



Initial Setup

Lab-session 0

Computer Vision and Image Processing

Prof: Luigi di Stefano - luigi.distefano@unibo.it
Tutor: Pierluigi Zama Ramirez - pierluigi.zama@unibo.it

Requirements



These slides will teach you how to configure and install the required libraries in your computer. Requirements:

- Windows or Linux-Ubuntu (macOS and other Linux distributions are not in this guide but they should work too.)
- Internet connection

We will install:

- Python 3.x
- Pip
- Python-Libraries: OpenCV, Matplotlib, Numpy
- Jupyter Notebook

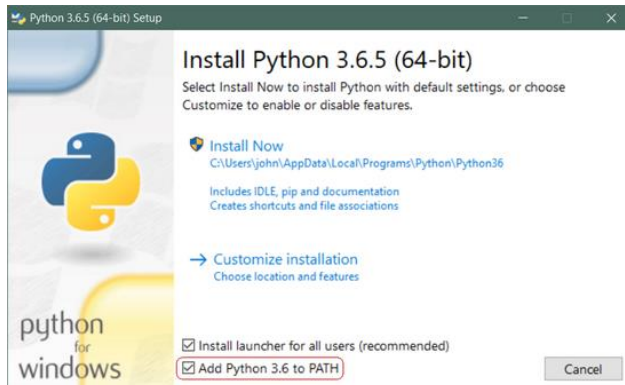
Step 1: Installing Python 3.x

Two main versions of Python: Python 2.x and Python 3.x are available.

The two version have several features in common, the two version are not fully compatible between each other and a Python 2.x program may not work for Python 3.x and vice versa.

Installing on Windows:

- Download the latest Python3 release from:
<https://www.python.org/downloads/>
- Run the installer. **Remind to Add Python to PATH**



Installing on Ubuntu:

- Ubuntu already have Python installed already. If you want to update it to the latest version open your terminal and run:
sudo apt-get install --upgrade python3

Step 2: Installing pip

PIP is a recursive acronym that stands for “PIP Installs Packages”. It’s a command-line utility that allows you to install, reinstall, or uninstall Python packages with a simple and straightforward command: **pip**

Installing on Windows:

- Download the get-pip.py installer script from here: <https://bootstrap.pypa.io/get-pip.py>
Right-click on the link and select Save As and save it to any location.
- Open the Command Prompt (better **Windows PowerShell**) and navigate to the folder containing the get-pip.py file (`cd path_to_folder`).
- Run the following command: ***python get-pip.py***

Installing on Ubuntu:

- Open your terminal and run the following:
sudo apt-get install python3-pip

Step 3: Install Python Libraries with pip



Installing on every OS:

Open your terminal (Windows PowerShell for Windows) and run the following commands:

- Install numpy (For array operations) :

pip3 install numpy

- Install matplotlib (For plotting and visualization)

pip3 install matplotlib

- Install OpenCV (Computer Vision Library):

pip3 install opencv-python==3.4.2.16

pip3 install opencv-contrib-python == 3.4.2.16

Step 4: Install and run Jupyter Notebook



The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. It is useful to explain how to code and to visualize interactively the results of an algorithm or code.

