DIT gentle introduction to Python

2nd Edition, February 2022

1. Basic concepts

Alberto Barrón-Cedeño

a.barron@unibo.it

@_albarron_

Overview

- Basics
- Algorithms
- Programming languages
- Baby steps into coding

What is a programming language?

A programming language is just another language

A formal language comprising a set of instructions that produce various kinds of output [given an input]

https://en.wikipedia.org/wiki/Programming_language

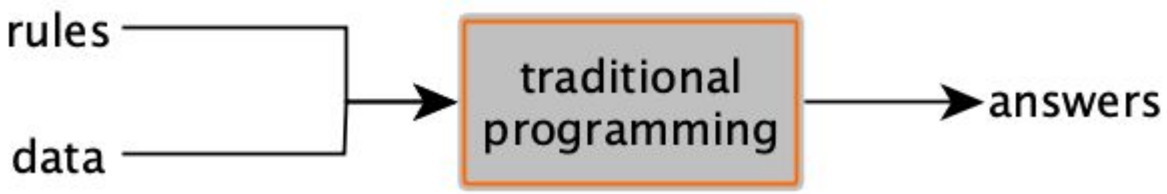


Diagram borrowed from L. Moroney's Introduction to TensorFlow for Artificial Intelligence, Machine Learning, and Deep Learning

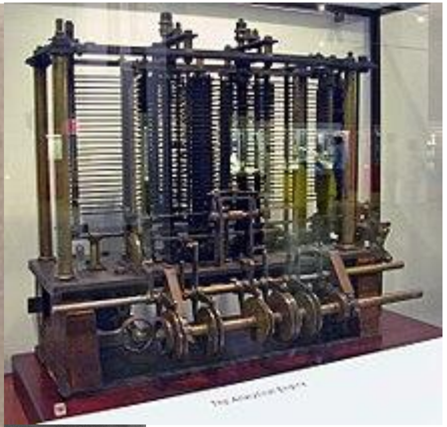
What is a programming language?

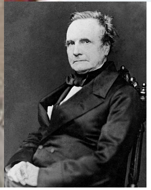
Programming languages are used in computer programming to implement an algorithm*

https://en.wikipedia.org/wiki/Programming_language

* derived from the 9th century Persian Mathematician Muhammad ibn Mūsā **al-Khwārizmī**

The *first* programmer





Ada Lovelace* (Mathematician) published the first algorithm for Charles Babbage's analytical engine

*Lord Byron's daughter

Algorithm

A finite sequence of well-defined computer-implementable instructions, typically to solve a class of problems or to perform a computation

https://en.wikipedia.org/wiki/Algorithm

Example: find if a number is odd or even*

Definitions

- A number is even if it can be divided by 2 without remainder
- A number is odd if it leaves a remainder when divided by 2

Examples

Even numbers: 2, 4, 6, 8, etc.

Odd numbers: 1, 3, 5, 7, etc.

* Adapted from
https://www.c-programming-simple-steps.com/algorithm-examples.html

Example: find if a number is odd or even*

Silly (useless) solution:

1. fill a bag with all even numbers and a second bag with all odd numbers.
2. Given an input number, look for it in both bags and return the label of the one in which you found it.

* Adapted from
https://www.c-programming-simple-steps.com/algorithm-examples.html

Example: find if a number is odd or even

Input/Output

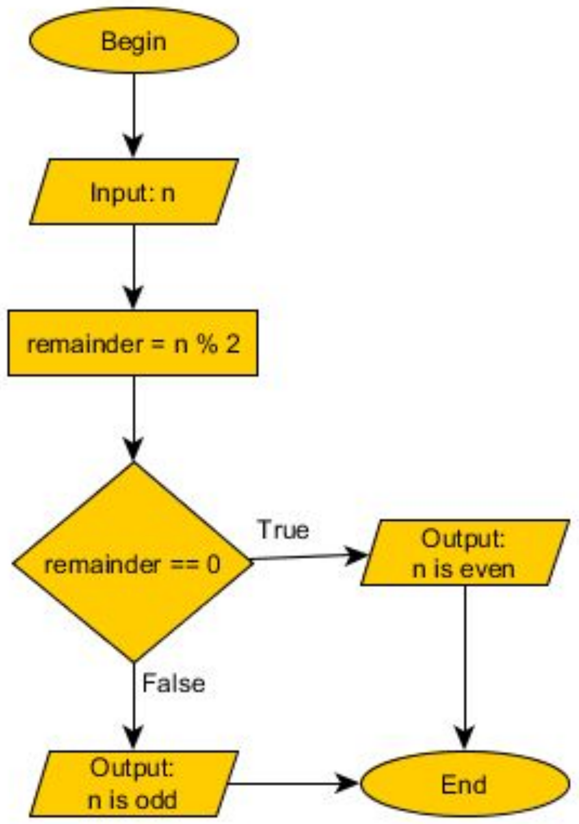
→ an integer (data)

