

Setting up the environment

Conda

- Each python project may require different set of packages with different versions.
- The correct way to handle such situations is by using **environments**
- We will use a popular environment manager called **conda**
- Conda is also a package manager (allows you to install additional packages)
- Conda also downloads and installs python for you

Installing conda

- Conda installer: miniconda / anaconda
 - Miniconda: includes conda (package and environment manager) and few basic packages
 - Anaconda: includes conda, many scientific packages and a GUI
- Platform: multiple OS - 64 / 32 bit
- Python version: install conda for python 3.8 or above

Anaconda

- Anaconda is the most popular Python data science platform
- Anaconda is a distribution of the Python and R programming languages for data science and machine learning related applications
- Includes a collection of over 1,000 open source data science packages
- Package versions are managed by the package management system conda

Installing Anaconda

- Go to <https://www.anaconda.com/products/individual>
- Download the Python 3.x version



Products ▾

Pricing

Solutions ▾

Resources ▾

Partners ▾

Blog

Company ▾

Contact Sales

Individual Edition is now

ANACONDA DISTRIBUTION

The world's most popular open-source Python distribution platform

Anaconda Distribution

Download 

For Windows

Python 3.9 • 64-Bit Graphical Installer • 621 MB

Get Additional Installers

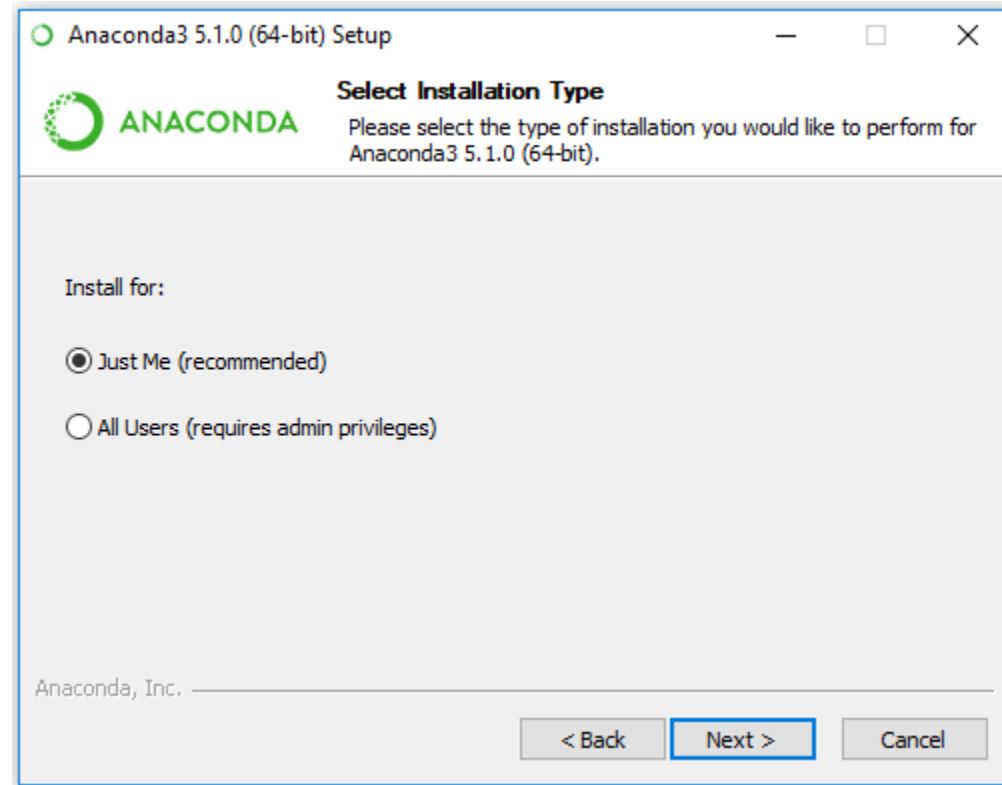


Installing Anaconda

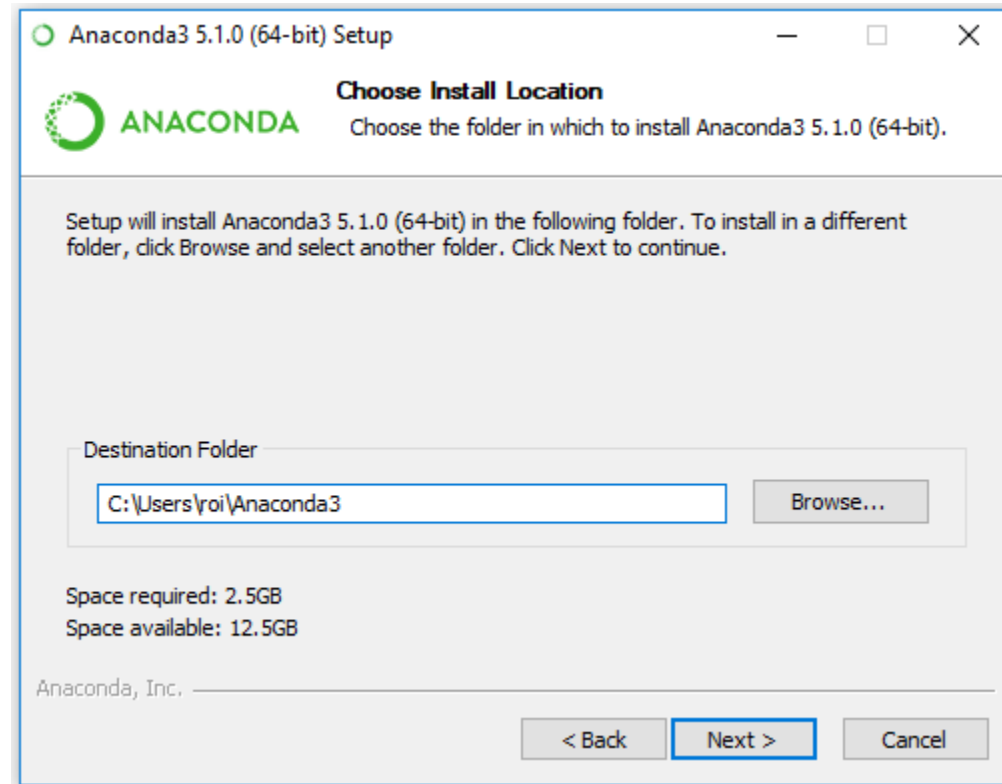
- Double click the executable file to start the installation



