Python For Economist

University of Bologna

April 17th 2024

What is Python?

A general-purpose programming language

You can do next to anything with python:

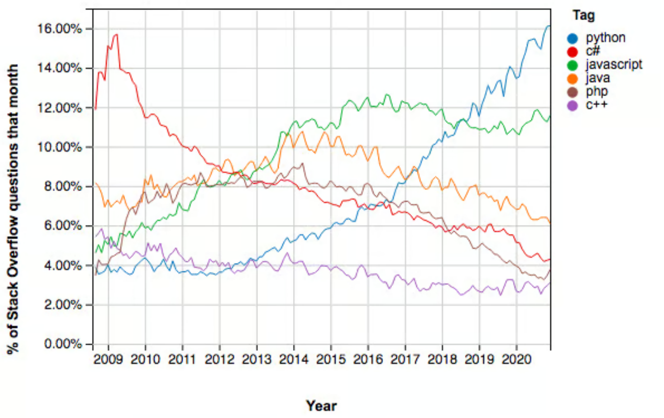
- Data Science (Natural Language Processing, Machine Learning).
  - Data Visualization.
- Web Scraping (BeautifulSoup, Scrapy).
- Web development (Instagram with Python-Django).
- Video game development (The Sims, Battlefield).
- Software development (Netflix).

There's little you cannot do with Python.

Why Learn Python?

One the most popular programming language

Figure: Growth of Python Queries in Stack Overflow



Why should Economist learn Python?

Horizontally differentiated from other languages.

Python differentiates itself from all the others:

- R/Matlab/Julia and Stata

The most similar (for us) is R, What (I think) Python does better:

- Webscraping.
- Natural Language Processing.
- Machine Learning.

You also access **correlated knowledge** with other programming language.

What parts of python do we learn?

The goals of this class

We'll learn the basics:

- Getting a general command of Python.

Then we'll progress to some must-haves for economists:

- Data Management and Handling.
- Data Analysis.
- Webscraping.
- Natural Language Processing.
- Geographical Data treatment.

If we have time, we'll get familiar with machine learning.

How do we learn Python?

Project-Based learning

I think you learn better by-doing.

- We'll try as much as possible to have hands-on classes.
- We learn a concept, and apply it directly.

Bringing your laptop to class is heavily encouraged.

How do we learn Python?

Evaluation

I want you to work on projects throughout the course:

- At the end of each week I'll give assignments.
    - You have to hand in at least 2-3 assignments (still to be decided).

The evaluation will be a project on Python that leverages on the course.

- I'll prepare a list of base projects.
- You are **strongly encouraged** to make up your own.

How do we learn Python?

Evaluation on Projects

Typical projects will be:

- Use the DB1B dataset on Airlines and study the relationship between market structure and price.
- Webscrape articles from repubblica and study how the writing differs with NLP.
- Use the DBnomics aggregator to get data and estimate a VAR to predict inflation in a country of your choice.
- Study Argentesi's case law with NLP.
- Get some environmental data and look at which provinces in a country are most affected/sensible to global warming.
- Take parliamentary speeches and study systematic differences between men and women.

Python Installation

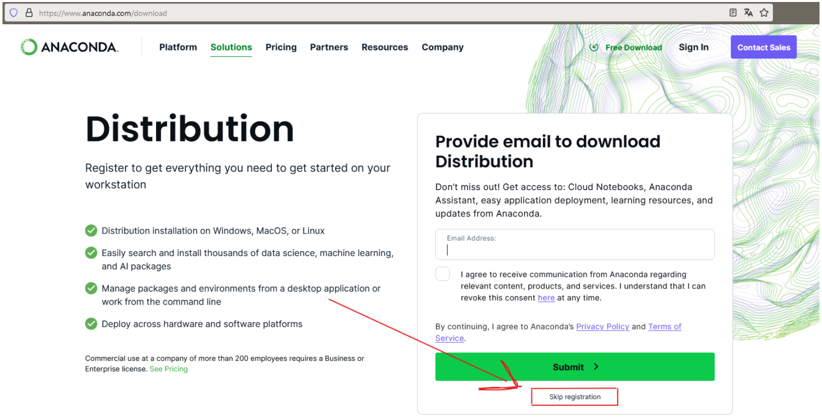
The Hardest past

Python has a convoluted architecture.

