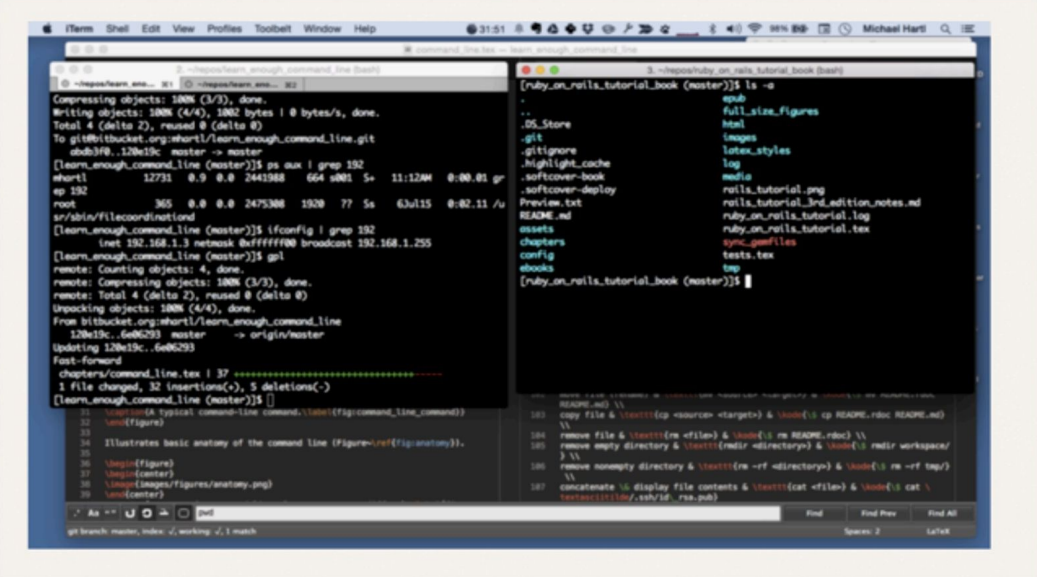
Reinforcement Learning

Setup

Introduction to Command Line

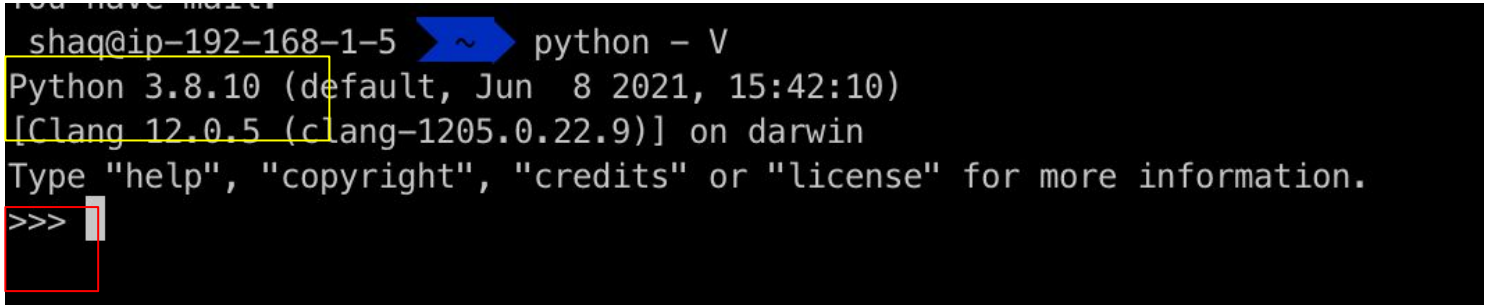
- The command line (also known as the Terminal, or Command Prompt) refers to a type of program that comes **pre-installed** with Windows, Linux, and Mac computers and allows you to execute commands, run programs and navigate through the folders on your computer.
- The command line is a **quick, powerful, text-based interface** developers use to more effectively and efficiently communicate with computers to accomplish a wider set of tasks. Learning how to use it will allow you to discover all that your computer is capable of!



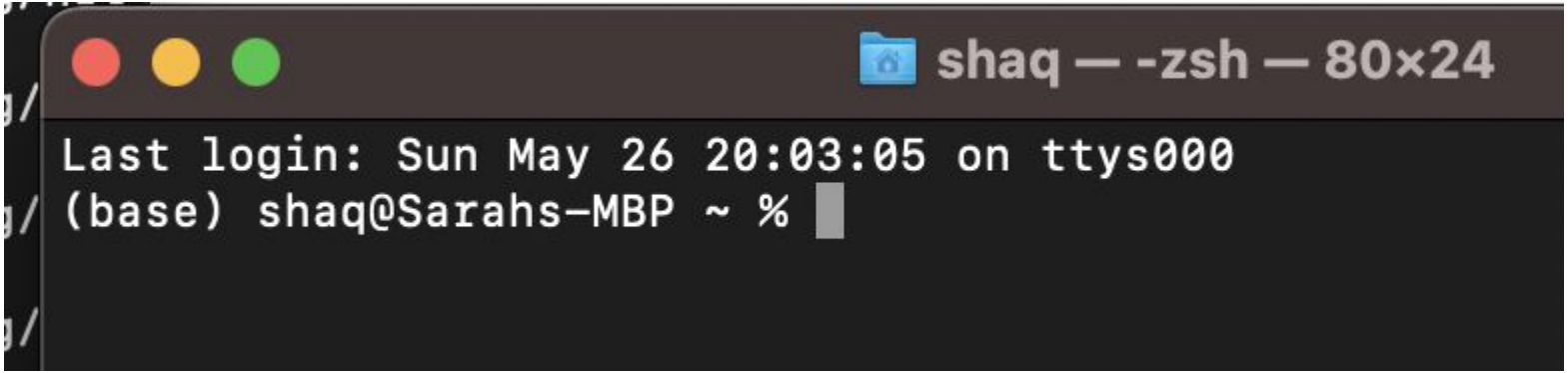
My terminal

Do you have Python Installed?

