

# Fundamentos da Programação

Computadores, Algoritmos e Programas  
Ambiente de Desenvolvimento

Aula 02

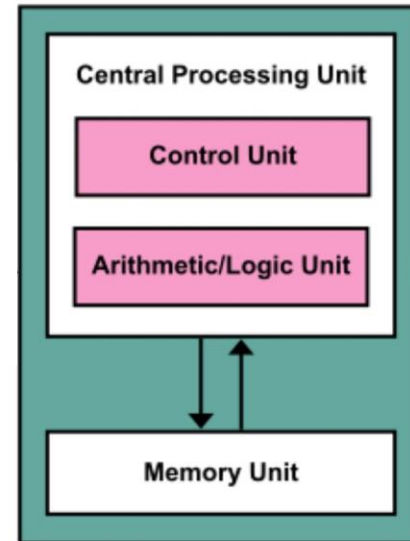
# Computadores, Algoritmos e Programas

“The single most important skill for a computer scientist is **problem solving**”,  
*Python for Software Design*

- Problema
- Algoritmo
- Computador
- Programa
- Linguagem de Programação

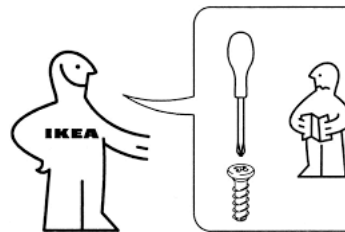
# Computadores

- Definição: Máquina que manipula informação
- Características:
  - Automático
  - Universal
  - Digital
  - Eletrónico
- Conjunto de instruções:
  - Entrada
  - Saída
  - Aritméticas
  - Condicionais
  - Repetição



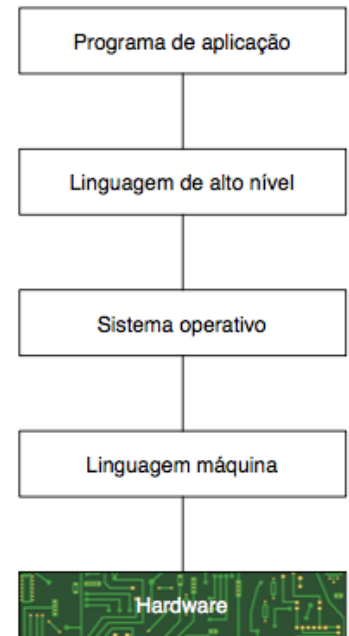
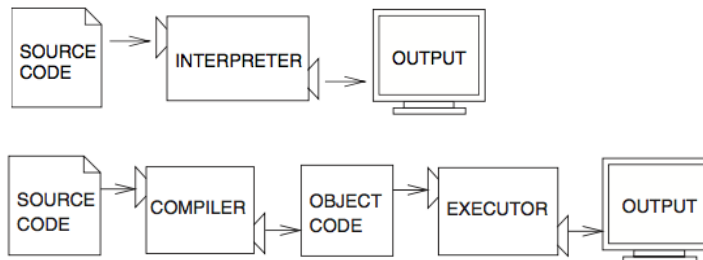
# Algoritmos

- Definição: Sequência finita de instruções bem definidas e não ambíguas, executáveis num período de tempo finito e com uma quantidade de esforço finita.
- Características:
  - rigoroso
  - eficaz
  - termina
- Exemplos informais:



# Programas

- Definição: Concretização de algoritmo numa linguagem de programação
- Linguagens de programação:
  - Assembly / Código Máquina
  - Alto-nível
- Processadores de linguagem:
  - Interpretados
  - Compilados
- Testar programas → depurar/*debugging*



- Definição linguagen
- Linguagem
  - Assembl
  - Alto-nív
- Processa
  - Interpre
  - Compila

92

9/9

0800 Antan started  
1000 stopped - antan ✓

1300 (033) HP-MC 1.2700 9.037 847 025  
(033) PRO 2 2.130476415 9.037 846 995 consch  
consch 2.130676415 4.615925059(-2)

Relays 6-2 in 033 failed special speed test  
in Relay 11.000 test.

