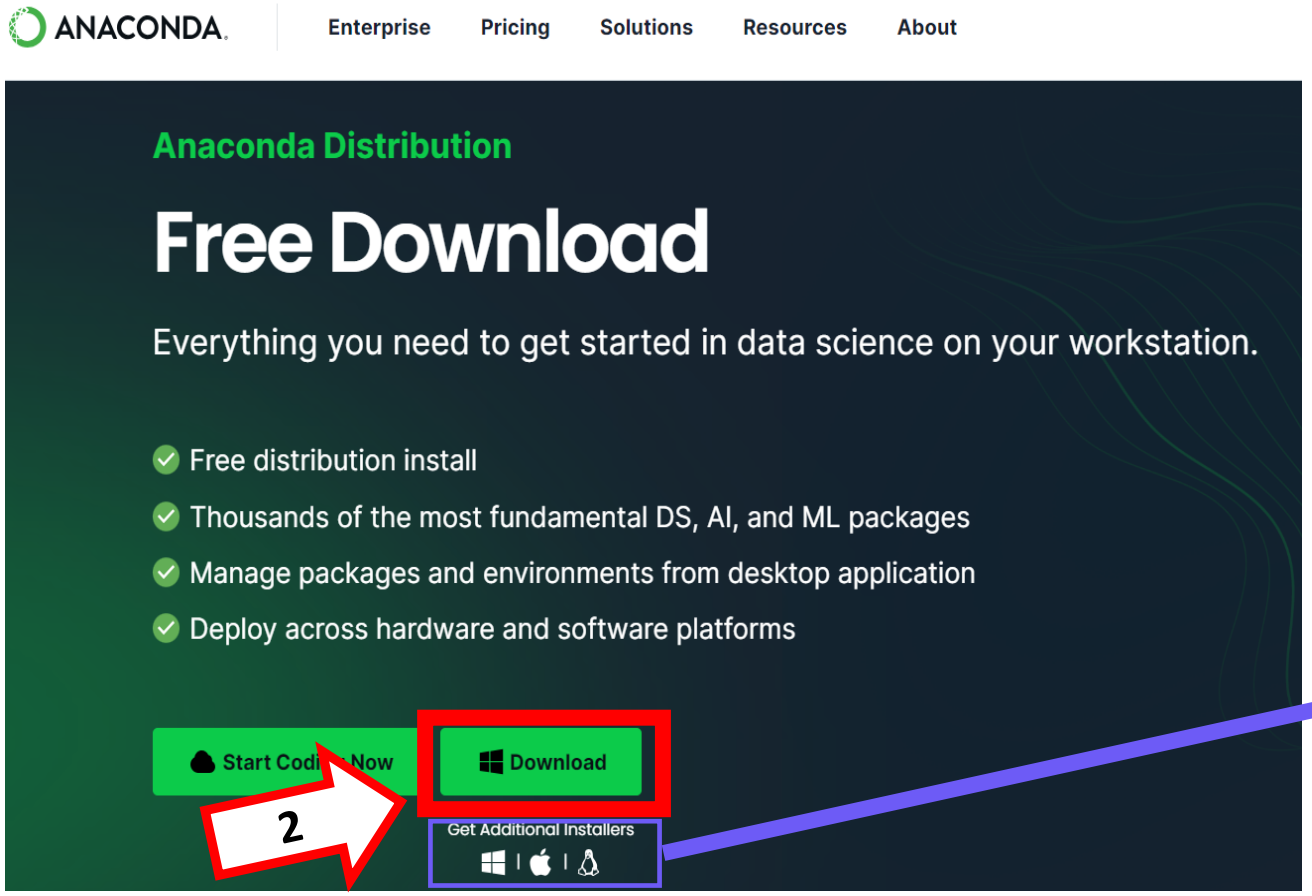
Anaconda & PyTorch

Quick Installation Tutorial

1. Download and