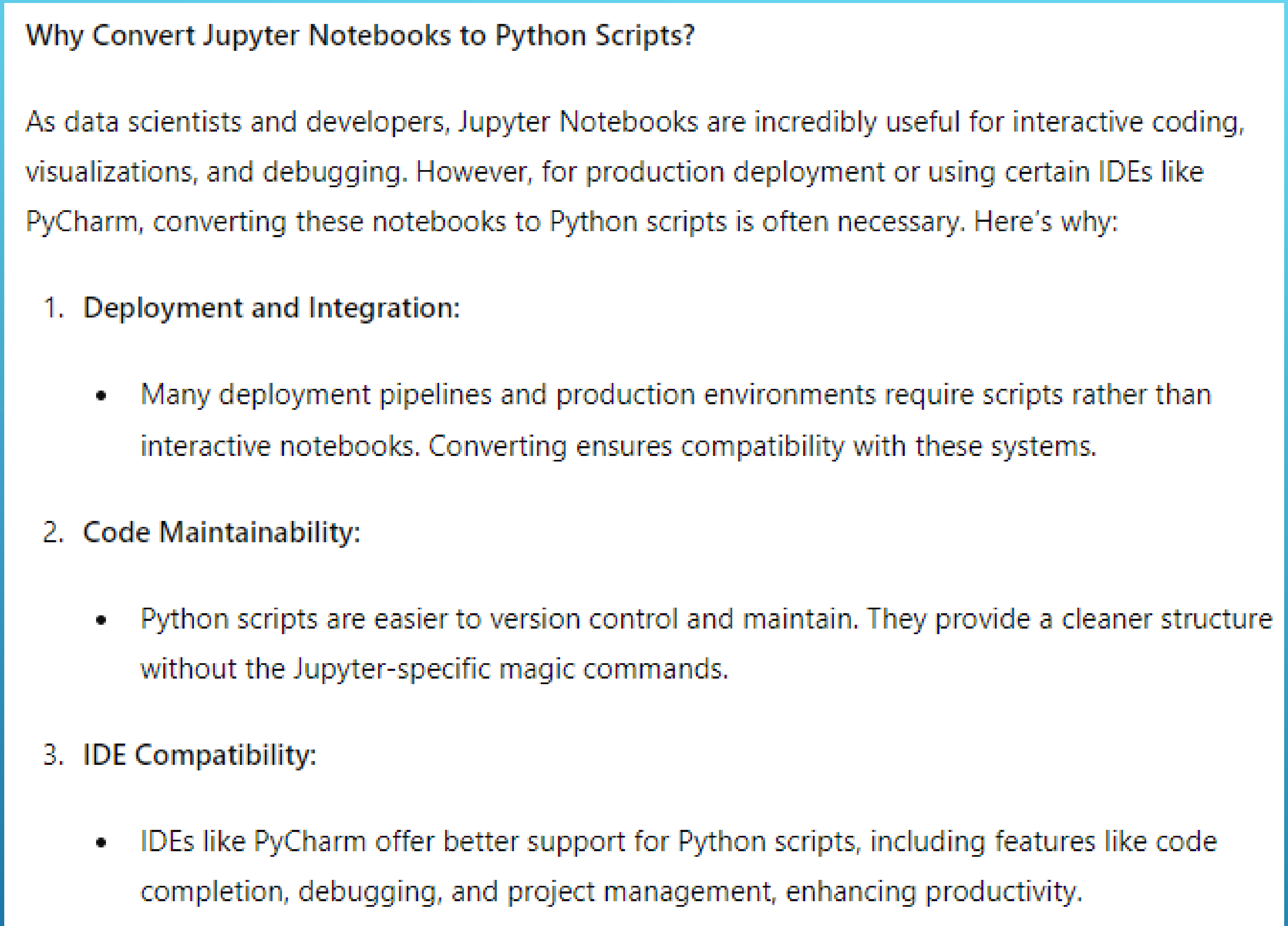
CONVERTING JUPYTER NOTEBOOKS TO PYTHON SCRIPTS:

A Guide for Beginners



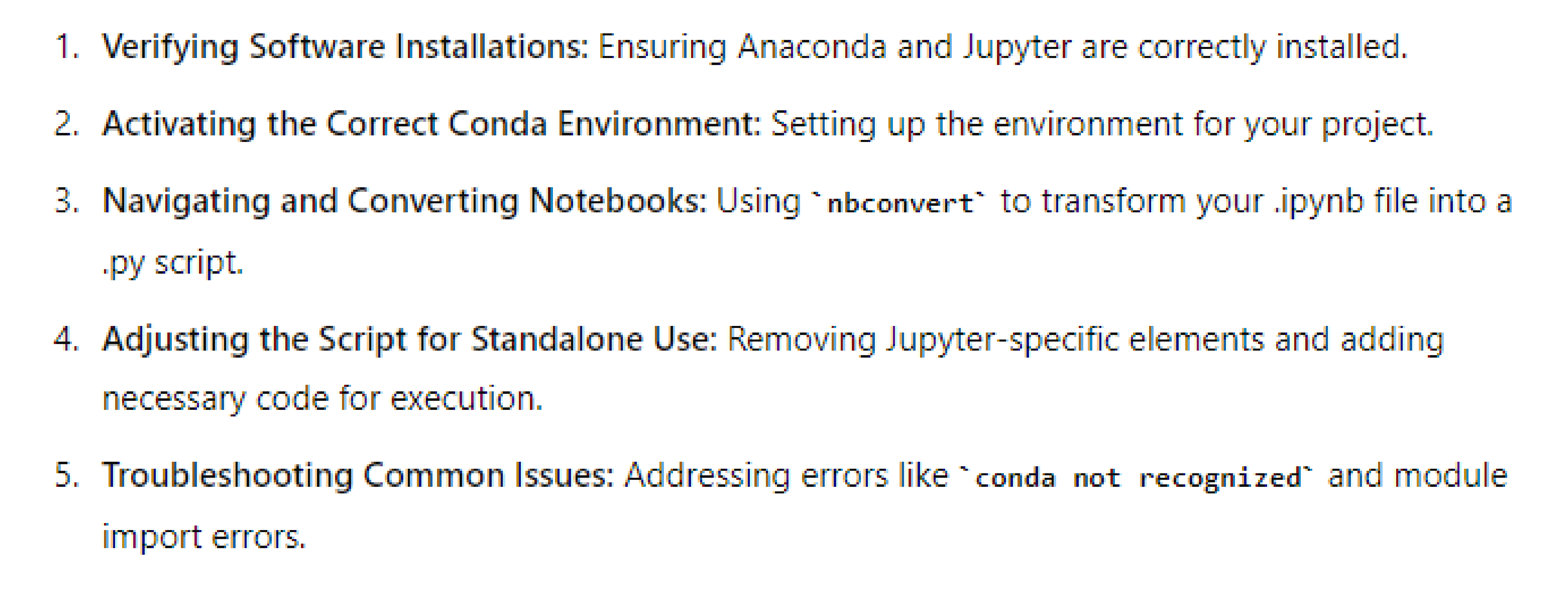
Why Convert Jupyter Notebooks to Python Scripts?

As data scientists and developers, Jupyter Notebooks are incredibly useful for interactive coding, visualizations, and debugging. However, for production deployment or using certain IDEs like PyCharm, converting these notebooks to Python scripts is often necessary. Here's why:

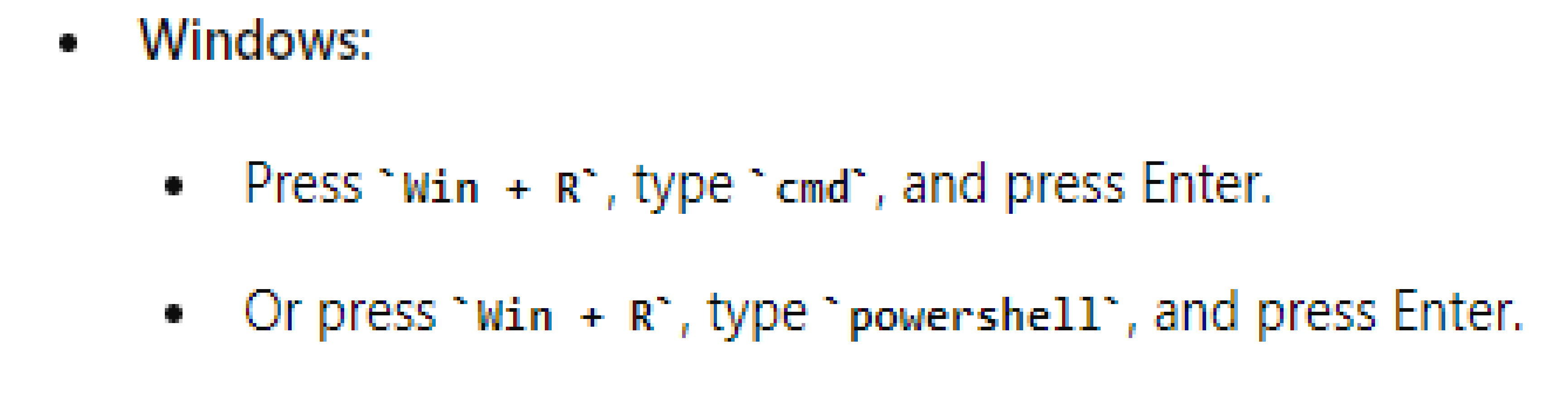
1. **Deployment and Integration:**
    - Many deployment pipelines and production environments require scripts rather than interactive notebooks. Converting ensures compatibility with these systems.
2. **Code Maintainability:**
    - Python scripts are easier to version control and maintain. They provide a cleaner structure without the Jupyter-specific magic commands.
3. **IDE Compatibility:**
    - IDEs like PyCharm offer better support for Python scripts, including features like code completion, debugging, and project management, enhancing productivity.