Relays changed

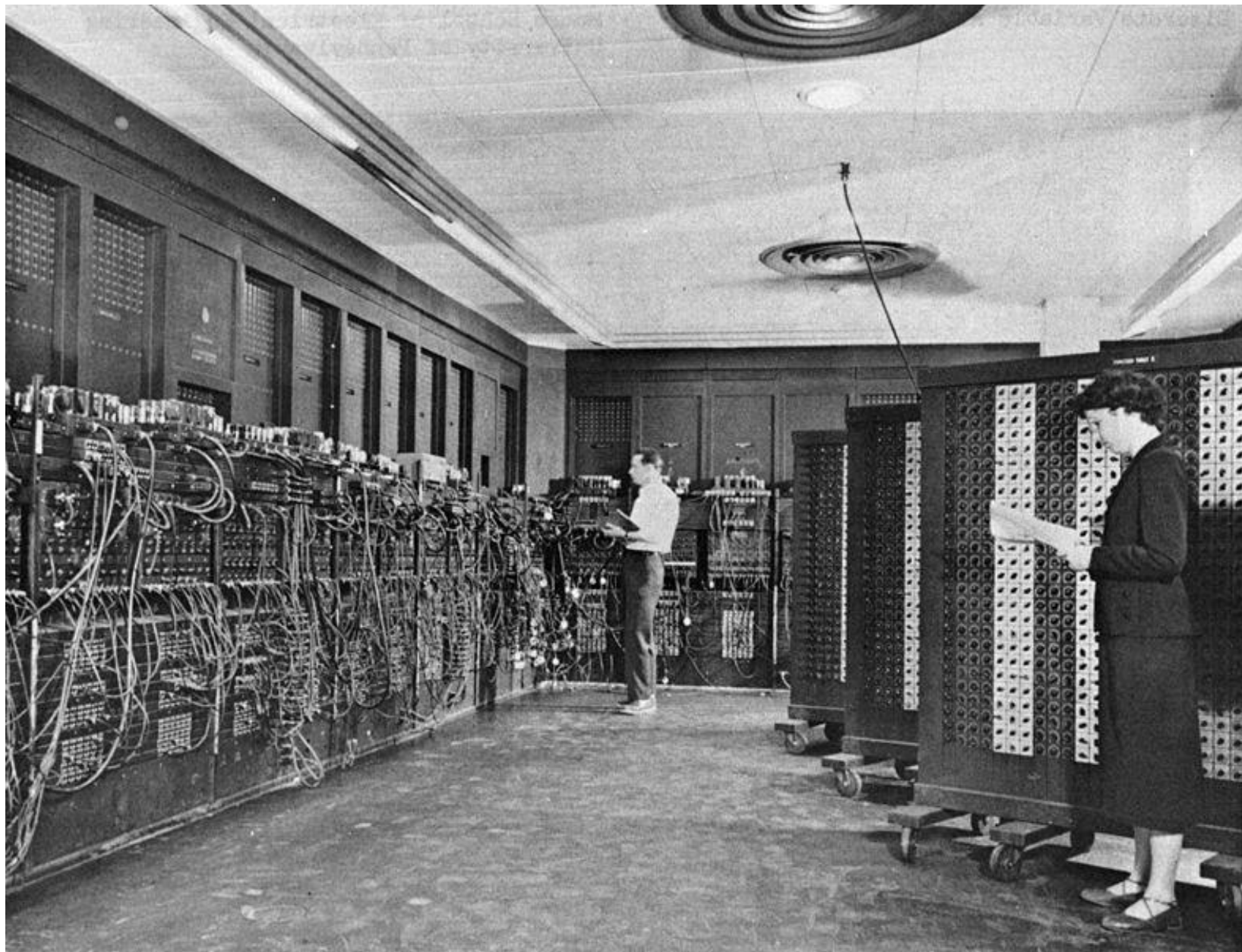
1100 Started Cosine Tap (Sine check)  
1525 Started Multi-Adder Test.

1545 Relay #70 Panel F  
(moth) in relay.

First actual case of bug being found.

1630 Antan started.  
1700 closed down.

- Testar programas → depurar/debugging





# Exemplo primeiro algoritmo/programa

Em matemática, o **fatorial** de um número natural  **$n$** , representado por  **$n!$** , é o produto de todos os inteiros positivos menores ou iguais a  **$n$** .



# The Python Programming Language

<https://www.python.org>



GO

Socialize

About

Downloads

Documentation

Community

Success Stories

News

Events

```
# Python 3: Simple output (with Unicode)
```

```
>>> print("Hello, I'm Python!")
```

```
Hello, I'm Python!
```

```
# Input, assignment
```

```
>>> name = input('What is your name?\n')
```

```
>>> print('Hi, %s.' % name)
```

```
What is your name?
```

```
Python
```

```
Hi, Python.
```



## Quick & Easy to Learn

Experienced programmers in any other language can pick up Python very quickly, and beginners find the clean syntax and indentation structure easy to learn. [Whet your appetite](#) with our Python 3 overview.

