

PhD Course on Python Language and Programming



Torino, 2022

Mauro Prencipe: mauro.prencipe@unito.it



Some of your possible motivations to take this course...

Because it's *fun*;

because it will eventually free you by the *slavery* and dependence from programs written by others

because you would like to understand what programs really do under the hood...

Some *less obvious* motivations...

Writing code to solve any problem means:

to be *forced* to understand the problem in depth

to be *forced* to understand the required algorithms very precisely

to strengthen your logic and skills in the analysis of problems

to *learn* to think *algorithmically*

What you will get from this course...

You will get the basic tools to learn to write codes to manipulate and to elaborate data

Essentially to understand and write (*esoteric*) code like that...

```
for iset, ix, iname in zip(set_list, x, set_name):  
    exec(iset + '= data_class(iname, ix)')  
  
l_set=list(eval(iset) for iset in set_list)  
sets=np.array(l_set, dtype='object')
```

but...

...but, if you really want to learn, you must practice a lot...

learning how to use a programming language is much like learning a foreign language:

just studying the *grammar*, the *syntax* and learning by heart a more or less rich *vocabulary* is *definitely not enough*: you must *practice the language* until, at least, you start to *think in that language*!

So, you should start...

... to think algorithmically

Imagine you have to *swap* the values stored in a pair of variables, **a1** and **a2**...

For instance, you start with **a1 = 2** and **a2 = 1** and you want to get **a1 = 1** and **a2 = 2**

Intuitively, you do **a1 = a2** and **a2 = a1**

Then, see what happens...

and what is the correct way to do that!

```
In [38]:
```

Why Python?

Python is *open source*

it is relatively young (*born* in the 90s), so probably it will last for many years to come

it is widely popular

it has a very large community of developers

You easily find help for any problem you might have

if you need of some particular feature or function, it is probable that someone else already developed it...

What is Python?

High level language

Interpreted

Supports *OOP* (Object Oriented Programming)

Main advantage over compiled languages: interactivity

main disadvantage: relatively low efficiency

Where to find (valuable) support...

Official sites

python.org

anaconda.org

numpy.org

scipy.org

pandas.pydata.org

sympy.org

Conferences and Tutorials

conference.scipy.org

YouTube playlist of tutorials

[2018](#)

[2016](#)

[2014](#)

Conferences by Bob Martin (Uncle Bob)

A must!

[Logics](#)

[Clean code](#)

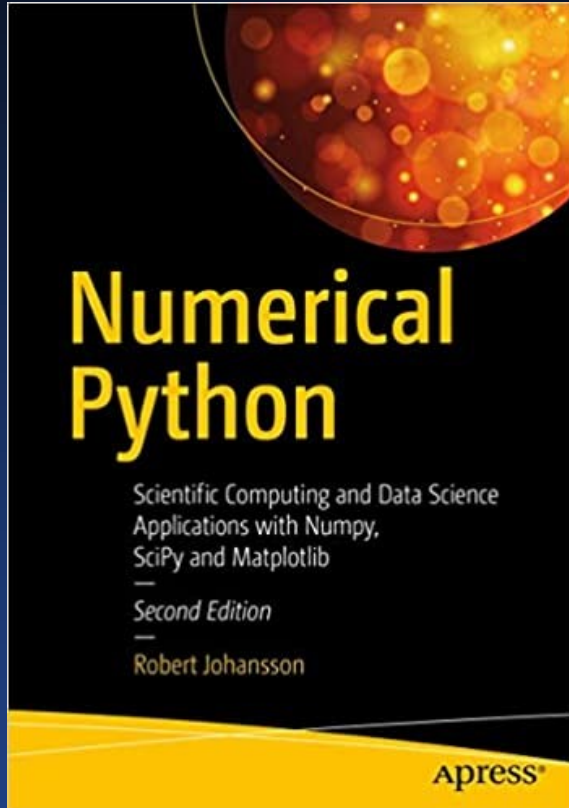
Specific topics

[Pandas \(David Chen 2019\)](#)

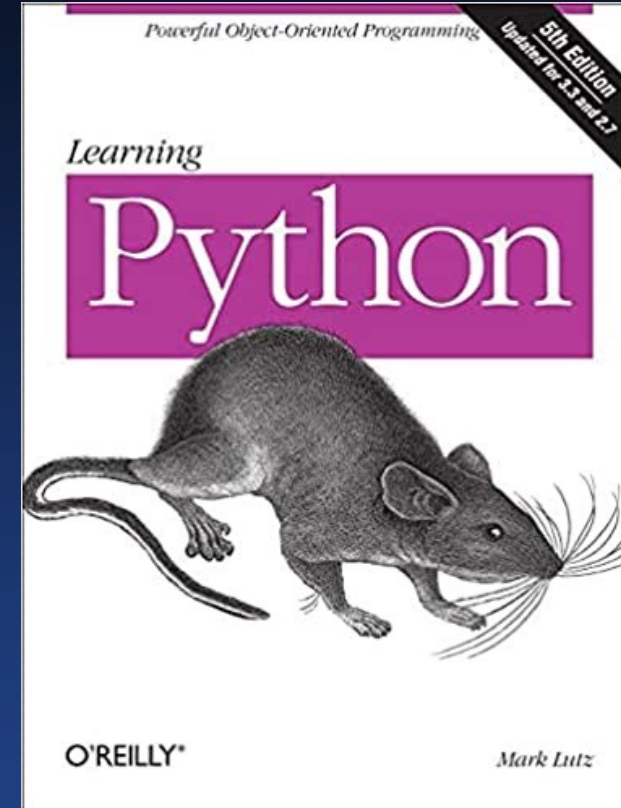
[Numpy \(Alex Chabot-Leclerc; 2019\)](#)



Books



Overview, recipes, practical advices, specialized topic in scientific computing



Very in depth analysis of the language: *logics*, grammar, syntax

Topics specifically covered in this course

Variables and types
conditional and cycles
functions
files
variable scoping

*Basic elements of the
language*

*Python, Numpy, Matplotlib
(Pandas)*

*Object Oriented Programming
Structuring the code*



classes
least squares and general
fits
Matplotlib

*Some specialized aspects
(Scipy)*