- We need first to install python.
- Then we need to install an Integrated Development Environment.
  - Spyder.
  - Jupyter Notebook.
  - PyCharm.
- Then we need to install packages.

Instead of doing it ourselves, we'll use **anaconda**.

Anaconda


https://www.anaconda.com/download

ANACONDA. Platform Solutions Pricing Partners Resources Company Free Download Sign In Contact Sales

Distribution

Register to get everything you need to get started on your workstation

- Distribution installation on Windows, MacOS, or Linux
- Easily search and install thousands of data science, machine learning, and AI packages
- Manage packages and environments from a desktop application or work from the command line
- Deploy across hardware and software platforms

Commercial use at a company of more than 200 employees requires a Business or Enterprise license. See Pricing

Provide email to download Distribution

Don't miss out! Get access to: Cloud Notebooks, Anaconda Assistant, easy application deployment, learning resources, and updates from Anaconda.

Email Address:

I agree to receive communication from Anaconda regarding relevant content, products, and services. I understand that I can revoke this consent here at any time.

By continuing, I agree to Anaconda's Privacy Policy and Terms of Service.

Submit >

Skip registration

Figure: https://www.anaconda.com/download

What is Anaconda?

While it downloads

Anaconda is a package manager:

- It is extremely useful if you work on Windows.
- It deals with the installation of packages and IDEs.

In Python, you almost always work on top of the work of others:

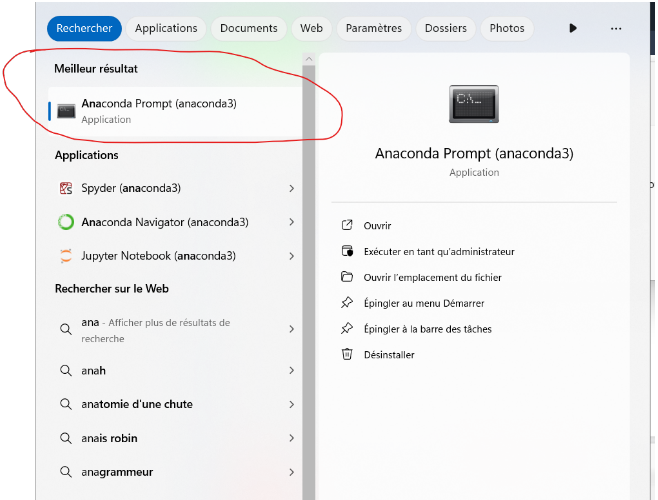
- The work of others is distributed in packages.
- These are library of functions that make your life easier.
- Packages are also built on other packages, and sometimes only on past versions of other packages.

Anaconda helps to manage this environment.

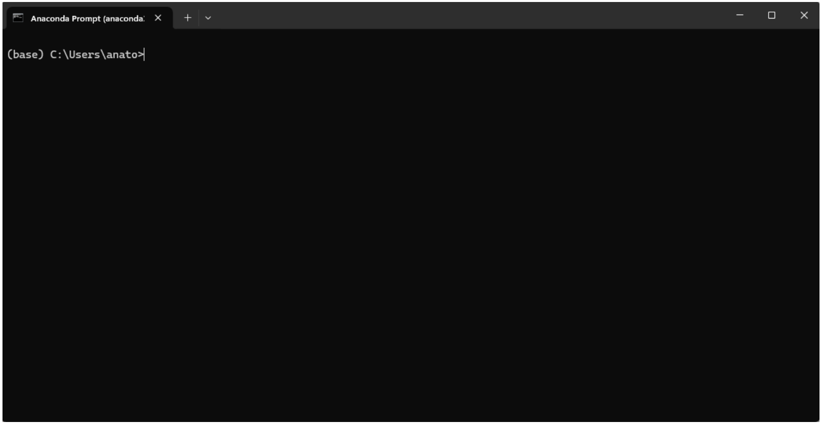
Virtual environment

With Anaconda, we'll create a virtual environment:

- Open the Anaconda Prompt:


Rechercher
Applications
Documents
Web
Paramètres
Dossiers
Photos
Meilleur résultat
Anaconda Prompt (anaconda3)
Application
Applications
Spyder (anaconda3)
Anaconda Navigator (anaconda3)
Jupyter Notebook (anaconda3)
Rechercher sur le Web
ana - Afficher plus de résultats de recherche
anah
anatomie d'une chute
anais robin
anagrammeur
Anaconda Prompt (anaconda3)
Application
Ouvrir
Exécuter en tant qu'administrateur
Ouvrir l'emplacement du fichier
Épingler au menu Démarrer
Épingler à la barre des tâches
Désinstaller