1

2

3

4

5

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

[https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))

<https://docs.python.org/3/>

# Porque o Python?

- Alto nível
- Oferece múltiplos paradigmas:
  - *functional, object-oriented, imperative, ...*
- Escalável/reutilizável/extensível
- Portável
- Interpretado
- Eficiente e rápido para prototipagem
- Gestão de memória e tipos dinâmicos
- Fácil de aprender e ler
- Muito utilizado

# Python: Interpreter

```
2. ist90700@sigma02.ist.utl.pt: /afs/ist.utl.pt/users/0/0/ist90700 (ssh)
alberto@caprica ~ $ ssh sigma.tecnico.ulisboa.pt -l ist90700
Password:
Linux sigma02.ist.utl.pt 4.9.0-8-amd64 #1 SMP Debian 4.9.110-3+deb9u4 (2018-08-21) x86_64
#####

      Bem vindo ao cluster sigma.

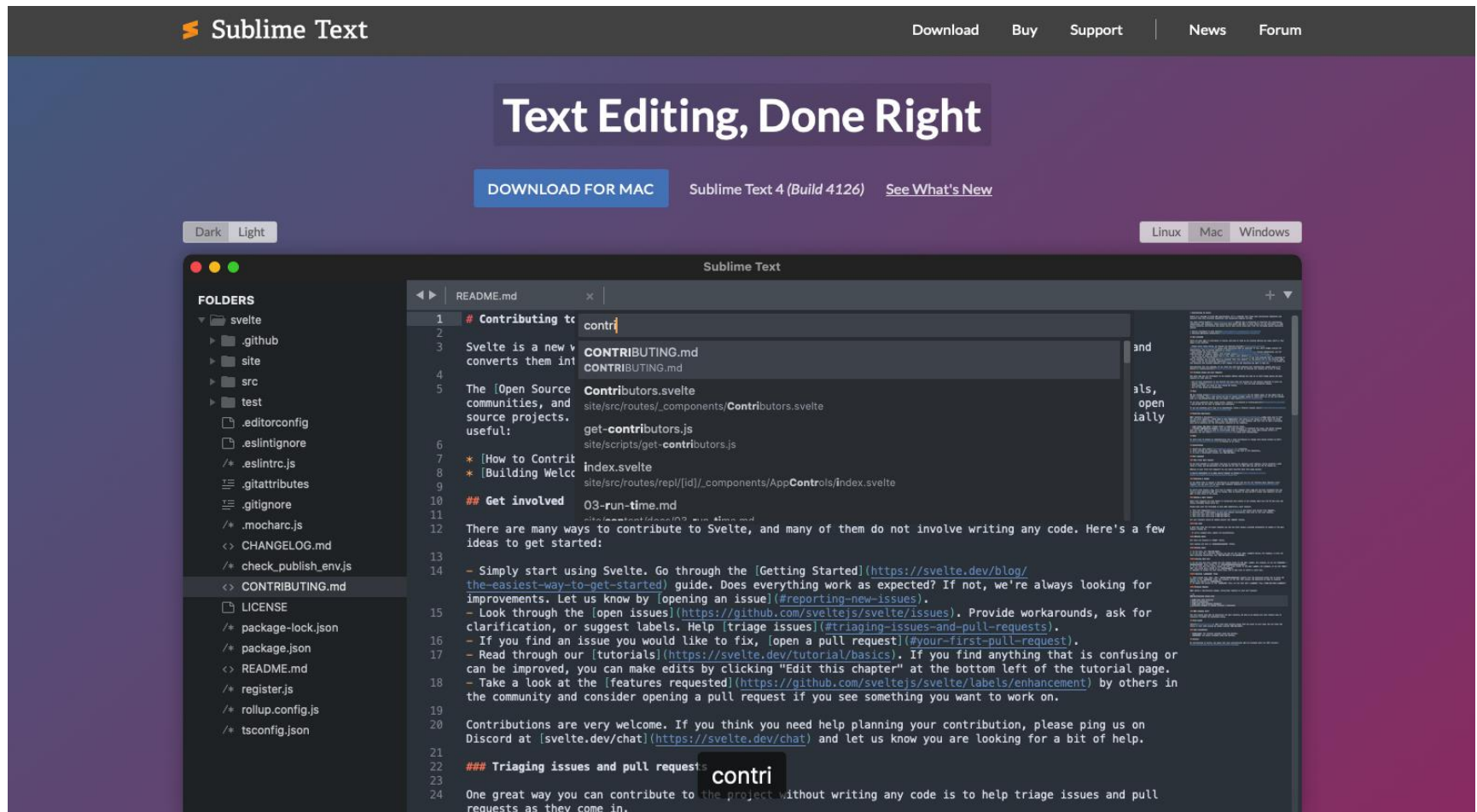
Informacoes : https://dsi.tecnico.ulisboa.pt/
Suporte      : dsi@tecnico.ulisboa.pt

#####
v6.4n
ist90700@sigma02:~$ python
Python 2.7.13 (default, Nov 24 2017, 17:33:09)
[GCC 6.3.0 20170516] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
ist90700@sigma02:~$ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170118] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> |
```