WHY CONVERT JUPYTER NOTEBOOKS TO PYTHON SCRIPTS?

IN THIS GUIDE, WE WILL COVER:



1. **Verifying Software Installations:** Ensuring Anaconda and Jupyter are correctly installed.
2. **Activating the Correct Conda Environment:** Setting up the environment for your project.
3. **Navigating and Converting Notebooks:** Using `nbconvert` to transform your .ipynb file into a .py script.
4. **Adjusting the Script for Standalone Use:** Removing Jupyter-specific elements and adding necessary code for execution.
5. **Troubleshooting Common Issues:** Addressing errors like `conda not recognized` and module import errors.

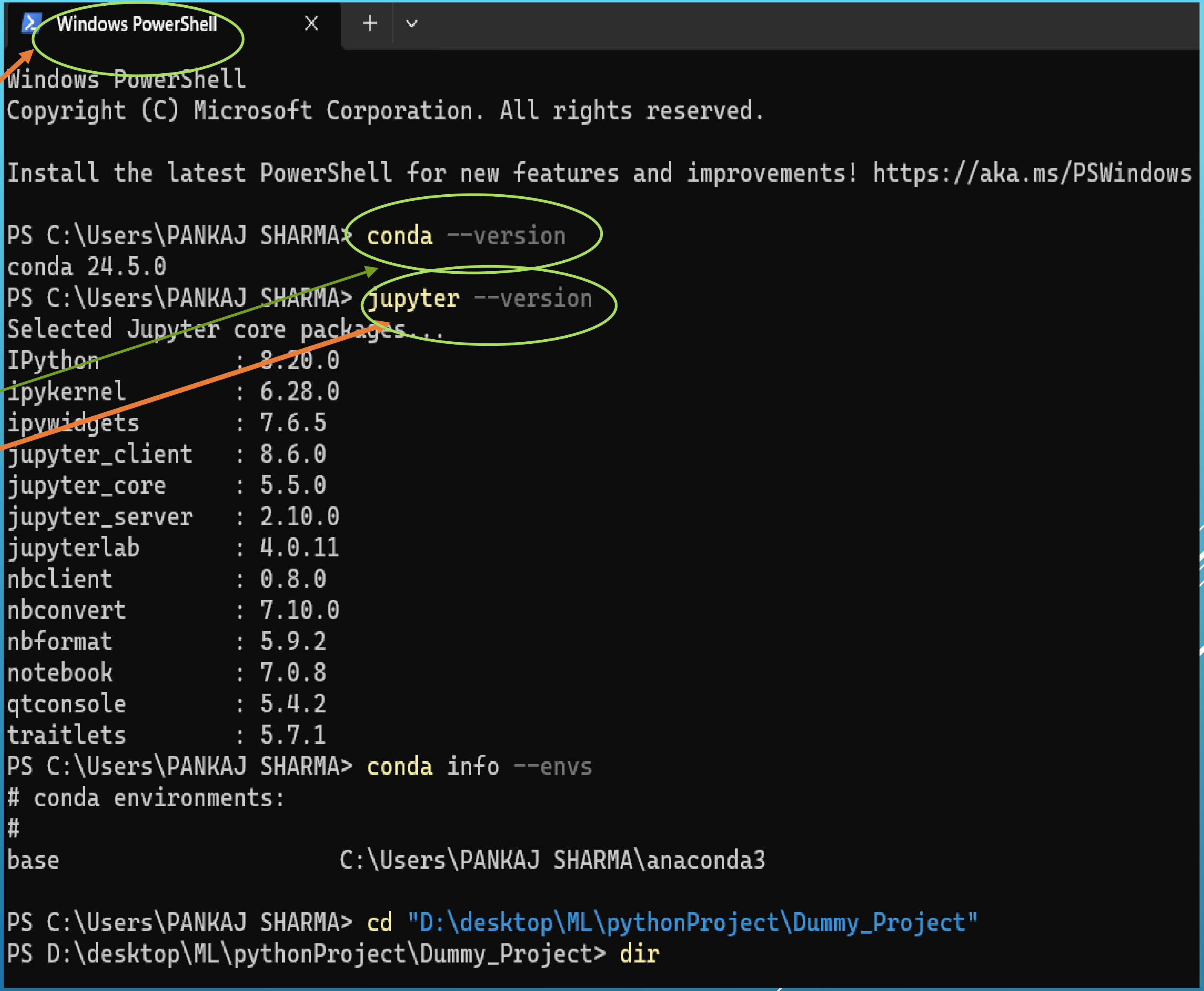


- Windows:
  - Press `Win + R`, type `cmd`, and press Enter.
  - Or press `Win + R`, type `powershell`, and press Enter.