- Open your **Command Prompt or Powershell (Windows)** or **Terminal (Mac)** or ….
  - This is a **command line interpreter application,** used to execute entered commands. Most of those commands automate tasks via scripts, in our case python scripts.
- Type **python** or **python3.**
  - **What version of Python do you have?** It is ok if we have different versions.
  - If you have less than 3.7, let's upgrade your Python.
  - The `>>>` shows that we are interacting with the Python interpreter. **If you don't see that, you need to [install Python](#).**



Do you see a (base) appearing in your terminal? Otherwise please install Python with miniconda. Instructions on next slide.



Installing Python - You must install using miniconda

Please make sure you are downloading Python for your operating system

https://docs.anaconda.com/free/miniconda/miniconda-install/

Activity: Print current working directory

- Mac users.. Type `pwd`, what do you see?
- Windows users.. Type `cd`, what do you see?

`pwd / cd` - prints the current working directory

When you open your command line, this is your home directory

Activity: View the folders and files in your current working directory

- Mac/linux users.. type `ls`. What do you see?
- Windows users, type `dir`. What do you see?

`ls/dir` - Gives the list of folders and files in your directory

Activity: Change the directory to a directory of your choice

- Type cd and the name of the folder of your choice, this changes the directory to the folder of your choice. Use `cd folder-name` to change the directory.
- Type `pwd/cd`, what do you see?
- Type `ls/dir`, what do you see?

Activity: Go back to the previous directory

- Type cd and the name of the folder of your choice, to go into another directory.
- To go back to the previous directory:
  - Mac users type `cd ..`
  - Windows users type `cd..`
- Type `pwd/cd`, what do you see?

Activity: Go back to your home directory

- Mac users.. type `cd`, and then `pwd`, what do you see?
- ~~Windows users.. Type `cd ~`, and then cd, what do you see?~~

`cd` - Goes back to the home directory i.e. the location when you open the command line

Activity: Can you find a suitable location and create a reinforcement_learning folder using your command line?

Activity: Create a 01Lecture folder in your reinforcement learning folder

- Make sure you are within the reinforcement_learning directory (double check with ls/dir or pwd/cd)
- Within that folder use **`mkdir`** to create a directory. For e.g. you could create a folder called 01lecture by executing the code **`mkdir 01Lecture`**.
- Type ls/dir, do you see your new folder?

Activity: Remove a folder from your newly created folder

- Use **cd** to change directory to go into your new folder.
- Create a new folder called test. Type ls/dir to double check it got created
- You could also use `rmdir` to remove a directory.
- Type `rmdir test` to remove the test folder. Type `ls/dir` to double check it got removed.

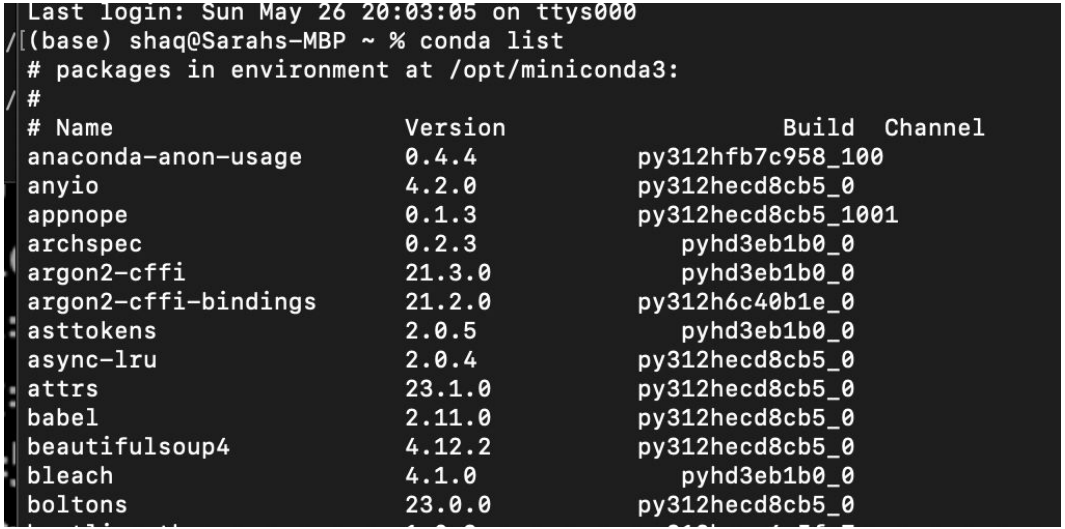
Recap: Command Line Code - 1

- **`cd` (windows) or `pwd` (mac)** stands for print working directory and it prints the "place" or directory we are currently at in the computer.
- **`ls` (mac) or `dir` (windows)** presents you the contents of the directory you're currently in. It will present you with both the files and other directories your current directory contains.
- **`cd`** is short for change directory, and it will take you from your current directory to another.
- If I want to go up one directory, meaning go to the directory that contains the current directory, you can enter (windows) **`cd..`** or (Mac) **`cd ..`**
- For mac users, if you enter **`cd`** it will take you straight to your home directory. ~~Windows users can try **`cd ~`**.~~

Recap and more: Command Line Code - 2

- **`mkdir`** stands for make directory and it will create a new directory for you. You have to pass the command the directory name parameter. If I wanted to create a new directory called "Test" would enter mkdir test.
- **`rmdir`** stands for Remove directory and it does just that. It needs the directory name parameter just as mkdir: rmdir test.
- **`touch`** allows you to create an empty file in your current directory. As parameters it takes the file name, like touch test.txt.
- **`rm`** allows you to delete files, in the same way rmdir allows you to remove directories.
  rm test.txt
- **`cp`** allows you to copy files or directories. This command takes two parameters: the first one is the file or directory you want to copy, and the second one is the destination of your copy (where do you want to copy your file/directory to).
- **`mv`** is short for move and allows us move a file or directory from one place to another. That is, create it in a new directory and delete it in the previous one (same as you could do by cutting and pasting).

Activity: Type `conda list` in your command line. What do you see? This command gives you a list of libraries you have already installed.



