



# Pattern Recognition

## Introduction

*Heraklion 2026*  
*Dr. Konstantinos Karampidis*



## Theory:

Where and when: Room 207 on Mondays (14:15 – 14:00)  
Room Γ2 on Tuesdays (14:15 – 14:00)

## Laboratory:

Where: Lab 3

When: The schedule will be announced in eclass course's  
agenda

## Grading

| Course unit                | Evaluation                                                            | % Theory | % Laboratory |
|----------------------------|-----------------------------------------------------------------------|----------|--------------|
| Theory                     | Theoretical project                                                   | 10%      | —            |
|                            | Final exam                                                            | 70%      | —            |
| Laboratory                 | Code, impementation, experiments, results, presentation and questions | —        | 100%         |
| Overall weight of the unit |                                                                       | 80%      | 20%          |



### Theoretical Project

| Status                       | % of the Project       |
|------------------------------|------------------------|
| Midterm Presentation         | 20%                    |
| Final Project & Presentation | 80%                    |
| Total                        | 10% of the total grade |

### Laboratory – 20% of the total grade

Code related to the theoretical project. During the final presentation of the theoretical project, the implementation will be graded, and lab questions will be answered.

### Requirements

| Requirement                                     | Minimum grade |
|-------------------------------------------------|---------------|
| Laboratory → eligibility to take the final exam | 4/10          |
| Successful completion of the course             | 5/10          |

### Example



| Unit                | Grade | Weight | Theory contribution | Laboratory contribution |
|---------------------|-------|--------|---------------------|-------------------------|
| Theoretical project | 8/10  | 10%    | 0,80                | —                       |
| Final exam          | 6/10  | 70%    | 4,20                | —                       |
| Theory total        | —     | 100%   | 5,00/10             |                         |
|                     |       |        |                     |                         |
| Laboratory          | 5/10  | 100%   | 5,00/10             | 1                       |

The final grade for the course is:

$$\begin{aligned} &5,00 \times 0,80 + 5,00 \times 0,20 \\ &= 4,00 + 1,00 = \boxed{5,00/10} \end{aligned}$$

Therefore, the student is graded **5,00/10**.

It satisfies the condition : Laboratory **5**  $\geq$  4

## Rules



- Deliver the theoretical project on time.
- Code Implementation with **Anaconda / Jupyter notebooks**.
- You can use only the python libraries used in the laboratory
- The theoretical project will be submitted strictly through **eclass**.
- To get a grade you must submit the theoretical project.
- Only one absence is allowed for the laboratory.
- If you use AI you must define how you have used it.
- Extensive use (more than 20%) of AI (theoretical project or/and code implementation) leads to course failure.
- All theoretical projects will be checked using Turnitin. A similarity report more than 20% leads to course failure.
- For any subject concerning the lab, you may contact me through email: [karampidis@hmu.gr](mailto:karampidis@hmu.gr)



# Syllabus

- 1 Introduction to Pattern Recognition**
- 2 Data Preprocessing & Feature Representation**
- 3 Distance & Similarity Measures**
- 4 Feature Selection, Feature Extraction & Dimensionality Reduction**
- 5 k-NN**
- 6 Bayesian Classification**
- 7 Linear Classifiers / Logistic Regression**
- 8 SVM**
- 9 Clustering**
- 10 Decision Trees & Random Forests**
- 11 Model Evaluation & Model Selection**
- 12 End-to-end Pattern Recognition Application**

# Laboratory software

We will use Anaconda for the laboratory. [Anaconda](#) is a Python distribution. A Python distribution provides the Python interpreter, together with a set of Python packages and sometimes other related tools, such as editors.

## Why Anaconda ?

- The open-source [Anaconda Distribution](#) is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 19 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:
- Quickly download 7,500+ Python/R data science packages
- Manage libraries, dependencies, and environments with [Conda](#)
- Develop and train machine learning and deep learning models with [scikit-learn](#), [TensorFlow](#), and [Theano](#)
- Analyze data with scalability and performance with [Dask](#), [NumPy](#), [pandas](#), and [Numba](#)
- Visualize results with [Matplotlib](#), [Bokeh](#), [Datashader](#), and [Holoviews](#)

- Visit <https://www.anaconda.com/products/individual>

and download the installer that corresponds to the operating system you are using

