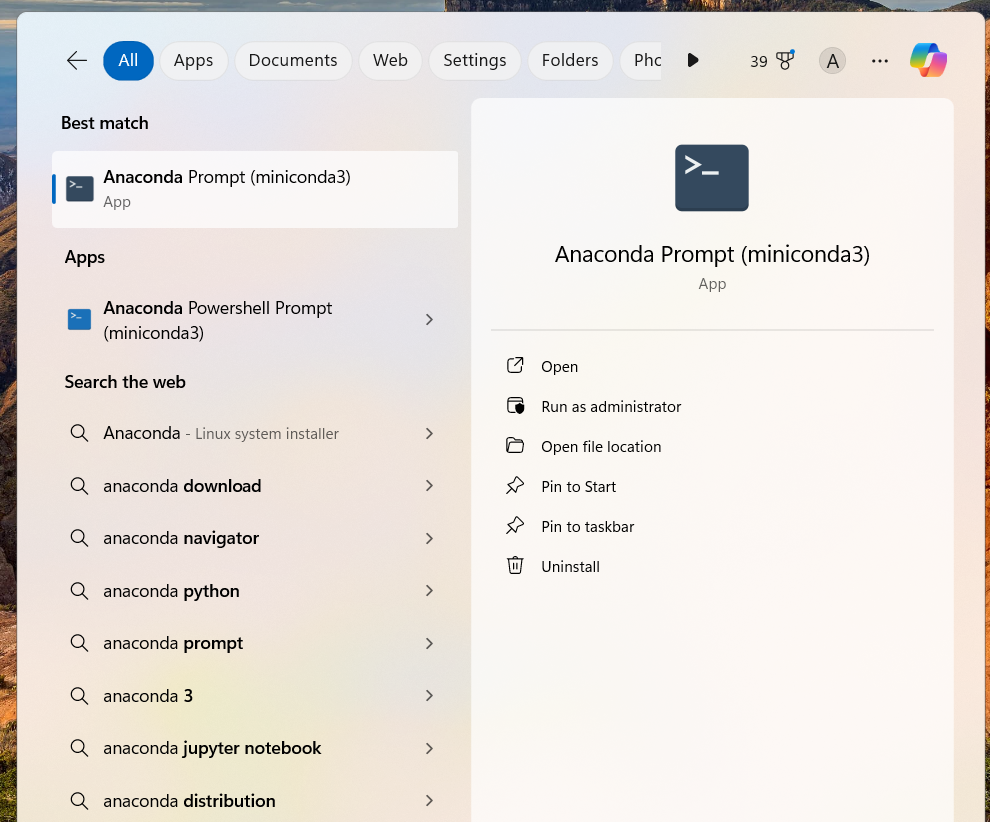
Today we will set the framework for upcoming Hands-on sessions in the course.

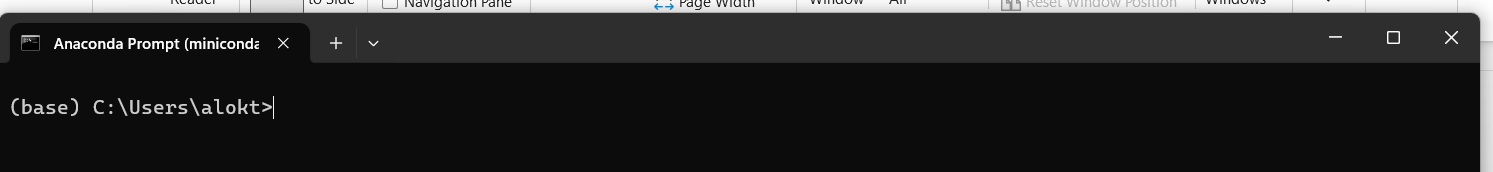
1. Download Miniconda from here: <https://docs.anaconda.com/miniconda/miniconda-install/>

Miniconda is a lightweight version of Anaconda. Anaconda is open-source distribution of Python well suited for data science. While installing, accept the defaults.

