



# Initial Setup

## Lab-session 0

### Computer Vision and Image Processing

Prof: Luigi di Stefano - [luigi.distefano@unibo.it](mailto:luigi.distefano@unibo.it)

Tutors: Andrea Amaduzzi - [andrea.amaduzzi4@unibo.it](mailto:andrea.amaduzzi4@unibo.it)

Iacopo Curti - [iacopo.curti2@unibo.it](mailto:iacopo.curti2@unibo.it)



# Working on your computer

# 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
- 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, but they are not fully compatible: a Python 2.x program may not work in Python 3.x and vice versa.

Installing on Windows:

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



Installing on Ubuntu:

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

# Step 2: 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***

***pip3 install opencv-contrib-python***

# 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.

**Installing** on every OS:

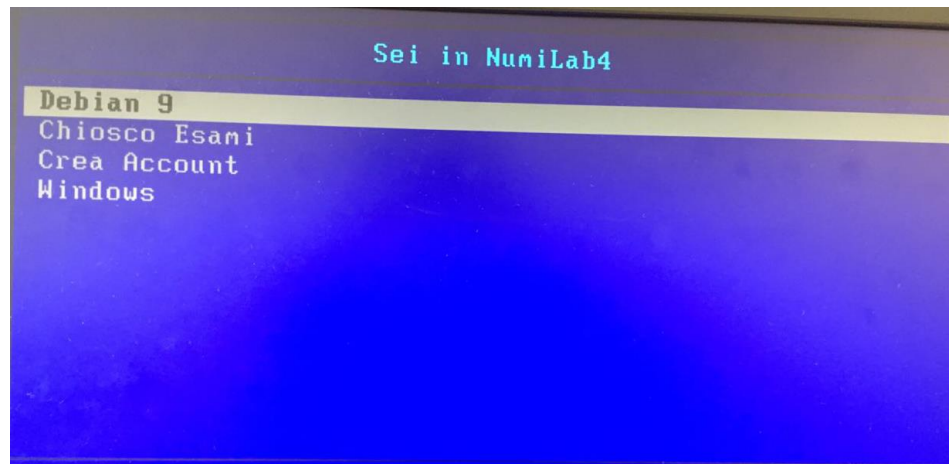
- Open your terminal and run:

***pip3 install jupyter***



# Working on Lab Computer

# Working on Lab computer



All the packages you need are already installed. If it is your first time in lab, perform the following steps:

1. Create your account
2. Login into **Debian 9** (only Linux will be supported in lab PCs) with username and password
3. run **startx** in the terminal to run the GUI





