Machine Learning and Programming in Python

Lecture for Master and PhD students

Chair of Data Science in Economics

Ruhr University Bochum

Summer semester 2024

Lecture 2

Outline

- How to download and install Python.
- What is Jupyter about?
- The main concepts of Python

- We use Python because is easy to learn, friendly and we have thousand of online resources in case we need help.
- The syntax is simpler than in C++, Java, JavaScript...
- We are going to start from the beginner level, to progressively increase the complexity of the module.

- Code quality.
  Highly readable code, making it reusable and maintainable.
- Developer Productivity.
  Coding style is considered as easy and elegant, using a correct English syntax that makes everything simpler than in other software. Also, the amount of code is smaller.
- Code Portability.
  The code can be imported by many used, with no need to change a line of code.
- Built-in and external libraries.
  This will make your life easier.

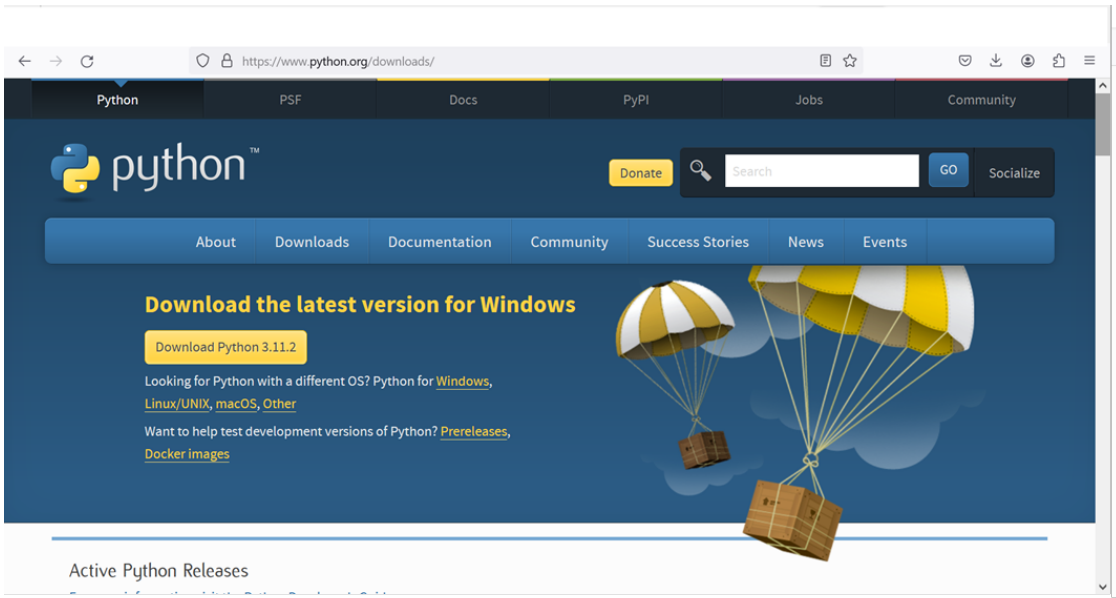
Main Python fields

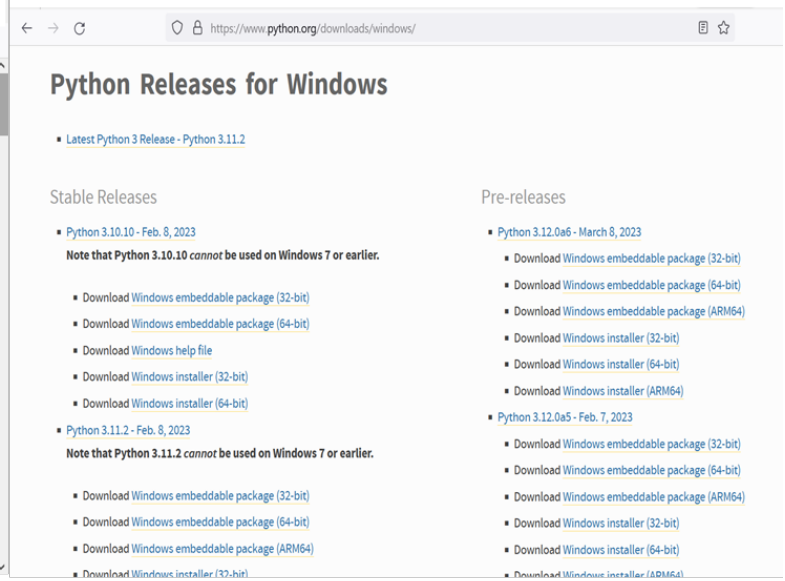
- Web application development
  Python boom is mostly attributed to web development: Tasks' simplification, backend data sets, handling online information and protocols like http, ftp, etc.