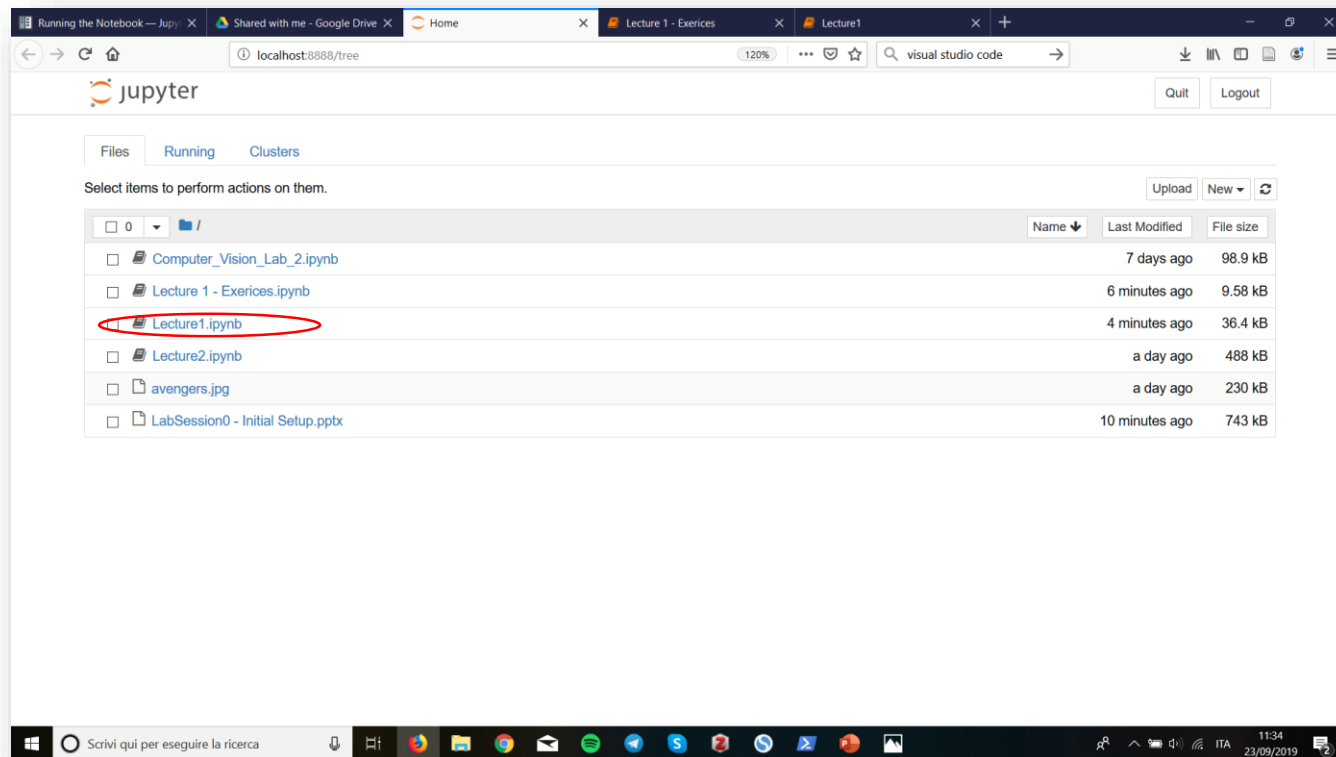
Install on every OS:

- Open your terminal and run: ***pip3 install jupyter***

Run it:

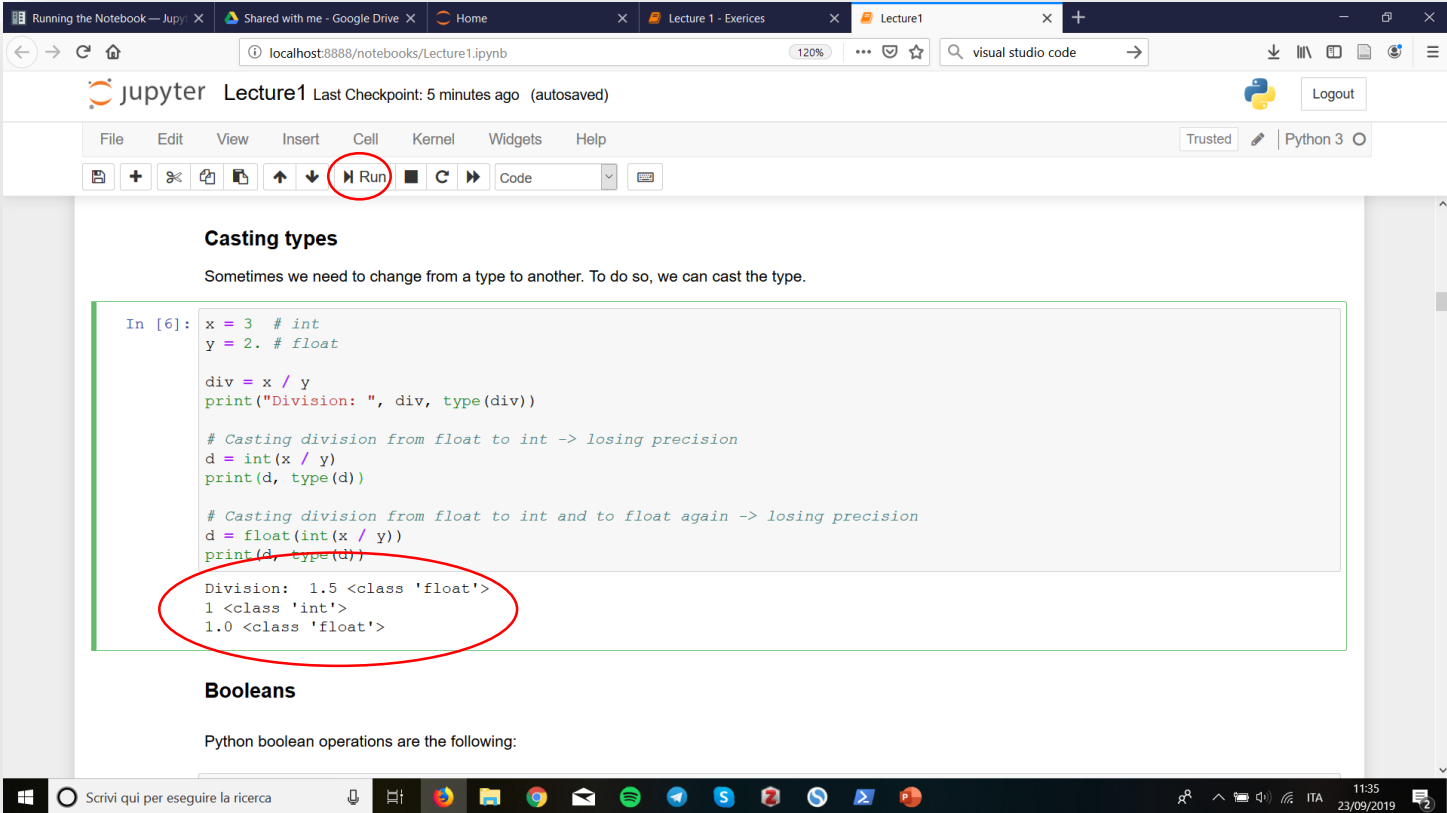
- **Navigate** to the folder containing the lab-session notebooks and **launch** the notebook server in a terminal (**do not close it!**) with the following command: ***jupyter notebook --port 8081***. You should see the notebook open in your browser. However, the terminal printed the URL of the server (***http://localhost:8081***)