- Localmente:
  - Utilizando o terminal em sistemas Unix-like
- Remotamente por **ssh**:
  - Ativar serviços de **afs** e **shell** no [self service](#) da DSI.
- Atenção à versão de Python ( $\geq 3.6$ )
- Interpretador: *interativo* vs *script*

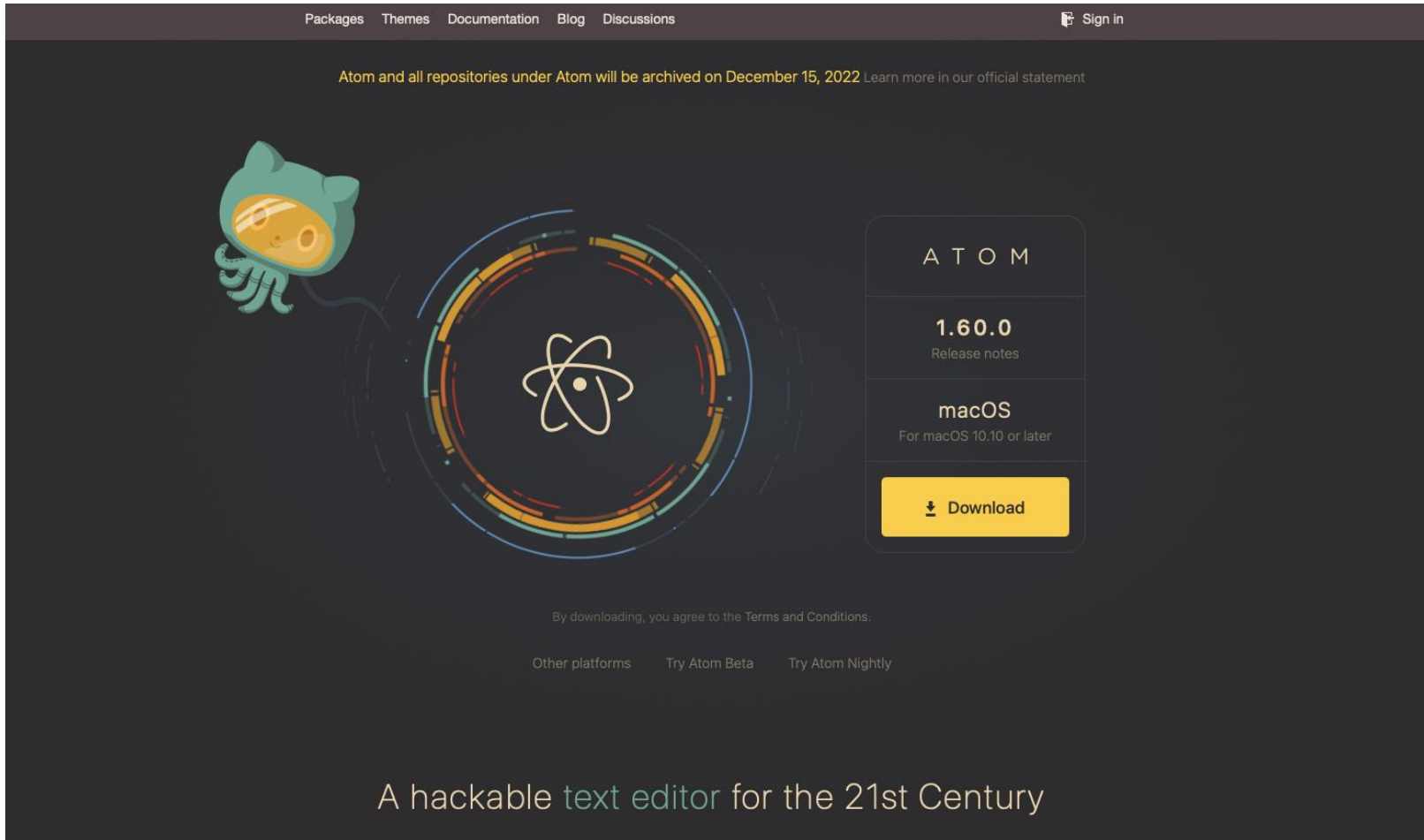
# Text Editors: SublimeText (and others)