Anaconda Prompt



```
(base) C:\Users\anato>
```

Environment Creation

Type the command:

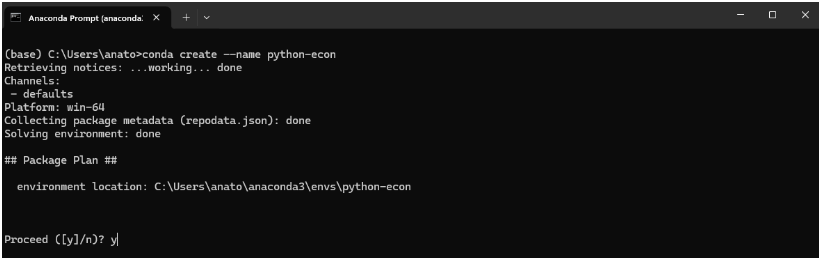
- **conda create −−name python-econ**

We are creating an environment. A box that will contain our work:

- Inside the box, we'll download all packages and libraries.
- We've named that box *python-econ*.

Confirmation

It will ask you confirmation, type **y** then press **Enter** key.



```
(base) C:\Users\anato>conda create --name python-econ
Retrieving notices: ...working... done
Channels:
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\anato\anaconda3\envs\python-econ



Proceed ([y]/n)? y
```

Spyder

Why Spyder?

Spyder is extremely similar to other softwares' IDEs:

- Almost the same IDE as RStudio.
- Very similar to Matlab.

What I think is the best IDE for economists:

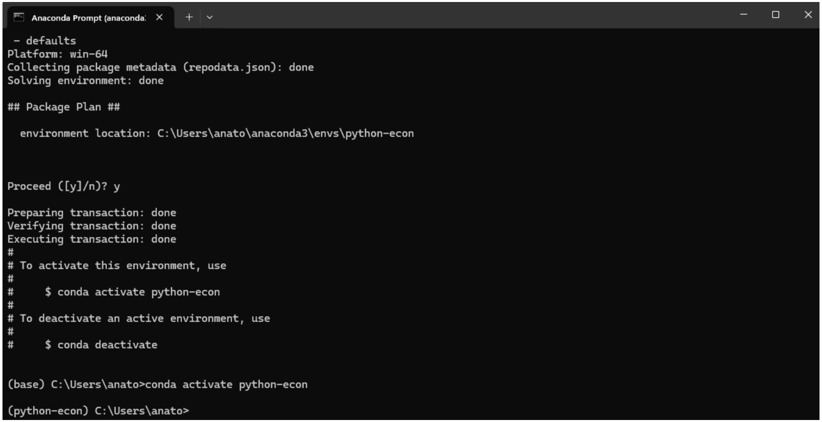
- Allows you see your data.
- Works well when you do not have too much data.

Not the most popular IDE otherwise, but great for academics and policy.

Installation

In the anaconda prompt, move inside your newly created environment:

- **conda activate python-econ**



```
 - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\anato\anaconda3\envs\python-econ



Proceed ([y]/n)? y

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate python-econ
#
# To deactivate an active environment, use
#
#     $ conda deactivate


(base) C:\Users\anato>conda activate python-econ

(python-econ) C:\Users\anato>
```

Install Spyder

To install spyder, now that you're inside your environment:

- **conda install spyder**

It will ask for confirmation again, type **y** then press **Enter**.

- We're ready to go! Either type **spyder**,
- or open Spyder from your computer itself:
    - **spyder (python-econ)**

Alternatives

If you cannot install spyder on your machine:

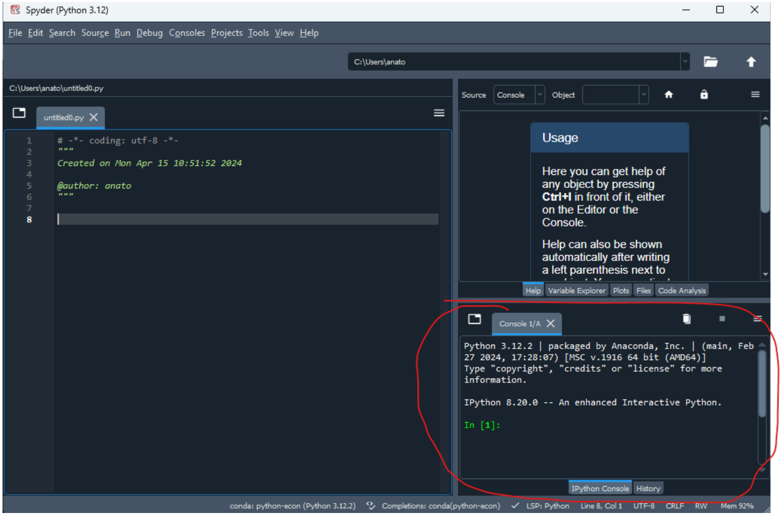
- Do not have admin rights.
- You absolutely hate it.

You can use **google Colab**:

- You use jupyter notebook on a google remote server.
  - It's free.
  - When it comes to python, it's the same.
- I would prefer you program on your local machine, for when we manipulate data.
- It's better if we all use the same IDE to follow the class better.

Spyder Interface (I)

The Console


Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\anato
C:\Users\anato\untitled0.py
untitled0.py
Source Console Object
Usage
Help Variable Explorer Plots Files Code Analysis
Console 1/A
IPython Console History
conda: python-econ (Python 3.12.2) Completions: conda(python-econ) LSP: Python Line 8, Col 1 UTF-8 CRLF RW Mem 92%

```
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 15 10:51:52 2024

@author: anato
"""

```

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to

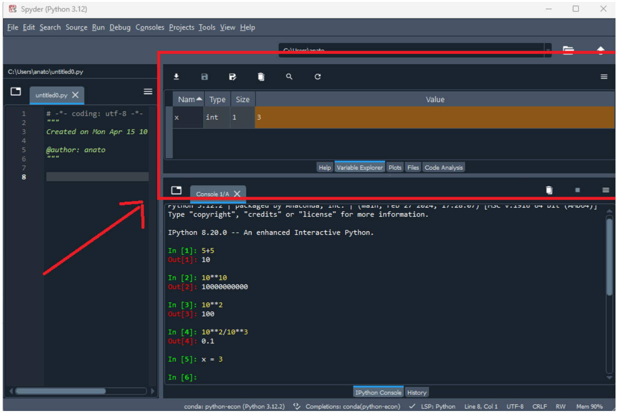
```
Python 3.12.2 | packaged by Anaconda, Inc. | (main, Feb
27 2024, 17:28:07) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more
information.

IPython 8.20.0 -- An enhanced Interactive Python.

In [1]:
```

Spyder Interface (II)

Variable Explorer

Variables:

- On Python, you can write variables in the memory.
  - **x=3**


Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\anato
C:\Users\anato\untitled0.py
untitled0.py
Help Variable Explorer Plots Files Code Analysis
Console 1/A
IPython Console History
conda: python-econ (Python 3.12.2) Completions: conda(python-econ) LSP: Python Line 8, Col 1 UTF-8 CRLF RW Mem 90%

```
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 15 10

@author: anato
"""

```

Nam	Type	Size	Value
x	int	1	3

```
Python 3.12.2 | packaged by Anaconda, Inc. | (main, Feb 27 2024, 17:28:07) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.20.0 -- An enhanced Interactive Python.

In [1]: 5+5
Out[1]: 10

In [2]: 10**10
Out[2]: 10000000000

In [3]: 10**2
Out[3]: 100

In [4]: 10**2/10**3
Out[4]: 0.1

In [5]: x = 3

In [6]:
```

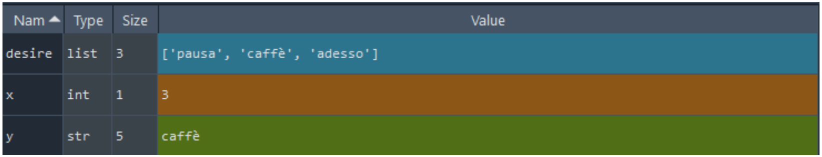
Base Python (I)

Now you can create different variables:

- String: **y = 'caffè'**
- List: **desire = ['pausa', 'caffè']**

Look at all these objects changing in your variable environment!

- **desire.append('tra poco')**
- You've modified your desire list and added one element!
- **desire[2] = 'adesso'**
- You've modified the third element of the list!

