

### Objectivo

Este projecto tem como objectivo iniciar a exploração da linguagem `python` (versão 3), e de alguns dos seus módulos de extensão na aplicação ao cálculo científico. Sempre que necessite recorra ao sistema de ajuda do `python` ou pesquize na *web*.

Para referência futura poderá guardar toda a sua sessão de trabalho num *ficheiro-py* (ficheiro executável com extensão “.py”). Para facilitar a sua leitura posterior, identifique as questões e adicione comentários apropriados nas linhas de comandos.

### Exercícios

- Da lista apresentada na tabela, teste e diga quais as designações que são válidas para nome de variáveis em `python`.

BC	F.	delta_T	período
bC''	L+4	delta T	delta-T
6x	and	raise	n_factorial
n!	(y)	A*B	\$doc
T56	T?56	_temp	assert
x_1	v:a	x%	x_max

- Com recurso às funções existentes no módulo `math`, calcule as seguintes quantidades:

a)  $3 \frac{\sqrt{5}-1}{(\sqrt{5}+1)^{1/5}} - 1$

b)  $e^3; \ln(e^3); \log_{10}(e^3); \log_{10}(10^5)$

c)  $\sin \frac{\pi}{4}; \sin(45^\circ); \sin^2\left(\frac{\pi}{4}\right)$

d)  $\sqrt{-2}; \sqrt{|-2|}; \frac{1+3i}{1-3i}$

e)  $e^{i\frac{\pi}{4}}$  e verifique a fórmula de Euler  $e^{ix} = \cos x + i \sin x$

- f) Sendo  $c = 3 + 4i$ , calcule o conjugado de  $c$  e o módulo de  $c$ , recorrendo às funções `python` apropriadas.

- Atendendo a que “as variáveis em `python` são apontadores para objectos”, interprete os resultados que obtém com a seguinte sequência de instruções.

```
x = [1, 2, 3]
y = x
print (y)
```

## Laboratório Numérico 2018-19

```
x.append(4)
print (y)

x = 'qualquer coisa'
print (y)
```

**4.** Importe o módulo de extensão numpy com o pseudônimo np.

**a)** Liste as funções disponíveis no módulo np e obtenha informação sobre as funções array e arange.

**b)** Reproduza o resultado que se obtém com a instrução abaixo, recorrendo à função arange do módulo np.

```
[val ** 2 for val in range(1, 10)]
```

**c)** Execute os dois blocos de código e explique os resultados obtidos

```
from numpy import array, sqrt, sin    from numpy import array
a = array([1.0, 4.0, 9.0, 16.0])      from math import sqrt
print(sqrt(a))                       a = array([1.0, 4.0, 9.0, 16.0])
print(sin(a))                         print(sqrt(a[1]))
                                       print(sqrt(a))
```

**5.** Utilizando a capacidade das funções do módulo numpy que operarem diretamente sobre matrizes, desenvolva uma versão do código abaixo que permita eliminar o ciclo for (vectorização).

```
from math import sqrt, sin, pi
x = 0.0; s = 0.0
for i in range(101):
    s = s + sqrt(x)*sin(x)
    x = x + 0.01*pi
print(s)
```