<https://www.sublimetext.com>



# Text Editors: SublimeText (and others)

<https://atom.io>



# Text Editors: SublimeText (and others)

<https://notepad-plus-plus.org>



Current Version 8.4.5

Home

- Download
- News
- Online Help
- Resources
- RSS
- Donate
- Author



Automatically fix C++ issues related to code readability, language semantics, performance & more. Always free!

ADS VIA CARBON

## What is Notepad++

Notepad++ is a free (as in “free speech” and also as in “free beer”) source code editor and Notepad replacement that supports several languages. Running in the MS Windows environment, its use is governed by GNU General Public License.

Based on the powerful editing component Scintilla, Notepad++ is written in C++ and uses pure Win32 API and STL which ensures a higher execution speed and smaller program size. By optimizing as many routines as possible without losing user friendliness, Notepad++ is trying to reduce the world carbon dioxide emissions. When using less CPU power, the PC can throttle down and reduce power consumption, resulting in a greener environment.

```
1 #include <GPL>
2 #include <free_software>
3
4 void Notepad4ever()
5 {
6     while (true)
7     {
8         Notepad++ ;
9     }
10 }
```

# Visual Studio Code: (Python) IDE

<https://code.visualstudio.com>



Version 1.71 is now available! Read about the new features and fixes from August.

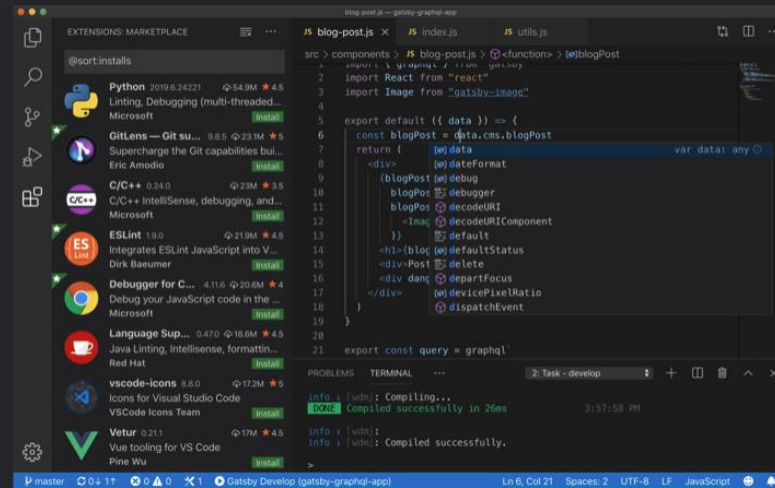
Code editing.  
Redefined.

Free. Built on open source. Runs everywhere.

Download Mac Universal  
Stable Build

Web, Insiders edition, or other platforms

By using VS Code, you agree to its  
license and privacy statement.