Installing Libraries in Python

How to install a library:

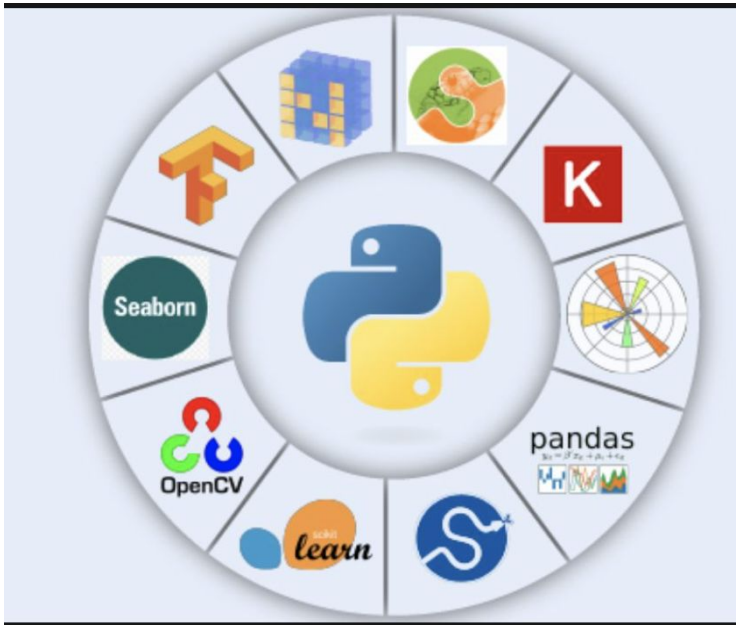
- `pip3 install` ***`package-name`***

  or

- `pip install` ***`package-name`***

  or

- `conda install` ***`package-name`***



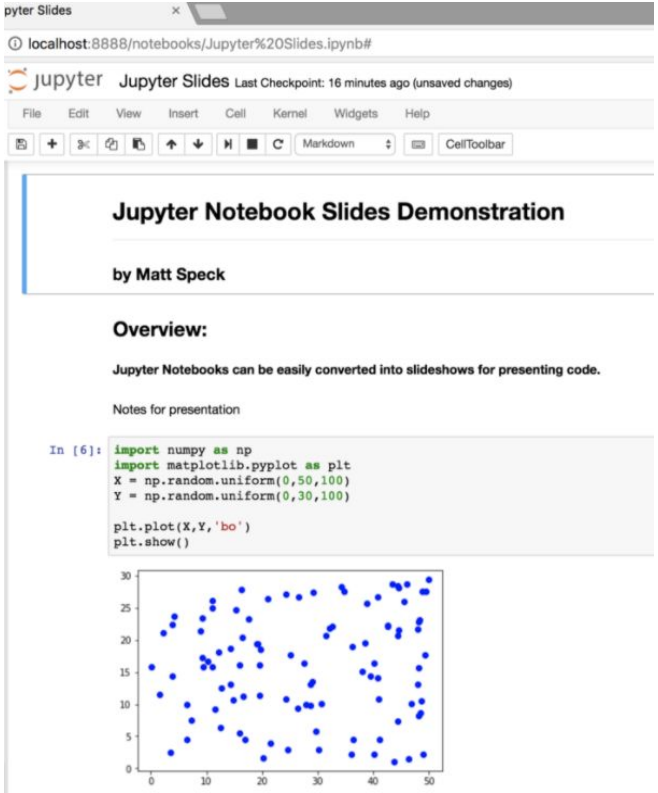
Python Libraries
Similar to apps on your phone

Activity: Can you install the Python library called jupyter?

```
conda install jupyter
```

We will work with “Jupyter Notebooks” in this course

- An **open source web application** that we can use to create and share documents that contain live code, equations, visualizations, and text.
- Jupyter Notebook is maintained by the people at Project Jupyter.
- The localhost is not a website but indicates that the content is being served from your *local* machine: your computer. Jupyter's Notebooks are web apps.



Launch your jupyter notebooks from your command line.

```
jupyter notebook
```

or

```
python3 -m notebook
```

If errors:

```
conda install jupyter
```

Installing Libraries in Python

How to install a library:

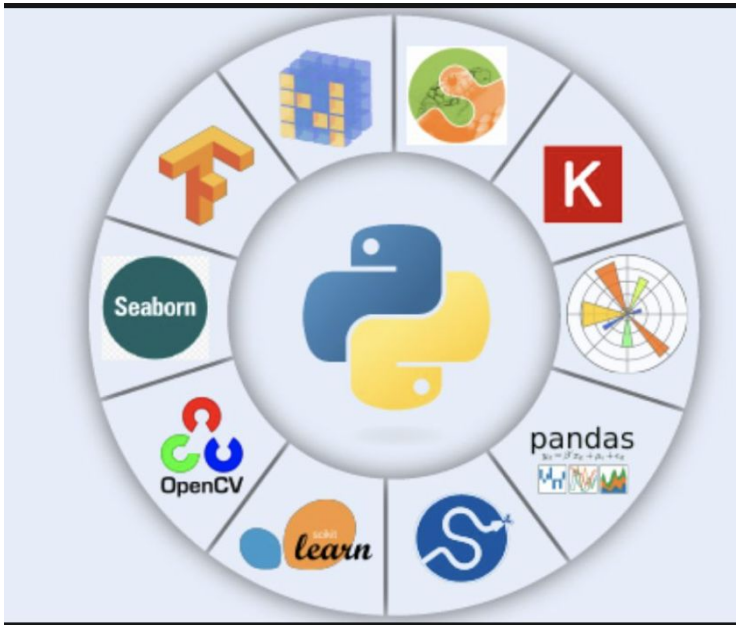
- `pip3 install` ***`package-name`***

  or

- `pip install` ***`package-name`***

  or

- `conda install` ***`package-name`***



Python Libraries
Similar to apps on your phone

The Dashboard of Jupyter Notebook

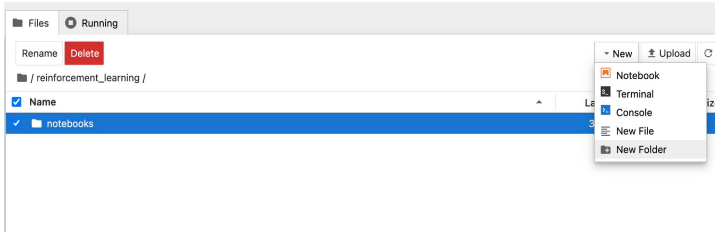
The Dashboard of Jupyter Notebook carries three tabs, which is shown in the following screenshot.

