

# AULA 1

Introdução ao curso. Regras de avaliação.  
Materiais de estudo. Revisão de conceitos  
básicos de programação em python.

# Laboratório Numérico 2017/2018

**Teóricas:** Pedro Miranda [pmmiranda@fc.ul.pt](mailto:pmmiranda@fc.ul.pt), (C8, 8.3.38)

**Práticas:** Fernando Soares, Pedro Mateus

## Objectivos

**Resolução de problemas** em Física Aplicada e Engenharia com utilização de métodos numéricos

## Linguagem

**python**

## Avaliação

Exame final: 50%, nota mínima 8

Trabalho de grupo: 50% *Projectos obrigatórios*

# Trabalho de grupo

1 **problema** por semana ( $\pm 9$ ). Divulgado na 5ª feira anterior.

Os problemas deverão ser feitos até ao final da **aula** e entregues (py file comentada)

Cada grupo só pode falhar **2** projectos

O trabalho prático (projectos) é **obrigatório**

**Práticas começam na semana de 26/2**

# Importante

O trabalho é em grupo mas a avaliação é individual

Grupos de  $\leq 2$

Há marcação de presenças nas aulas práticas

# Estudo

Notas do curso e ppts das aulas  
(progressivamente...)

phoenix

Help: google “python anything”

# python

Existem 2 versões (python 2 e python 3) não totalmente compatíveis... Quando se utiliza código já escrito pode ser necessário ter os 2.

Vamos usar a versão **3.6**, na distribuição **anaconda**, recorrendo à interface gráfica **spyder**

A versão 3.6 é mais moderna e consistente, mas existem aplicações que só correm na 2.7.

# spyder

Interface gráfica para o python: inclui editor, linha de comandos (ipython), visualizador de variáveis (debug)

Pode ser necessário limpar a memória do spyder, de ve em quando:

```
reset #apaga tudo
```

```
plt.close('all') #apaga todas as figuras
```

Nota: os scripts \*.py são ficheiros de texto e podem ser corridos na linha de comandos (sem acumular lixo na memória):

```
python script.py
```

# Objetos

O python processa **objetos**

Cada objeto é identificado por um nome:

`X, x, xis, a22, b_2`

Notar que  $x \neq X$

Cada objeto, consoante a **classe** a que pertença, pode conter conjuntos de números, textos e pedaços de Código

Os objetos mais simples são **escalares**: contêm um número. Esse número é de um dado **tipo**

# Tipos numéricos

*Plain integer (32 bit)*

*Long integer (n bit)*

*Floating point (64 bit)*

*Complex (2 x 64bit)*

*Boolean (0/1)*

# Listas

As listas são objetos constituídos por outros objetos

Exemplos:

```
x=[0,1]
```

```
Y=[0,10.,'texto']
```

```
z=[x,Y]
```