IntelliSense



Run and Debug



Built-in Git



Extensions



# iPython & Jupyter: Advanced interpreter and interactive notebooks

<http://jupyter.org>



```
pip3 install jupyter
```

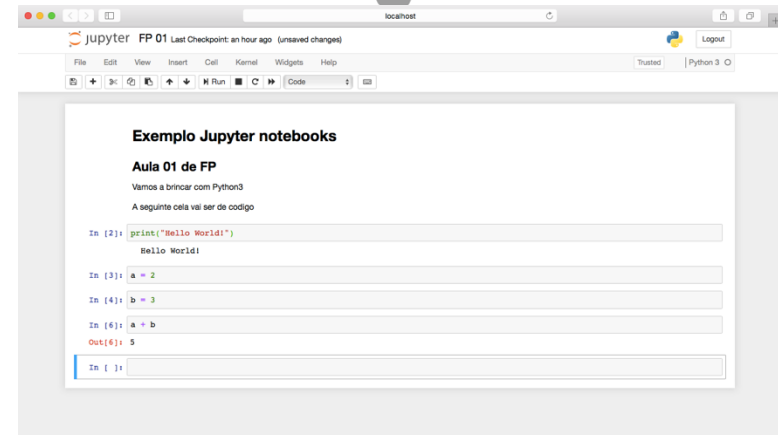
**IP[y]:** IPython  
Interactive Computing



```
1. IPython: Users/alberto (Python)
alberto@DemosAir ~ $ ipython
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 26 2018, 23:26:24)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.5.0 -- An enhanced Interactive Python. Type '?' for help.

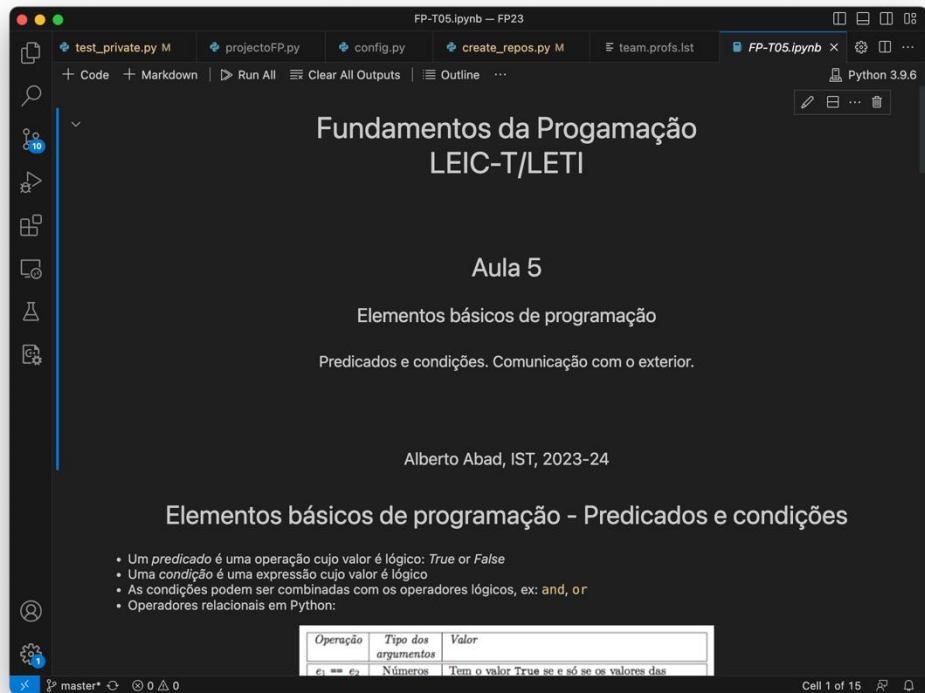
In [1]:
In [1]:
In [1]: print("Hello World!")
Hello World!

In [2]:
```



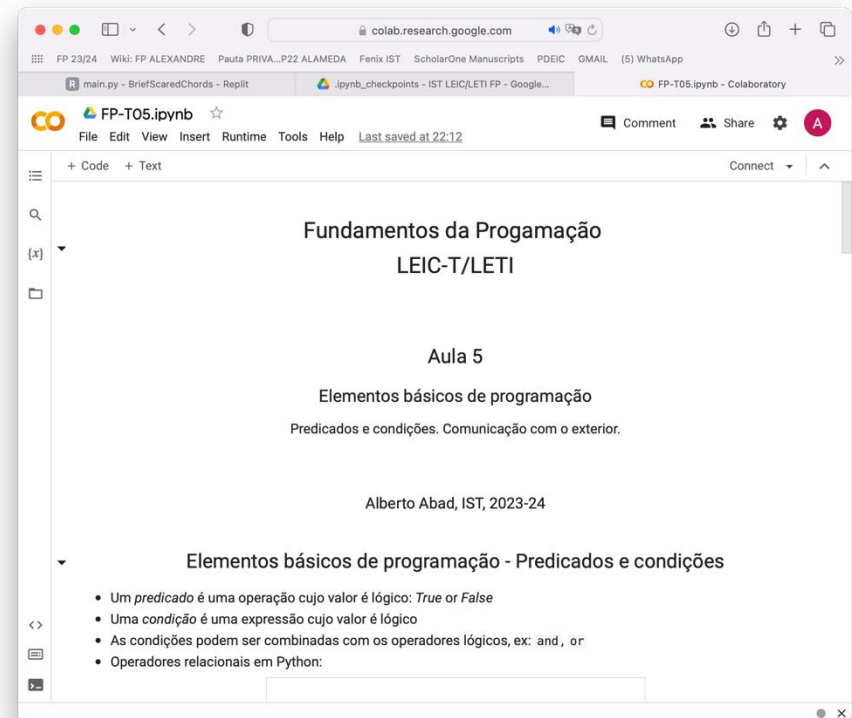
# Outras formas de utilizar Jupyter Notebooks