install Anaconda

1

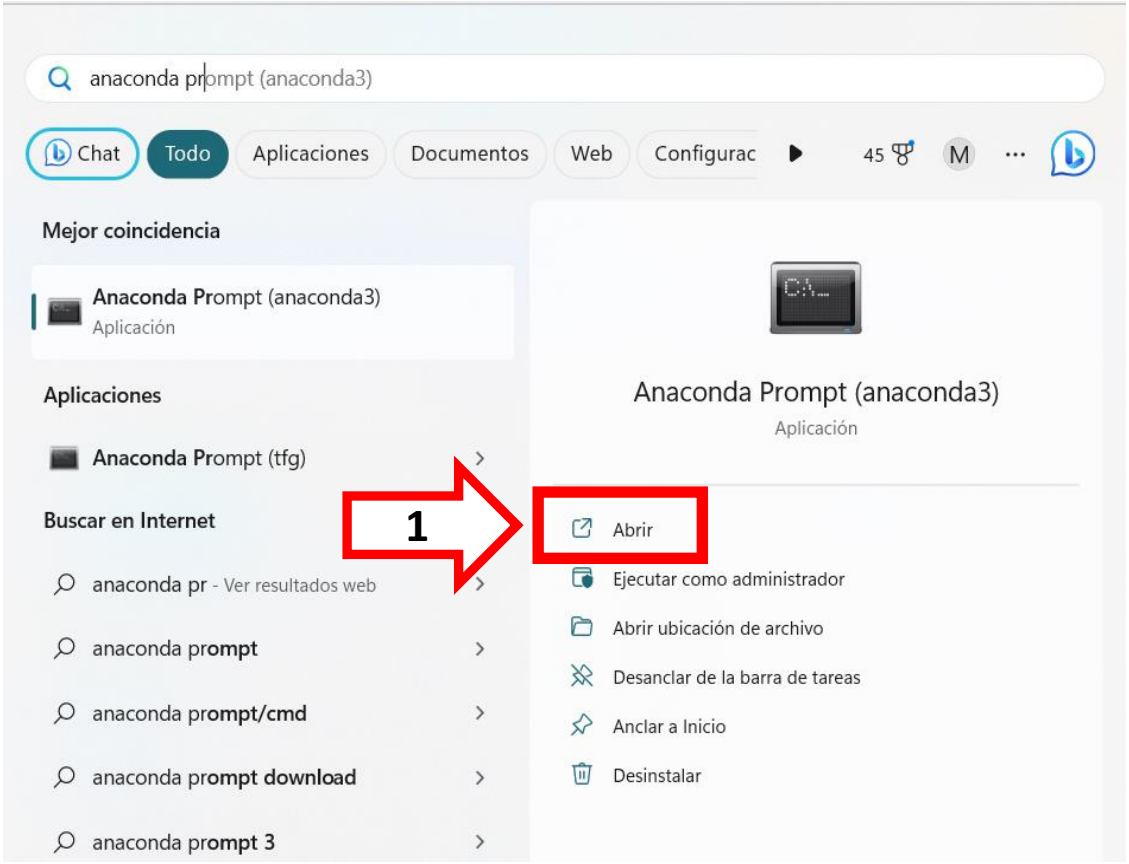
Visit the website:
https://www.anaconda.com/