Files Tab: The **Files Tab** is applied to shows all **files** and **folders** in the current directory. It also used an **upload** button to upload a file in the notebook server, and also contain a **new** button to create a new notebook in the notebook server.

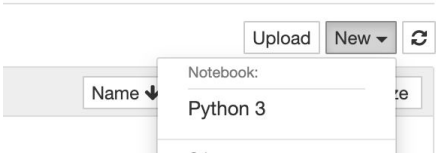
Your dashboard might look a little different, but it shows our files and folders from where you launched the notebook.

Activity: Create your notebook in a notebooks folder

- Please make sure you are in your reinforcement_learning folder.
- Create a new folder. Call it notebooks. Please lowercase it:
  - notebooks



- Open this folder and open a new jupyter notebook - this is where we will do our activities today.

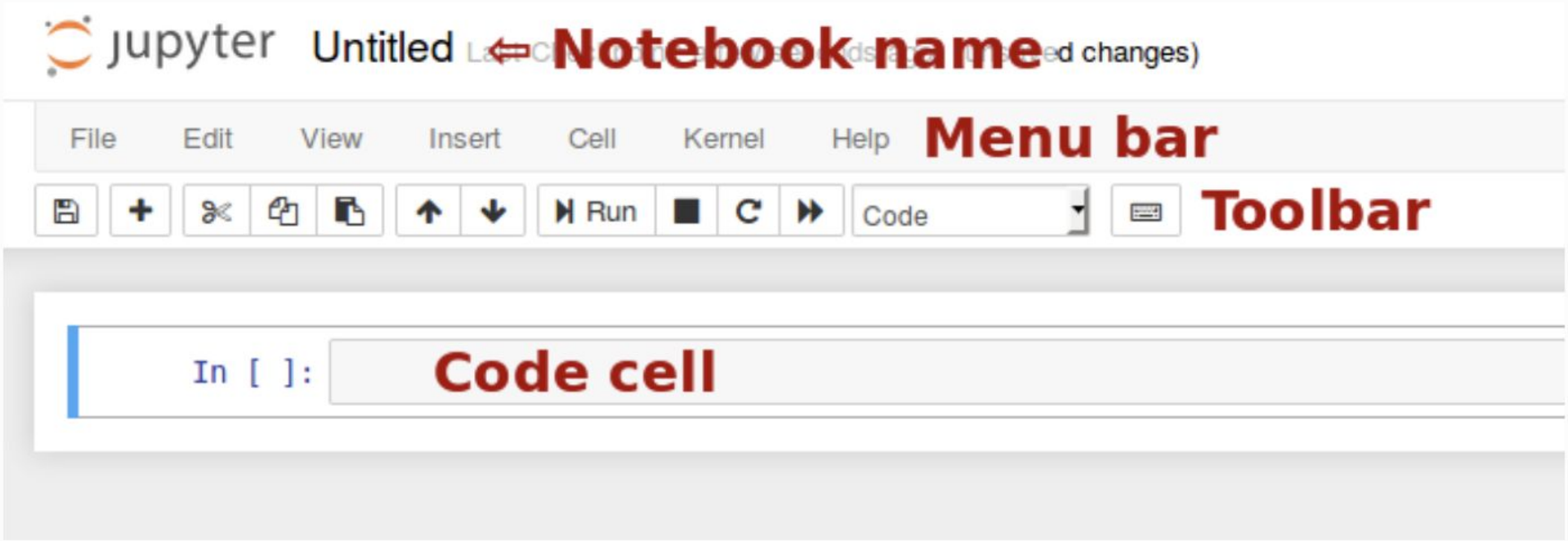


If you need more support, watch this video:

https://www.youtube.com/watch?v=9YmkldQSHE4

Notebook user interface

When you create a new notebook document, you will be presented with the **notebook name**, a **menu bar**, a **toolbar** and an empty **code cell**.



