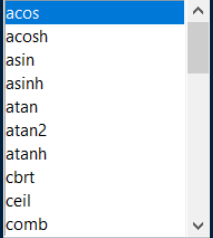
 Começa por **digitar** o seguinte **código** no teu IDE:

```
*8import.py - H:\Escola18ano\12anoAPIb\Python\8import.py (3.11.5)*
File Edit Format Run Options Window Help
import math
print(math.factorial(4))
```

 Corre o **programa**. Verás o **resultado** do **fatorial** de **4**.


 Tal como esta **função** existem **outras** que te podem **ajudar** na **disciplina** de **matemática**. Sabendo que para **aceder** às **funções disponíveis** na biblioteca **math** tens de escrever **math** **ponto** e esperar que uma **lista apareça**:


```
import math
print(math.factorial(4))
print(math.
```




Calcula e imprime o seguinte por **linha**:

- Cosseno de 0.
- Raiz quadrada de 16.
- Potência (pow) de 5^2
- Valor de e^2
- Logaritmo $\log_{10}(10)$
- Logaritmo $\ln(10)$
- Valor de Pi

 Guarda o teu **trabalho**.

 Cria um **novo programa** de nome **8 importGrafico.py**.

 Nem **sempre** as **bibliotecas** estão **disponíveis** para serem **utilizadas sem pré-instalação**. É o caso da **biblioteca** para fazer **gráficos** que vais **utilizar** neste **programa**.

Abre uma janela de **CMD** e **corre** o seguinte **comando**:

```
Linha de comandos
Microsoft Windows [Version 10.0.19042.631]
(c) 2020 Microsoft Corporation. Todos os direitos reservados.
C:\Users\RS>pip install matplotlib
```

 Quando **terminar a instalação** terás o seguinte **ecrã**:



```
Linha de comandos
Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
----- 247.7/247.7 kB 3.8 MB/s eta 0:00:00
Collecting six>=1.5 (from python-dateutil)>=2.7->matplotlib
Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
----- 11.0/11.0 kB 3.2 MB/s eta 0:00:00
Downloading matplotlib-3.8.2-cp311-cp311-win_amd64.whl (7.6 MB)
----- 7.6/7.6 MB 3.2 MB/s eta 0:00:00
Downloading contourpy-1.2.0-cp311-cp311-win_amd64.whl (187 kB)
----- 187.6/187.6 kB 2.8 MB/s eta 0:00:00
Downloading cycler-0.12.1-py3-none-any.whl (8.3 kB)
----- 8.3/8.3 kB 3.2 MB/s eta 0:00:00
Downloading fonttools-4.47.0-cp311-cp311-win_amd64.whl (2.2 MB)
----- 2.2/2.2 MB 3.2 MB/s eta 0:00:00
Downloading kiwisolver-1.4.5-cp311-cp311-win_amd64.whl (56 kB)
----- 56.1/56.1 kB 2.9 MB/s eta 0:00:00
Downloading numpy-1.26.3-cp311-cp311-win_amd64.whl (15.8 MB)
----- 15.8/15.8 MB 3.2 MB/s eta 0:00:00
Downloading packaging-23.2-py3-none-any.whl (53 kB)
----- 53.0/53.0 kB 2.9 MB/s eta 0:00:00
Downloading pillow-10.2.0-cp311-cp311-win_amd64.whl (2.6 MB)
----- 2.6/2.6 MB 3.2 MB/s eta 0:00:00
Downloading pyparsing-3.1.1-py3-none-any.whl (103 kB)
----- 103.1/103.1 kB 2.9 MB/s eta 0:00:00
Installing collected packages: six, pyparsing, pillow, packaging, numpy, kiwisolver, fonttools, cycler, python-dateutil,
contourpy, matplotlib
Successfully installed contourpy-1.2.0 cycler-0.12.1 fonttools-4.47.0 kiwisolver-1.4.5 matplotlib-3.8.2 numpy-1.26.3 pac
kaging-23.2 pillow-10.2.0 pyparsing-3.1.1 python-dateutil-2.8.2 six-1.16.0

[notice] A new release of pip is available: 23.2.1 -> 23.2.2
[notice] To update, run: python.exe -m pip install --upgrade pip
C:\Users\RS>
```



Digita o seguinte código no teu IDE (exercício retirado do livro Código 12 – Porto Editora):

```
Bimportgrafico.py - H:\Escola18ano\12anoAPIb\Python\Bimportgrafico.py (3.11.5)
File Edit Format Run Options Window Help
import matplotlib.pyplot as plt #plt passa a ser abreviatura para ser mais fácil
plt.plot(["Ana", "Bruno", "Inês", "Hugo", "Rita"], [23.5, 9.6, 24.3, 26.2, 25])
plt.title("Índice de massa corporal")
plt.xlabel("Nome")
plt.ylabel("IMC")
plt.show()
```

Explicação:

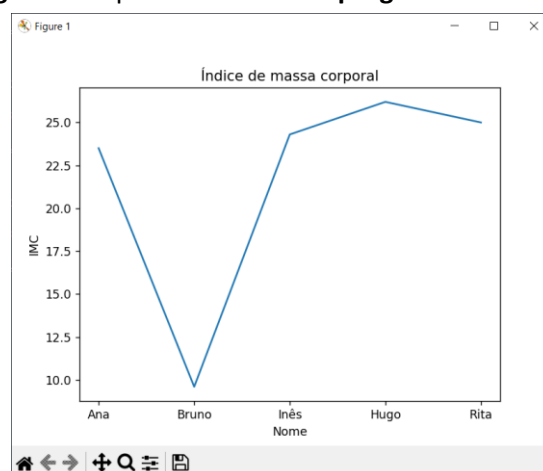
Depois de importares **matplotlib** terias de **andar** com **este nome** para acederes às **funções disponibilizadas** por esta **biblioteca**. Ao acrescentar **“as plt”** passa a ser **criado** um **Alias** que **substitui** a palavra grande **matplotlib** por apenas **plt**.

Na **segunda linha** indicas a **informação** sobre a qual o **gráfico** vai ser **construído** (igual ao Excel quando se selecionam as células antes de escolher o tipo de gráfico)

Na **terceira linha** é o **título** do gráfico e na **4ª** e **5ª** respetivamente o **título** dos eixos **XX** e **YY**. **Finalmente** manda **mostrar** o gráfico. Sem esta linha não mostra nada.



Irás **obter** algo deste **género** depois de **correres** o **programa**:



Adiciona o teu nome e o teu **IMC** ao **programa**. **Volta** a **correr** o **programa**.




Clica sobre a **diskette** e **guarda** o **gráfico** no teu **PC** (quem está noutra IDE pode fazer um print screen).

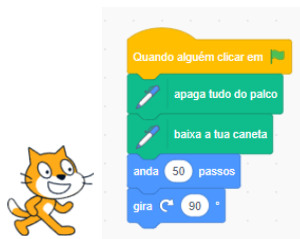


Guarda o teu **trabalho**.



 Cria um novo programa de nome **8importTurtle.py**

 Esta biblioteca permite-te **desenhar**. Para quem já utilizou a **Caneta em Scratch** este código é muito semelhante.



 Para já, passa o seguinte código para o teu IDE:

```
8importTurtle.py - H:/Escola18ano/12anoAPIb/Python/8importTurtle.py (3.11.5)
File Edit Format Run Options Window Help
import turtle as t
t.reset()
t.speed(2)
t.pendown()
t.forward(100)
```

Explicação:

O **reset()** limpa o ecrã.

O **speed(2)** é a **velocidade** com que se **vê** a **desenhar**.

Pendown() baixa a caneta, i.e. fica pronta a desenhar (penup()) é para levantar)

Forward(100) é para se **deslocar 100 pixels**, mas como a caneta está para baixo irá fazer um risco de 100 pixels.

 Corre o programa:




Neste caso fez uma **aresta** de um **quadrado**.

 Acrescenta a seguinte linha:

```
t.right(90)
```

Esta linha **permite alterar a orientação da caneta**. Neste caso **fica virada para baixo**:





 Está **pronta** para **desenhar** outra **aresta** do **quadrado**. Achas que **consegues desenhar** esta **aresta sem ajuda**? E **completar** o **quadrado**?

PRO: depois de terminares o quadrado identifica no código o que se repete e simplifica utilizando um ciclo que ter permita desenhar o mesmo quadrado.

HACKER: cria uma função com o código de criar o quadrado que aceite um argumento. Este argumento é o tamanho da aresta do quadrado introduzido pelo utilizador.




 **Guarda o teu trabalho.**


 **Cria um novo programa de nome `8importRandom.py`**

 Neste programa vais **aprender a sortear números (aleatórios).**


 **Começa por copiar as seguintes linhas de código para o teu IDE.**

```
8importrandom.py - H:\Escola18ano\12anoAPIb\Python\8importrandom.py (3.11.5)
File Edit Format Run Options Window Help
import random
print(random.randrange(1,10))
```

 Este **programa** sorteia n^{os} inteiros entre 1 e 9. Sabendo isso, **altera o programa** para que **sorteie um número de 0 a 19 exclusive.**

 **Acrescenta código** que te **permita, após o sorteio verificar** se o número é **par ou ímpar** e dar esse **feedback** ao **utilizador.**


O número que saiu no sorteio foi o X e trata-se de um número ímpar/par.

 **Altera o programa** para que o **sorteio de n^{os} seja realizado** entre **dois valores escolhidos** pelo **utilizador.** Exemplo:


Quer sortear um valor entre que números?


Valor menor:

Valor maior:


 **Guarda o programa.**

 **Cria um novo programa de nome `8randomlista.py`.**

 **Pretende-se que geres 10 números aleatórios e os adiciones** a uma **lista de nome sorteio.** **Atenção** **deves utilizar ciclos.** No **final** **deves mostrar no ecrã** todos os **números sorteados recorrendo** à chamada de uma **função** para o efeito.

 **Guarda o teu programa.**

 **Chama o teu professor para avaliar. Mostra também** os outros programas **que realizaste.**

 **Envia todos os programas** para a tua pasta **Programas** na **cloud.**