# After Initial Setup

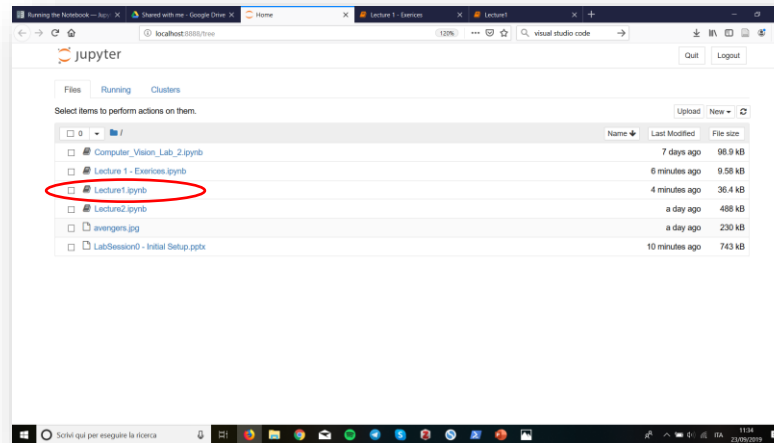
# Jupyter Notebook: An overview

**Run Jupyter Notebook:**

**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:

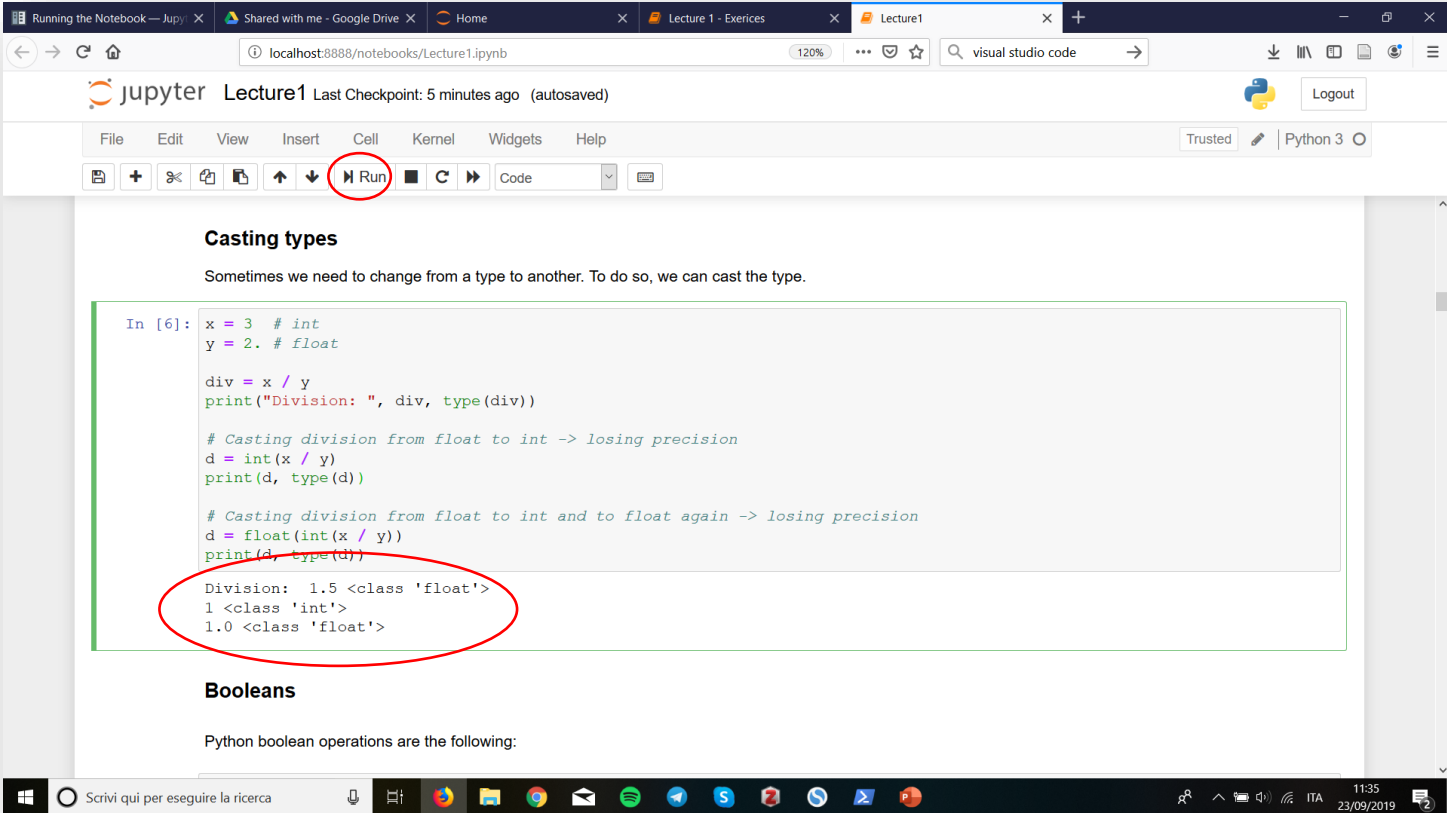
***jupyter notebook***

You should see the notebook open in your browser. If this is not the case, just point your browser to the URL printed on the terminal (default: **http://localhost:8888**)