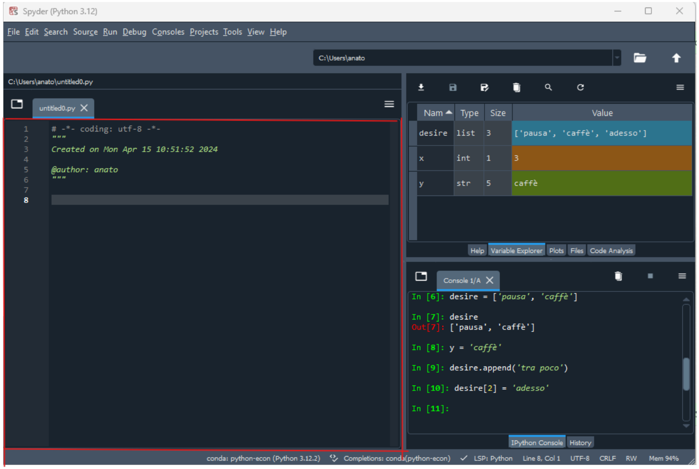


Nam ▲	Type	Size	Value
desire	list	3	['pausa', 'caffè', 'adesso']
x	int	1	3
y	str	5	caffè

Spyder Interface (III)

We will start writing code we will want to remember:

- Use the *script* tool of python!


Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\anato
C:\Users\anato\untitled0.py
untitled0.py
Help Variable Explorer Plots Files Code Analysis
Console 1/A
IPython Console History
conda: python-econ (Python 3.12.2) Completions: conda(python-econ) LSP: Python Line 8, Col 1 UTF-8 CRLF RW Mem 94%

```
# -*- coding: utf-8 -*-
"""
Created on Mon Apr 15 10:51:52 2024

@author: anato
"""

```

Nam	Type	Size	Value
desire	list	3	['pausa', 'caffè', 'adesso']
x	int	1	3
y	str	5	caffè

```
In [6]: desire = ['pausa', 'caffè']

In [7]: desire
Out[7]: ['pausa', 'caffè']

In [8]: y = 'caffè'

In [9]: desire.append('tra poco')

In [10]: desire[2] = 'adesso'

In [11]:
```

Spyder Interface (IV)

Try and write some code in the script tool:

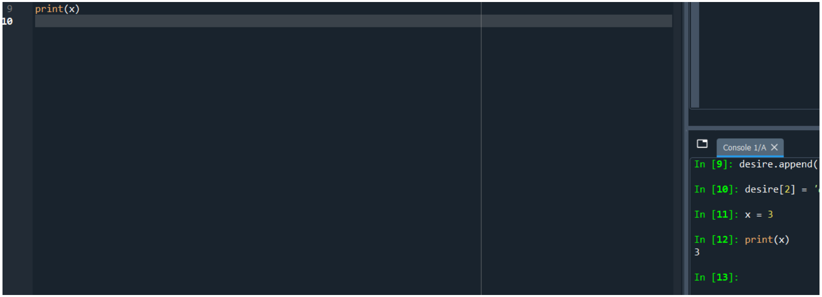
- **x = 3**
- **print(x)**

To "run" the cell, or automatically input it into the console:

- Shortcuts: F9 for me.
- the "Run" tab above.
- Define the shortcuts that are better for you:
  - Tools − > Preferences − > Shortcuts
  - In the Search tab, type "run selection" − > select the key you want
  - Type "run" select the one where name is only run − > select key

Spyder - Shortcuts

When you use the run selection shortcut:



```
print(x)
```

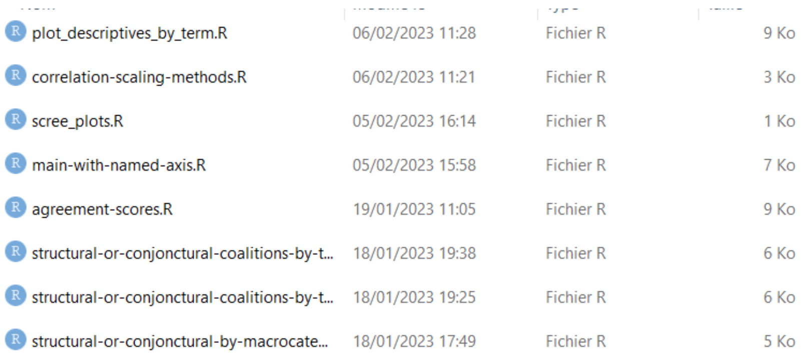
```
In [9]: desire.append(

In [10]: desire[2] = '

In [11]: x = 3

In [12]: print(x)
3

In [13]:
```

The lines where your cursor is pointing (or the line you highlighted) is inputed to the console.