- Data Science and Machine Learning
  Machine learning and the automatisation of many economic processes (also business intelligence).

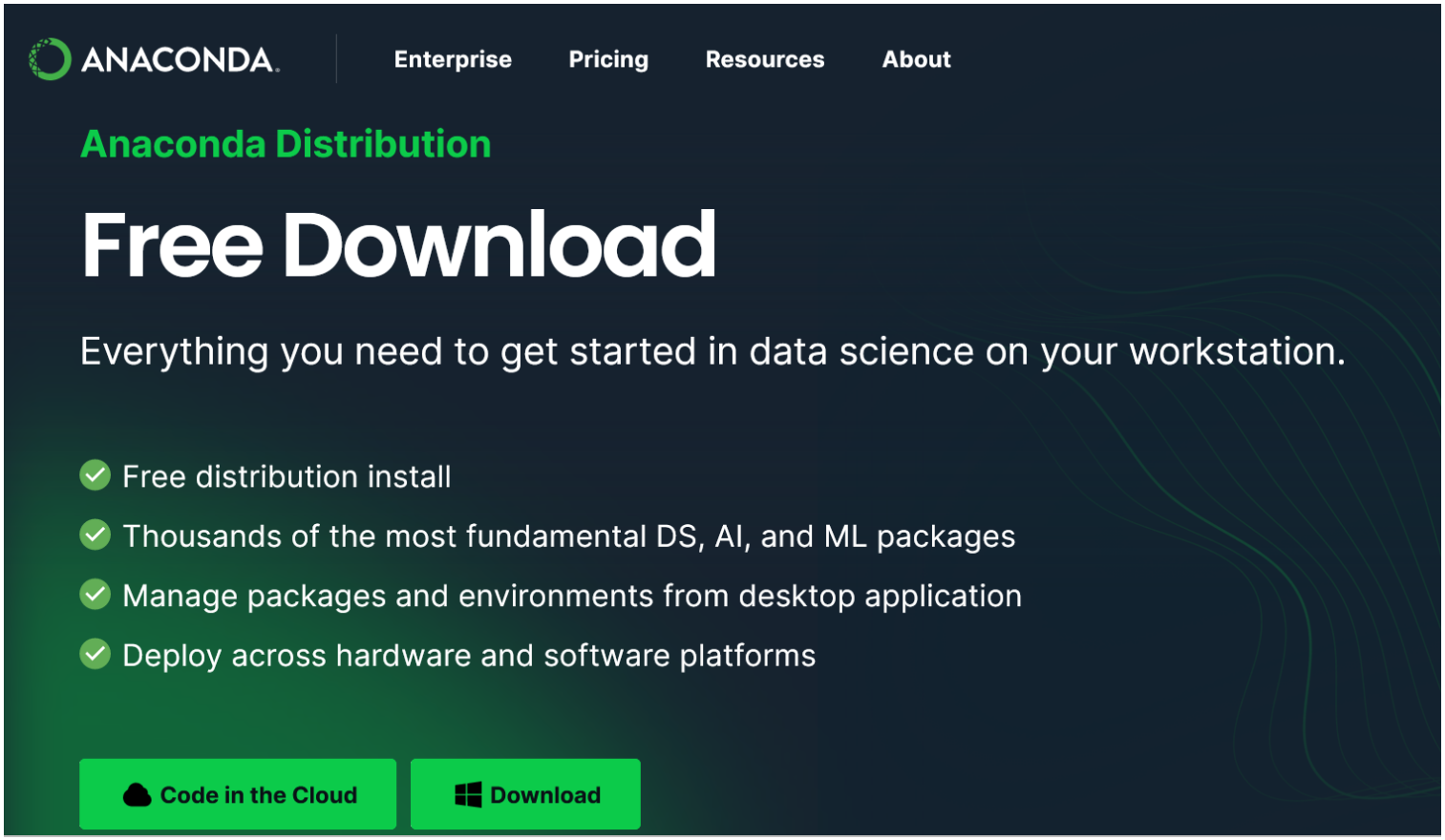
Installing Python either from python.org (https://www.python.org/downloads/) or via the platform Anaconda from anaconda.com