Installing Anaconda

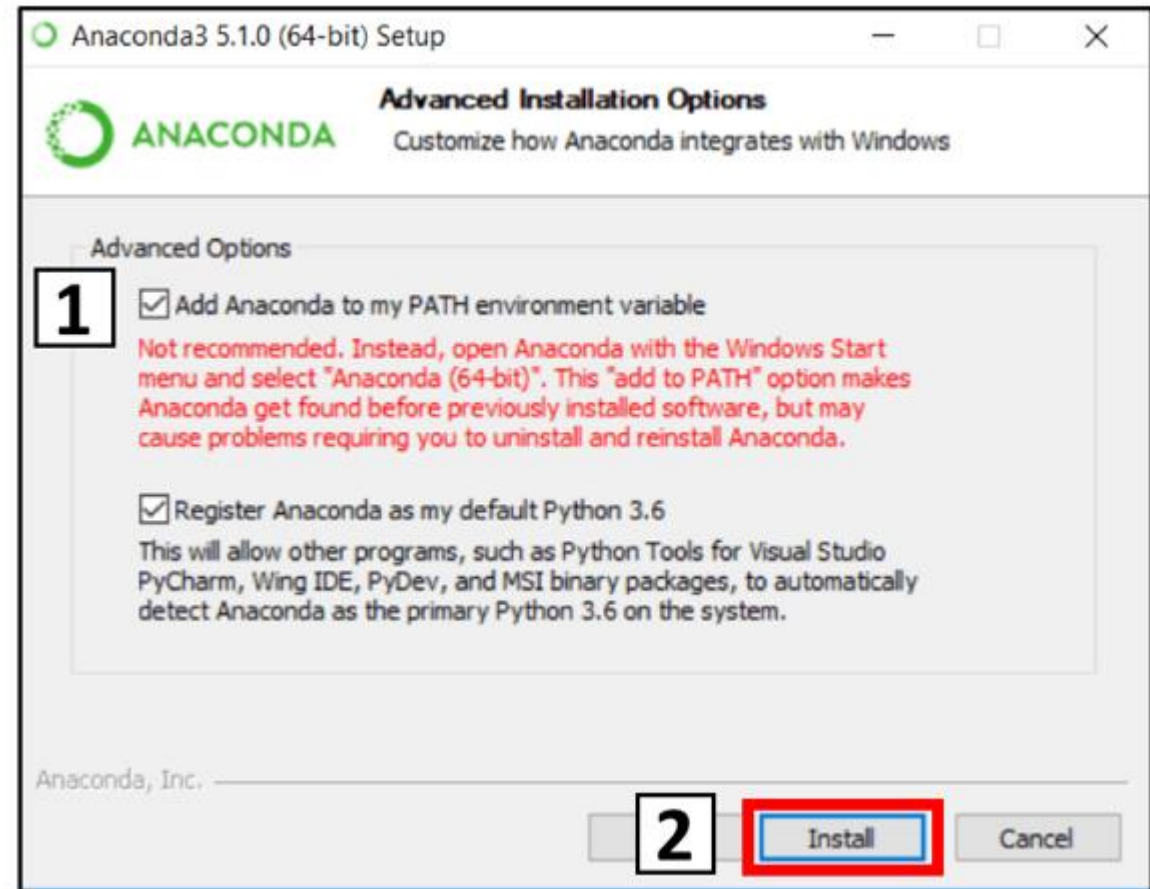


Installing Anaconda

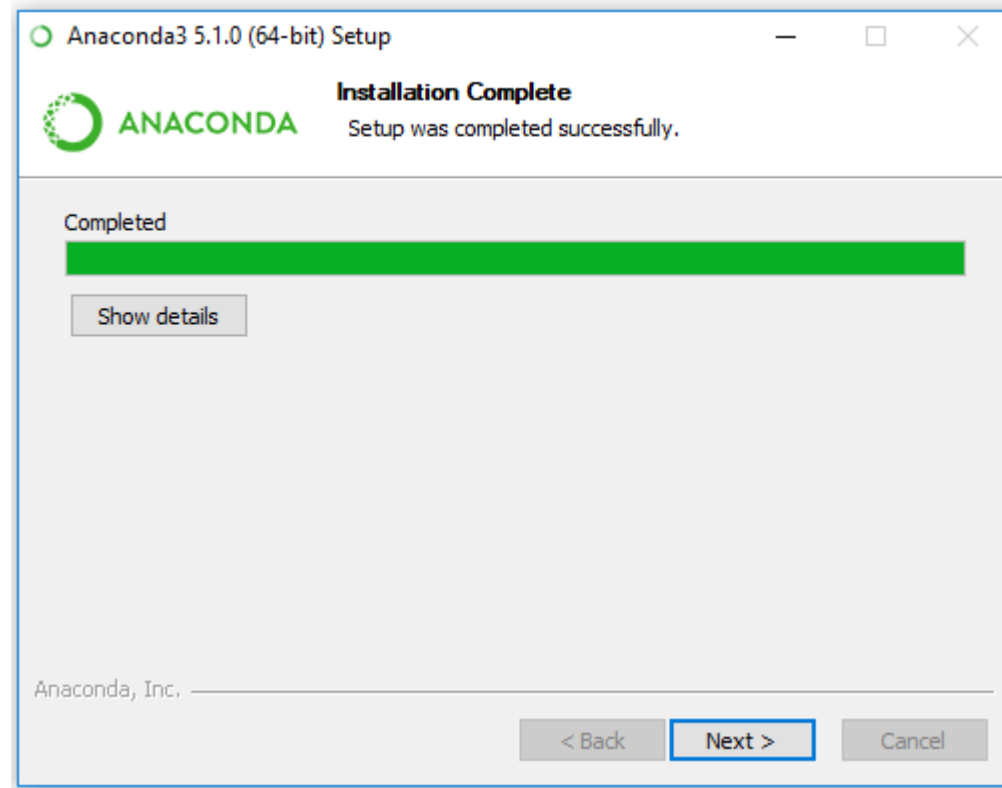


Installing Anaconda

- The first check box enables you to use Anaconda in your command prompt.