Reference: https://jupyter-notebook.readthedocs.io/en/stable/notebook.html

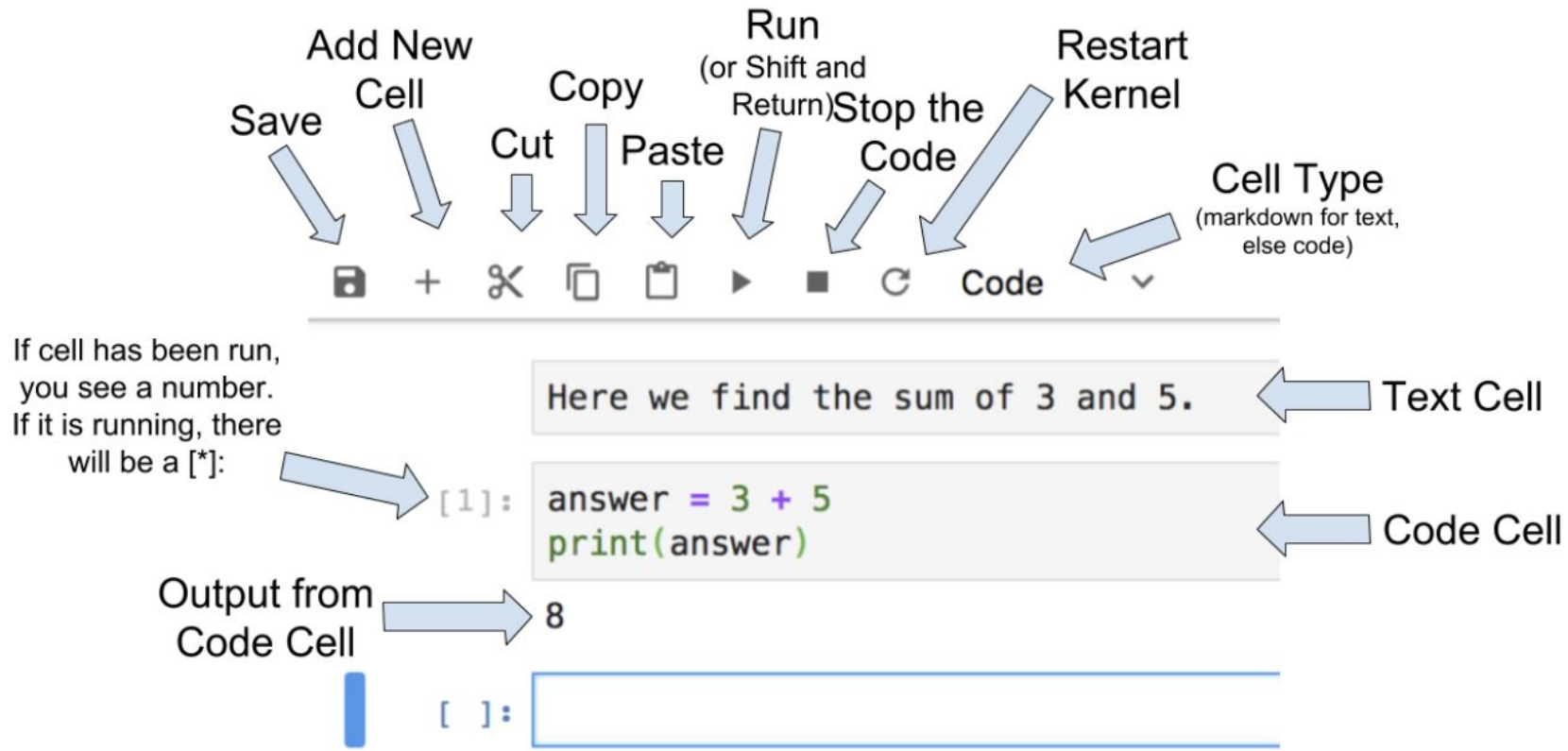
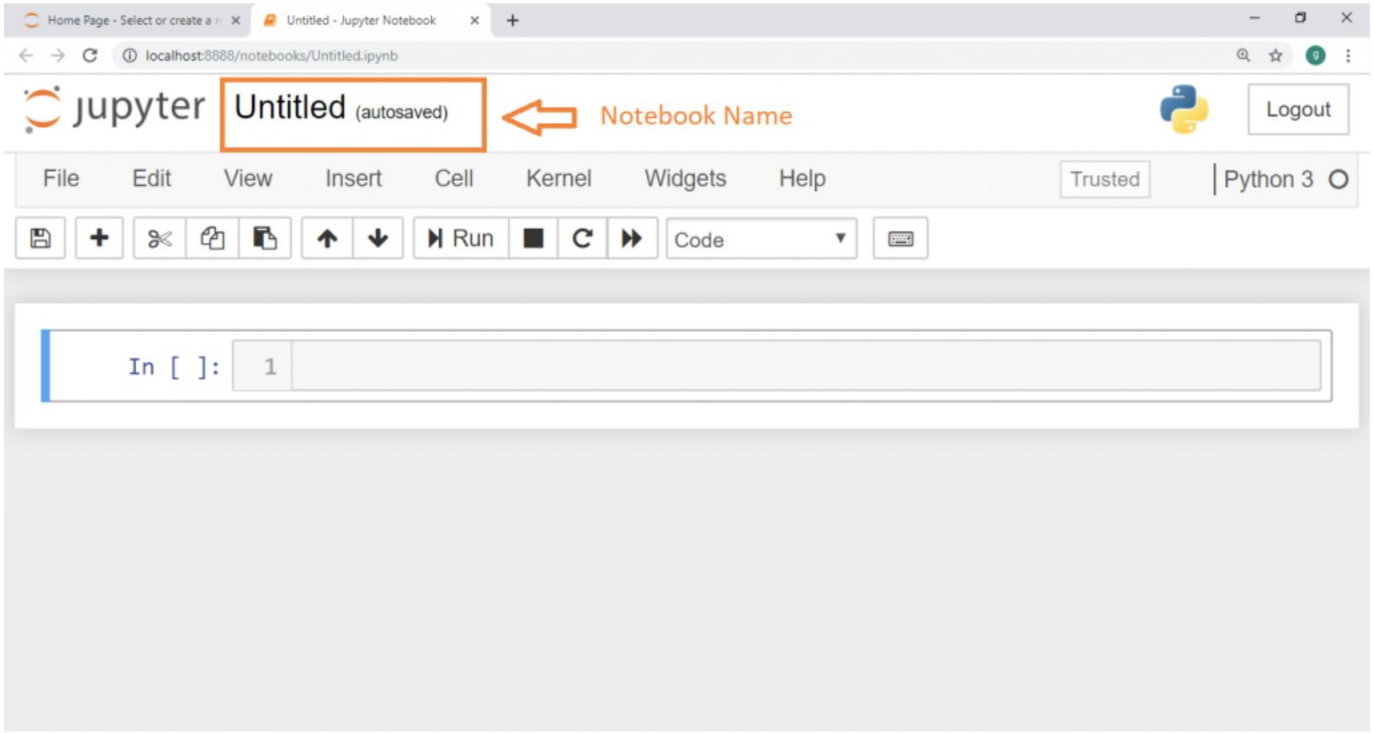


Figure 1: Anatomy of JupyterLab Notebook

Reference: https://runestone.academy/ns/books/published/httlads/MovieData/jupyter_getting_started.html

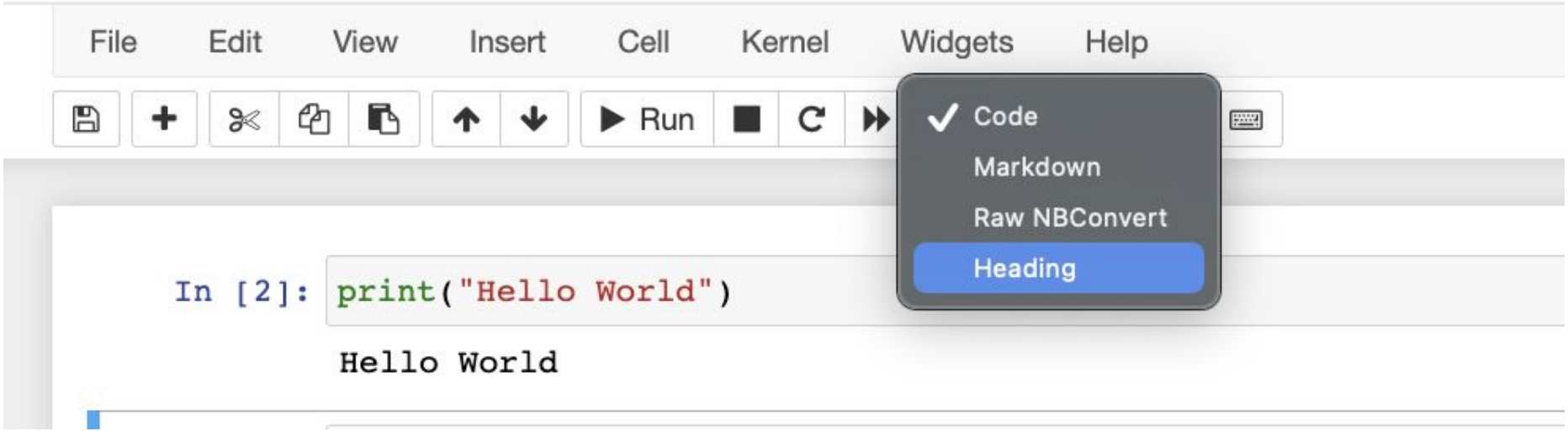
Activity: Rename your jupyter notebook to 01-intro