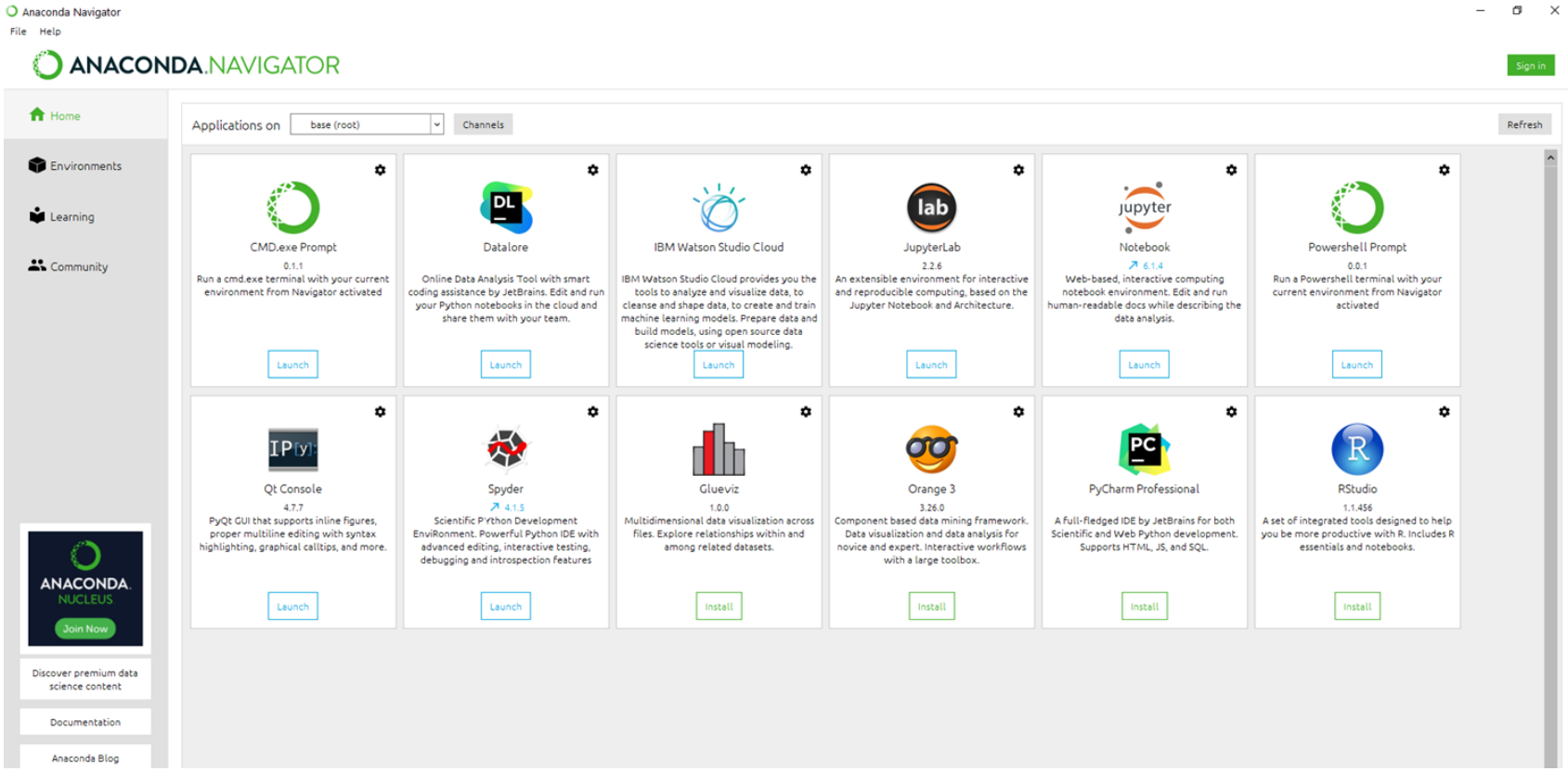
Recommendation to do it via Anaconda (see next slide)

