

Programação I / Introdução à Programação

João Pedro Pedroso

2024/2025

Programação I / Introdução à Programação

Primeiro curso de programação em diversas licenciaturas da Faculdade de Ciências



Pretende-se que os estudantes:

- 1 Se familiarizem com a utilização de computadores pessoais com sistema operativo da família GNU/Linux;
- 2 Aprendam a programar na linguagem Python e a executar os programas num terminal;
- 3 Adquiram competência para codificar algoritmos elementares;
- 4 Adquiram bons hábitos de programação e de estruturação dos programas;
- 5 Apreendam algumas estruturas de dados e algoritmos fundamentais;
- 6 Adquiram competências para testar e corrigir erros dos seus programas.

- 1 Percepção do papel da programação na resolução de problemas na licenciatura.
- 2 Domínio dos componentes básicos de uma linguagem de programação recente.
- 3 Capacidade de escrever programas que permitam realizar objetivos úteis.
- 4 Confiança no uso da linguagem Python e da biblioteca padrão.

Programa

- 1 Breve introdução aos computadores
- 2 Variáveis, expressões, instruções
- 3 Fluxo de um programa
- 4 Funções
- 5 Tipos de dados
- 6 O módulo numpy
- 7 Ficheiros
- 8 Módulos Python
- 9 Mais tipos de dados
- 10 Recursão
- 11 Classes e objetos
- 12 Exceções

Programação: O que é um programa?

Um **programa** é uma sequência de instruções que determina como efetuar um cálculo

- matemático
- simbólico
- ...

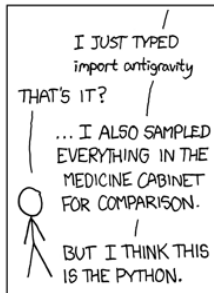
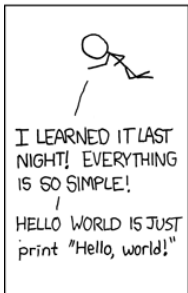
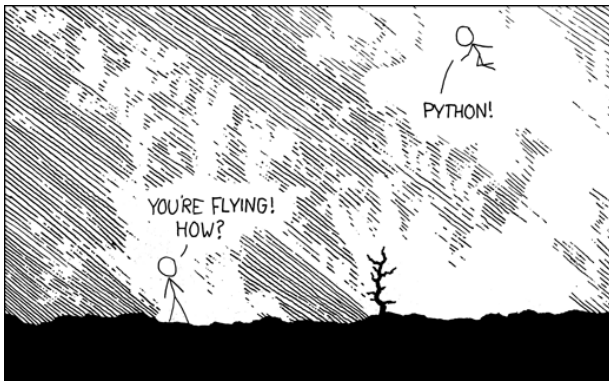
Instruções básicas, presentes em praticamente todas as linguagens:

- **input** → obter dados do teclado ou de outro dispositivo
- **output** → escrever dados no ecrã ou nouro dispositivo
- **cálculo matemático** → operações como adição e multiplicação
- **execução condicional** → testar uma condição e decidir quais as instruções a executar de acordo com o resultado
- **repetição** → executar um conjunto de instruções repetidamente, geralmente com pequenas variantes

Lingagem que iremos utilizar neste curso: **Python**

Python: porquê?

- ① fácil de programar
 - rapidez de programação
 - programas curtos e fáceis de ler
 - menos erros
- ② portabilidade
 - podem ser executadas em computadores diferentes
- ③ atualmente, a linguagem mais utilizada em cálculo científico
- ④ biblioteca de apoio muito completa



- Ver página no sistema de informação da Universidade:
https://sigarra.up.pt/fcup/pt/UCURR_GERAL.FICHA_UC_VIEW?pv_ocorrencia_id=487782
- Notar:
 - Condições de Obtenção de frequência:
 - Cálculo da nota / forma de avaliação
 - Docentes
 - Horário