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 is titled 'Lecture1' and shows the last checkpoint was 5 minutes ago. The code cell is titled 'Casting types' and contains the following Python 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))
```

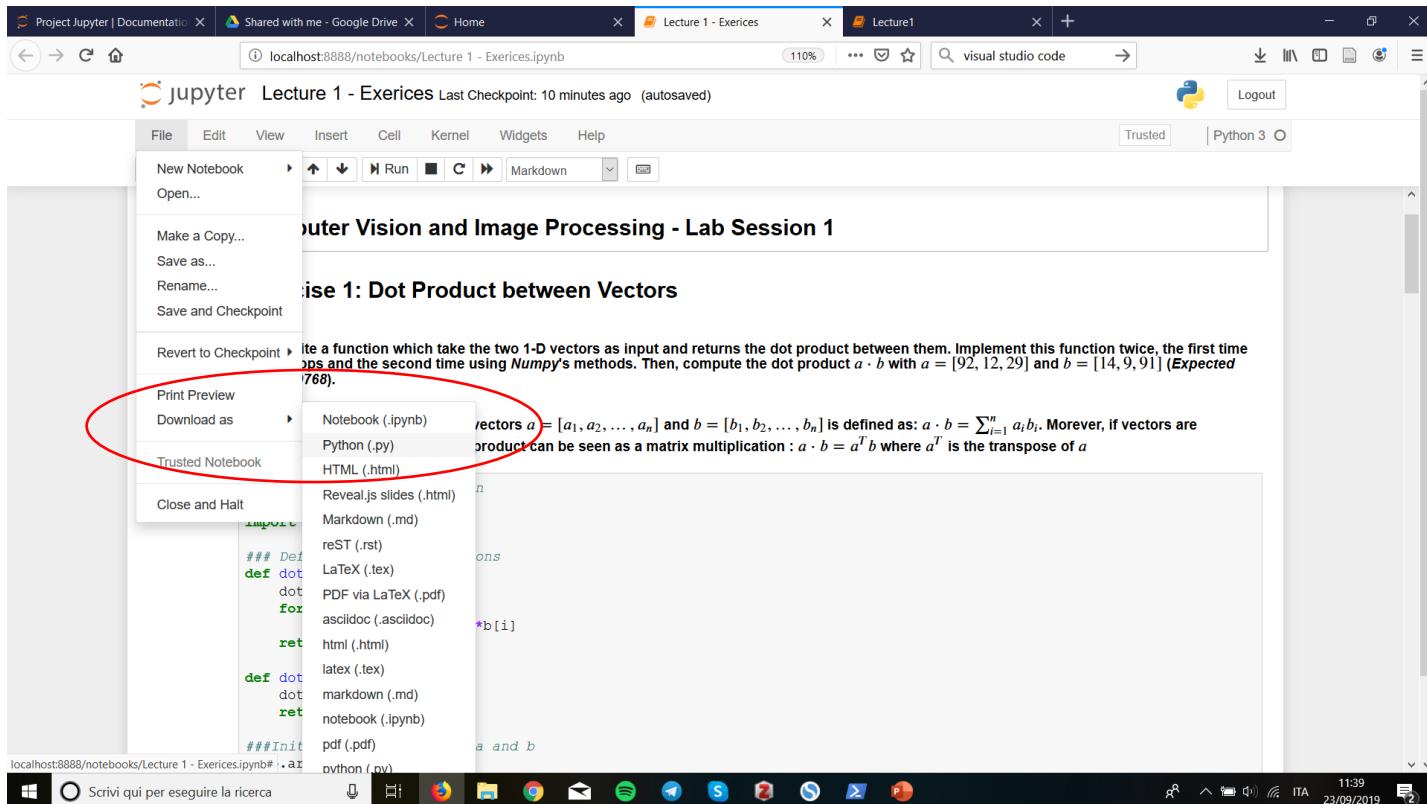
The output of the code is:

```
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 all the code cells of a notebook as a single **.py file**. After that, you can run it as a standard Python script.

# Working from home: IDE and terminal



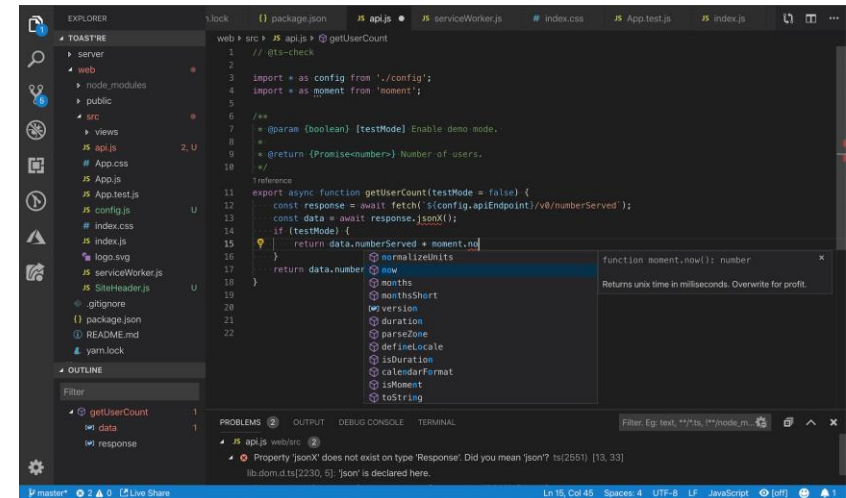
When you work with python you typically do not program directly on Jupyter Notebook but it is common to use a Source Code Editor such as

**Visual Studio Code:**

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

or **Pycharm:**

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



Visual Studio Code

When you finish writing your script, navigate in your terminal 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](https://docs.opencv.org/trunk/d6/d00/tutorial_py_root.html)