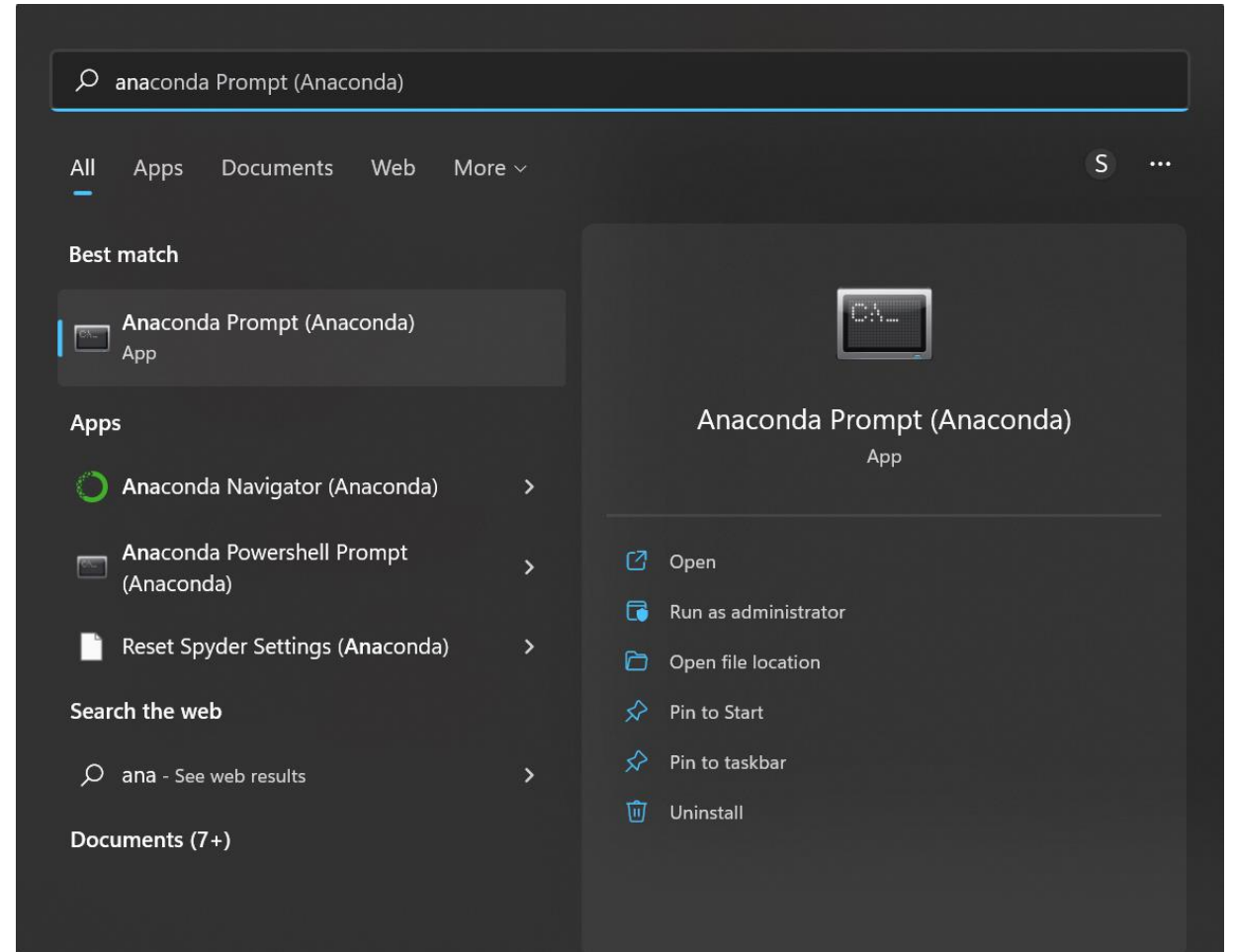


Installing Anaconda



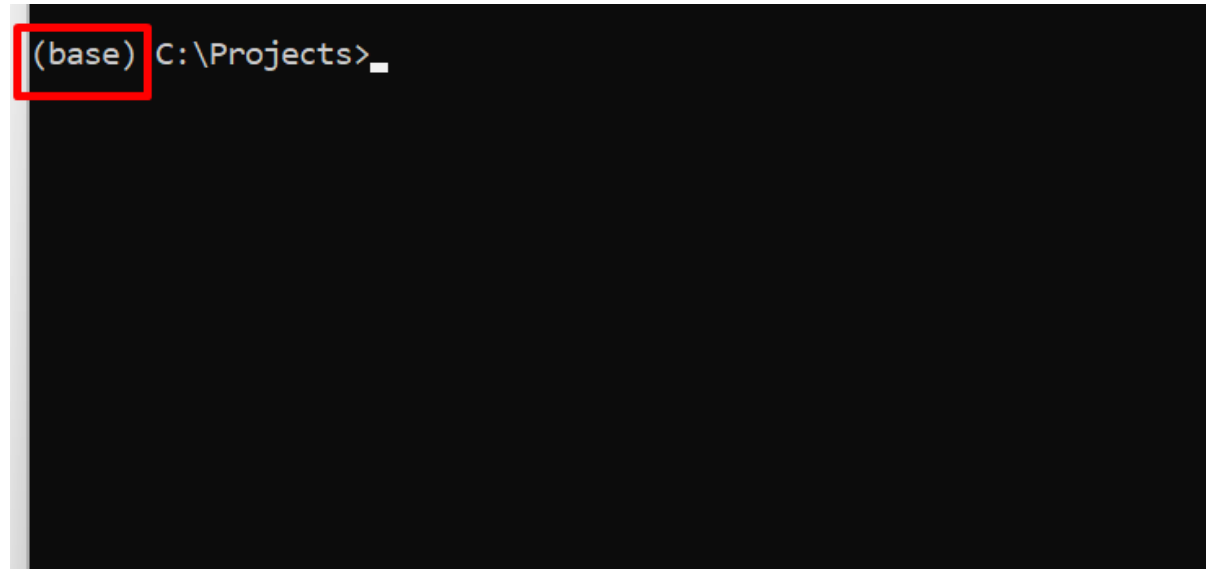
Conda

- You can open conda prompt by searching for it in the OS search tool (windows, mac, etc.)
- There are many other ways to open anaconda prompt.



Conda

- Miniconda / Anaconda sets up two things for you: **Conda** and the **root environment**
- On startup, the 'base' environment (in red) will be activated.

A screenshot of a terminal window with a black background. The prompt shows the text "(base) C:\Projects>". The text "(base)" is highlighted with a red rectangular box, indicating that the base environment is currently active. The rest of the prompt "C:\Projects>" is in white text.

