Getting Started: Installing Key Python Libraries for Machine Learning

The goal of these instructions is to prepare you with the essential Python libraries for data science, machine learning, and deep learning experiments.

Essential Libraries

- NumPy: Core library for n-dimensional array operations and mathematical functions.
- SciPy: Extends NumPy with additional scientific computing capabilities.
- Matplotlib: Comprehensive library for creating static, animated, and interactive visualizations.
- Pandas: Advanced data structures and data analysis tools built on NumPy.
- Scikit-Learn: Efficient library for implementing machine learning algorithms and data mining.
- TensorFlow: Google's open-source library for deep learning, also supports a variety of traditional machine learning tasks.
- Keras: High-level deep learning API, included within TensorFlow as tf.keras, simplifying model creation and training.

Installation Instructions: macOS Users

Step 1: Download Anaconda

- Go to the Anaconda downloads page: https://www.anaconda.com/download
- Download the latest Python 3 graphical installer for macOS.

Step 2: Install Anaconda

- Open the downloaded *.pkg* installer file and follow the on-screen instructions.
- Ensure the installer adds Anaconda to your PATH environment variable.
- Verify the installation by opening Terminal and typing: `conda --version`
- Check the installed Python version by typing: `python --version`

Step 3: Set Up a New Conda Environment

- Create a new environment to keep your projects organized and dependencies isolated: `conda create --name myenv python=X.Y`
- Replace X.Y with your preferred Python version (e.g., 3.11). Choose the latest version for new features or an older version for stability and widespread library support.

Step 4: Activate the Conda Environment

- Activate your environment with: `conda activate myenv`
- Your Terminal prompt should now show (myenv), indicating the environment is active.

Step 5: Install Required Libraries

- Anaconda includes Jupyter Notebook, NumPy, Pandas, Matplotlib, and Scikit-learn by default. No need for separate installation.
- Install TensorFlow and Keras with: `conda install tensorflow keras`
- Alternatively, you may use Python's own packaging system "pip": `pip install tensorflow keras`
- pip is a recursive acronym that can stand for either "Pip Installs Packages" or "Pip Installs Python". pip is a package management system used to install and manage software packages written in Python. Python 3.4 and later include pip (pip3 for Python 3) by default.
- Display a list of all packages installed in the active environment, including their versions, by typing: `conda list`

Step 6: Install Additional Libraries

- To install additional libraries, use either *conda* or *pip* within your active environment: `conda install <library-name>` or `pip install <library-name>`

Step 7: Deactivate the Conda Environment

- When finished working, deactivate the environment by typing: `conda deactivate`

Step 8: Launch Jupyter Notebook

- To launch Jupyter Notebook, first activate the conda environment by opening the Terminal and typing: `conda activate my-env`
- Then, type the following command in the Terminal and press Enter: `jupyter notebook`
- This command will start the Jupyter Notebook server and open the Jupyter Notebook dashboard in your default web browser.

Installation Instructions: Windows Users

Step 1: Download and Install Anaconda

- Go to the Anaconda installation guide for Windows: https://docs.anaconda.com/anaconda/install/windows/
- Follow the instructions to install Anaconda, which includes Python and all necessary packages.

Step 2: Install Required Libraries

- Anaconda includes NumPy, Pandas, Matplotlib, Scikit-learn, and Jupyter Notebook by default. No need for separate installation.
- For installing other libraries such as TensorFlow and Keras, open the "Anaconda Prompt" application (https://docs.anaconda.com/anaconda/install/verify-install/).
- Install TensorFlow and Keras with: `conda install tensorflow keras`
- View installed libraries by typing: `conda list`

Step 3: Launch Jupyter Notebook

- Use the Anaconda prompt to navigate to the directory containing your Jupyter Notebook files using: cd directory_name
- Start Jupyter Notebook by typing: jupyter notebook

Create Your First Jupyter Notebook

Jupyter Notebook is a versatile, web-based application designed for creating and running interactive notebooks. These notebooks allow you to write and execute code, visualize data, and include explanatory text, all within a single document. Jupyter Notebook can be run locally on your machine or on a remote server.

After completing the installation of Jupyter Notebook, follow these steps to create and work with your first notebook:

Starting the Jupyter Notebook App

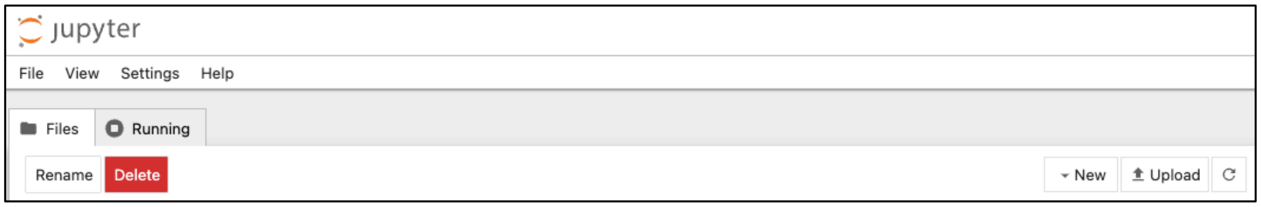
- Launch the Jupyter Notebook app as previously instructed. Once started, a Jupyter server will run on your command line interface (e.g., Terminal on macOS or Anaconda Prompt on Windows), listening to port 8888.
- You can access this server by opening your web browser and navigating to http://localhost:8888/. This usually happens automatically when the server starts. You'll be presented with the Jupyter Notebook dashboard, which displays the files and folders in your current working directory.