## Anaconda Individual Edition

Download 

For Windows

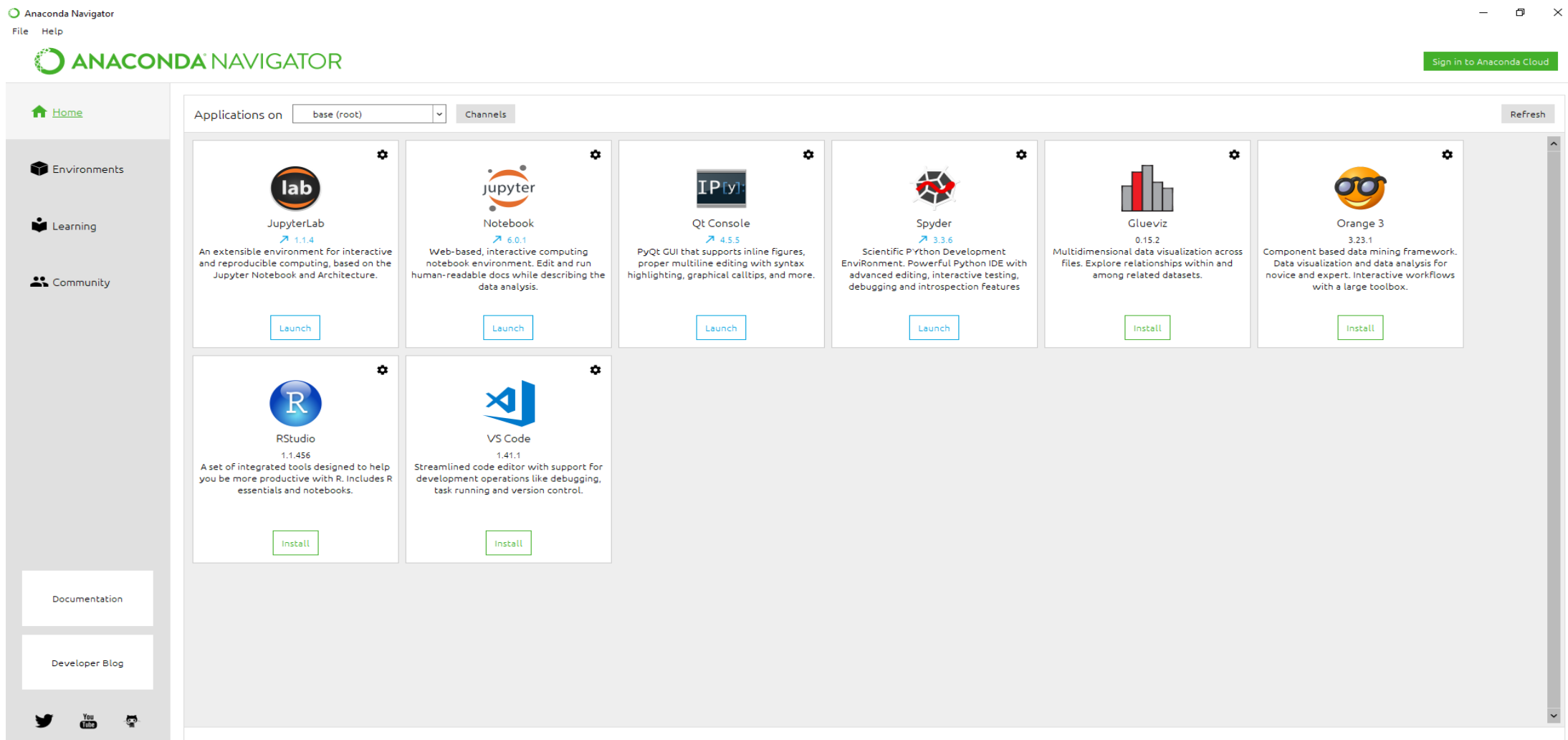
Python 3.8 • 64-Bit Graphical Installer • 477 MB

Get Additional Installers



# Anaconda

After installing and executing the program you will see this:



# Anaconda Installation

Depending on your operating system you can find a detailed installation guide below

[Windows](#)

[Linux](#)

[macOS](#)

# Anaconda Enviroments

Navigator uses conda to create separate environments containing files, packages, and their dependencies that will not interact with other environments.

- Create a new environment named snowflakes and install a package in it
- In Navigator, click the Environments tab, then click the Create button. The Create new environment dialog box appears.
- In the Environment name field, type a descriptive name for your environment.

Note that in this step you will choose the Python version you want to use. Therefore, you can have enviroments with different versions of Python according to your needs.

Also, you can choose to install R instead of Python.

# Anaconda Enviroments

Create new environment

X

Name:

snowflakes

Location:

*C:\Users\builder\Anaconda3\envs\snowflakes*

Packages:

