

Python Data Analysis

2023 – 2024

General informations

Louie Corpe

louie.corpe@cern.ch



Romain Madar

romain.madar@cern.ch



Material for the lecture

1. On **moodle** platform (ENT, UCA account access required):

<https://ent.uca.fr/moodle/course/view.php?id=26834>

2. On **github** platform (free access):

<https://github.com/rmadar/lecture-python>

Material for the lecture

1. On **moodle** platform (ENT, UCA account access required):

<https://ent.uca.fr/moodle/course/view.php?id=26834>

2. On **github** platform (free access):

<https://github.com/rmadar/lecture-python>

Content of the lecture -- full PDF

There are a lot of information in this lecture. In order to help you to focus on important aspect, each chapter start with a list of expected skills that you should take away, ranked with three levels: *basic*, *medium*, *expert*.

0. Practical Introduction to Jupyter Notebooks. This section is not present in the final PDF but is presented during the lecture.

1. Practical Introduction to Python. This first section is dedicated to basic object type and operation in python. Functions will also be described but object oriented programming will not be covered.

2. Introduction to numpy. Differences between usual python objects and numpy objects will be introduced.

3. Three tools to know. This section gives a glimpse of `matplotlib`, `pandas` and `scipy` packages allowing powerful data analysis.

4. Multidimensional data manipulation. Non-trivial operation for multidimensional data using the full power of numpy. Most of these operation can be performed with existing tools but it is intrusive to do it once with native numpy.

5. Introduction to image processing. Very first steps of image processing (definition, plotting, operation) including basic filters application (noising, sharpen, border detection).

Other practical examples. Depending on the remaining time (and the people taste), we can go through different topics among the following ones. Some of them can be also used as a project performed by students.

- Fourier analysis
- Principal component analysis (PCA)
- Random Forest regression
- Gaussian processes

List of previous exams with corrections

- 2019 : Analysis of an electric pulse --> [exam / correction](#)
- 2020 : Ising model (more details on this topic [here](#)) --> [exam / correction](#)
- 2021 : Coupled harmonic oscillators (more details on this topic [here](#)) --> [exam / correction](#)
- 2022 : Random walk --> [exam / correction](#)

How to get prepared

1. Get familiar with python. I would recommend two links: [w3school tutorial](#) (both basic and complete) and <https://www.learnpython.org> (code can be ran directly within your web browser).

2. Install python with anaconda. In order to run python on your own machine, you should install it. I would recommend [anaconda](#) for this, which also includes jupyter-notebook.

3. Install git. This is a versioning software which can be installed following these [instructions](#). This whole repository can be *cloned* using `git clone https://github.com/rmadar/lecture-python` command.

4. Get familiar with notebooks. This represents a nice environment combining codes, notes and plots. This is very powerful to learn something and play with it. You can checkout [this video](#) or [this post](#).

Material for the lecture

1. On **moodle** platform (ENT, UCA account access required):

<https://ent.uca.fr/moodle/course/view.php?id=26834>

2. On **github** platform (free access):

<https://github.com/rmadar/lecture-python>

rmadar Updating the preamble			a4daf84	3 days ago	🕒 157 commits
📁 assignment	Getting up to date	2 weeks ago			
📁 data	Getting up to date	2 weeks ago			
📁 documentation	Updating the preamble	3 days ago			
📁 exam	Getting up to date	2 weeks ago			
📁 exercises	Clearing exercise directory	last year			
📁 lectures	Set the repo up-to-date before 2022 modifications	last year			
📄 .gitignore	added assignment	4 years ago			
📄 README.md	Update README	3 days ago			

Content of the lecture -- full PDF

There are a lot of information in this lecture. In order to help you to focus on important aspect, each chapter start with a list of expected skills that you should take away, ranked with three levels: *basic*, *medium*, *expert*.

0. Practical Introduction to Jupyter Notebooks. This section is not present in the final PDF but is presented during the lecture.

1. Practical Introduction to Python. This first section is dedicated to basic object type and operation in python. Fonctions will also be described but object oriented programming will not be covered.

2. Introduction to numpy. Differences between usual python objects and numpy objects will be introduced.

3. Three tools to know. This section gives a glimpse of `matplotlib`, `pandas` and `scipy` packages allowing powerful data analysis.

4. Multidimensional data manipulation. Non-trivial operation for multidimensional data using the full power of numpy. Most of these operation can be performed with existing tools but it is intrusive to do it once with native numpy.

5. Introduction to image processing. Very first steps of image processing (definition, plotting, operation) including basic filters application (noising, sharpen, border detection).

Other practical examples. Depending on the remaining time (and the people taste), we can go through different topics among the following ones. Some of them can be also used as a project performed by students.