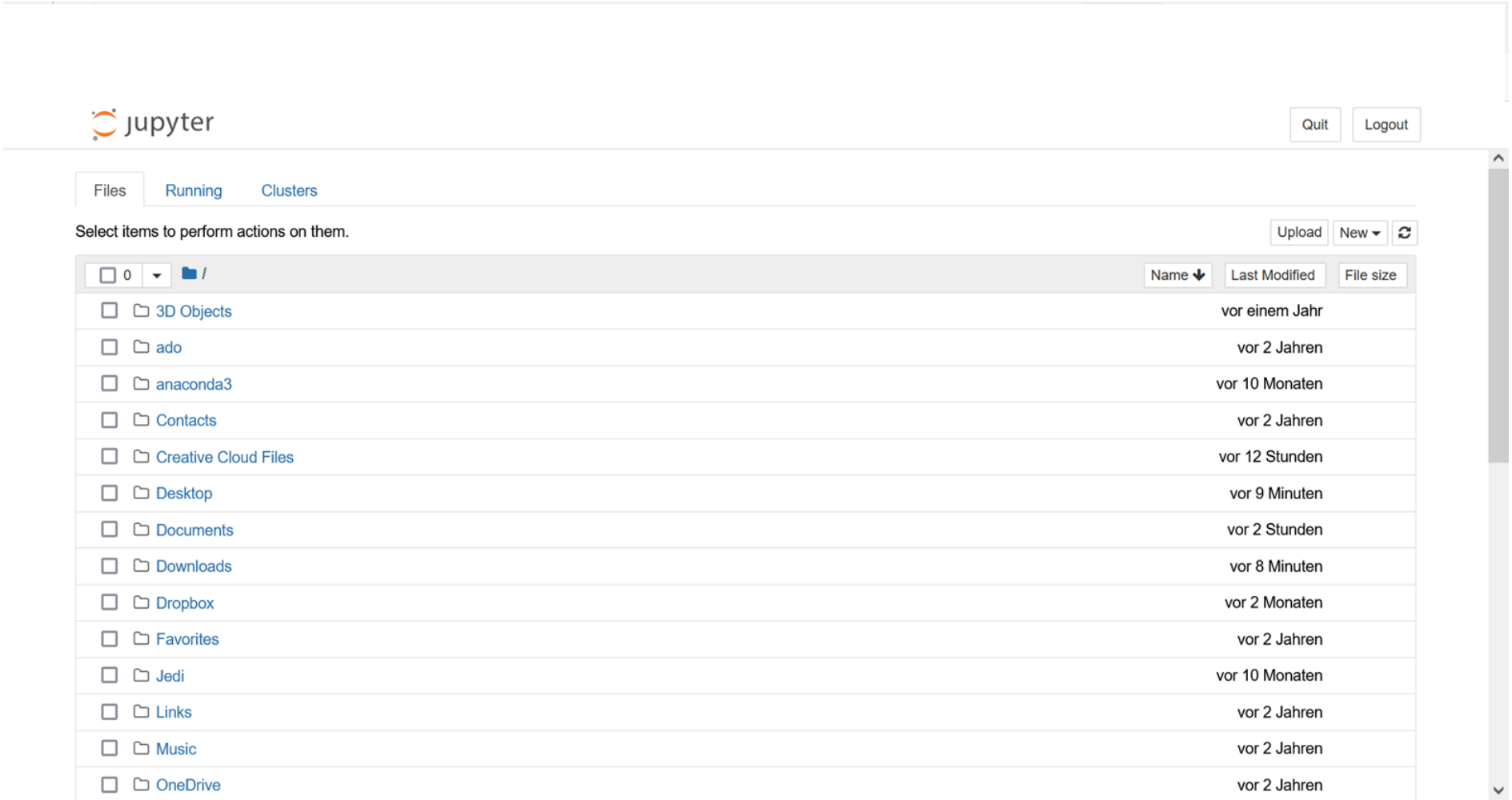




https://www.anaconda.com/products/distribution




Anaconda Navigator
File Help
ANACONDA.NAVIGATOR
Sign in
Home
Environments
Learning
Community
Applications on
base (root)
Channels
Refresh
CMD.exe Prompt
0.1.1
Run a cmd.exe terminal with your current environment from Navigator activated
Launch
Datalore
Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.
Launch
IBM Watson Studio Cloud
IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.
Launch
JupyterLab
2.2.6
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch
Notebook
6.1.4
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch
Powershell Prompt
0.0.1
Run a Powershell terminal with your current environment from Navigator activated
Launch
Qt Console
4.7.7
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch
Spyder
4.1.5
Scientific Python Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch
Glueviz
1.0.0
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install
Orange 3
3.26.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
Install
PyCharm Professional
A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.
Install
RStudio
1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.
Install
ANACONDA. NUCLEUS
Join Now
Discover premium data science content
Documentation
Anaconda Blog

