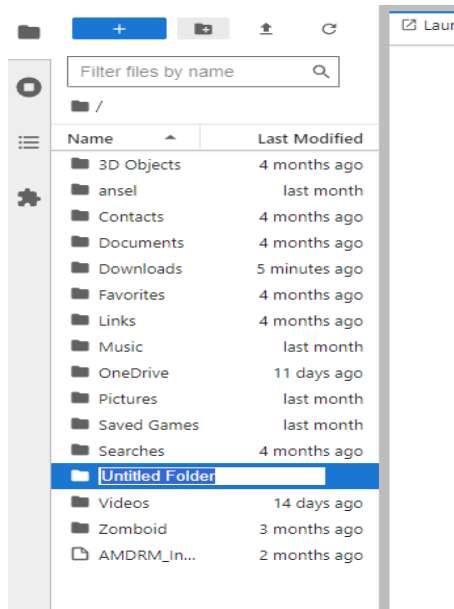
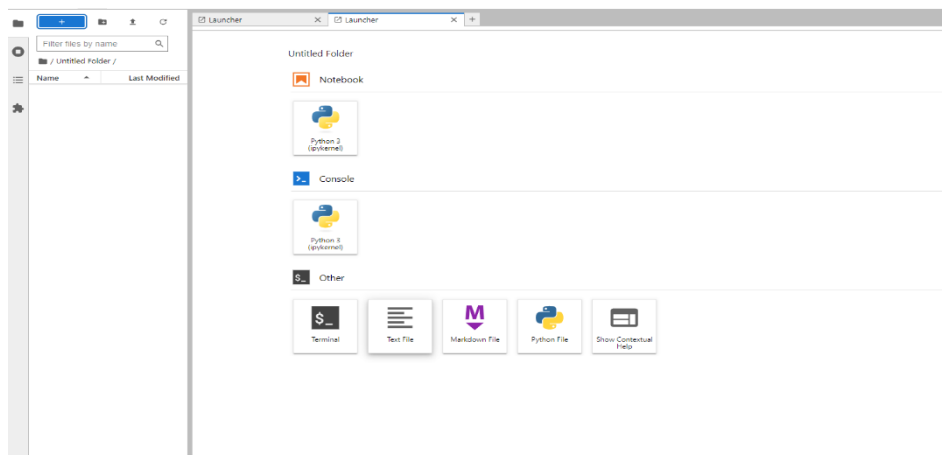


1. Click the add folder icon



2. After opening the folder you created click the text file icon to create text file



3. Launch JupyterLab: Open JupyterLab either through Anaconda Navigator, command line, or any other method you prefer.

Create a new Notebook: Click on the "+" button in the launcher tab or the file browser tab to create a new notebook.

Import Libraries: In the first cell of your notebook, import the necessary libraries. For working with CSV files, you'll primarily use Pandas.

4. you can work with dictionaries by creating them with key-value pairs enclosed in curly braces, accessing elements using square brackets, employing methods like `keys()`, `values()`, and `items()` to gather dictionary information, adding or modifying entries with square brackets, deleting entries using `del` or `pop()`, and iterating over keys, values, or key-value pairs using loops.
5. To create a directory in Jupyter Notebook, import the `os` module, then use the `os.makedirs()` function, providing the desired directory path as the argument. After execution, the directory will be created at the specified location.
6. To import libraries in Jupyter Notebook, use the `import` keyword followed by the name of the library. If you want to import a specific function or submodule from a library, you can use `from library_name import function_name` or `from library_name import submodule_name`. After importing, you can use the functions and modules from the library in your notebook.
7. To use a CSV file for data in Jupyter Notebook, you can first ensure that the CSV file is present in your working directory or provide the full path to the CSV file. Then, use libraries like `pandas` to read the CSV file into a `DataFrame`. You can do this by using `pd.read_csv('file.csv')`, where `file.csv` is the name of your CSV file. Once the CSV file is read into a `DataFrame`, you can perform various data analysis and manipulation tasks using `pandas` methods and functions.
8. Import Libraries: Start by importing the necessary libraries for data analysis and visualization. Common libraries include `pandas` for data manipulation, `matplotlib` and `seaborn` for visualization, and `numpy` for numerical operations.
9. Finding Data: Locate the CSV file or dataset you want to analyze. Ensure it's accessible from your Jupyter Notebook environment.
10. Importing Data: Use the `pd.read_csv()` function from the `pandas` library to import the CSV file into a `DataFrame`. Provide the path to the CSV file as an argument to this function.
11. Data Attributes: Once the data is imported, explore its attributes such as the shape (number of rows and columns), column names, data types, and summary statistics. You can use `DataFrame` methods like `info()`, `describe()`, and attribute accessors like `shape` and `columns`.