- Fourier analysis
- Principal component analysis (PCA)
- Random Forest regression
- Gaussian processes

List of previous exams with corrections

- 2019 : Analysis of an electric pulse --> [exam / correction](#)
- 2020 : Ising model (more details on this topic [here](#)) --> [exam / correction](#)
- 2021 : Coupled harmonic oscillators (more details on this topic [here](#)) --> [exam / correction](#)
- 2022 : Random walk --> [exam / correction](#)

How to get prepared

1. Get familiar with python. I would recommend two links: [w3school tutorial](#) (both basic and complete) and <https://www.learnpython.org> (code can be ran directly within your web browser).

2. Install python with anaconda. In order to run python on your own machine, you should install it. I would recommend [anaconda](#) for this, which also includes jupyter-notebook.

3. Install git. This is a versioning software which can be installed following these [instructions](#). This whole repository can be *cloned* using `git clone https://github.com/rmadar/lecture-python` command.

4. Get familiar with notebooks. This represents a nice environment combining codes, notes and plots. This is very powerful to learn something and play with it. You can checkout [this video](#) or [this post](#).

Lecture structure & technicals

A typical day:

- presentation of a new chapter, with some little exercises for you to practice
- practical sessions with larger exercises

You need to have a running notebook with a proper python environment

(2 options : UCA computer, your laptop)

How to get prepared

1. Get familiar with python. I would recommend two links: [w3school tutorial](#) (both basic and complete) and <https://www.learnpython.org> (code can be ran directly within your web browser).

2. Install python with anaconda. In order to run python on your own machine, you should install it. I would recommend [anaconda](#) for this, which also includes jupyter-notebook.

3. Install git. This is a versioning software which can be installed following these [instructions](#). This whole repository can be *cloned* using `git clone https://github.com/rmadar/lecture-python` command.

4. Get familiar with notebooks. This represents a nice environnement combining codes, notes and plots. This is very powerful to learn something and play with it. You can checkout [this video](#) or [this post](#).

Skills and Evaluations

[lecture pdf]

Chapter 1

Practical Introduction to Python

Skills to take away

- *basic*: int/float/str, list/dictionary, indexing/slicing, loops, functions, reading/writing files
- *medium*: docstring, comprehension, zip()/enumerate(), sorting dictionary
- *expert*: packing/unpacking, parsing file with correct casting, basic plotting

1.1 General information

Python can be installed using [anaconda](#). [Jupyter-notebook](#) (also coming with anaconda) is probably the easiest way to follow this lecture and make your own notes. The goal of this first chapter is to give a *very quick introduction basis*, but practice is mandatory to get comfortable with python objects and syntax. Practicing is possible with a web browser only using <https://www.learnpython.org>. A more complete tutorial (but not interactive) can be found in [w3school python tutorials](#). I recommend to follow the last tutorial up to *Arrays*.

In python, there is one instruction per line. Variable assignment is done with `=`, indentation is used to group instructions together under a loop or a condition block: there is no bracket (like in C++) or equivalent. Comments (*i.e.* uninterpreted text) start with `#`. Importation of external modules or function can be done with three different ways: `import module`, `import module as m` or `from module import this_function`.

[lecture notebook]

jupyter
nbviewer

lecture-python / lectures

JUPYTER FAQ

Practical Introduction to Python

Skills to take away

- *basic*: int/float/str, list/dictionary, indexing/slicing, loops, functions, reading/writing files
- *medium*: docstring, comprehension, zip()/enumerate(), sorting dictionary
- *expert*: packing/unpacking, parsing file with correct casting, basic plotting

General information

Python can be installed using [anaconda](#). [Jupyter-notebook](#) (also coming with anaconda) is probably the easiest way to follow this lecture and make your own notes. The goal of this first chapter is to give a *very quick introduction basis*, but practice is mandatory to get comfortable with python objects and syntax. Practicing is possible with a web browser only using <https://www.learnpython.org>. A more complete tutorial (but not interactive) can be found in [w3school python tutorials](#). I recommend to follow the last tutorial up to *Arrays*.

In python, there is one instruction per line. Variable assignment is done with `=`, indentation is used to group instructions together under a loop or a condition block: there is no bracket (like in C++) or equivalent. Comments (*i.e.* uninterpreted text) start with `#`. Importation of external modules or function can be done with three different ways: `import module`, `import module as m` or `from module import this_function`.

In the following example, the result of the command will be printed so that people can check that the computer is doing what is expected. The instruction `print(x)` will print the content of `x`. If several variables are printed, it is convenient to use `print('x={}' and y={}'.format(x, y))` syntax that will print `x` and `y` in bracket fields with one command - even if they have different types.