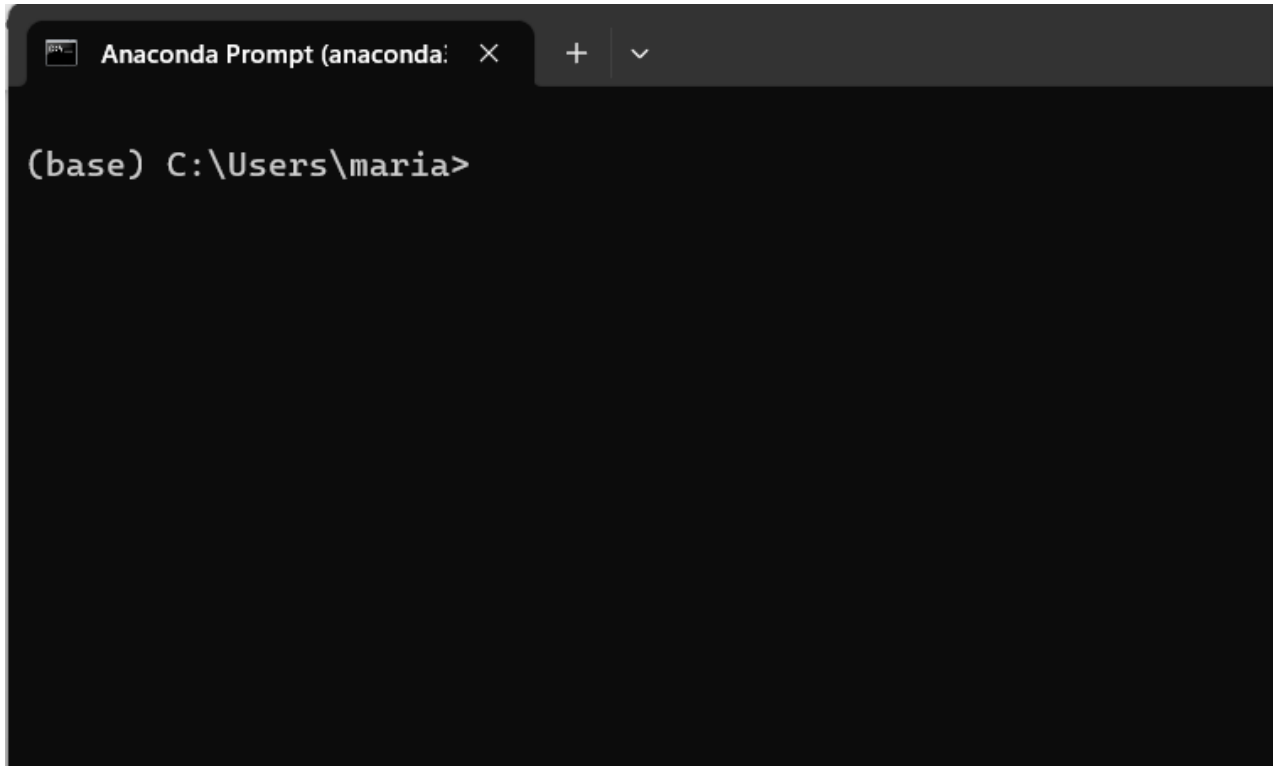


Available Anaconda distribution installation packages across platforms and processor type:

2. Set up Anaconda environment

Step 1: Open Anaconda prompt:





2. Set up Anaconda environment

Step 2: Create a new environment

- Run the following command to create a new environment with a specific Python version:

```
conda create -n env_name python=python_version
```

```
(base) C:\Users\maria>conda create -n ttt_dl python=3.10.10
```

- Display the names of the existing environments. Apart from the default environment called base, verify that the newly created one appears in the list:

```
conda env list
```

```
(base) C:\Users\maria>conda env list
# conda environments:
#
base                  *  C:\Users\maria\anaconda3
ttt_dl                   C:\Users\maria\anaconda3\envs\ttt_dl
```

To remove the environment:

```
conda env remove -n env_name
```

2. Set up Anaconda environment

Step 3: Activate your environment

- Switch to the new environment. After entering the command, you should see that **(base)** has now changed to **(your environment name).** This means that any commands entered from this point on will be executed within that environment:

```
conda activate env_name
```

```
(base) C:\Users\maria>conda activate ttt_dl

(ttt_dl) C:\Users\maria>
```

For Mac and Linux you can use the command above or also:

```
source activate env_name
```

```
(base) marialuque@anya:~$ source activate ttt_dl
(ttt dl) marialuque@anya:~$
```

- Additional information about Anaconda environments can be found in this link.