← even or odd (more data)

Process

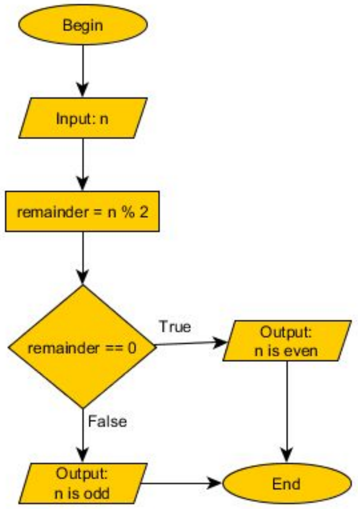
A series of instructions and

routines

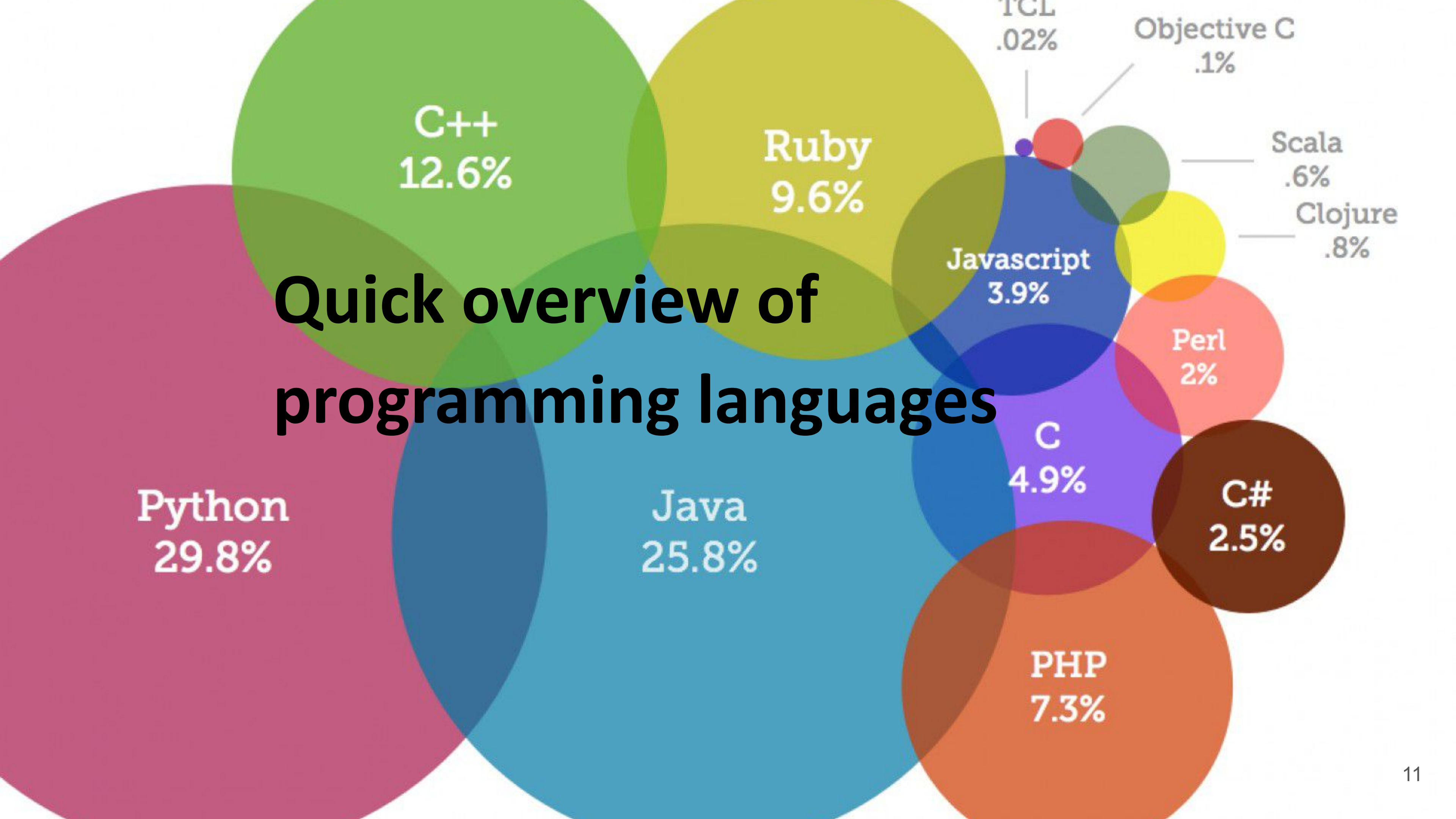


Example: find if a number is odd or even*

From the algorithm into the implementation



```
if n % 2 == 0:
    print('even')
else:
    print('odd')
```



Quick overview of programming languages

History of (some) flagship languages

year	language	highlights
1957	Fortran	compiled, imperative
1959	Lisp*	Object-oriented, popular in AI, recursive functions
1964	Basic*	Procedural, object-oriented (“goto”)
1970	Pascal*	Imperative, procedural, lists, trees
1972	C*	Procedural, recursion, static type system
1983	C++*	Object-oriented, compiled, functional

* language I “speak” (or “spoke” at some point in time)

History of (some) flagship languages

year	language	highlights
1989	Python*	Interpreted, object-oriented, code readability
1995	Java*	compiled, object-oriented
1995	Javascript	Just-in-time-compiled, object-oriented, WWW
1995	PHP*	Scripting, Web-oriented
2001	Visual Basic .NET	Object-oriented, .NET framework
2009	Go	Compiled, C-like (safer)

* language I “speak” (or “spoke” at some point in time)

Python is (among other things)...

General-purpose