- Aulas práticas:
 - exercícios com correção automática, disponíveis apenas nos laboratórios
 - sistema funciona por níveis
 - só é corrigido o nível $n + 1$ depois de se ultrapassar o nível n
 - no total, haverá 10 níveis; é necessário ultrapassar pelo menos 5 para obter frequência
- Aulas teóricas:
 - exercícios com correção automática, disponíveis:
 - no final da aula (últimos 10 minutos)
 - no final do dia, para quem não pode ir à aula (das 20:00 às 20:15)
 - quizzes (questões de escolha múltipla em cada uma das aulas)
 - é necessário ter pelo menos 50% da cotação total para obter frequência

- Feito no computador, com correção automática
- Salas com a mesma configuração das práticas
- Revisão pelo docente → pontuação pela *elegância* dos programas

Recursos para esta unidade curricular

- **SIGARRA**: sistema de informação da Universidade
- Páginas no **Moodle** (devem ter recebido convite)
 - links para textos de aulas teóricas e práticas
 - onde estudantes devem colocar perguntas
- **Página web** cc1008 + cc1015 + ccinf1001
 - slides de aulas, folhas de exercícios
 - <http://www.dcc.fc.up.pt/~jpp/P1>
- **Codex**: Página com exercícios das aulas teóricas
 - <https://codex.dcc.fc.up.pt/cc1015t/>

- Teóricas:
 - exposição da matéria
 - momentos para esclarecimento de dúvidas
 - "quizzes", perguntas simples sobre a matéria dada
 - exemplo: <https://codex.dcc.fc.up.pt/cc1015t/>
- Práticas:
 - Início: esta semana
 - avaliação contínua: exercícios com correção automática
 - no máximo, 3 tentativas ("*submissões*") por cada exercício
 - cada semana tem vários exercícios de um **nível**
 - têm de se fazer para "passar" o nível
 - se não passar, repete na semana seguinte
 - como nos jogos

Acesso:

- Utilizar **credenciais de acesso ao SIGARRA**
 - aula prática desta semana: verificar acesso
 - Codex nos labs: só a partir da próxima semana
- As credenciais serão as mesmas para o **Codex**

- Obrigatória

- Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers
[How to Think Like a Computer Scientist: Learning with Python 3](#)
download
- Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers
[How to Think Like a Computer Scientist, 2012](#) *link*

- Complementar

- Versão interativa desse livro ([recomendada](#)): *link*
- Allen B. Downey [Think Python](#) *link*

Até à próxima semana:

- responder ao *quiz* da aula 1 em <https://codex.dcc.fc.up.pt/cc1015t/>
- instalar Python no vosso computador
- escolher e instalar um editor de texto

Instalação do Python:

- Página de referência: <https://www.python.org>
 - seguir link *Download*
 - disponível para Linux, macOS, Windows, ...
- Outra possibilidade: *anaconda*
<https://www.anaconda.com/download>
 - por omissão, coloca todos os ficheiros em 'home'
 - não é necessário permissões de administrador
 - não interfere com outras versões de Python que possam estar instaladas
 - vem com muitas bibliotecas científicas pré-instaladas

[Python](#)[PSF](#)[Docs](#)[PyPI](#)[Jobs](#)[Community](#)

[Donate](#)

[GO](#)

[Socialize](#)

[About](#)[Downloads](#)[Documentation](#)[Community](#)[Success Stories](#)[News](#)[Events](#)

```
# Python 3: Fibonacci sequence
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while b < n:
>>>         a, b = b, a + b
>>>     print(a)
>>> fib(1000)
0 1 1 2 3 5 8
987
```

[All releases](#)[Source code](#)[Windows](#)[Mac OS X](#)[Other Platforms](#)[License](#)[Alternative Implementations](#)

Download for Mac OS X

Python 3.8.6

Not the OS you are looking for? Python can be used on many operating systems and environments. [View the full list of downloads.](#)

ing functions.
ments, keyword
s. [More about](#)

Python is a programming language that lets you work quickly and integrate systems more effectively. [>>> Learn More](#)

Get Started

Whether you're new to programming or an experienced developer, it's easy to learn and use Python.

Download

Python source code and installers are available for download for all versions!

