



PyQB

Monga

Why Python

Python
fundamentals

Programming in Python¹

Mattia Monga

Dip. di Informatica
Università degli Studi di Milano, Italia

`mattia.monga@unimi.it`

Academic year 2022/23, I semester



PyQB

Monga

Why Python

Python
fundamentals

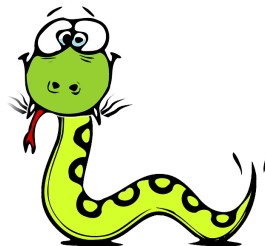
Lecture I: Programming in Python for quantitative biologists

Programming in Python (for quantitative biologists)



The course introduces imperative programming by referring to the Python language.

- 1 Python3 and its object-oriented features;
- 2 Python3 libraries that can be useful in scientific computation and data analysis, in particular NumPy and pandas.



Everything will be available on:

mattiamon.ga/pyqb

(a.k.a. <https://mameli.docenti.di.unimi.it/pyqb>)

PyQB

Monga

Why Python

Python
fundamentals



PyQB

Monga

Why Python

Python
fundamentals

- Tuesdays: 10:30 Aula Farina, Thursday: 8:30 G15, Fridays: 8:30 Lab Delta **but the scheduling is complex: check the website**
- Lectures: 40h, Labs: 16h
- Labs always on Friday
- We will explore different setups: (1) a “scaffolded” one for the first steps, (2) the plain python interpreter, and finally (3) the notebooks popular in scientific practice
- Tutor: dott. Rita Folisi (computer science master student)
- Text: every Python3 reference/book/tutorial is ok, you can access freely to the book linked on the website
- Final test: write (small) python programs without help

Why Python?



PyQB

Monga

Why Python

Python
fundamentals

Programming can be approached in many “languages”, the fundamental skills are general. . . but you cannot learn without referring to a specific language.

- A precise requirement of the teaching committee
- Very popular in the scientific landscape
- Easy to learn, many useful libraries, free software
- Alternatives: Fortran, C, Matlab, Mathematica, R, Julia, . . .
- Python is slower, but it is considered easier to understand and manage

Which Python?



PyQB

Monga

Why Python

Python
fundamentals

We will use Python3 (current version is 3.10): be careful when looking around, Python2 is still very common (but deprecated) and incompatible. Python supports different “paradigms”, we will focus on:

- Imperative programming: programs describe **changes** in *registers* and the *executing environment*;
- Object-oriented: complex (imperative) programs are organized around **objects** in order to hide and isolate complexity.

This is a **programming course**: I will try to propose example that I believe could be useful in your daily practice, but I'm not a biologist.



PyQB

Monga

Why Python

Python
fundamentals

Programming in science can serve two (almost opposite) goals:

- 1 Understanding every detail of a computational process;
- 2 Compose computational process by assembling powerful build blocks of which you understand very little.

Most of the current popularity of programming is related to goal 2... with many *sorcerer's apprentices*. But this course will focus mainly on goal 1. In the last part of the course we will bend towards 2, hopefully with a solid background.

Programming can be both hard and addictive: [Teach Yourself Programming in Ten Years](#)

Fundamental concepts of Python



The programmer describes computational processes in terms of:

objects : all the entities manipulated by the program, each has an **identity** (can be distinguished) and a **value**, that is an element in a specific **type** (a set of values together with the operations that make sense on them)

basic types : integers (**int**), floats, strings (**str**), functions; they can be composed in more complex types

variables : **names** used to refer to objects; the same name can refer to different objects during the same process

special commands : the only way to change the execution environment (i.e., the “virtual machine” provided by the operating system) is to use **system calls**; syscalls change from system to system (e.g., Linux vs. Windows), but Python wraps them and they appear like the functions written by the programmers (e.g., **print**), even if they could not be programmed in Python.

PyQB

Monga

Why Python

Python
fundamentals

Let's try!



PyQB

Monga

Why Python

Python
fundamentals

<https://python.di.unimi.it/>

You can use it without any personal account, but if you want support you must create one, putting me as the “guru”: `mmonga`

This platform will be used for the first lessons, since it requires no setup at all: everything happens in the browser (and the server).

(Thanks to the University of Waterloo, Canada for providing the CS Circles)