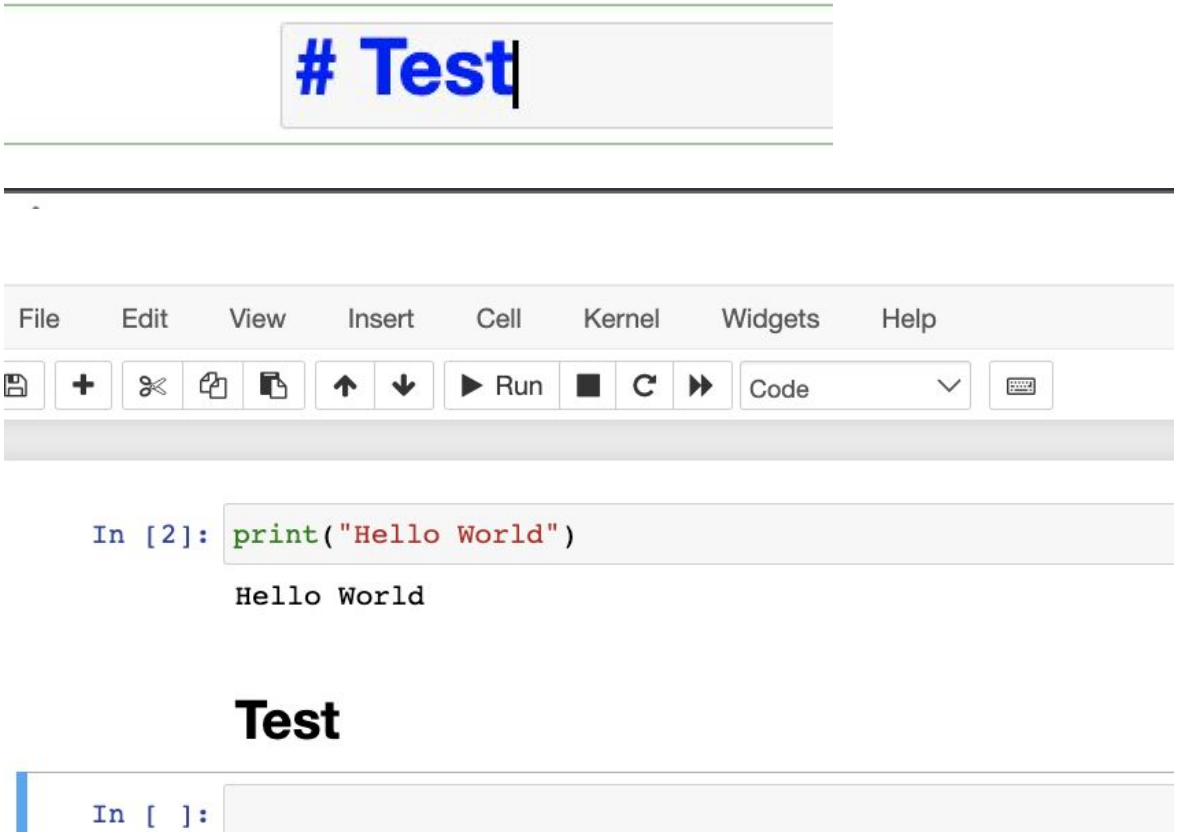
Activity: Write code that prints hello world and execute this code.

Click on the run button or shift enter. If you click on shift enter then you see hello world being printed and a new cell.

Activity: Create a heading in your jupyter notebooks - 1



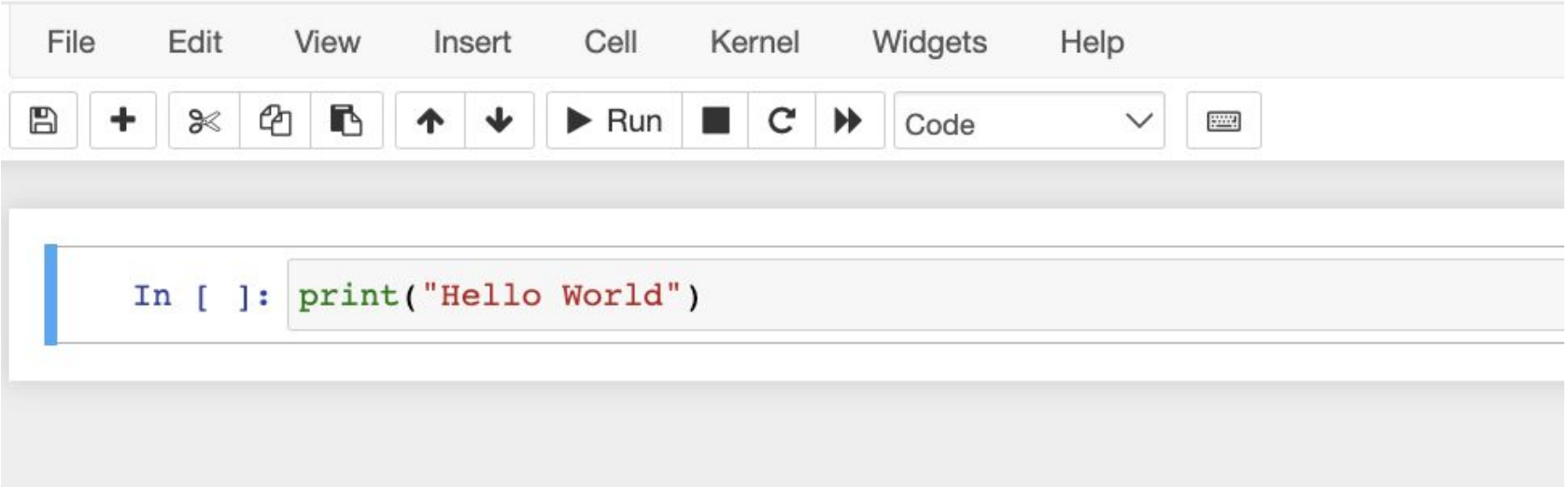
Activity: Create a heading in your jupyter notebooks - 1



You can adjust the size of the heading by adding another hashtag and another...