OPEN TERMINAL OR COMMAND PROMPT:

VERIFY ANACONDA AND JUPYTER INSTALLATION

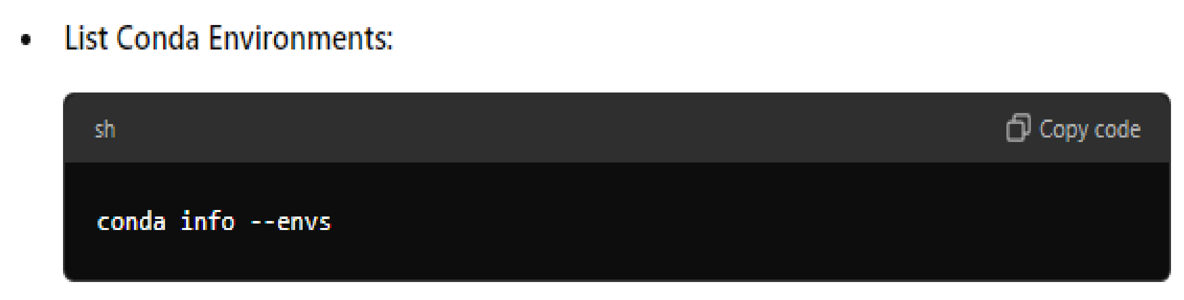
- Go to power shell
- conda --version
- jupyter --version



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\PANKAJ SHARMA> conda --version
conda 24.5.0
PS C:\Users\PANKAJ SHARMA> jupyter --version
Selected Jupyter core packages...
IPython          : 8.20.0
ipykernel        : 6.28.0
ipywidgets       : 7.6.5
jupyter_client   : 8.6.0
jupyter_core     : 5.5.0
jupyter_server   : 2.10.0
jupyterlab       : 4.0.11
nbclient         : 0.8.0
nbconvert        : 7.10.0
nbformat         : 5.9.2
notebook         : 7.0.8
qtconsole        : 5.4.2
traitlets        : 5.7.1
PS C:\Users\PANKAJ SHARMA> conda info --envs
# conda environments:
#
base                     C:\Users\PANKAJ SHARMA\anaconda3