Applicable across application domains

High-level

Strong abstraction from the computer (hardware)

Interpreted

No previous compilation into machine-level instructions necessary

(Not-necessarily) object-oriented paradigm

An object contains data (attributes) and procedures (methods)

Some notable features (1/2)

- Elegant syntax (indentation-based) → easy to read
- Simple and ideal for prototyping
- It has a large standard library for diverse tasks (e.g., web servers, text search and processing, file reading/modifying)
- Interactive mode → continuous snippet testing

https://wiki.python.org/moin/BeginnersGuide/Overview

Some notable features (2/2)

- Extendable with modules in compiled languages (e.g., C++)
- Multi-platform (e.g., Mac OS X, GNU Linux, Unix, MS Windows)
- Free: zero-cost to download/use; open-source license
- Large and friendly community

https://wiki.python.org/moin/BeginnersGuide/Overview

Some programming-language features

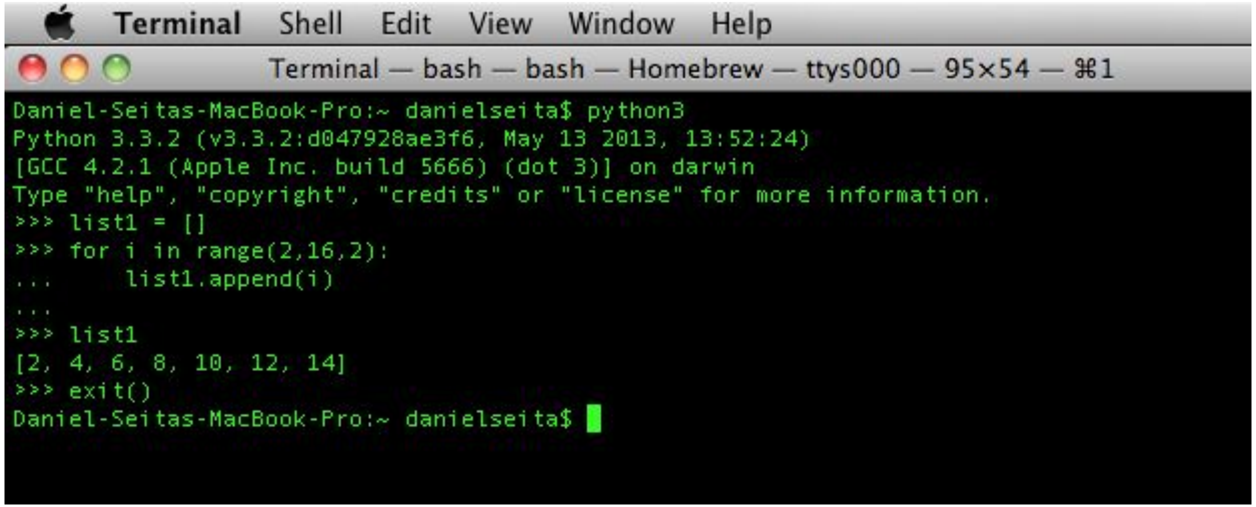
- A variety of basic data types are available:
  - numbers (floating point, complex, integers) ← later today
  - strings (both ASCII and Unicode)
  - Lists
  - Dictionaries

- It supports object-oriented programming ← 3rd session (?)