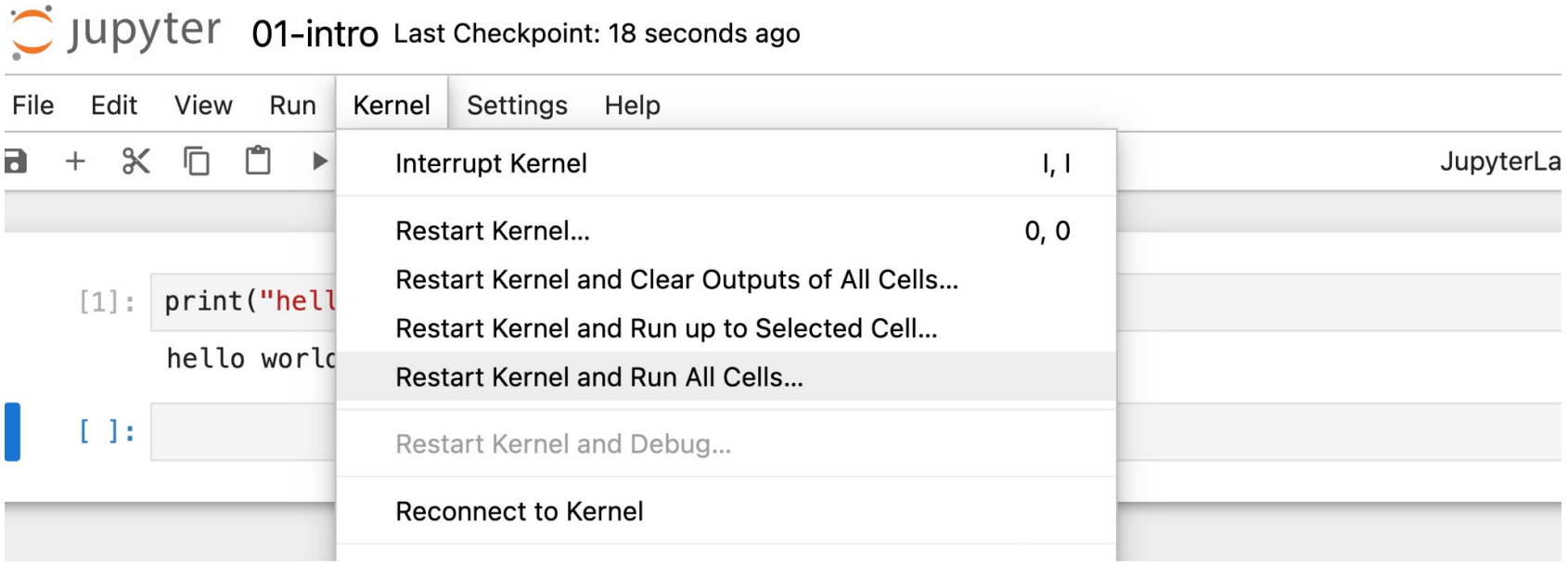
Activity: Delete the cell that contains the heading

- Click on edit and delete cells
- Or try the shortcut DD

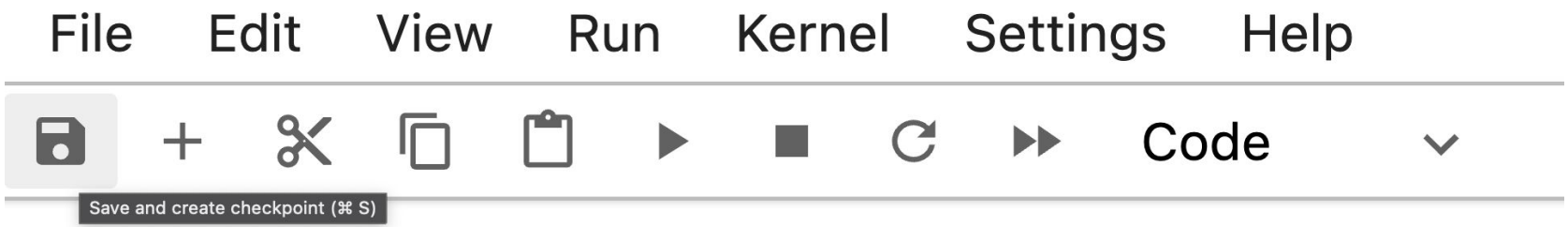


Activity: Restart your notebooks and clear all the output

- Click on Kernel and select the restart all output option



Activity: Don’t forget to save your notebooks and close them after the class




```
8af17c
[I 2024-04-08 14:14:14.463 ServerApp] Use Control-C to stop this server and shut down all kernels
irmation).
[C 2024-04-08 14:14:14.481 ServerApp]
```

```
96dc-b08b4adab854
^C[I 2024-04-08 14:19:33.455 ServerApp]
[I 2024-04-08 14:19:33.456 ServerApp] Se
cs
    1 active kernel
    Jupyter Server 2.8.0 is running at:
    http://localhost:8888/tree?token=5c
        http://127.0.0.1:8888/tree?toker
Shutdown this Jupyter server (y/[n])? y
```

Useful references for using Jupyter Notebooks

- https://www.edureka.co/blog/wp-content/uploads/2018/10/Jupyter_Notebook_CheatSheet_Edureka.pdf

