Python Tutorial 2023

2-5 May

Welcome!

Hello and welcome, everyone! We're excited that you chose to embark in this Python journey

Over **four** days, we'll see a lot of Python. **Two** fundamental topics each day

Day 1

1. Basic syntax and data types
2. Control flow, loops, and exceptions

Day 2

1. Functions
2. Working with files

Day 3

1. Functional programming
2. Object-oriented programming

Day 4

1. Modules and packages
2. Questions, exercises, more questions…

Although this is an introductory course, we will be going quite in-depth to cover as much ground as possible. Our aim is to equip you with a strong foundation in Python programming that you can build on in the future. So, let's get started!

Timetable

TL;DR: Every morning, 9:00-13:00, on Zoom

. . .

More in detail:

- 8:45-9:00 — Welcome, warm-up questions
- 9:00-10:50 — Topic 1 (theory, hands-on, Q&A)
- *10:50-11:10 — Break*
- 11:10-13:00 — Topic 2
- – Practice – Offline discussions on Slack

How to interact

Each block is 20% theory and 80% hands-on. It's an online workshop, so interaction between us is **essential**

A few tips to interact:

On Zoom

- Raise your hand then ask
- Write in the chat
- We will open a break-out room to address longer questions. **Anyone** interested can jump in

On Slack

- Ask your question in the **corresponding channel**. Use `#help` for generic questions
- Reply **in a thread** if you want to add a comment to another question
- Use `@Sasha`, `@Simone`, `@Edoardo` if you want to mention the tutorial instructors directly

Although the chat will be distributed together with the recording, Slack is much better suited to hold continued Q&A and other discussions. Please, use it.

How to use notebooks

1. The link to open Binder is on Slack in `#announcements` and in Zoom's chat
2. Open the `index.ipynb` file and click on the current topic link
3. Run the code, modify, experiment

Exercises

For each topic we prepared a bunch of exercises to practice the concepts

In the notebooks, you will see cells like

```
%%ipytest

def solution_to_exercise(input_arg):
    """
    Write your solution here
    """
    pass
```

Unless otherwise noted, you can delete everything *below* the line starting with `def` and write your solution

The "chili" symbol estimates the effort required. In general, more chilis mean more work and thinking to solve the exercise

Double-check before deleting everything in a solution's cell. There might be useful hints to solve the exercise!

Let's look at one important detail about exercises: the variables you are expected to work with in **your solution**.

```
%%ipytest

def solution_longest_sequence(numbers: list[int]) -> int:
    """
    You should treat `numbers` as a "list of integers".
    It will be already defined when you run this function
    """
    pass
```

> Important
>
> The input arguments to the `solution_*` function **is already available**. We also tell you what **type** you should expect: a string, an integer, a file path, or anything else.

Remember: **ask** if you need help, either on Zoom or on Slack. These four days should be a practical workshop where we can answer to most of your questions you have as a beginner Python user. Let's make the most out of this time!

Hands-on sessions

There will be plenty of time to work on the exercises.

Here's how these time slots will be organized **on Zoom**:

Main room

- Anyone who is not yet comfortable with Python
- Walkthrough of a few challenging exercises
- **Interactive**: you should participate with suggestions

Break-out rooms

- Anyone who wants to work on their own
- Discussion with someone who wants to share their code
- Specific Q&A

Let's get started!