## Visual Studio Code (local)



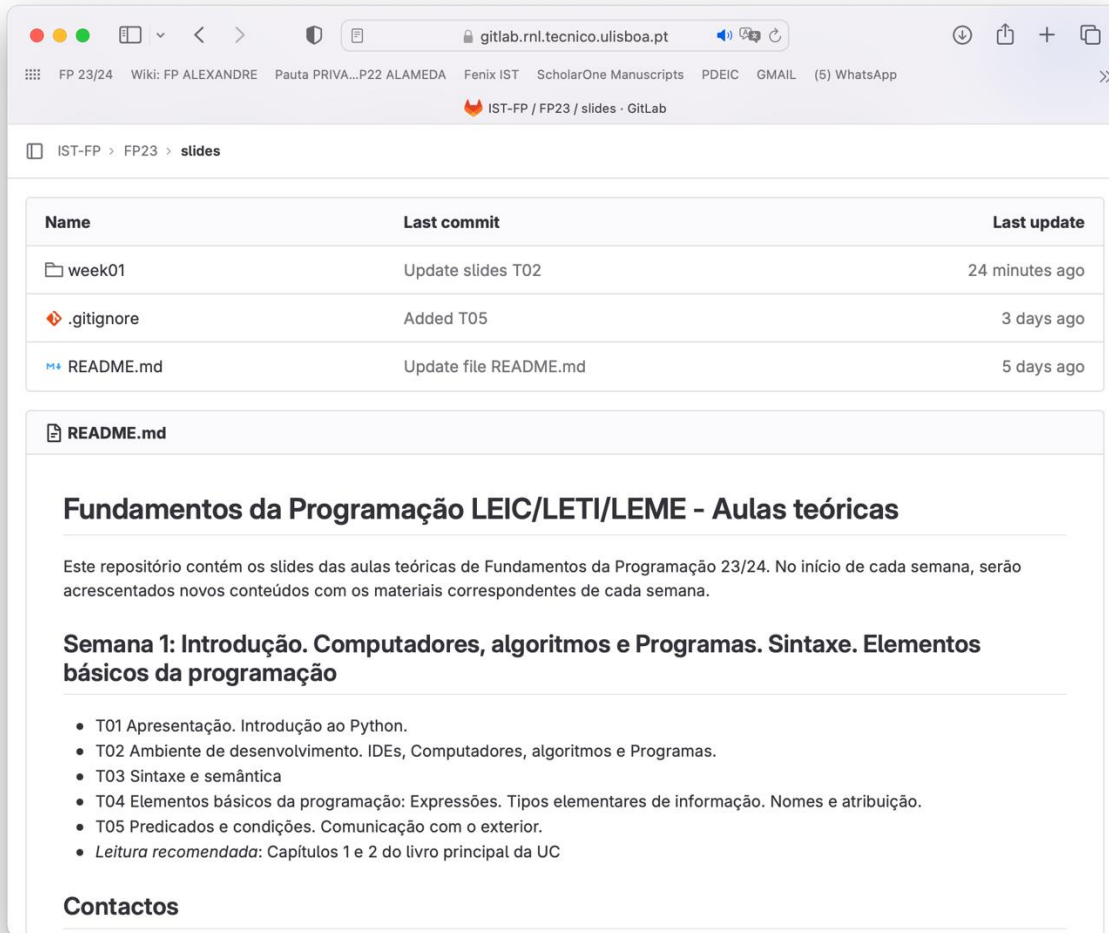
## Google Colab (remoto)

<https://colab.research.google.com>



# GitLab: Software repository (and more)

<https://gitlab.rnl.tecnico.ulisboa.pt>



The screenshot shows a web browser displaying a GitLab repository page. The browser's address bar shows the URL `gitlab.rnl.tecnico.ulisboa.pt`. The page title is "IST-FP / FP23 / slides - GitLab". The breadcrumb navigation shows "IST-FP > FP23 > slides".