- Code can be grouped into modules and packages

There are many ways to code/launch a Python program

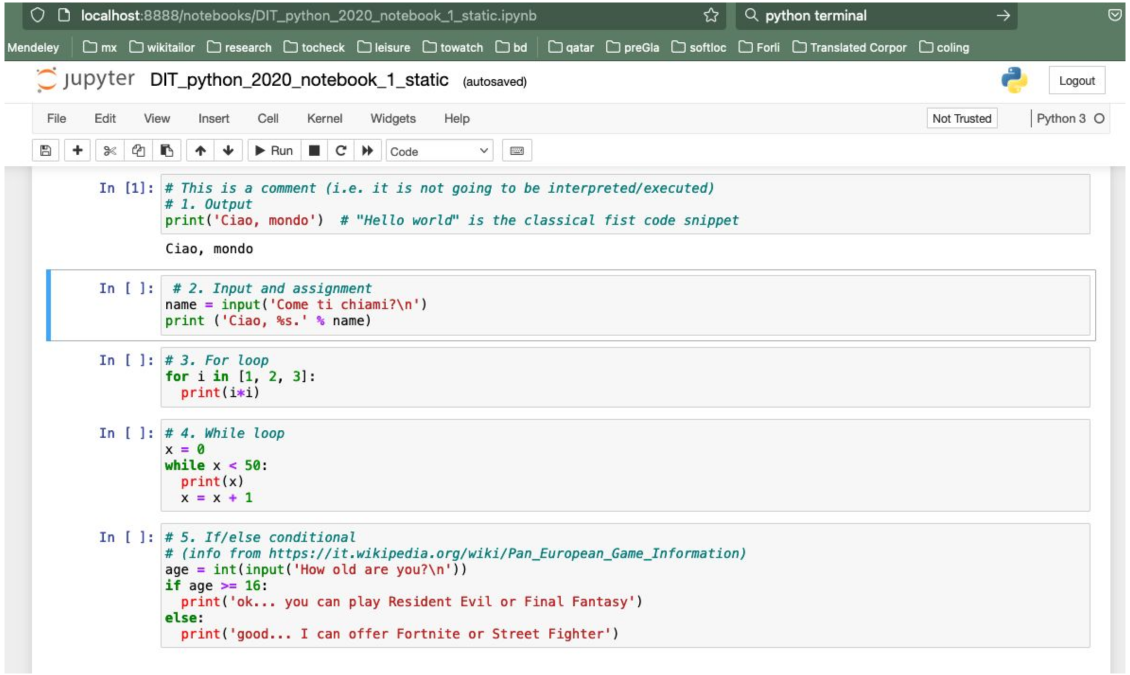
From the UNIX/GNU Linux/Windows terminal


Terminal Shell Edit View Window Help
Terminal — bash — bash — Homebrew — ttys000 — 95×54 — ⌘1

```
Daniel-Seitas-MacBook-Pro:~ danielseita$ python3
Python 3.3.2 (v3.3.2:d047928ae3f6, May 13 2013, 13:52:24)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> list1 = []
>>> for i in range(2,16,2):
...     list1.append(i)
...
>>> list1
[2, 4, 6, 8, 10, 12, 14]
>>> exit()
Daniel-Seitas-MacBook-Pro:~ danielseita$
```