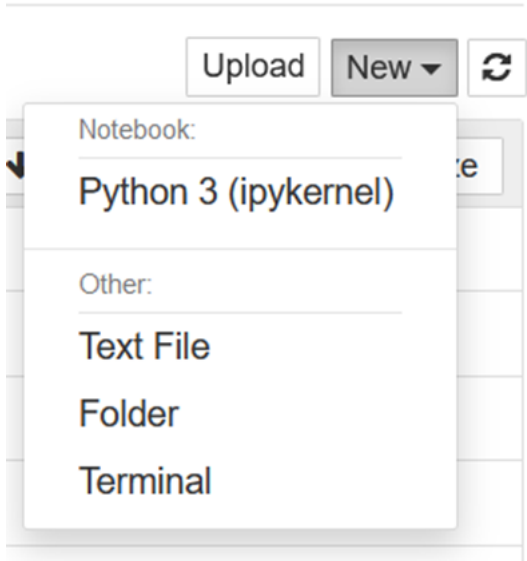

jupyter
Quit
Logout
Files
Running
Clusters
Select items to perform actions on them.
Upload
New
0
/
Name
Last Modified
File size
3D Objects
vor einem Jahr
ado
vor 2 Jahren
anaconda3
vor 10 Monaten
Contacts
vor 2 Jahren
Creative Cloud Files
vor 12 Stunden
Desktop
vor 9 Minuten
Documents
vor 2 Stunden
Downloads
vor 8 Minuten
Dropbox
vor 2 Monaten
Favorites
vor 2 Jahren
Jedi
vor 10 Monaten
Links
vor 2 Jahren
Music
vor 2 Jahren
OneDrive
vor 2 Jahren

Let's run it!

- Jupyter is both a notebook and a codebook; that is, it is used both to keep track of our script and to run estimates and algorithms in a friendly Python environment.

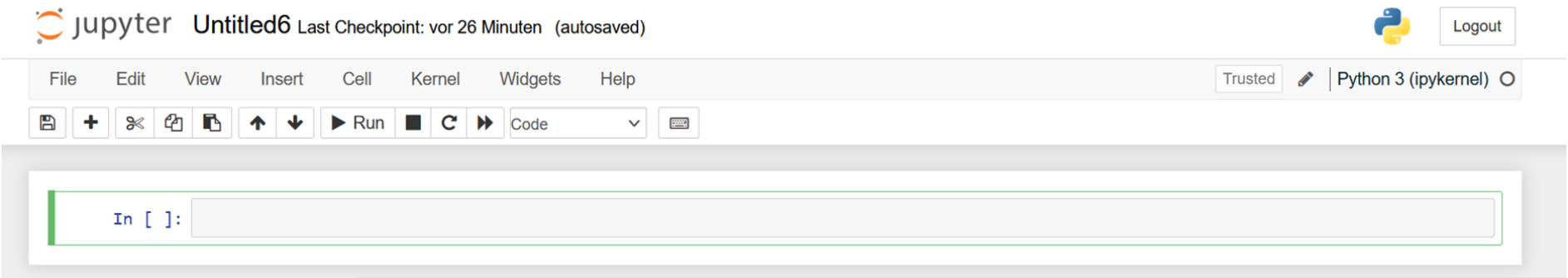
- The Dashboard gives you access to those files and folders created under the Jupyter environment, and you can change it if you prefer so.

- "New", gives you access to the different possibilities provided by Jupyter: Working in a Python environment, creating a new text file, or a folder.
- "Python 3" refers to your Jupyter Notebook.
- Your file will by default be saved as ".ipynb". All the information generated, including strings, floats, lists, etc. will be saved in your notebook.