1. Anaconda includes conda, a package manager that helps you install, update, and manage Python libraries and dependencies. Conda can handle packages from the Anaconda repository as well as other repositories, and it resolves complex dependencies to ensure compatibility.
2. Anaconda provides tools for creating isolated environments with different sets of packages and versions. This allows you to manage multiple projects with different requirements on the same system without conflicts.
3. The Anaconda distribution comes with many popular libraries and tools preinstalled, including NumPy, pandas, SciPy, scikit-learn, Jupyter Notebook, and more.
4. Anaconda supports the conda-forge channel, which is a community-maintained collection of packages.
5. Anaconda includes a graphical user interface called Anaconda Navigator. It provides a user-friendly way to manage environments and packages, launch applications like Jupyter Notebook and Spyder, and perform other tasks without using the command line.
6. Once installed. Open Anaconda Prompt by typing it in Windows Explorer. If you have Mac, you can simply open the Terminal.



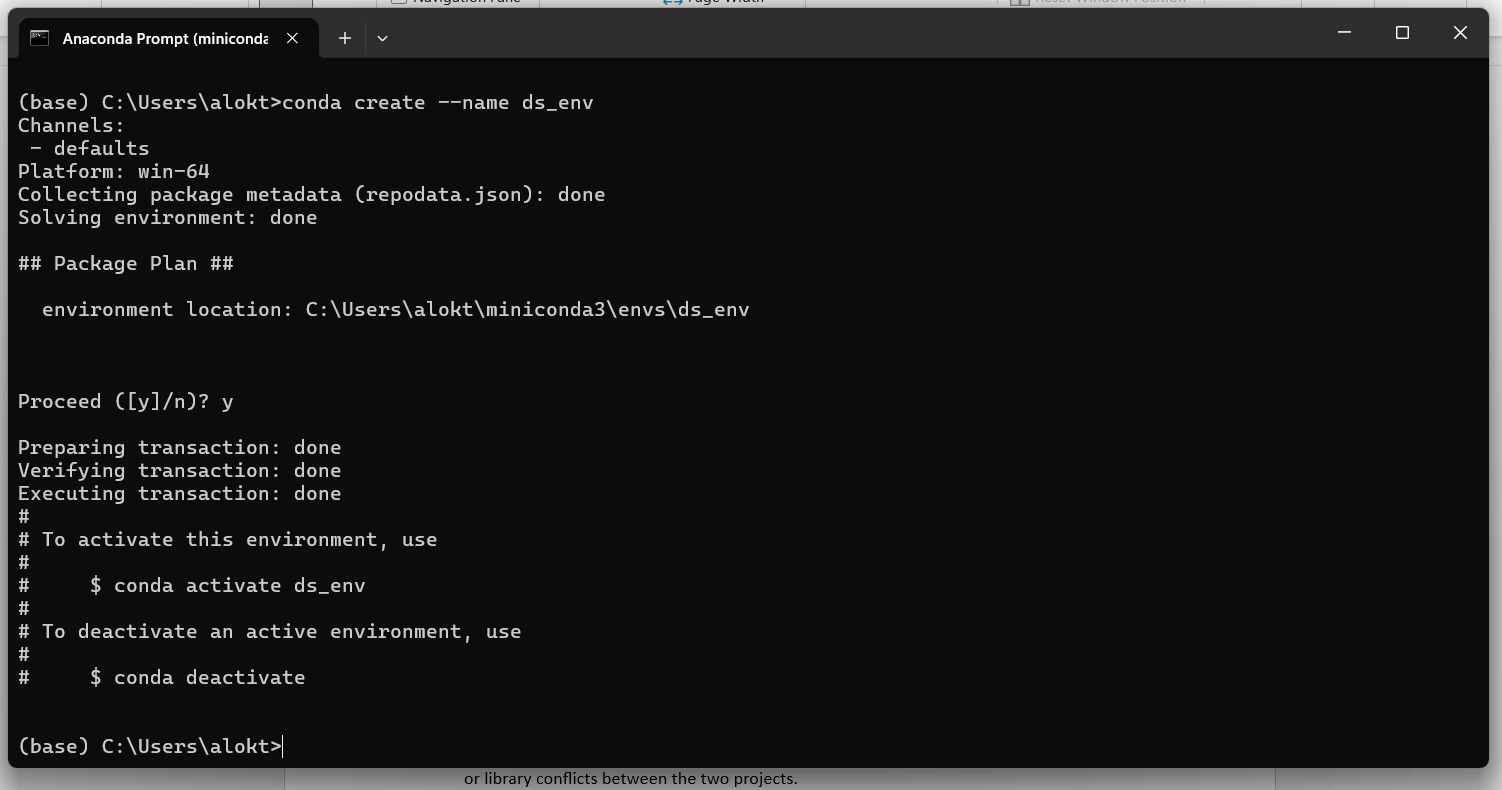
1. Note that ‘ (base)’ is written at the prompt. This signifies that you are at the base environment. In Python, an "environment" generally refers to a self-contained workspace where you can install specific versions of Python and its packages without affecting other projects or the system-wide Python installation.