There are many ways to code/launch a Python program

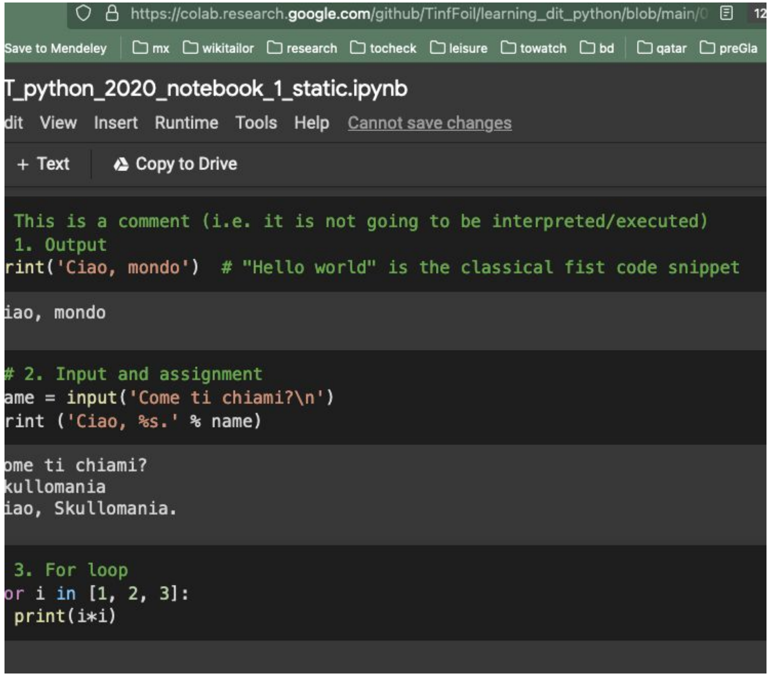
From your web browser (local; offline)



```
In [1]: # This is a comment (i.e. it is not going to be interpreted/executed)
        # 1. Output
        print('Ciao, mondo')  # "Hello world" is the classical fist code snippet

        Ciao, mondo

In [ ]:  # 2. Input and assignment
        name = input('Come ti chiami?\n')
        print ('Ciao, %s.' % name)

In [ ]: # 3. For loop
        for i in [1, 2, 3]:
          print(i*i)

In [ ]: # 4. While loop
        x = 0
        while x < 50:
          print(x)
          x = x + 1

In [ ]: # 5. If/else conditional
        # (info from https://it.wikipedia.org/wiki/Pan_European_Game_Information)
        age = int(input('How old are you?\n'))
        if age >= 16:
          print('ok... you can play Resident Evil or Final Fantasy')
        else:
          print('good... I can offer Fortnite or Street Fighter')
```

There are many ways to code/launch a Python program

From your web browser on Google’s colab (remotely online)


https://colab.research.google.com/github/TinfFoil/learning_dit_python/blob/main/0
Save to Mendeley
mx
wikitailor
research
tocheck
leisure
towatch
bd
qatar
preGla
T_python_2020_notebook_1_static.ipynb
dit View Insert Runtime Tools Help Cannot save changes
+ Text
Copy to Drive

```
 This is a comment (i.e. it is not going to be interpreted/executed)
 1. Output
rint('Ciao, mondo')  # "Hello world" is the classical fist code snippet
```

```
iao, mondo
```

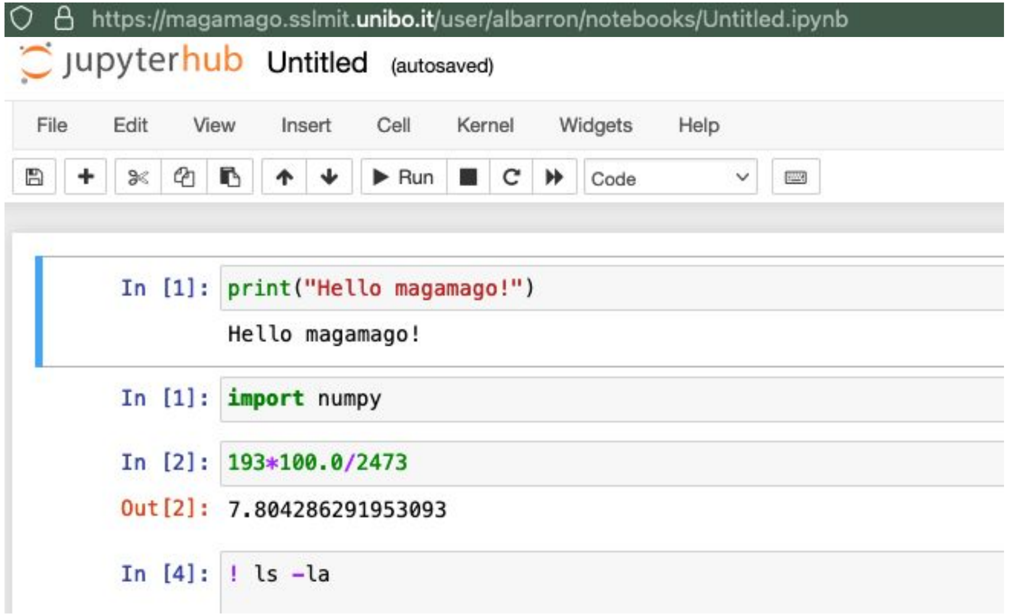
```
# 2. Input and assignment
ame = input('Come ti chiami?\n')
rint ('Ciao, %s.' % name)
```

```
ome ti chiami?
kullomania
iao, Skullomania.
```

```
 3. For loop
or i in [1, 2, 3]:
 print(i*i)
```