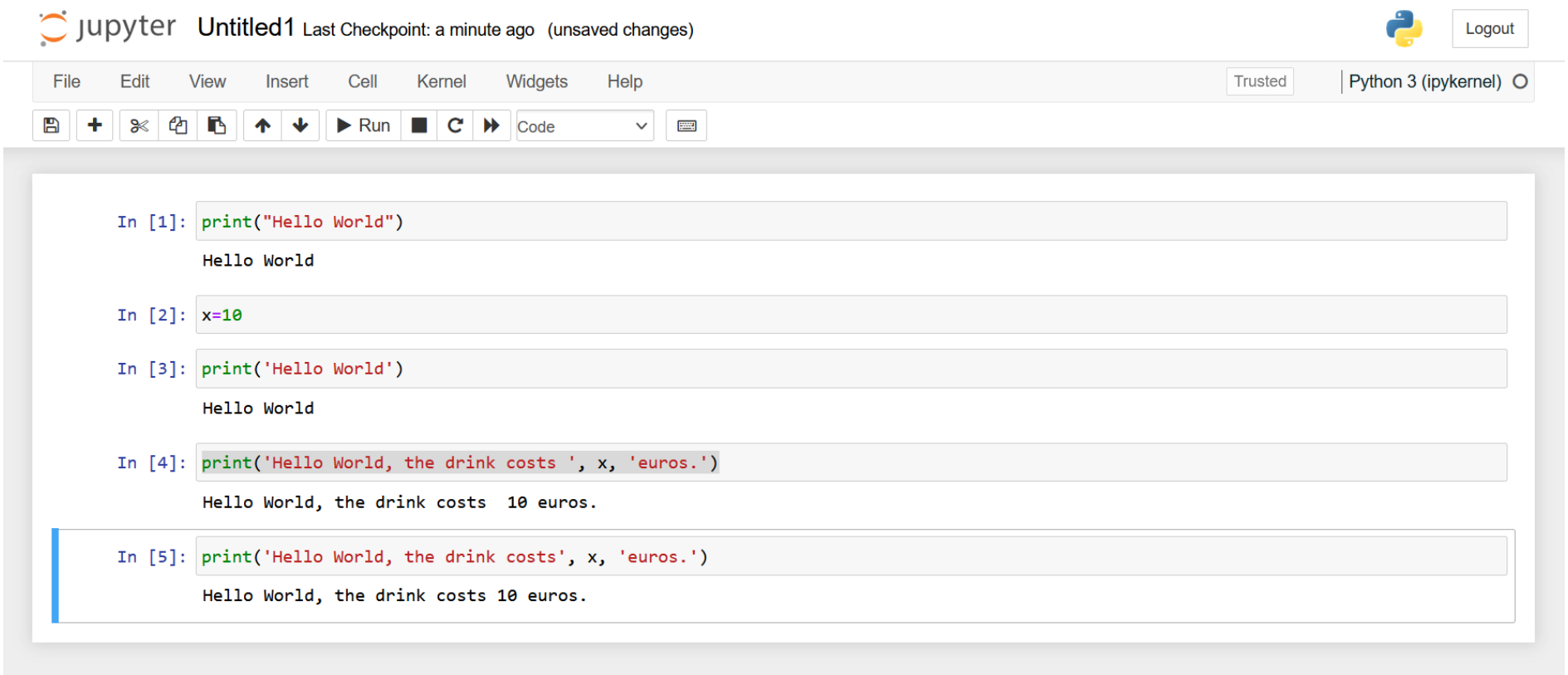


- Looks like the standard Microsoft Word environment (to make it more friendly).
- You have many options: Run your code, edit your text, view it, save it...