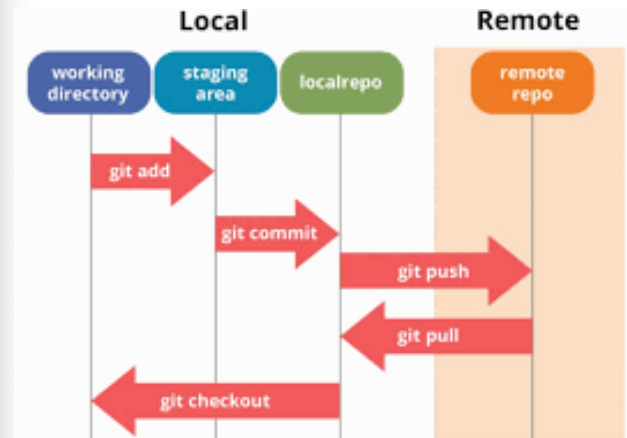
Name	Last commit	Last update
week01	Update slides T02	24 minutes ago
.gitignore	Added T05	3 days ago
README.md	Update file README.md	5 days ago

Below the table, there is a section for the `README.md` file. The title is "Fundamentos da Programação LEIC/LETI/LEME - Aulas teóricas". The text describes the repository's purpose: "Este repositório contém os slides das aulas teóricas de Fundamentos da Programação 23/24. No início de cada semana, serão acrescentados novos conteúdos com os materiais correspondentes de cada semana."

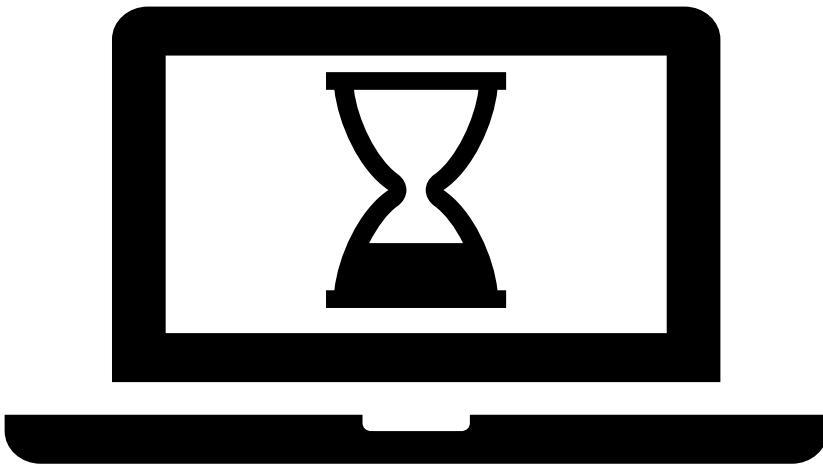
**Semana 1: Introdução. Computadores, algoritmos e Programas. Sintaxe. Elementos básicos da programação**

- T01 Apresentação. Introdução ao Python.
- T02 Ambiente de desenvolvimento. IDEs, Computadores, algoritmos e Programas.
- T03 Sintaxe e semântica
- T04 Elementos básicos da programação: Expressões. Tipos elementares de informação. Nomes e atribuição.
- T05 Predicados e condições. Comunicação com o exterior.
- *Leitura recomendada:* Capítulos 1 e 2 do livro principal da UC

**Contactos**



# Tempo de prática: Tarefas



- Explorar site da cadeira
- Registrar no Slack e deixar uma mensagem
- Instalar Python3
- Abrir um interpretador  
“Olá mundo”
- Instalar um IDE (*VS Code* recomendado)
- Instalar Jupyter notebooks
- Aceder ao GitLab da RNL
- ...

# Tarefas antes da próxima aula

- Explorar site da cadeira
- Registrar no Slack
- Instalar Python3
- Abrir um interpretador
  - “Olá mundo”
- Instalar um IDE (*VS Code* recomendado)
- Instalar Jupyter notebooks
- Aceder ao GitLab da RNL
- Ler o Capítulo 1 do livro da disciplina