There are many ways to code/launch a Python program

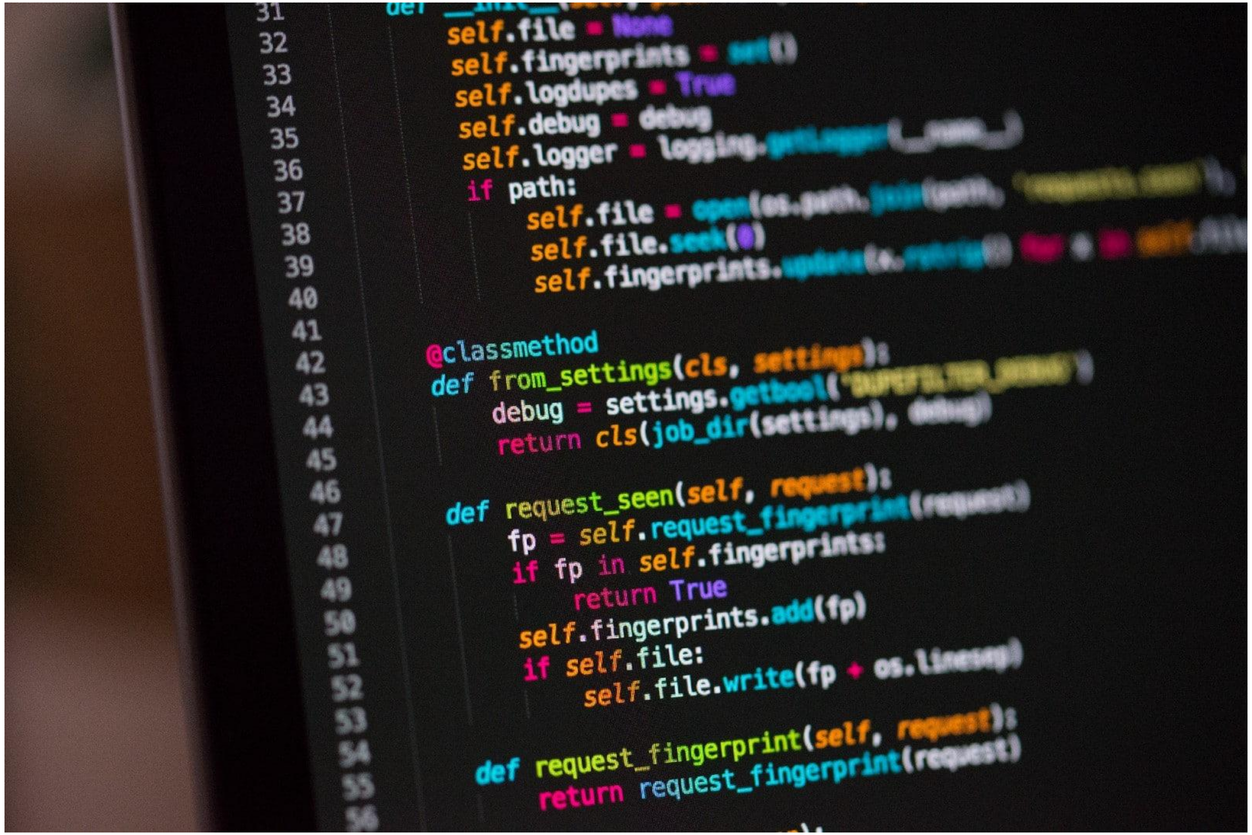
From your web browser on DIT’s magamago (remotely online)*


https://magamago.sslmit.unibo.it/user/albarron/notebooks/Untitled.ipynb
jupyterhub Untitled (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Run Code

```
In [1]: print("Hello magamago!")
        Hello magamago!

In [1]: import numpy

In [2]: 193*100.0/2473
Out[2]: 7.804286291953093

In [4]: ! ls -la
```

* Open to advanced students only

Enough! Let us look at some code!