☒ Python

3.7

▼

☐ R

mro

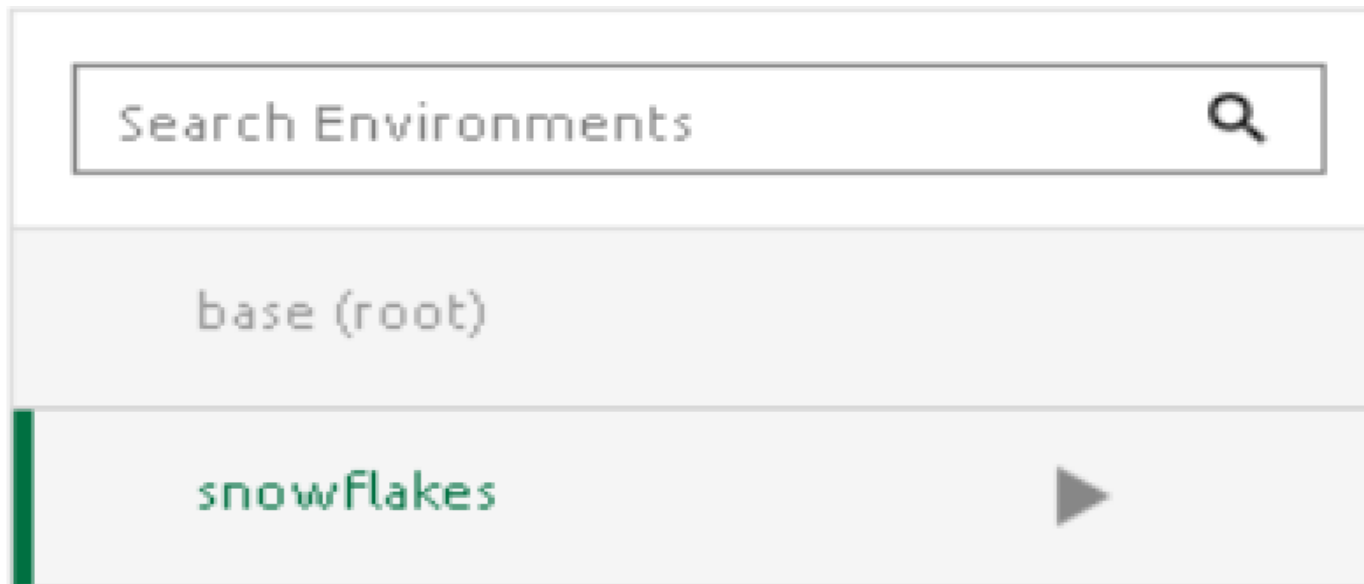
▼

Cancel

Create

# Anaconda Enviroments

Click Create. Navigator creates the new environment and activates it.



Now you have two environments, the default environment base (root), and snowflakes.

Switch between them (activate and deactivate environments) by clicking the name of the environment you want to use.

Return to the other environment by clicking its name.

# Anaconda Enviroments

## *Managing packages*

In this section, you check which packages you have installed, check which are available, and look for a specific package and install it.

To find a package you have already installed, click the name of the environment you want to search. The installed packages are displayed in the right panel.

You can change the selection of packages displayed in the right panel at any time by clicking the drop-down box above it and selecting Installed, Not Installed, Updatable, Selected, or All.

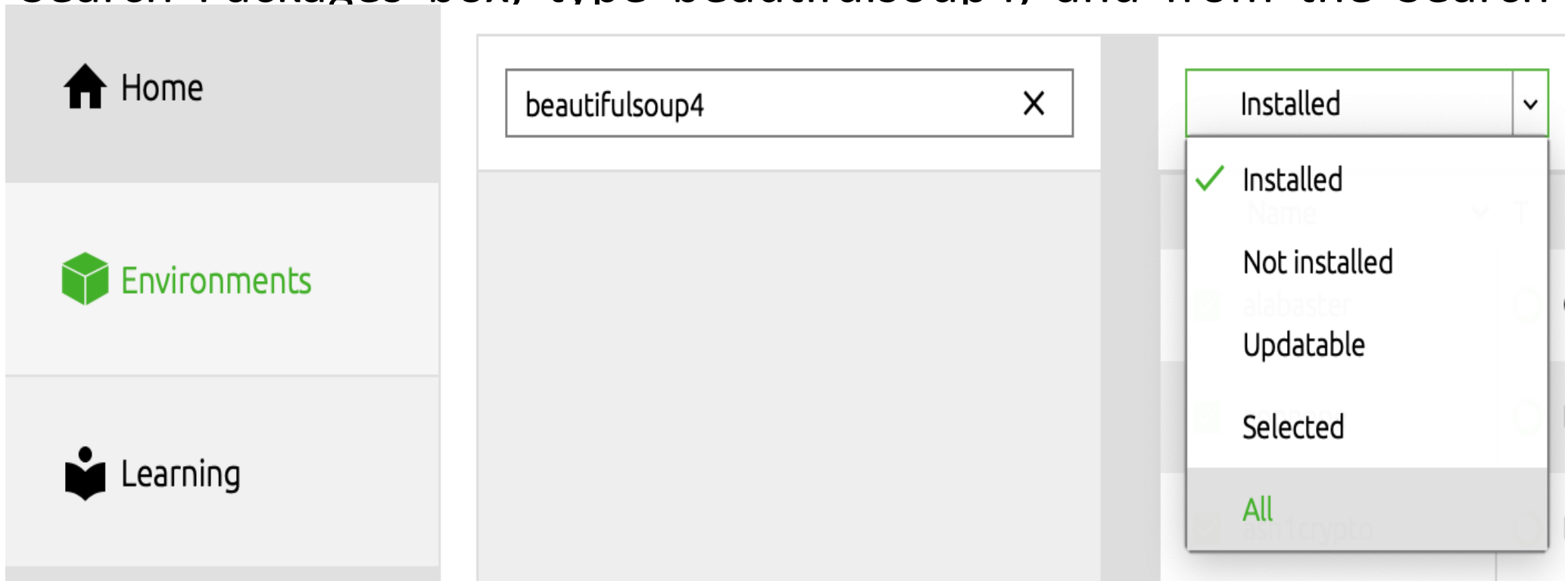
# Anaconda Enviroments

The screenshot displays the Anaconda Environments application window. At the top, there is a header bar with a dropdown menu currently set to 'Installed'. To the right of the dropdown are buttons for 'Channels', 'Update index...', and a 'Search Packages' field with a magnifying glass icon. Below the header, a table lists installed packages. The first row shows 'alabaster' with a description 'Configurable, python 2+3 compatible sphinx theme.' The second row shows 'Disable app nap on os x 10.9'. The third row shows 'asn1crypto' with a description 'Python asn.1 library with a focus on performance and a pythonic api'. The 'Installed' dropdown menu is open, showing options: 'Installed' (checked with a green checkmark), 'Not installed', 'Updatable', 'Selected', and 'All'.