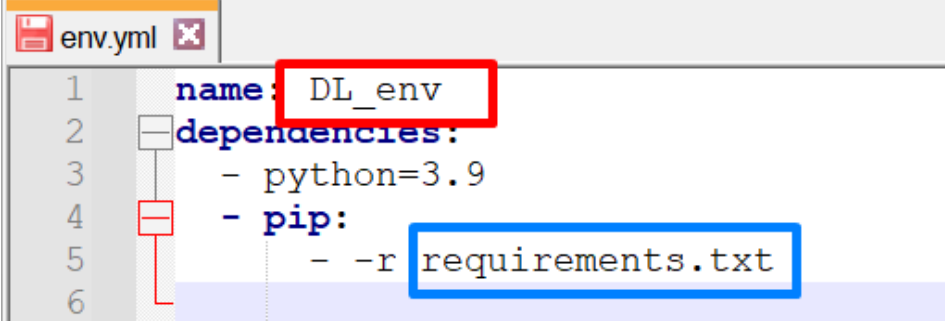
```
(base) C:\Projects>
```

Creating the course environment

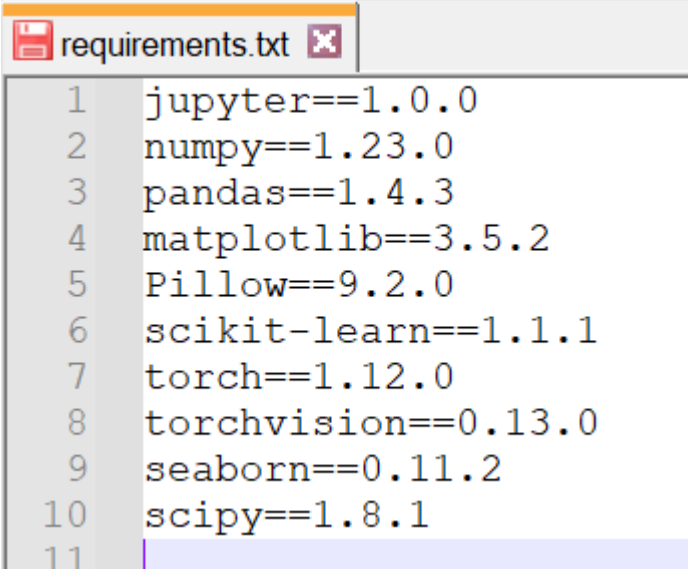
- In order to be able to execute your code and replicate your results it's important that we have consistent environment (same packages, same python version, etc.).
To this end, you'll need to create the course environment.
- You were provided with a yalm file (env.yml)
 - Configuration file
 - Specifying the required dependencies
 - We use it to create the necessary environment for the course assignments

Creating the course environment

- Naming the environment (red).
- Using python 3.9
- Installing the required packages (blue)
 - All the required packages are listed in requirements.txt file you were provided.

A screenshot of a code editor showing the 'env.yml' file. The file contains a YAML configuration for a conda environment. The 'name' field is set to 'DL_env' and is highlighted with a red box. The 'dependencies' section lists 'python=3.9' and 'pip' with a sub-dependency '-r requirements.txt'. The 'requirements.txt' file name is highlighted with a blue box. Line numbers 1 through 6 are visible on the left side of the editor.

```
1 name: DL_env
2 dependencies:
3   - python=3.9
4   - pip:
5     - -r requirements.txt
6
```

A screenshot of a code editor showing the 'requirements.txt' file. The file contains a list of data science packages and their versions, each on a new line. Line numbers 1 through 11 are visible on the left side of the editor.

```
1 jupyter==1.0.0
2 numpy==1.23.0
3 pandas==1.4.3
4 matplotlib==3.5.2
5 Pillow==9.2.0
6 scikit-learn==1.1.1
7 torch==1.12.0
8 torchvision==0.13.0
9 seaborn==0.11.2
10 scipy==1.8.1
11
```

Creating the course environment

- In order to create the environment and install all the required packages follow these steps:
 - Make sure you are connected to the internet.
 - Open the Anaconda prompt and navigate to the project folder containing both env.yml and requirements.txt files
 - Run the following command:

```
conda env create -f {your environment yaml file}
```