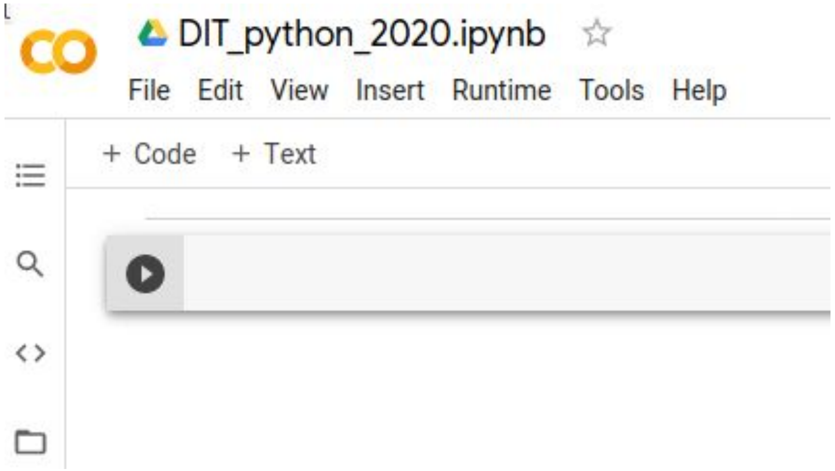


```
    def __init__(self, pa
        self.file = None
        self.fingerprints = set()
        self.logdupes = True
        self.debug = debug
        self.logger = logging.getLogger(__name__)
        if path:
            self.file = open(os.path.join(path, 'requests.seen'),
            self.file.seek(0)
            self.fingerprints.update(x.rstrip() for x in self.fil

    @classmethod
    def from_settings(cls, settings):
        debug = settings.getbool('DUPEFILTER_DEBUG')
        return cls(job_dir(settings), debug)

    def request_seen(self, request):
        fp = self.request_fingerprint(request)
        if fp in self.fingerprints:
            return True
        self.fingerprints.add(fp)
        if self.file:
            self.file.write(fp + os.linesep)

    def request_fingerprint(self, request):
        return request_fingerprint(request)
```

Google’s colab

“a free Jupyter notebook environment that runs in the cloud and stores its notebooks on Google Drive”