| Name                         | Description                                                         |
|------------------------------|---------------------------------------------------------------------|
| alabaster                    | Configurable, python 2+3 compatible sphinx theme.                   |
| Disable app nap on os x 10.9 |                                                                     |
| asn1crypto                   | Python asn.1 library with a focus on performance and a pythonic api |

# Anaconda Enviroments

Check to see if a package you have not installed named “beautifulsoup4” is available from the Anaconda repository (must be connected to the Internet). On the Environments tab, in the Search Packages box, type beautifulsoup4, and from the Search



# Anaconda Enviroments

To install the package into the current environment, check the checkbox next to the package name, then click the bottom Apply

Not installed

Channels

Update index...

Search Packages

| Name                                             | T | Description                                                                  | Version |
|--------------------------------------------------|---|------------------------------------------------------------------------------|---------|
| <input type="checkbox"/> _ipyw_jlab_nb_ex...     |   | A configuration metapackage for enabling anaconda-bundled jupyter extensions | 0.1.0   |
| <input checked="" type="checkbox"/> _mutex_mxnet |   |                                                                              | 0.0.40  |
| <input type="checkbox"/> _nb_ext_conf            |   |                                                                              | 0.4.0   |
| <input type="checkbox"/> _py-xgboost-mutex       |   |                                                                              | 2.0     |
| <input checked="" type="checkbox"/> _r-mutex     |   |                                                                              | 1.0.0   |
| <input type="checkbox"/> _r-xgboost-mutex        |   |                                                                              | 2.0     |
| <input type="checkbox"/> _tflow_1100_select      |   |                                                                              | 0.0.2   |

1783 packages available2 packages selected

ApplyClear

# Jupyter Notebook

The Jupyter Notebook is an interactive programming environment , in which you can combine Python code and execution results with annotations, equations, figures, links, etc.

The Jupyter Notebook will run in the internet browser of our computer and does not require internet access, it will access to our local files via the browser interface.

Notebooks are similar to Word documents and we can edit them interactively in the internet browser. Notebooks are saved in its own format with the '. ipynb' extension, but can be also downloaded as PDFs, HTML pages or Python code.

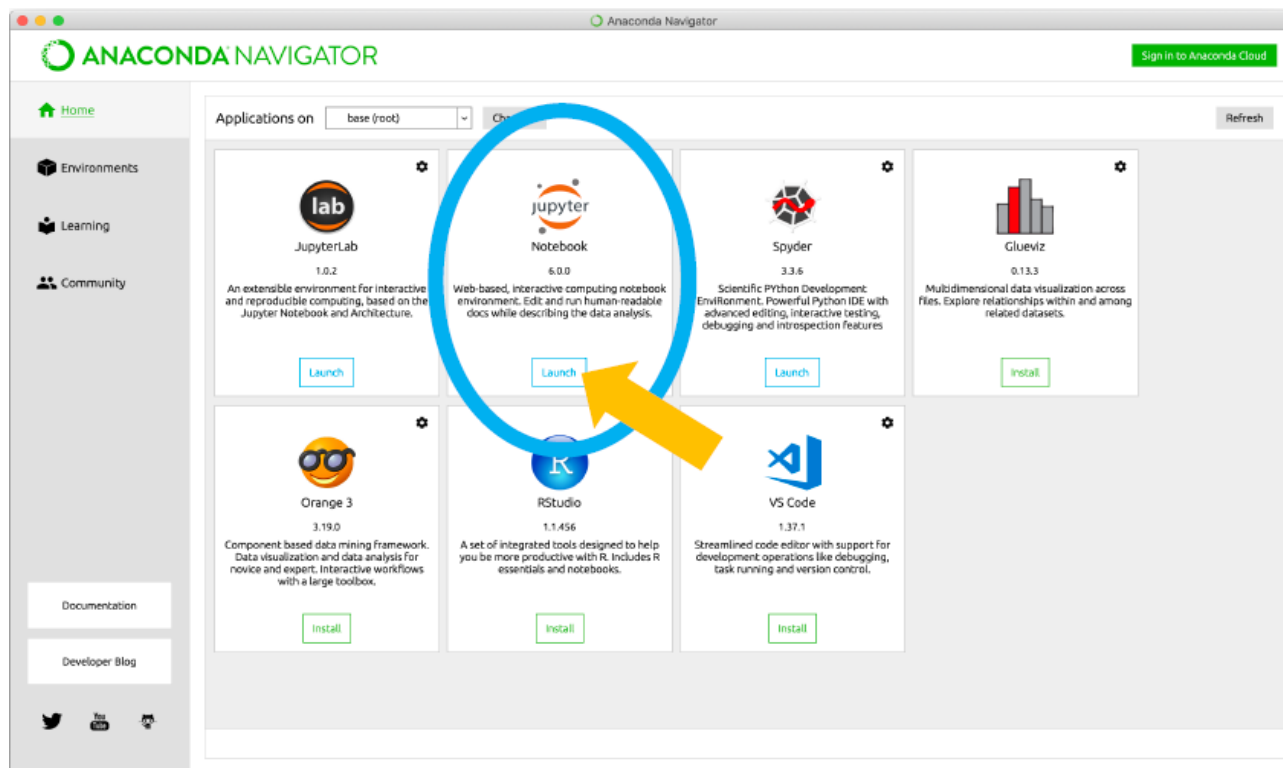
# Jupyter Notebook

There are 3 ways to launch Jupyter Notebook:

1) Using Anaconda Navigator

a) Open the application called Anaconda Navigator

b) Click on “Launch” in the Jupyter Notebook box

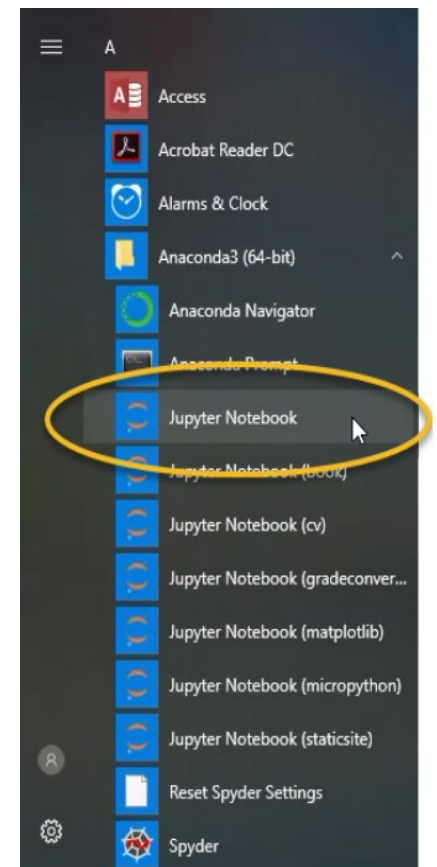


# Jupyter Notebook

## 2) Using Anaconda Prompt

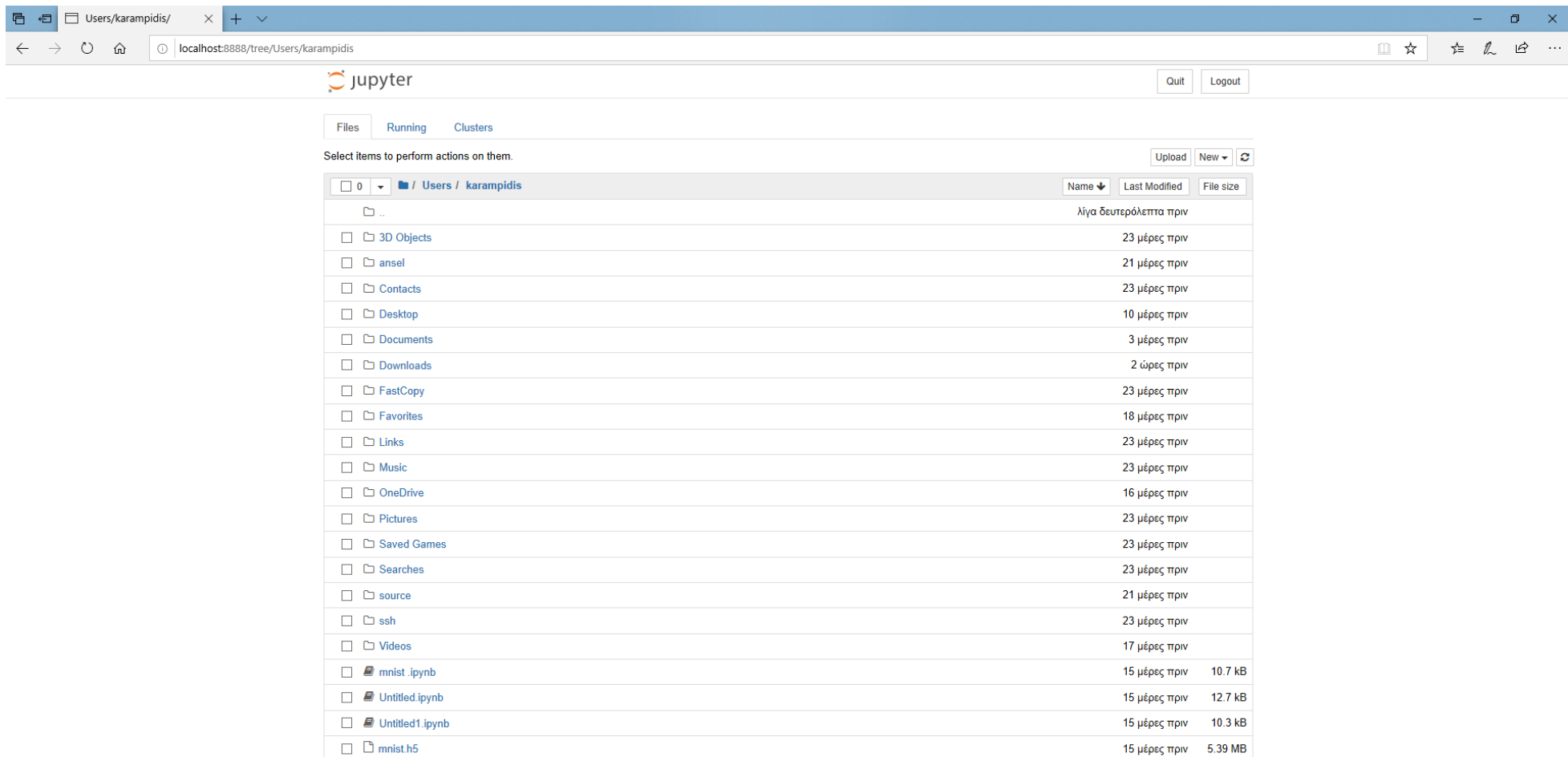
- a) Open the application called Anaconda Prompt
- b) Type “jupyter notebook” (without quotes) and hit the return key