Good Practice (I)

Always work with scripts – and save them with appropriate names!

- If you only type in the console there's a good chance you won't remember.
  - I made that mistake so many times... be smarter.
- The name of your script should reflect what your script does.



plot_descriptives_by_term.R	06/02/2023 11:28	Fichier R	9 Ko
correlation-scaling-methods.R	06/02/2023 11:21	Fichier R	3 Ko
scree_plots.R	05/02/2023 16:14	Fichier R	1 Ko
main-with-named-axis.R	05/02/2023 15:58	Fichier R	7 Ko
agreement-scores.R	19/01/2023 11:05	Fichier R	9 Ko
structural-or-conjonctural-coalitions-by-t...	18/01/2023 19:38	Fichier R	6 Ko
structural-or-conjonctural-coalitions-by-t...	18/01/2023 19:25	Fichier R	6 Ko
structural-or-conjonctural-by-macrocate...	18/01/2023 17:49	Fichier R	5 Ko

Good Practice (II)

COMMENTS

You want to comment most of what you write:

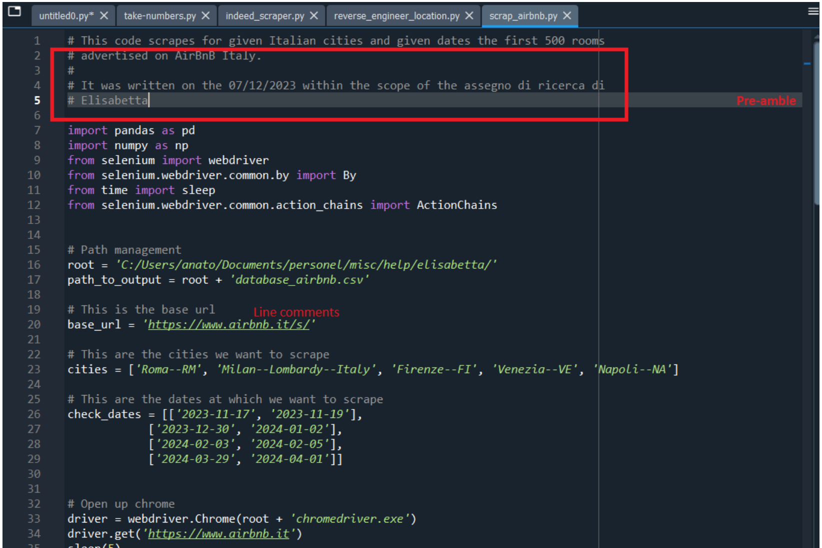
- Comments are inserted with # in python and R.
- A good rule is one comment per action.

It is crucial that you comment your work well:

- You're gonna have many projects, and you'll go back to your codes maybe 2 months after you wrote them.
- Other people will read your work (me).
- **comment a lot please**.

How much to comment?

A lot



```python
# This code scrapes for given Italian cities and given dates the first 500 rooms
# advertised on AirBnB Italy.
#
# It was written on the 07/12/2023 within the scope of the assegno di ricerca di
# Elisabetta

import pandas as pd
import numpy as np
from selenium import webdriver
from selenium.webdriver.common.by import By
from time import sleep
from selenium.webdriver.common.action_chains import ActionChains


# Path management
root = 'C:/Users/anato/Documents/personel/misc/help/elisabetta/'
path_to_output = root + 'database_airbnb.csv'

# This is the base url
base_url = 'https://www.airbnb.it/s/'

# This are the cities we want to scrape
cities = ['Roma--RM', 'Milan--Lombardy--Italy', 'Firenze--FI', 'Venezia--VE', 'Napoli--NA']

# This are the dates at which we want to scrape
check_dates = [['2023-11-17', '2023-11-19'],
              ['2023-12-30', '2024-01-02'],
              ['2024-02-03', '2024-02-05'],
              ['2024-03-29', '2024-04-01']]


# Open up chrome
driver = webdriver.Chrome(root + 'chromedriver.exe')
driver.get('https://www.airbnb.it')
sleep(5)
```