https://colab.research.google.com

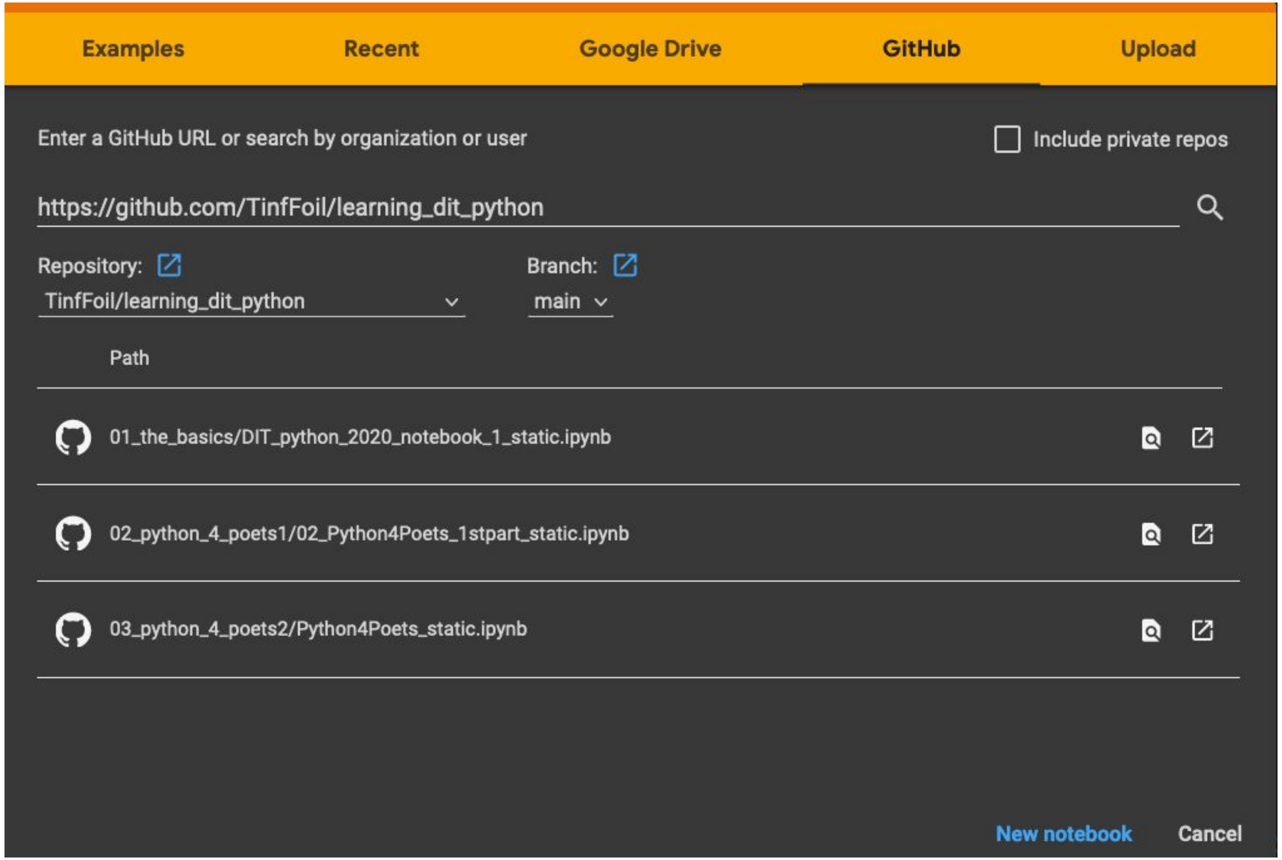


Let’s go to our first jupyter notebook

Google’s colab: baby steps

1. Visit https://colab.research.google.com
2. Click on Github
3. Type https://github.com/TinfFoil/learning_dit_python
4. Press search
5. Select **DIT_python_2022_notebook_1_static.ipynb**

Google's colab: baby steps



Enter a GitHub URL or search by organization or user

☐ Include private repos

https://github.com/TinfFoil/learning_dit_python

Repository: TinfFoil/learning_dit_python

Branch: main

Path

01_the_basics/DIT_python_2020_notebook_1_static.ipynb

02_python_4_poets1/02_Python4Poets_1stpart_static.ipynb

03_python_4_poets2/Python4Poets_static.ipynb

What we know so far: input/output

- print() displays stuff to the screen

- input() captures information from the user

What we know so far: variables

x = 5	- x is a variable - We assign values to a variable with **=**	
	- x = 5 - x = 5.5 - x = ‘ciao’ - x = “ciao” - x = ‘5’	is an integer is a float is a string is also a string **is what?**
x = x * 3	- We can apply operators to variables - We can assign the output to a variable	

What we know so far: flow control - conditionals

```
if (condition):
   execute something
elif (condition):
   execute something
else:
   execute something
```

Only one of these three snippets is executed

How is this different?

```
if (condition):
   execute something
if (condition):
   execute something
else:
   execute something
```

What we know so far: flow control - loops

```
for (iterator):                    while (condition):
   execute something                  execute something
```

The code snippet will be executed during a number of iterations

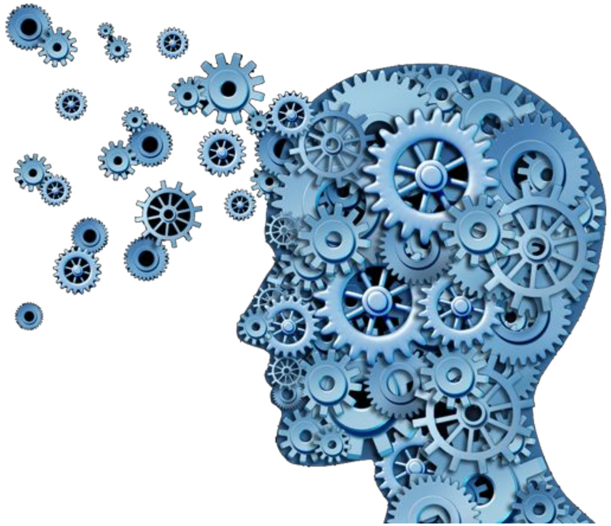
Danger: if you make a mistake, a loop could run forever

What we know so far: basic formatting

```
# my code
x = 0
while x < 50:
 for i in range(x):
   print('x', end="")
 print()
 x += 1
```

- Comments start with #
- A line break is enough to close an instruction (in Java or C, we need ;)
- Colon opens a *special* code snippet
- Indentation is crucial

You know a lot already!



It is your turn to play with the notebook

DIT gentle introduction to Python

2nd Edition, February 2022

Alberto Barrón-Cedeño

a.barron@unibo.it

@_albarron_