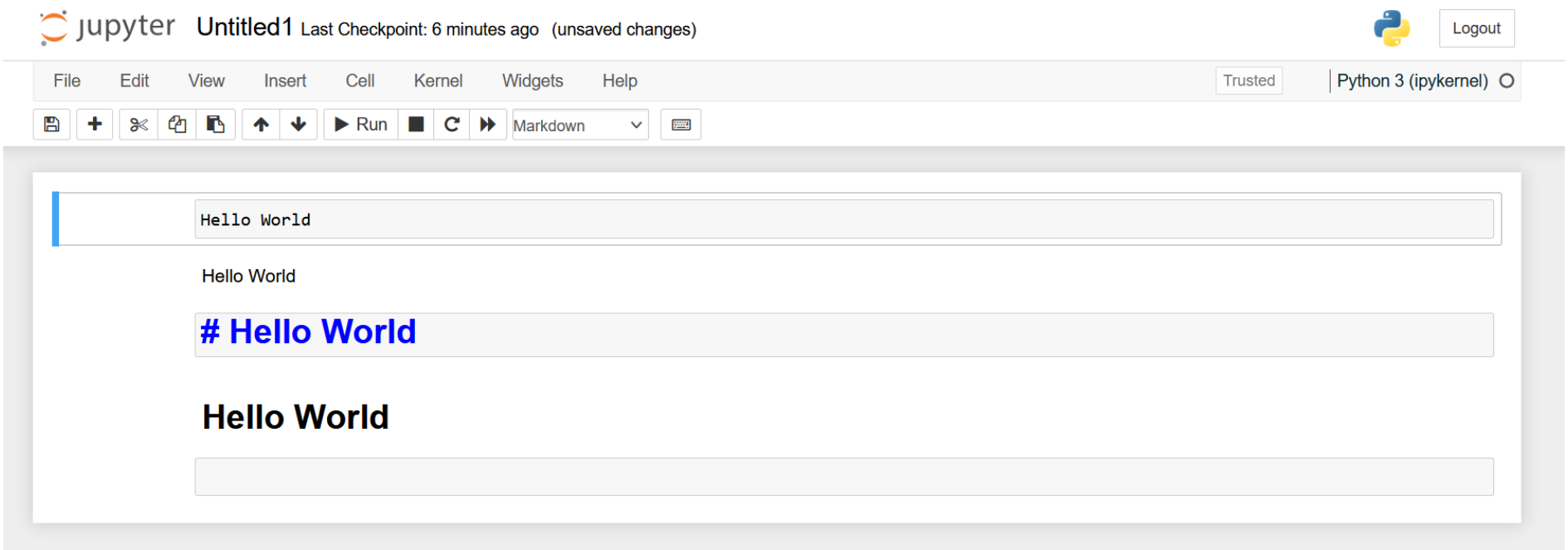
- Two concepts:
  - Kernel: The "engine" that is going to execute the code contained in our notebook.
  - Cell: What "contains" our text; to be fixed, modified, displayed, executed by Kernel...



1. Code Cell: Is the cell containing the code to be executed by our kernel. If everything is correct, the notebook is going to display the main results.



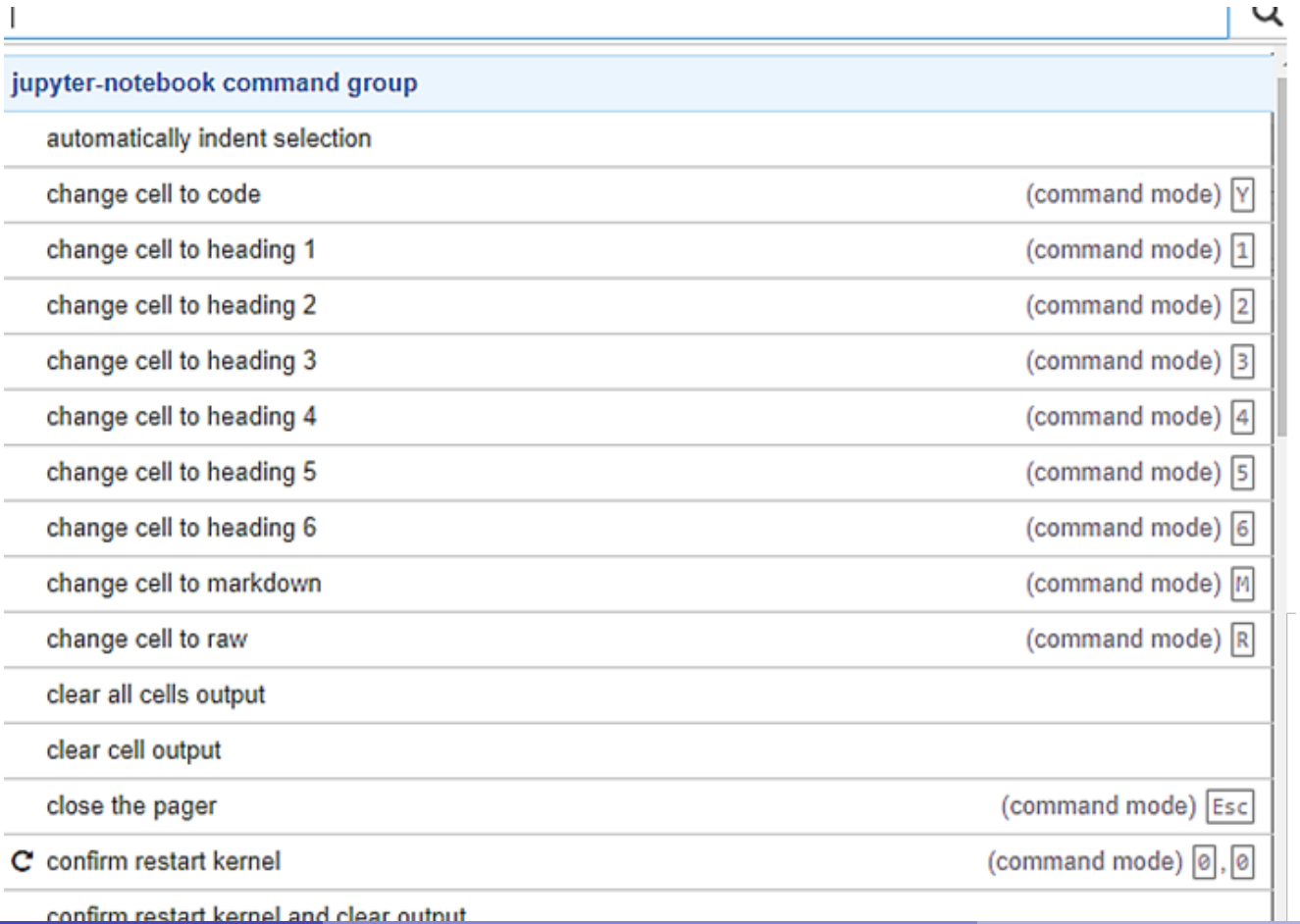
2. Markdown Cell: Contains text. Inserted just as it is, or with hashs. The output is displayed in a different way and not dependent on whether computations or logical operators are correct.