## 3) Clicking on the Jupyter Notebook App in the Start Menu



# Jupyter Notebook

You must see a page similar to this



The screenshot shows the Jupyter Notebook web interface in a browser window. The address bar indicates the URL is `localhost:8888/tree/Users/karampidis`. The interface includes a top navigation bar with 'Files', 'Running', and 'Clusters' tabs. Below this, there's a section for file management with 'Upload', 'New', and 'Refresh' buttons. The main area displays a directory tree for the user 'karampidis'. The tree shows a list of folders and files, including standard system folders like 'Desktop', 'Documents', and 'Downloads', as well as user-specific files like 'mnist.ipynb' and 'mnist.h5'.

| Name            | Last Modified          | File size |
|-----------------|------------------------|-----------|
| ..              | λίγα δευτερόλεπτα πριν |           |
| 3D Objects      | 23 μέρες πριν          |           |
| ansel           | 21 μέρες πριν          |           |
| Contacts        | 23 μέρες πριν          |           |
| Desktop         | 10 μέρες πριν          |           |
| Documents       | 3 μέρες πριν           |           |
| Downloads       | 2 ώρες πριν            |           |
| FastCopy        | 23 μέρες πριν          |           |
| Favorites       | 18 μέρες πριν          |           |
| Links           | 23 μέρες πριν          |           |
| Music           | 23 μέρες πριν          |           |
| OneDrive        | 16 μέρες πριν          |           |
| Pictures        | 23 μέρες πριν          |           |
| Saved Games     | 23 μέρες πριν          |           |
| Searches        | 23 μέρες πριν          |           |
| source          | 21 μέρες πριν          |           |
| ssh             | 23 μέρες πριν          |           |
| Videos          | 17 μέρες πριν          |           |
| mnist.ipynb     | 15 μέρες πριν          | 10.7 kB   |
| Untitled.ipynb  | 15 μέρες πριν          | 12.7 kB   |
| Untitled1.ipynb | 15 μέρες πριν          | 10.3 kB   |
| mnist.h5        | 15 μέρες πριν          | 5.39 MB   |

# Jupyter Notebook

- Navigate through your folders until you get to the directory you want to save your scripts in.
- Open a new Notebook file by clicking on the “New” menu on the upper right

[Quit](#)[Logout](#)[Files](#)[Running](#)[Clusters](#)

Select items to perform actions on them.

[Upload](#)[New](#)

☐ 0 [/ Users / karampidis](#)

Name

λίγα

..

☐ [3D Objects](#)

☐ [ansel](#)

☐ [Contacts](#)

☐ [Desktop](#)

☐ [Documents](#)

Notebook:

**Python 3**

Other:

[Text File](#)

[Folder](#)

[Terminal](#)