In our case:

```
conda env create -f env.yml
```

Creating the course environment

- This could take some time.

```
Anaconda Prompt (Anaconda) - conda env create -f env.yml
(base) C:\Projects\Deep Learning>conda env create -f env.yml
Warning: you have pip-installed dependencies in your environment file, but you do not list pip itself as one of your conda dependencies. Conda may not use the correct pip to install your packages, and they may end up in the wrong place. Please add an explicit pip dependency. I'm adding one for you, but still nagging you.
Collecting package metadata (repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.10.3
  latest version: 22.9.0

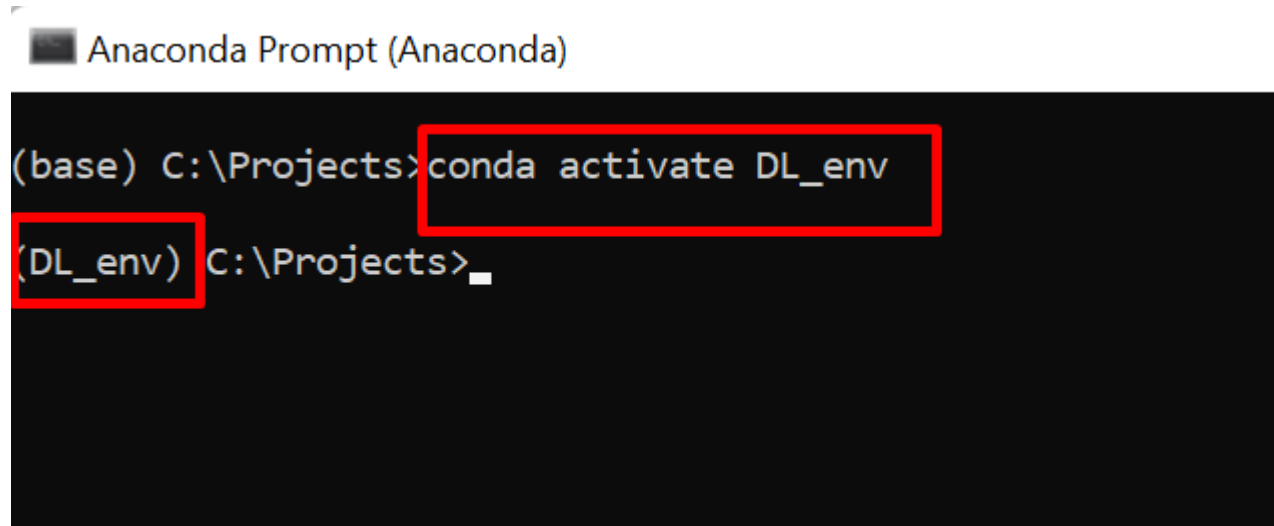
Please update conda by running

  $ conda update -n base -c defaults conda

Downloading and Extracting Packages
python-3.9.13      | 17.1 MB | ##### | 100%
pip-22.2.2        | 2.3 MB  | ##### | 100%
sqlite-3.39.3     | 804 KB  | ##### | 100%
openssl-1.1.1q    | 4.8 MB  | ##### | 100%
tzdata-2022e     | 107 KB  | ##### | 100%
certifi-2022.9.24 | 154 KB  | ##### | 100%
setuptools-63.4.1 | 1.0 MB  | ##### | 100%
ca-certificates-2022 | 123 KB  | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```


Creating the course environment

- After the installation is complete, you'll be able to activate the course environment by the command `conda activate DL_env`