Latest: [Python 3.9.0](#)

Docs

Documentation for Python's standard library, along with tutorials and guides, are available online.

Jobs

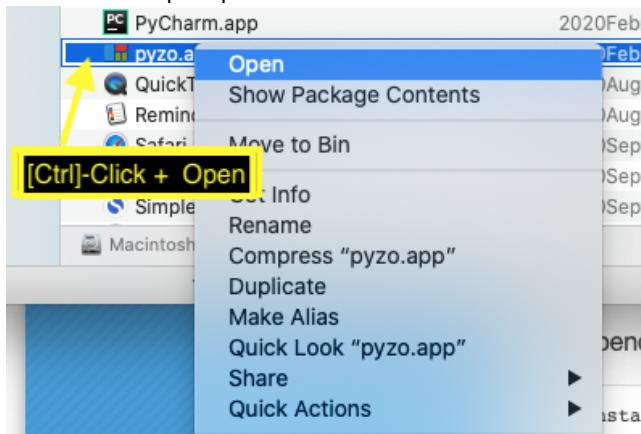
Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run job board** is the

Instalação de um editor

- Editor recomendado: **Pyzo**
`https://pyzo.org`
 - instruções para instalação: `https://pyzo.org/start.html`
- Outras possibilidades:
 - idle (editor simples, distribuído com o Python)
 - atom: `https://atom.io`
 - emacs (utilização mais avançada)
 - PyCharm (IDE completo, incluindo *debugger*)

Instalação e utilização do Pyzo

- Windows
 - Download + Install
- MacOS
 - Download + Install
 - Para executar pela primeira vez:



Pyzo detected a Python interpreter, but for scientific programming we recommend [miniconda](#). If you want to manually specify the interpreter, set the exe in the [shell config](#).

Click one of the links above, or [refresh](#).



Click star to bookmark current dir

/Users/jpp

- ▶ .adobe
- ▶ .android
- ▶ .astropy
- ▶ .atom
- ▶ .bash_sessions
- ▶ .byobu
- ▶ .cabal
- ▶ .cache

!*.pyc



Search in files

Python

+ Add config

name Python

exe

ipython

gui

pythonPath

startupScript

startDir

argv

✓

/opt/local/Library/Frameworks/Python.framework/Versions/3.7/bin/python

/Library/Frameworks/Python.framework/Versions/3.7/bin/python3.7 [v

/Library/Frameworks/Python.framework/Versions/3.6/bin/python3.6

/opt/local/Library/Frameworks/Python.framework/Versions/2.7/bin/python

/opt/local/Library/Frameworks/Python.framework/Versions/2.7/bin/pythonw2.7

/opt/local/lib/pypy/bin/pypy [v2.7.13]

/usr/bin/python [v2.7.10]

/System/Library/Frameworks/Python.framework/Versions/2.7/bin/python2.7 [v

/System/Library/Frameworks/Python.framework/Versions/2.7/bin/pythonw2.7

/path/to/script.py

- ☒ File to run at startup
- ☐ Code to run at startup

/path/to/your/pyth...

arg1 arg2 "arg wit...

Cancel

Done

<tmp 1>

```
1 def y(v0, t):  
2     '''Posição de um corpo em queda.'''  
3     g = 9.8  
4     return v0 * t - 1 / 2 * g * t * t  
5
```

Shells

Python

Python 3.7.4 (default, Sep 7 2019, 19:52:29) on da
rwin (64 bits).
This is the Pyzo interpreter with integrated event
loop for ASYNCIO.
Type 'help' for help, type '?' for a list of *magic
* commands.

>>> (executing lines 1 to 4 of "<tmp 1>")

>>> y(1,2)
-17.6

>>>

Source str...

File Browser



Click star to bookmark current dir

y0

/Users/jpp

- ▶ .adobe
- ▶ .android
- ▶ .astropy
- ▶ .atom
- ▶ .bash_sessions
- ▶ .byobu
- ▶ .cabal
- ▶ .cache

!*.pyc



Search in files

- Introdução aos computadores
- Iniciação à programação em Python