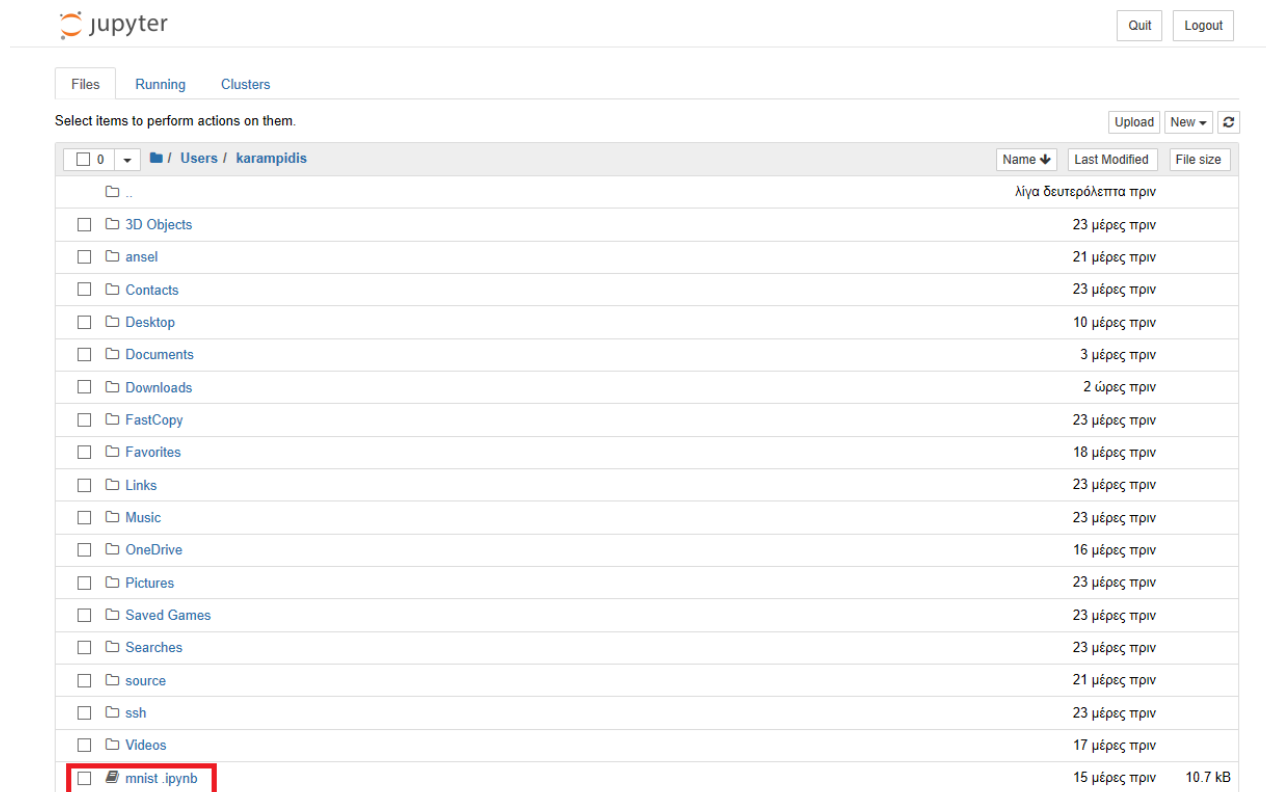
10 μέρες πριν

3 μέρες πριν

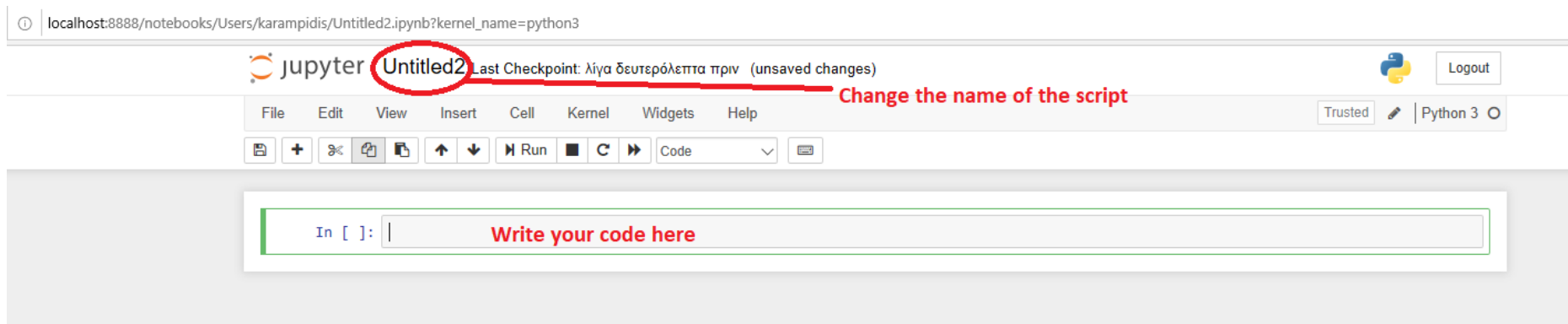
# Jupyter Notebook

Open a previously saved Notebook file by clicking on the name of the file

The extension for a Jupyter Notebook file is “.ipynb”, which is short for “interactive python notebook”