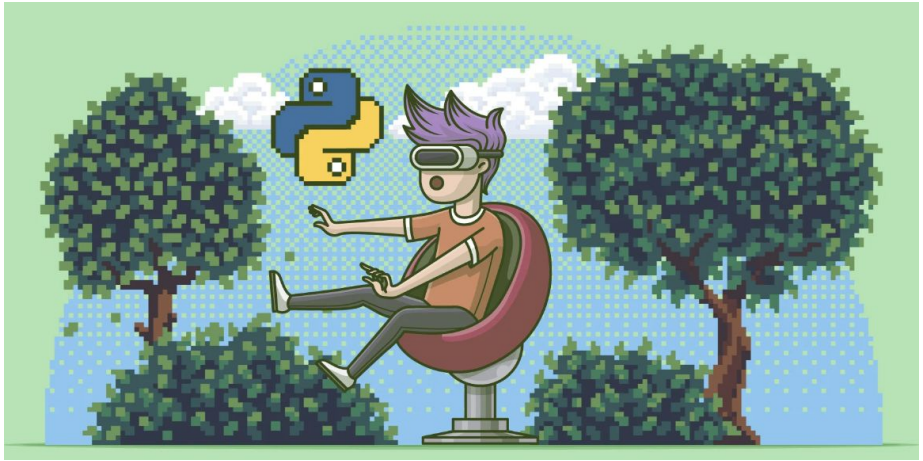
- https://medium.com/analytics-vidhya/the-ultimate-markdown-guide-for-jupyter-notebook-d5e5abf728fd

- https://jupyter-notebook.readthedocs.io/en/stable/notebook.html

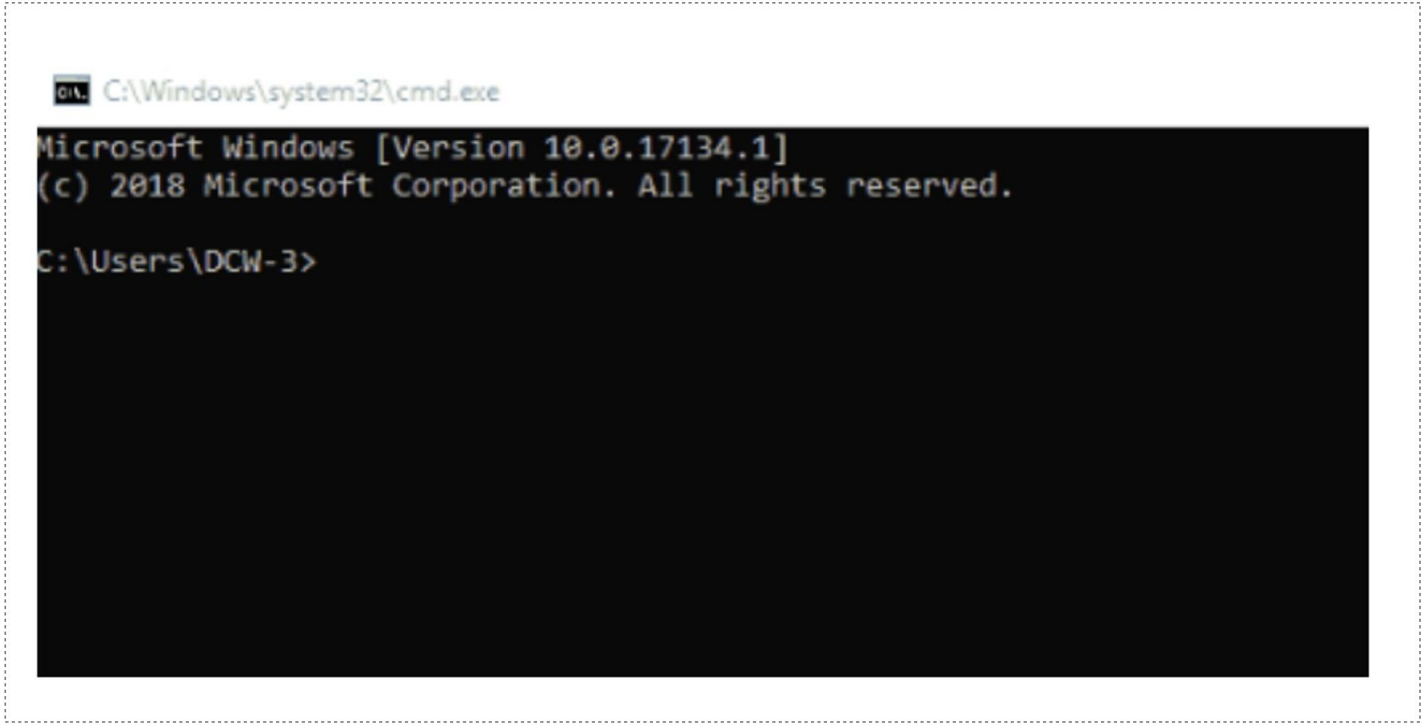
Virtual Environments

Virtual Environments

- A virtual environment is a Python tool for dependency management and project isolation
- They allow Python site packages (third party libraries) to be installed locally in an **isolated directory** for **a particular project**, as opposed to being installed globally (i.e. as part of a system-wide Python)



Open your terminal / command prompt?



Activity: Create a Virtual Environment

- Locate your reinforcement learning folder from your command line
- Create a new directory called 01Lecture in your reinforcement learning folder
- Change directory to your 01Lecture folder. Double check with pwd / cd

If you installed python with conda…
Create your own conda environment and install a package

- Create your own environment: `conda create -n test python=3.11`
- Activate environment: `conda activate test`
- Check list of packages in new environment: `conda list`
- Check the python version
- Deactivate your environment: `conda deactivate`

Cheatsheet:
https://docs.conda.io/projects/conda/en/4.6.0/_downloads/52a95608c49671267e40c689e0bc00ca/conda-cheatsheet.pdf

(Optional) Do you have a Python Code Editor installed?

- Install the **Community Edition** of PyCharm for your operating system. PyCharm is an **Integrated Development Environment** (IDE) used for programming in Python.

(Optional) Important: The Project set up

- Step 1: Create a new folder/directory called reinforcement_learning in a location of your choice.
- Step 2: Open PyCharm and click on open - locate your reinforcement_learning folder