Creating a New Notebook

To create a new notebook, click on the "New" button in the top-right corner of the Jupyter dashboard. A dropdown menu will appear—select the appropriate Python version for your environment (e.g., Python 3).

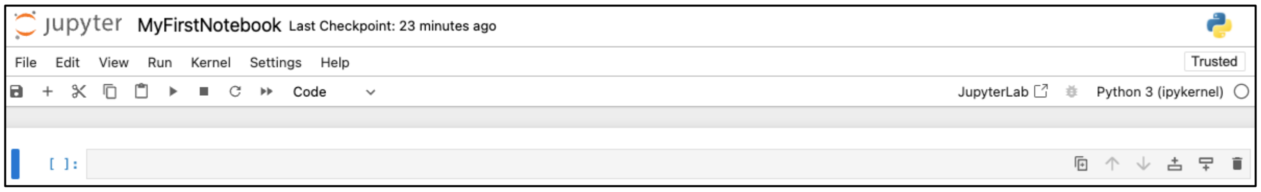
This action does three things:

- It creates a new notebook file named Untitled.ipynb in your workspace.
- It starts a Jupyter Python kernel that will run the code in this notebook.
- It opens the notebook in a new tab of your web browser.



Renaming Your Notebook

- The default name of your notebook is "Untitled". It is a good practice to rename it to something more meaningful.
- Click on the name "Untitled" at the top of the notebook interface and type in a new name, such as "MyFirstNotebook."
- This will automatically rename the file to MyFirstNotebook.ipynb.

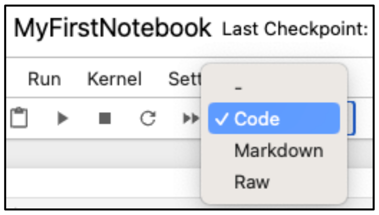

jupyter
MyFirstNotebook Last Checkpoint: 23 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)
[ ]:

Working with Cells

A Jupyter notebook is composed of cells. Each cell can contain either executable code or formatted text (written in Markdown).

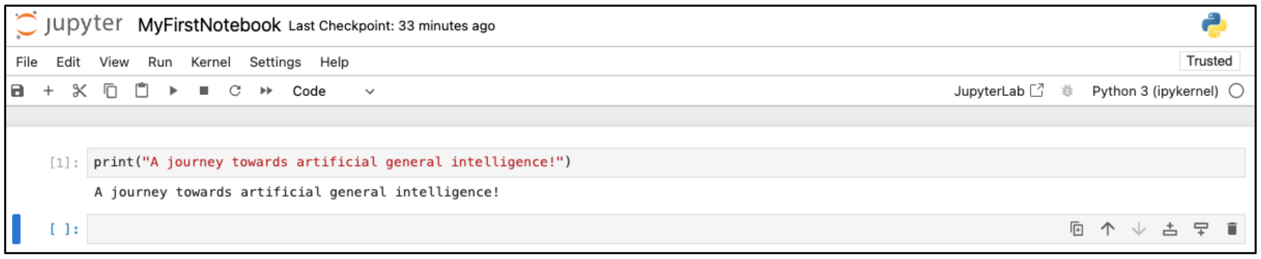
- Code Cell: This is the default cell type where you can write and execute Python code. To ensure a cell is set to code, click on the cell, and in the toolbar, choose "Code" from the dropdown menu.
- Markdown Cell: If you want to write formatted text, such as explanations, headers, or lists, you need to switch the cell type to Markdown. To do this, click on the cell, and in the toolbar, select "Markdown" from the dropdown menu.

After selecting the desired type, you can begin writing your code or Markdown text within the cell.



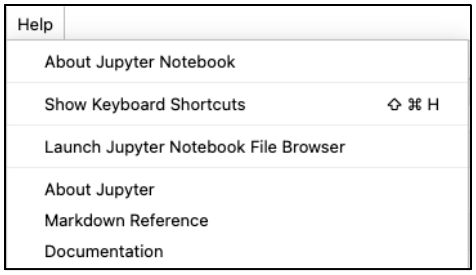
Write And Execute Your First Python Code

- When you first open your new notebook, it will contain one empty code cell labeled "In [1]:".
- To get started, type the following Python code into the cell: print("A journey towards artificial general intelligence!")
- Execute the cell by clicking the play button in the toolbar or pressing Ctrl+Enter (or Shift+Enter).
- The code will run, and the output will be displayed directly below the cell.
- Since the notebook is interactive, a new cell will be automatically created after the output.


jupyter
MyFirstNotebook Last Checkpoint: 33 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Code
JupyterLab
Python 3 (ipykernel)

```
[1]: print("A journey towards artificial general intelligence!")
     A journey towards artificial general intelligence!
[ ]:
```

Exploring the User Interface
To familiarize yourself with the Jupyter Notebook interface, it is recommended to go through the resources available under the Help menu. This will provide a quick overview of the various features and shortcuts available within the notebook environment.


Help
About Jupyter Notebook
Show Keyboard Shortcuts ⇧ ⌘ H
Launch Jupyter Notebook File Browser
About Jupyter
Markdown Reference
Documentation