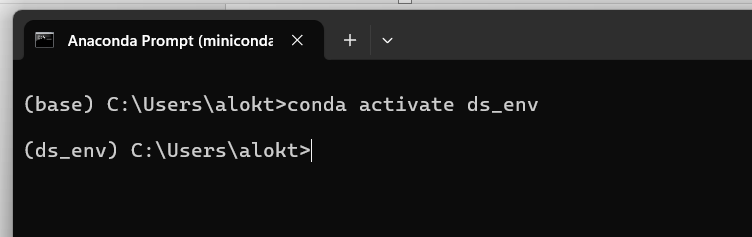
1. Type following commands to create the environment for Data Science. In the Data Science, environment you can install libraries dedicated to doing only data science. Suppose, in parallel you are doing a Data Engineering Project that require completely different set of Python libraries. So, you can create another Python environment for latter. This will avoid any version or library conflicts between the two projects.

Creation of environment: **conda create --name myenv**

Replace ‘ myenv’ with ‘ ds\_env’



Activate the environment: **conda activate ds\_env**

****

This activates the environment and you enter your data science environment

Now you can install libraries in the environment.

Numpy: <https://anaconda.org/anaconda/numpy>

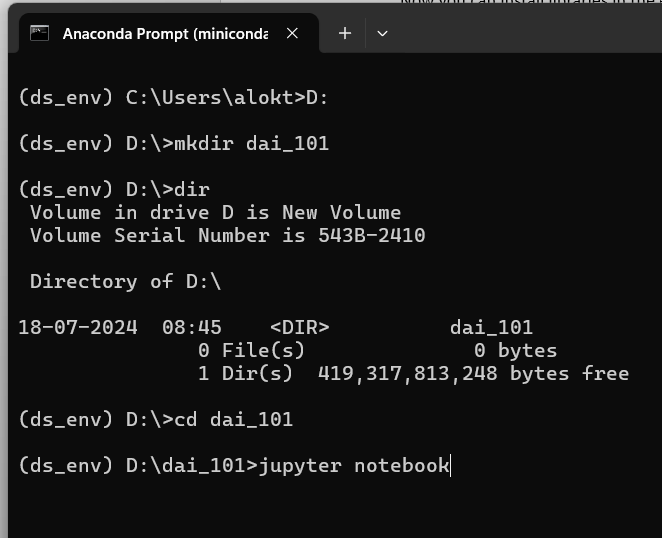
Pandas: <https://anaconda.org/anaconda/pandas>