```
Anaconda Prompt (Anaconda)
(base) C:\Projects>conda activate DL_env
(DL_env) C:\Projects>_
```

The screenshot shows a terminal window titled "Anaconda Prompt (Anaconda)". The prompt is "(base) C:\Projects>". The command "conda activate DL_env" is entered and highlighted with a red box. The prompt changes to "(DL_env) C:\Projects>" and is also highlighted with a red box.

Conda cheatsheet

- For other conda commands I recommend using the [conda-cheatsheet](#).
- It provides a summary of the most useful conda operations in one place.
- If you're running into technical problems, there are a lot of solutions online (stackoverflow, etc.), I highly recommend to google it.

IDE (integrated development environment)

- **Visual Studio Code**
 - **PyCharm**
 - Atom
 - Spyder
-
- I personally prefer the first two, but you are free to use any development environment / code editor you want
 - All debugging and development can be performed from Jupyter Notebook

Jupyter notebook

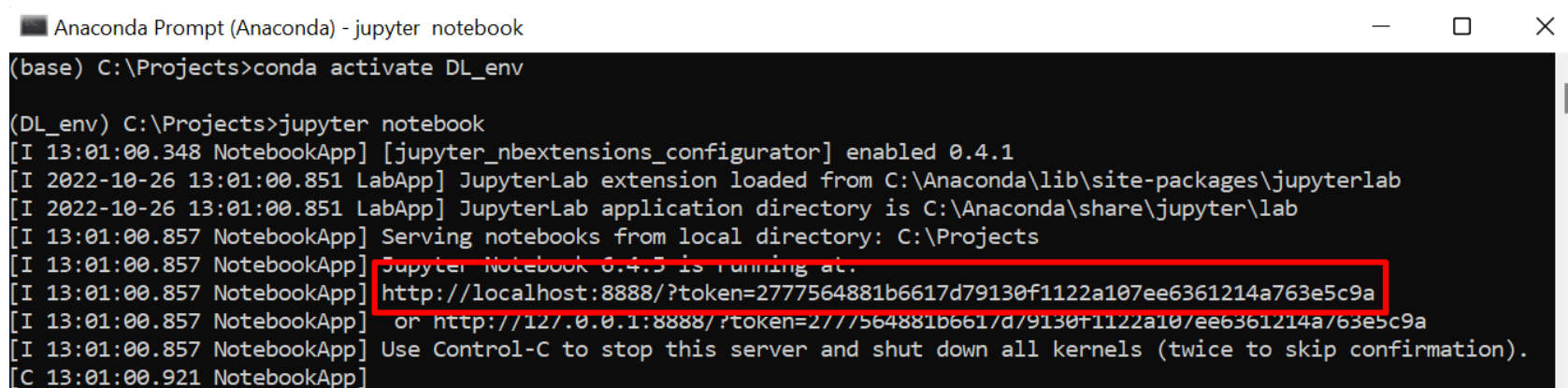
- “The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more”
- Jupyter is installed in the course environment.

Jupyter notebook

- To activate jupyter run the following command in the anaconda prompt (**make sure the course environment is activated**):