To switch back to the base environment:

```
conda deactivate
```

```
(ttt_dl) C:\Users\maria>conda deactivate

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

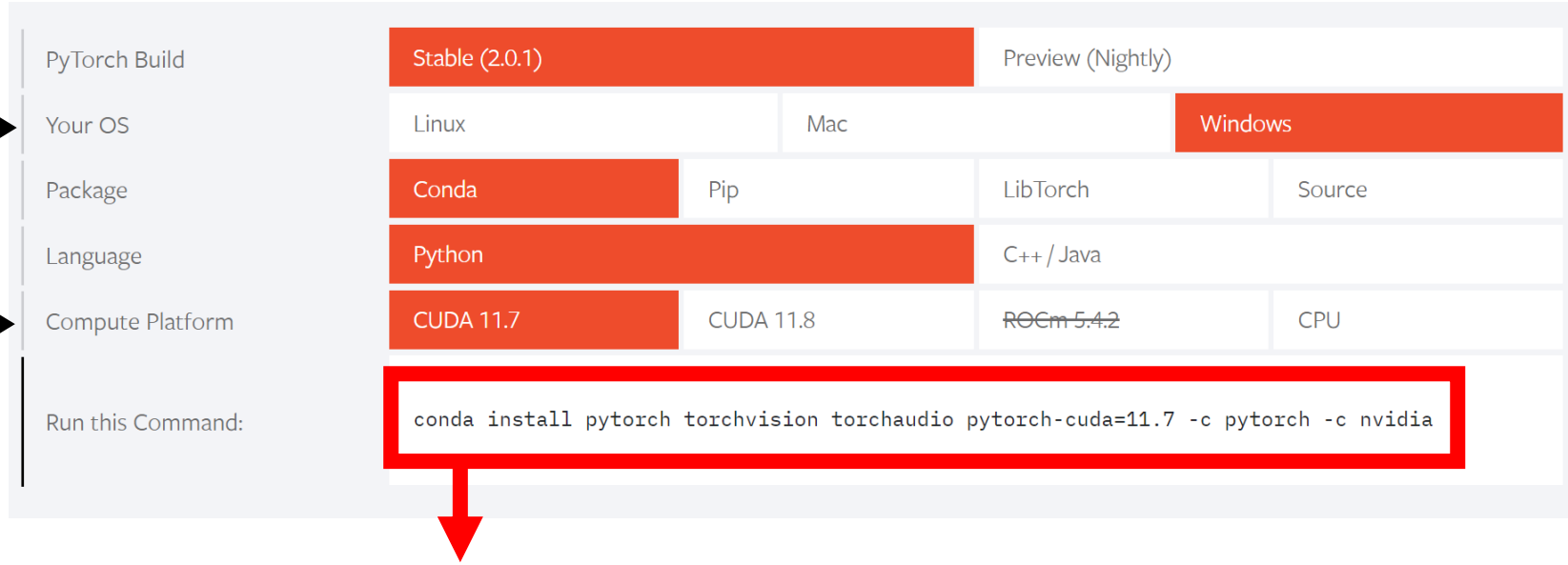
2. Set up Anaconda environment

Step 4: Install PyTorch

- Visit the website: https://pytorch.org/ and scroll down until you find the section “Install PyTorch”. Specify the appropriate configuration options for your particular device.

Select your OS

Select the cuda version compatible with your NVIDIA GPU. If you don’t have a supported NVIDIA GPU, choose the option “CPU” or “Default”



- Run the presented command in the terminal to install PyTorch

```
(ttt_dl) C:\Users\maria>conda install pytorch torchvision torchaudio pytorch-cuda=11.7 -c pytorch -c nvidia
```

- Wait until the installation is finished. It may take a few minutes

2. Set up Anaconda environment

- Verify that PyTorch is installed and check the version:

```
conda list pytorch
```

```
(ttt_dl) C:\Users\maria>conda list pytorch
# packages in environment at C:\Users\maria\anaconda3\envs\ttt_dl:
#
# Name                    Version                   Build  Channel
pytorch                   2.0.1           py3.10_cuda11.7_cudnn8_0    pytorch
pytorch-cuda              11.7                 h16d0643_5    pytorch
pytorch-mutex             1.0                        cuda    pytorch
```

Python version

cuda version

- Check if cuda is available on your system:

```
python -c "import torch; print(torch.cuda.is_available())"
```

```
(ttt_dl) C:\Users\maria>python -c "import torch; print(torch.cuda.is_available())"
True
```

Output will be **True** if cuda is enabled.

CUDA is not required to use PyTorch. However, GPU acceleration can lead to significant speed improvements, especially for deep learning tasks that involve large datasets or complex neural networks.

2. Set up Anaconda environment

Step 5: Install the required libraries

- Locate the file called "requirements.txt" in the GitHub repository and download it. In the Anaconda prompt, move to the path where the file is located and install the libraries listed.

```
cd requirements_path
```

```
pip install -r requirements.txt
```

```
(ttt_dl) C:\Users\maria>cd C:\D\TTT_DLBootcamp\resources

(ttt_dl) C:\D\TTT_DLBootcamp\resources>pip install -r requirements.txt
```

With the previous step we specify the packages and libraries that need to be installed to follow the course. In these tutorials, we will be using Jupyter Notebooks as our coding environment to facilitate interactive data analysis, code execution, and the integration of explanations and visualizations. We will also rely on a set of powerful libraries to manipulate data efficiently, such as matplotlib, scikit-learn, pandas, numpy.....

3. Getting Started with PyTorch in Jupyter Notebook

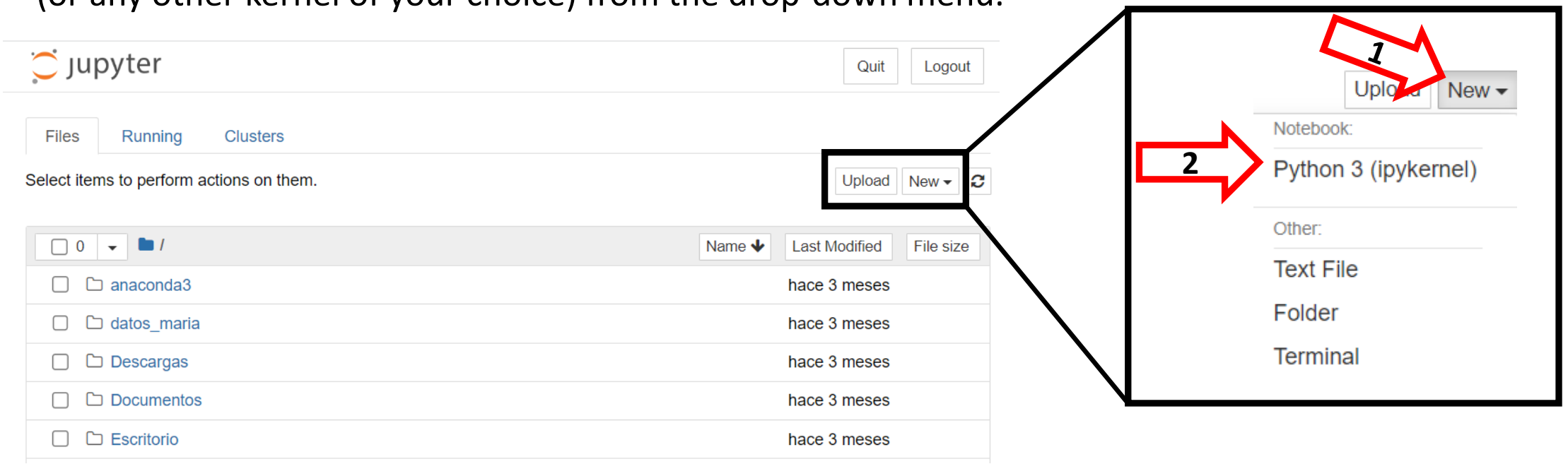
Step 1: Launch Jupyter Notebook

- To start Jupyter Notebook, open the Anaconda command prompt and run the following command:

```
jupyter notebook
```

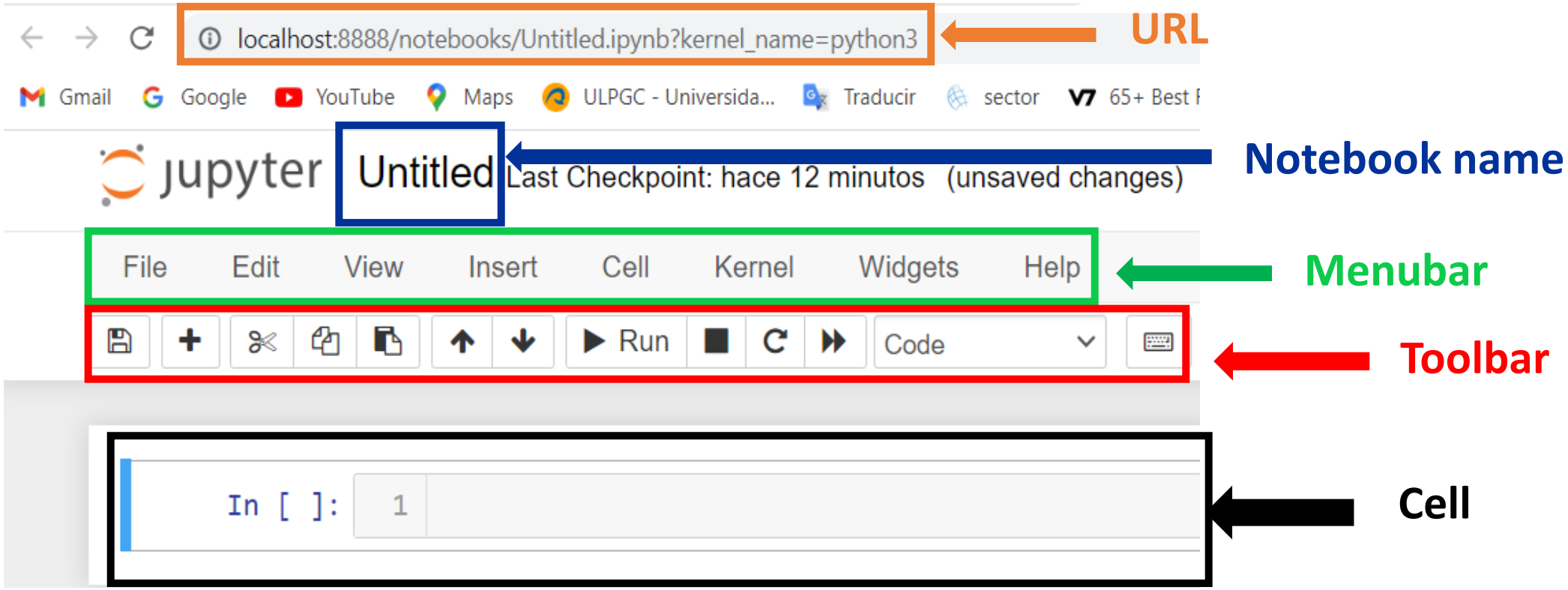
```
(ttt_dl) C:\Users\maria>jupyter notebook
```

- After entering the command above, a Jupyter notebook server should launch via your default browser. To create a new notebook, click on the "New" button located at the top right corner and select "Python 3" (or any other kernel of your choice) from the drop-down menu.



3. Getting Started with PyTorch in Jupyter Notebook

Step 2: Familiarize yourself with the UI



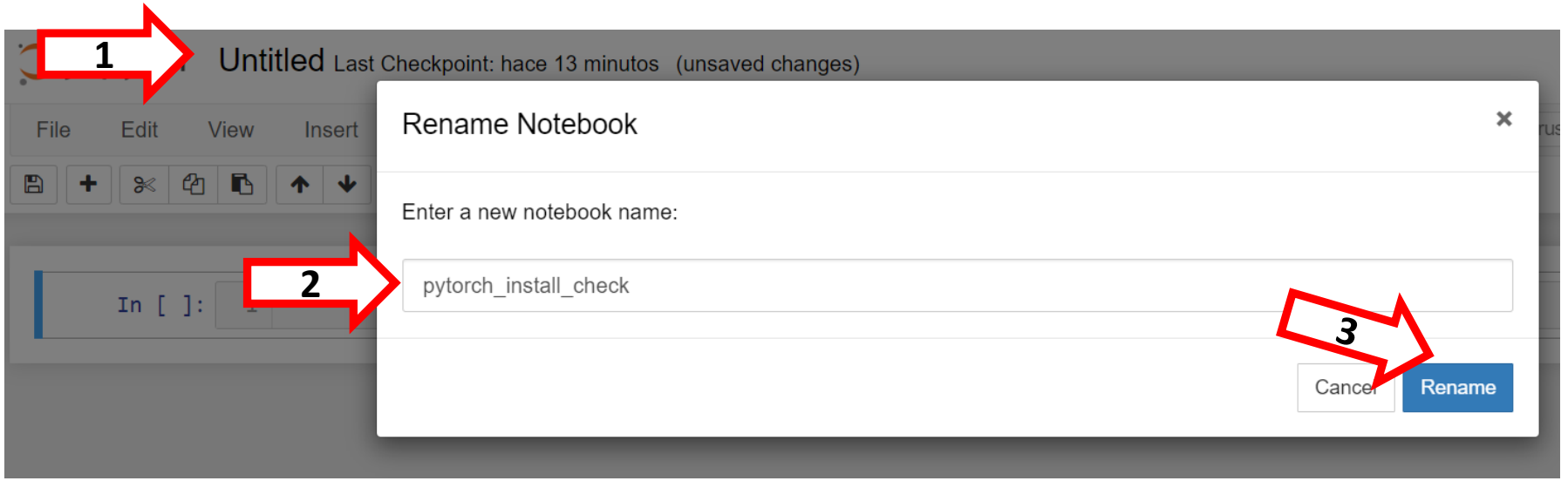
- If you need additional information about Jupyter Notebook, check out this link.

3. Getting Started with PyTorch in Jupyter Notebook

- Toolbar options:

Save	Add a new cell bellow	Delete the selected cell	Copy the selected cell	Paste the duplicated cell bellow	Move the location of a selected cell	Execute the cell and select the cell below	Interrupt the execution	Restart kerel	Restart kerel and re-run the entire notebook	Select the type of cell

- Give your notebook a meaningful name by clicking on the "Untitled" text at the top of the page and entering a desired name.



3. Getting Started with PyTorch in Jupyter Notebook

Step 3: Edit your notebook

- Run this simple code to start working with PyTorh in Jupyter Notebook:

In [1]:
```
1 import torch
```

In [2]:
```
1 # Print the version of the PyTorch library
2 print(torch.__version__)
```

```
2.0.1
```

In [3]:
```
1 # Check CUDA availability for GPU acceleration
2 if torch.cuda.is_available():
3     print(f"CUDA version: {torch.version.cuda}")
4 else:
5     print("CUDA is not available")
```

```
CUDA version: 11.7
```

Commands:

```
import torch

torch.__version__

torch.cuda.is_available()

torch.version.cuda
```