Jupyter commands/ shortcuts: p

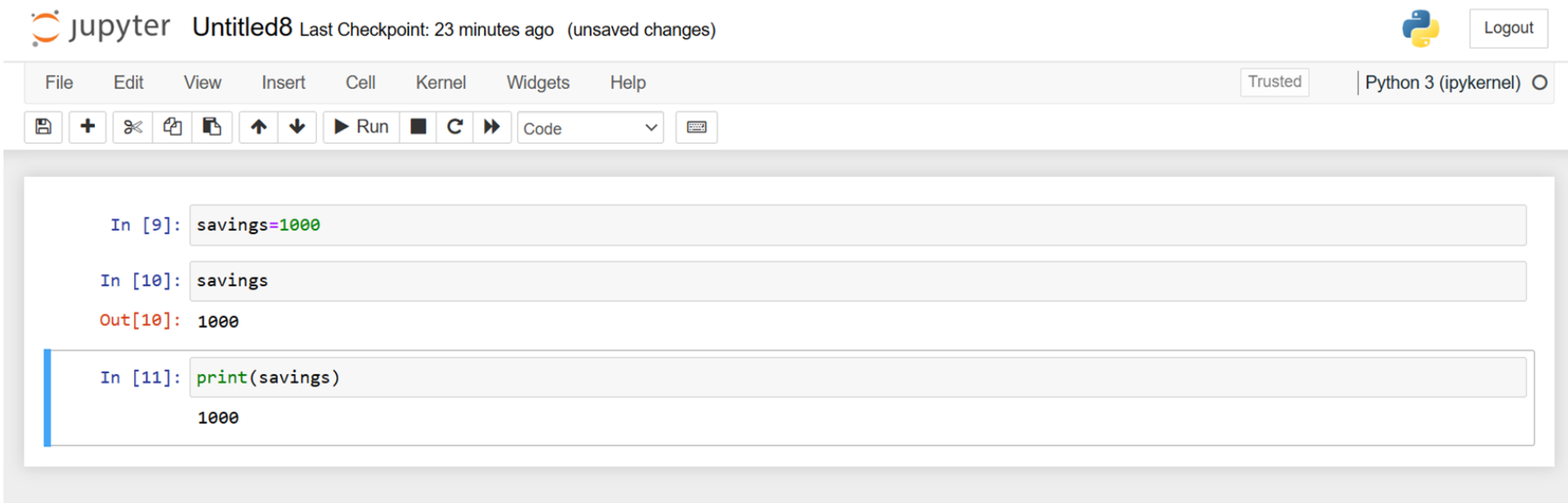
Run code: Ctrl and Enter

Print code/ Save as pdf: Ctrl and p



Variables:

Variables are what make our code and ideas feasible, in statistical and mathematical terms. We can set up the value of any variable just by "writing" it.



Data Types:

This refers to the format in which the variable has been created (string, float, etc.). In the previous example, savings was an integer.

There are many data types; floats, integers, strings, booleans, lists, tuples, sets, dicts, etc.

```
In [12]: type(savings)
Out[12]: int

In [13]: 2+3
Out[13]: 5

In [14]: "ab"+"cd"
Out[14]: 'abcd'
```