```
jupyter notebook
```

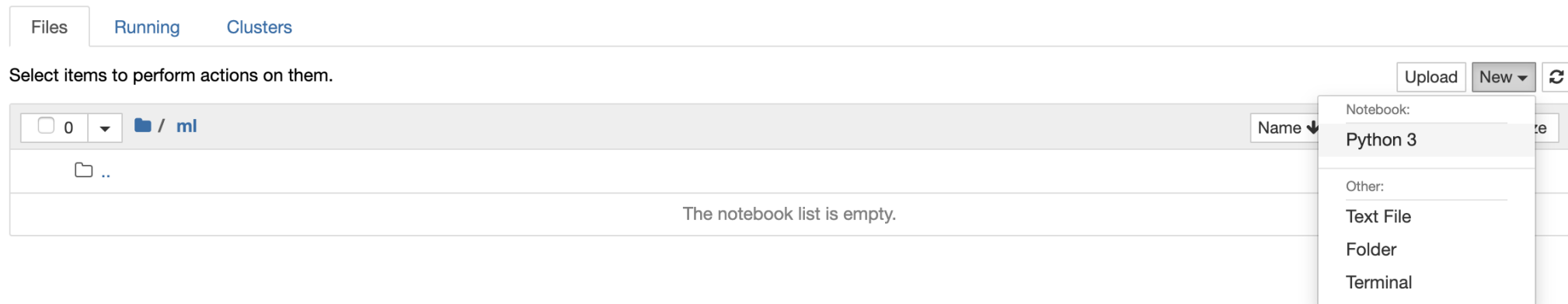
- This will create a local server on your machine which you can access from your browser, usually from `http://localhost:8888`:



```
Anaconda Prompt (Anaconda) - jupyter notebook
(base) C:\Projects>conda activate DL_env
(DL_env) C:\Projects>jupyter notebook
[I 13:01:00.348 NotebookApp] [jupyter_nbextensions_configurator] enabled 0.4.1
[I 2022-10-26 13:01:00.851 LabApp] JupyterLab extension loaded from C:\Anaconda\lib\site-packages\jupyterlab
[I 2022-10-26 13:01:00.851 LabApp] JupyterLab application directory is C:\Anaconda\share\jupyter\lab
[I 13:01:00.857 NotebookApp] Serving notebooks from local directory: C:\Projects
[I 13:01:00.857 NotebookApp] Jupyter Notebook 6.4.0 is running at:
[I 13:01:00.857 NotebookApp] http://localhost:8888/?token=2777564881b6617d79130f1122a107ee6361214a763e5c9a
[I 13:01:00.857 NotebookApp] or http://127.0.0.1:8888/?token=2777564881b6617d79130f1122a107ee6361214a763e5c9a
[I 13:01:00.857 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 13:01:00.921 NotebookApp]
```

Jupyter notebook

- The notebook should open automatically. Otherwise, copy and paste the link from the console to your browser (see red square in previous page).
- Google chrome works best.
- Create a new notebook:



Jupyter notebook

- Type the following into the first cell and press 'shift+enter' to run:
 - import numpy as np
 - import pandas as pd
 - import matplotlib.pyplot as plt
- If the cell run without errors, you should see the following

```
In [1]: import numpy as np  
import pandas as pd  
import matplotlib.pyplot as plt
```

```
In [ ]:
```

The end

- Now you can start doing the HW by open the assignment in your jupyter.
- The assignment is a jupyter notebook file with missing code that you should complete.