Note For python version greater than 3.6, we can also use *f-strings* which simplify a bit the print commands. This works as follow:

```
print(f'x={x} and y={y}')
```

where the `x` and `y` in bracket are actual python variables.

Object types

Numbers. There are three type of numbers: int, float and complex. The usual operations (`+`, `-`, `*`, `/`) are available. In addition, there is also `a**b` (which means a^b), `a // b` and `a % b` (which are the result of integer divisions - see example below).

```
In [1]: # Basic numbers and operations
a = 2
b = 3.14
print(a+b)
print(a**b)

5.1406000000000001
8.815240927012887
```

Evaluation : 2h exam on a computer in classroom (beginning of november for the masters, end of september for DUs).

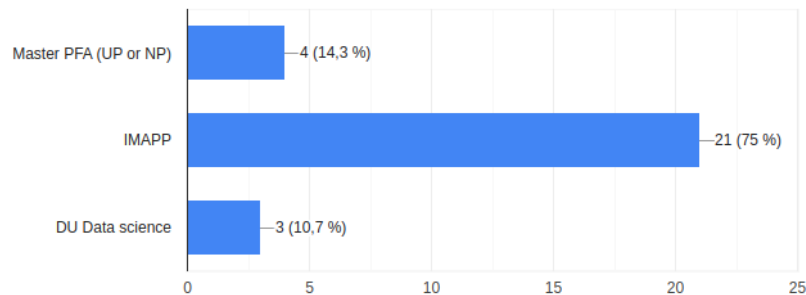
Allowed material : official python documentations of all tools (**lecture material not allowed**).

The group

In which context will you follow the "python for data analysis" lectures ?

 Copier

28 réponses

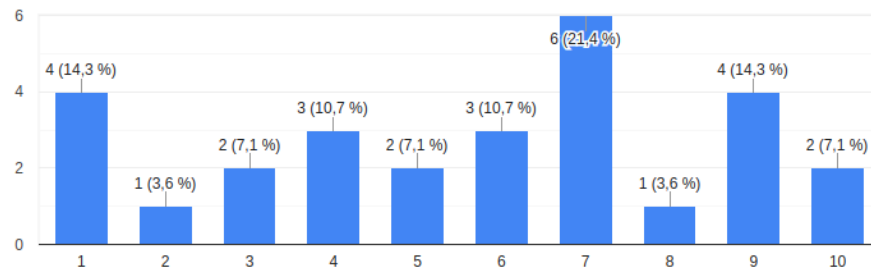


Self evaluation of python skills

How would you evaluate your python skills ? On the scale below, '1' means you have never practiced python and '10' means that you can do more or less anything, including taking new tools in hands (with some documentation).

 Copier

28 réponses



	Lundi 11/09/2023	Mardi 12/09/2023	Mercredi 13/09/2023	Jeudi 14/09/2023	Vendredi 15/09/2023
07h30					
08h00					
08h30					
09h00					
09h30	Data analysis Python (Romain Madar) 1/9 09h30 - 12h30 Intervenant à préciser SCI_008 SCI_007 M2 UP M2 NP DU Data Scientist	Data analysis Python (Romain Madar) 2/9 09h30 - 12h30 Intervenant à préciser SCI_008 SCI_007 M2 UP M2 NP DU Data Scientist	Data analysis Python (Romain Madar) 3/9 09h30 - 12h30 Intervenant à préciser SCI_008 SCI_007 M2 UP M2 NP DU Data Scientist	Data analysis Python (Romain Madar) 4/9 09h30 - 12h30 Intervenant à préciser SCI_008 SCI_007 M2 UP M2 NP DU Data Scientist	Data analysis Python (Romain Madar) 5/9 10h30 - 12h30 Intervenant à préciser SCI_008 SCI_007 M2 UP M2 NP DU Data Scientist
10h00					
10h30					
11h00					
11h30					
12h00					
12h30			<div>Free</div> <div>Ends at 12h00</div>		
13h00					
13h30					
14h00					
14h30					
15h00	Data analysis Python (Romain Madar) 2/9 14h00 - 17h00 Intervenant à préciser SCI_008 SCI_007 M2 UP M2 NP DU Data Scientist	Data analysis Python (Romain Madar) 4/9 14h00 - 17h00 Intervenant à préciser SCI_008 SCI_007 M2 UP M2 NP DU Data Scientist		Data analysis Python (Romain Madar) 7/9 14h00 - 17h00 Intervenant à préciser SCI_008 SCI_007 M2 UP M2 NP DU Data Scientist	Data analysis Python (Romain Madar) 9/9 14h00 - 17h00 Intervenant à préciser SCI_008 SCI_007 M2 UP M2 NP DU Data Scientist
15h30					
16h00					
16h30					
17h00					