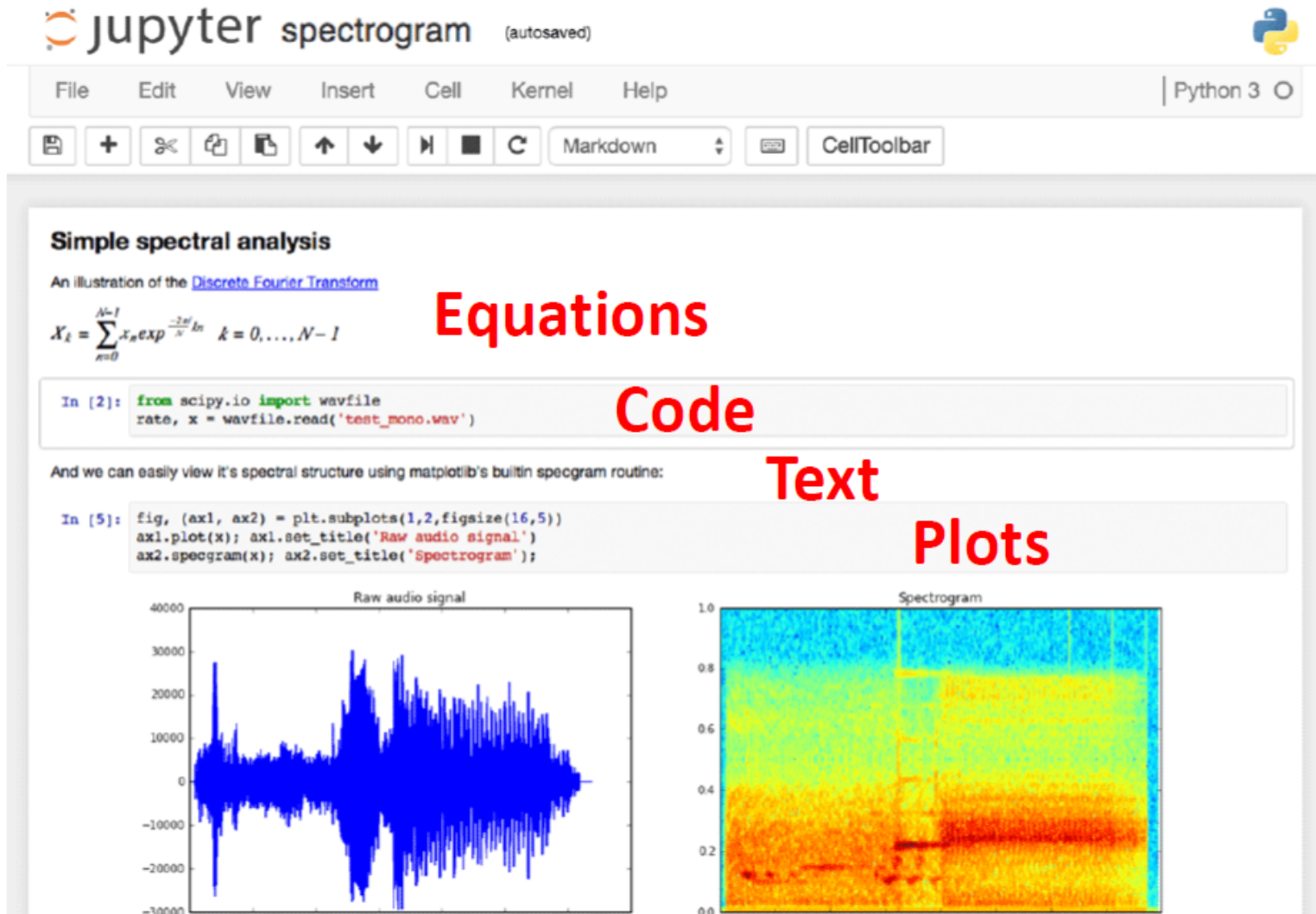


# Jupyter Notebook

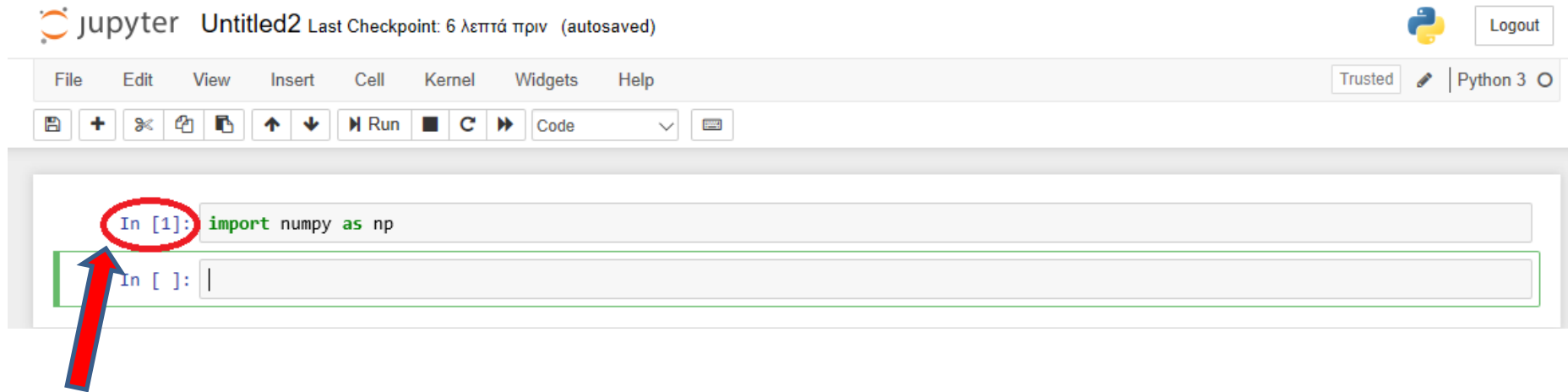


In a jupyter notebook you can have equations, text , code , plots etc.

# Jupyter Notebook



# Jupyter Notebook



If there is a number , then the cell has been evaluated! This means that the code written in that cell has been run. A cell is evaluated by either clicking “Run” or by typing Shift+Return.

# Jupyter Notebook

In order to work flawlessly this semester you need to:

- Create a new environment
- Select the new environment and install these packages:
  - numpy
  - matplotlib
  - pandas
  - scipy
  - scikit-learn
  - yellowbrick

Then you have to install jupyter notebook on this environment. The same procedure can be performed through Anaconda prompt.

```
conda create --name pr
conda activate pr
conda install numpy matplotlib pandas seaborn scipy
pip install -U scikit-learn
pip install yellowbrick
conda install -c conda-forge notebook
```

# Google Colab

Alternatively, you can use Google Colab (connect through your institutional account). To upload files to Colab use the following code:

```
from google.colab import files  
uploaded = files.upload()
```

Afterwards you can read the uploaded file as a dataframe:

```
df = pd.read_excel('file.xlsx') or  
df = pd.read_csv('file.csv')
```

To run any command line script in Colab, just add a ! **preceding** the line.

```
! pip install tensorflow
```

! denotes that what follows is a command line command and not a python code.

**Thank you**

**Questions ?**