Jupyter Notebook: An overview



Notebook dashboard. Navigate to your notebook (.ipynb file) and open it.

Jupyter Notebook: An overview



The screenshot shows a Jupyter Notebook interface in a web browser. The browser tabs include 'Running the Notebook - Jupyter', 'Shared with me - Google Drive', 'Home', 'Lecture 1 - Exercises', and 'Lecture1'. The address bar shows 'localhost:8888/notebooks/Lecture1.ipynb'. The Jupyter interface has a menu bar with 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. Below the menu bar is a toolbar with icons for file operations, a 'Run' button (circled in red), and a 'Code' dropdown. The notebook content includes a section titled 'Casting types' with a paragraph: 'Sometimes we need to change from a type to another. To do so, we can cast the type.' Below this is a code cell with the following code:

```
In [6]: x = 3 # int
        y = 2. # float

        div = x / y
        print("Division: ", div, type(div))

        # Casting division from float to int -> losing precision
        d = int(x / y)
        print(d, type(d))

        # Casting division from float to int and to float again -> losing precision
        d = float(int(x / y))
        print(d, type(d))
```

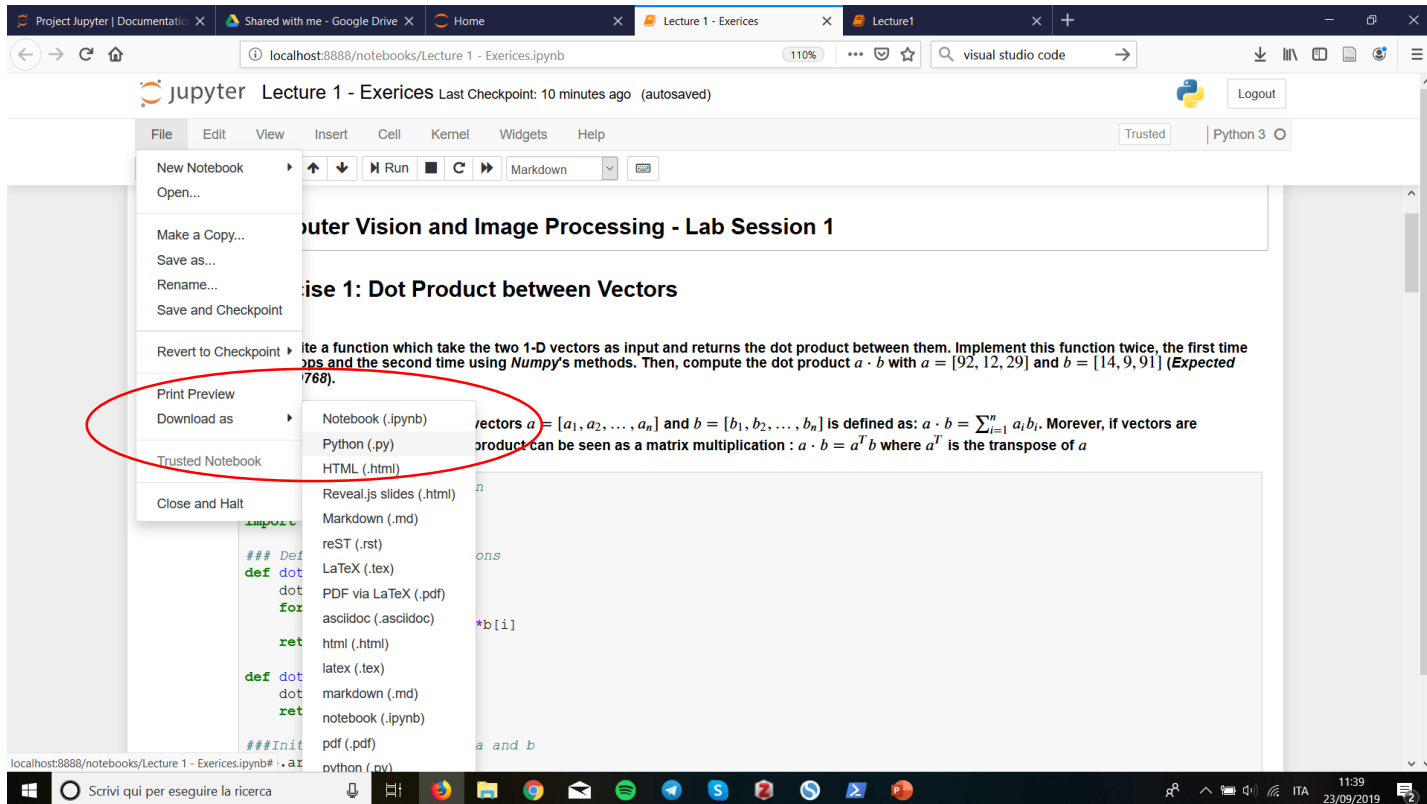
The output of the code cell is:

```
Division: 1.5 <class 'float'>
1 <class 'int'>
1.0 <class 'float'>
```

The output is circled in red. Below the code cell is a section titled 'Booleans' with the text: 'Python boolean operations are the following:'.

Run single code instruction and see the result interactively!

Export python .py from notebook file



You can download the all notebook as a single **.py** file! After that, you can run it as normal Python script.

Working from home: IDE and terminal



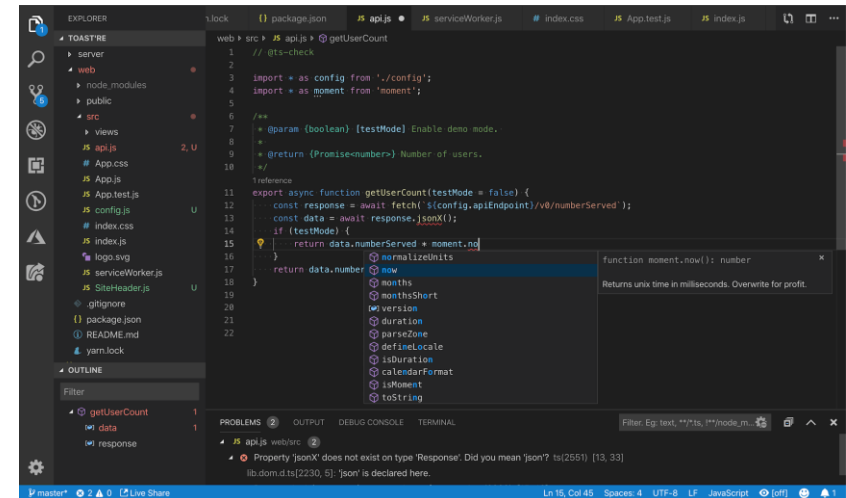
When you work with python you typically do not program directly on Jupyter Notebook but it is common to use a Source Code Editor such as

Visual Studio Code:

<https://code.visualstudio.com/>

or **Pycharm:**

<https://www.jetbrains.com/pycharm/>



Visual Studio Code

When you finish writing your script from your terminal you can navigate to your script folder and run:

python my_script.py

References

- Python3 documentation:

<https://docs.python.org/3/c-api/index.html>

- Jupyter documentation:

<https://jupyter.readthedocs.io/en/latest/>

- Numpy documentation:

<https://docs.scipy.org/doc/numpy/reference/index.html>

- Matplotlib documentation:

<https://matplotlib.org/3.1.1/api/index.html>

- OpenCV documentation:

https://docs.opencv.org/trunk/d6/d00/tutorial_py_root.html