Jupyter Notebook: <https://anaconda.org/conda-forge/jupyter>

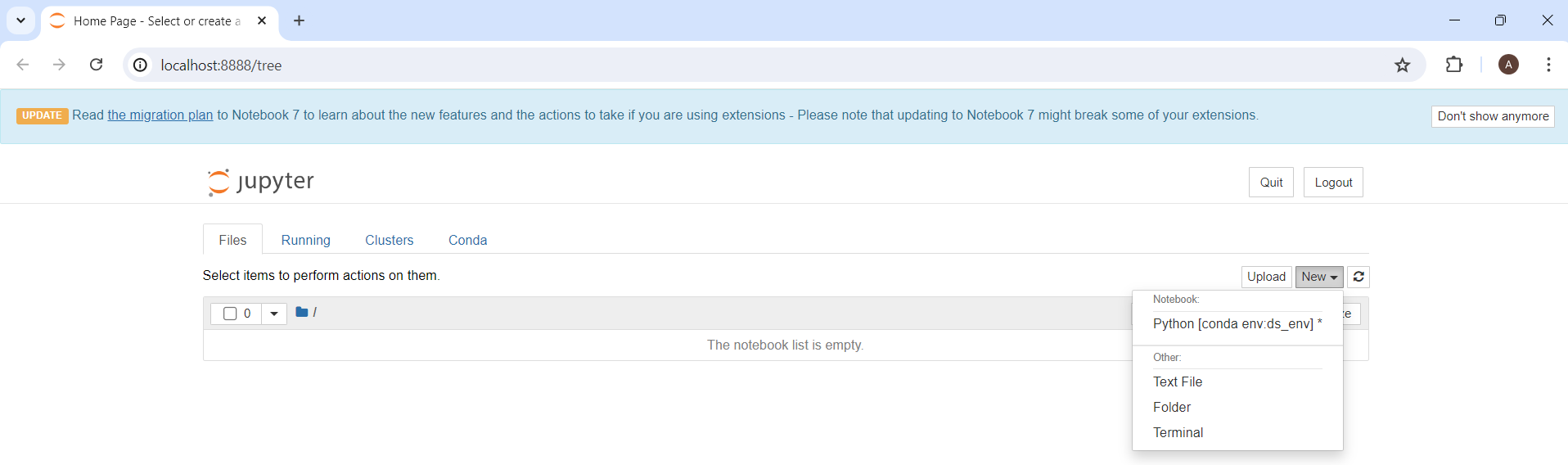
Nb\_conda: <https://anaconda.org/conda-forge/nb_conda>

Nb\_conda helps you to visualize different environments in Jupyter notebook.

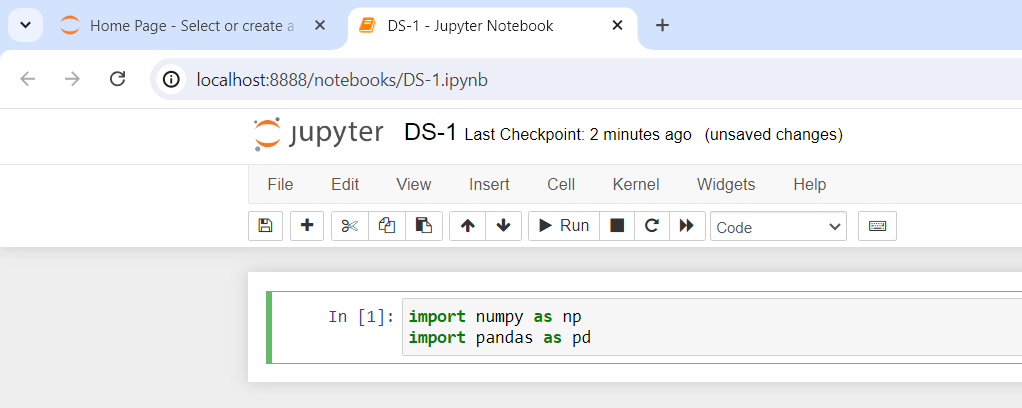
1. After all the above steps, its time to open the Jupyter notebook. For this, first create a directory in your local machine where you will be working with Data Science. Go to that directory in the anaconda command prompt.
2. Type **jupyter notebook**



1. You will see a browser window open with jupyter notebook. Note the **ds\_env** when you click on **New** button



1. Click on the ds\_env to create a new jupyter notebook. Click on Untitled to rename it. I have renamed it to DS-1.
2. Type the commands shown in picture. It should work fine. If not then either there is a problem with setting up the environment or installation of the libraries or not opening jupyter notebook from ds\_env.



1. Congratulations on setting up the environment and basic setup to do hands-on for this course.