PS C:\Users\PANKAJ SHARMA> cd "D:\desktop\ML\pythonProject\Dummy_Project"
PS D:\desktop\ML\pythonProject\Dummy_Project> dir
```

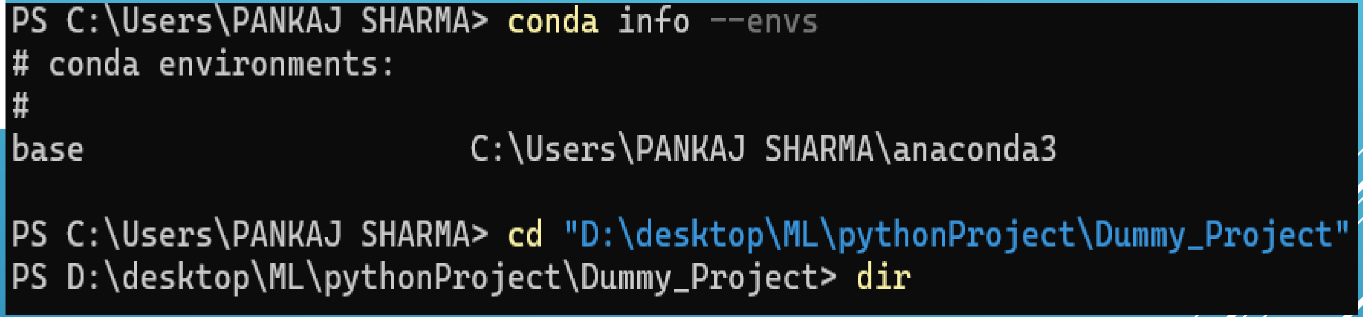
ACTIVATE THE CORRECT CONDA ENVIRONMENT

- **Objective:** Activate the Conda environment where Jupyter and required packages are installed.

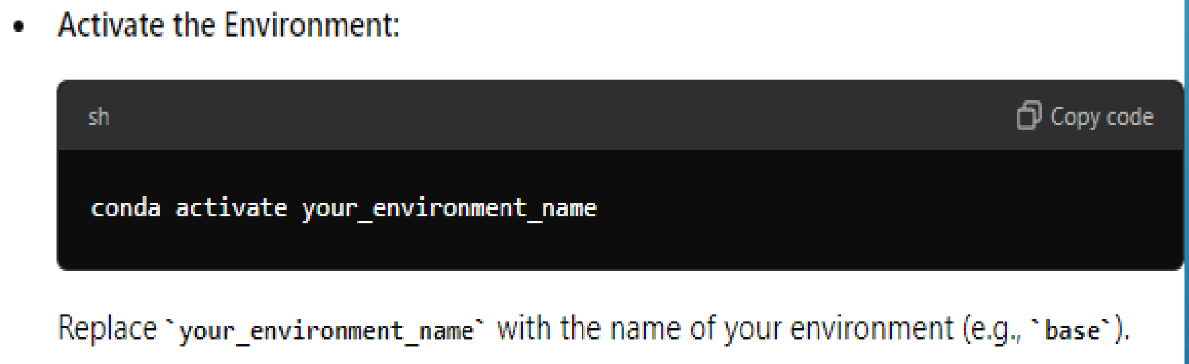


- List Conda Environments:

```sh
conda info --envs
```



```
PS C:\Users\PANKAJ SHARMA> conda info --envs
# conda environments:
#
base                     C:\Users\PANKAJ SHARMA\anaconda3

PS C:\Users\PANKAJ SHARMA> cd "D:\desktop\ML\pythonProject\Dummy_Project"
PS D:\desktop\ML\pythonProject\Dummy_Project> dir
```



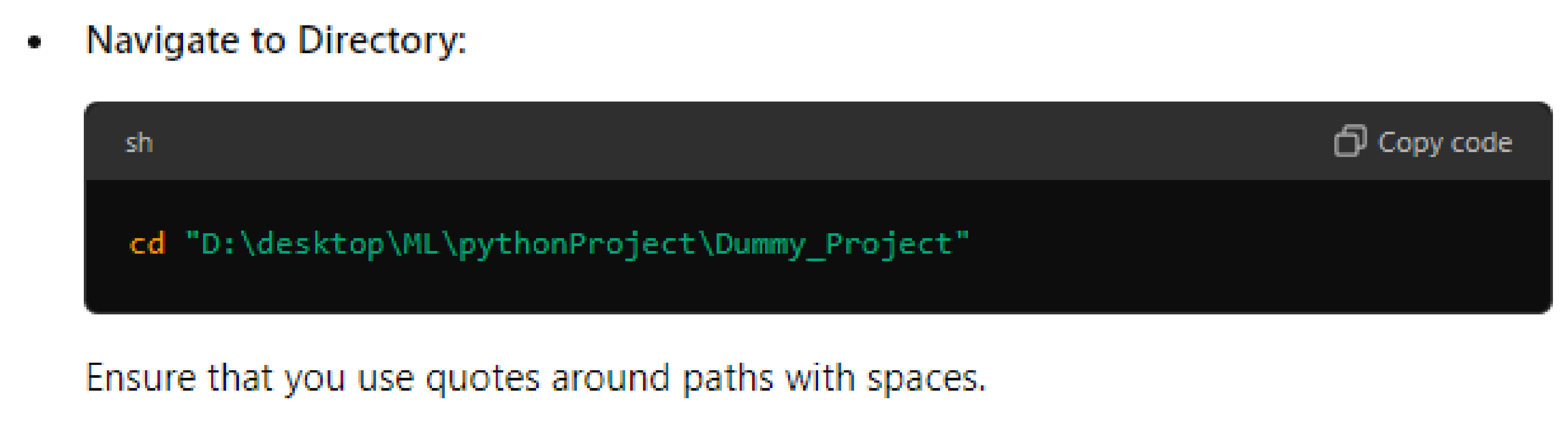
- Activate the Environment:

```sh
conda activate your_environment_name
```

Replace `your_environment_name` with the name of your environment (e.g., `base`).

NAVIGATE TO THE DIRECTORY CONTAINING YOUR NOTEBOOK

Objective: Change the directory to where your .ipynb file is located.

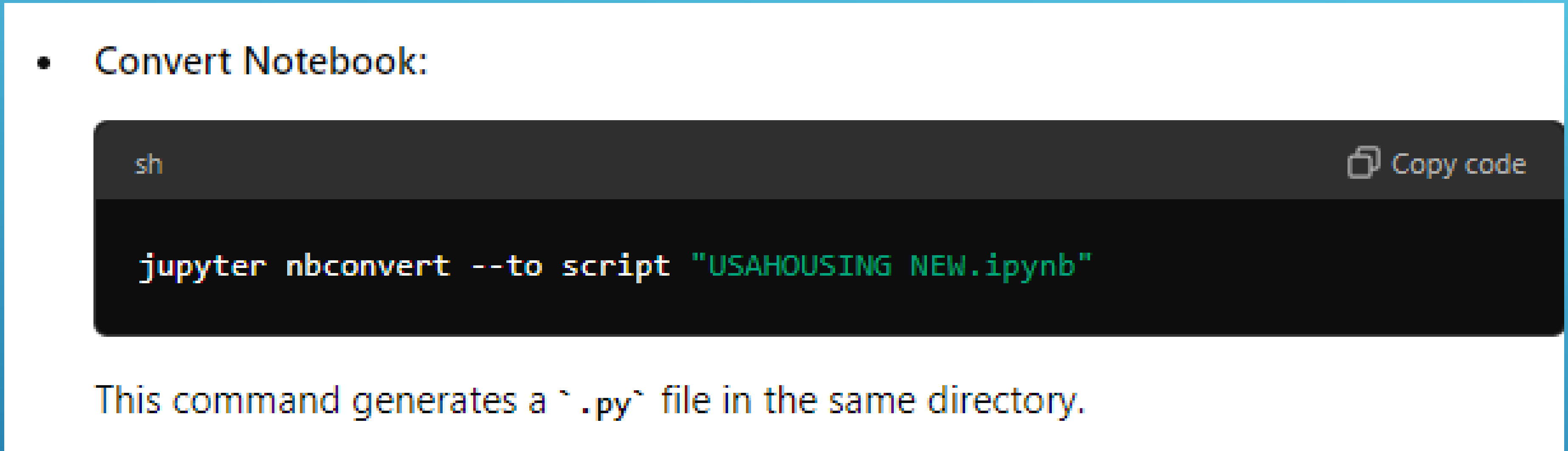


- Navigate to Directory:

```sh
cd "D:\desktop\ML\pythonProject\Dummy_Project"
```

Ensure that you use quotes around paths with spaces.

Objective: Use the ‘nbconvert’ command to convert the Jupyter Notebook to a Python script.



- Convert Notebook:

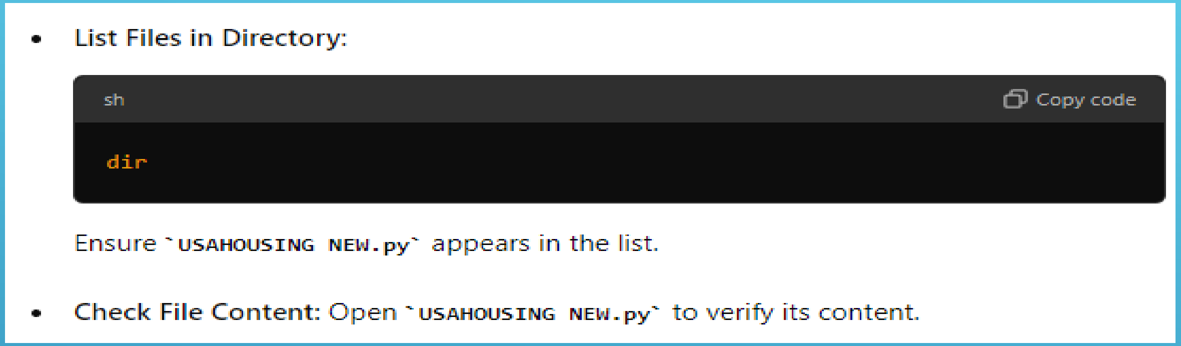
```sh
jupyter nbconvert --to script "USAHOUSING NEW.ipynb"
```

This command generates a `.py` file in the same directory.

CONVERT THE NOTEBOOK TO A PYTHON SCRIPT

VERIFY THE CONVERSION

➢**Objective:** Confirm that the .py file has been created.

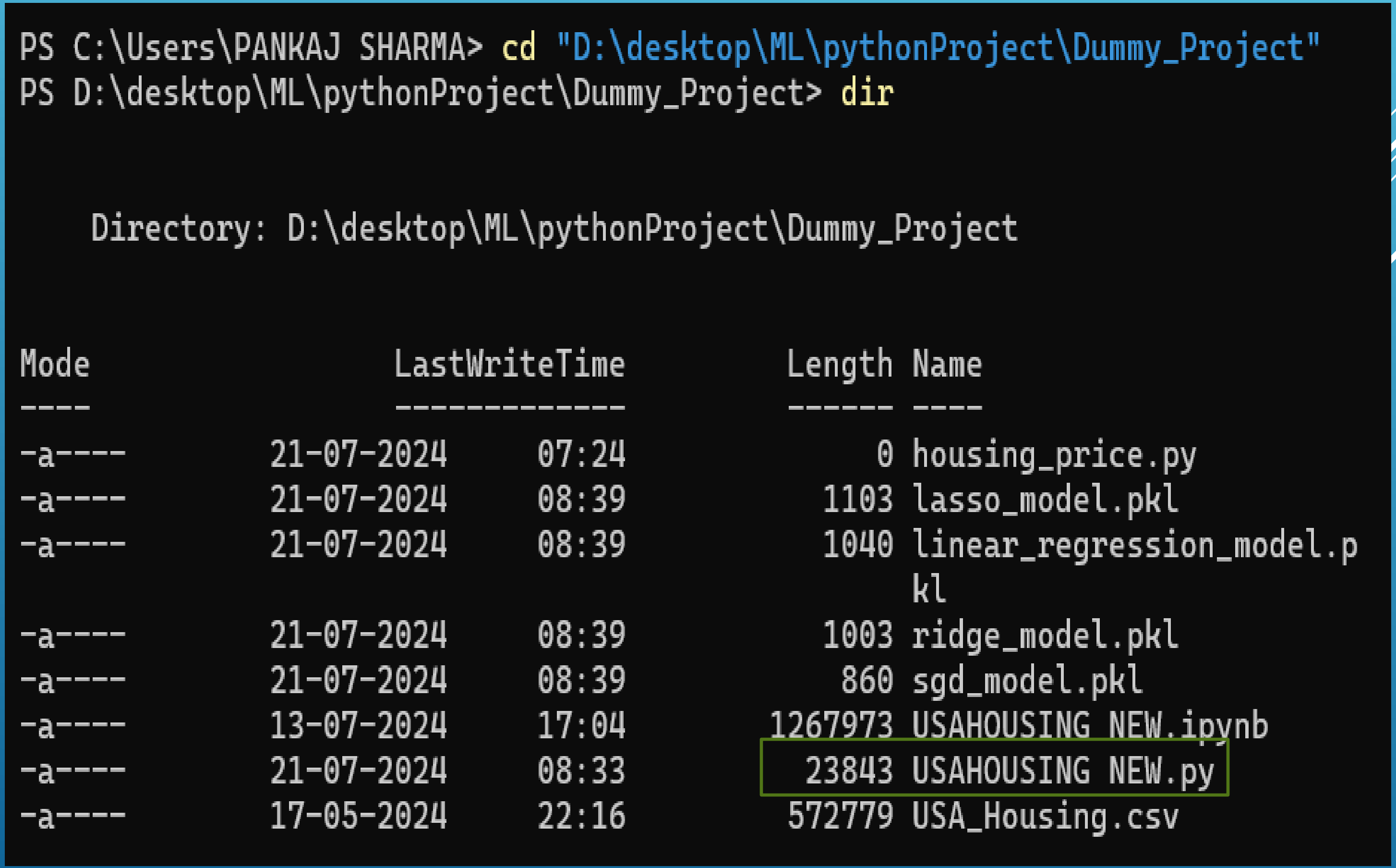


- List Files in Directory:

```sh
dir
```

Ensure `USAHOUSING NEW.py` appears in the list.

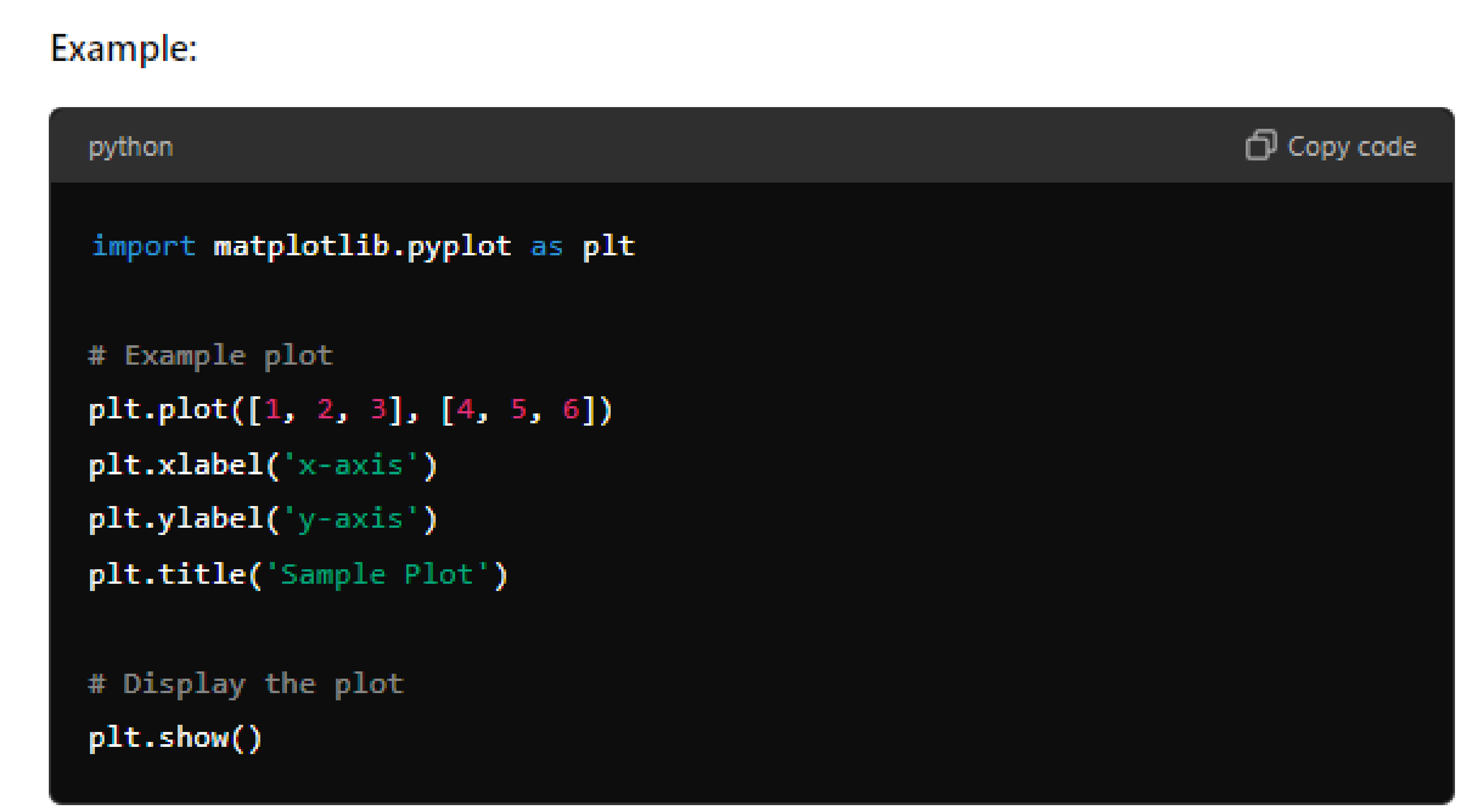
- Check File Content: Open `USAHOUSING NEW.py` to verify its content.



```
PS C:\Users\PANKAJ SHARMA> cd "D:\desktop\ML\pythonProject\Dummy_Project"
PS D:\desktop\ML\pythonProject\Dummy_Project> dir


    Directory: D:\desktop\ML\pythonProject\Dummy_Project


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        21-07-2024     07:24              0 housing_price.py
-a----        21-07-2024     08:39           1103 lasso_model.pkl
-a----        21-07-2024     08:39           1040 linear_regression_model.p
                                                  kl
-a----        21-07-2024     08:39           1003 ridge_model.pkl
-a----        21-07-2024     08:39            860 sgd_model.pkl
-a----        13-07-2024     17:04        1267973 USAHOUSING NEW.ipynb
-a----        21-07-2024     08:33          23843 USAHOUSING NEW.py
-a----        17-05-2024     22:16         572779 USA_Housing.csv
```

- **Objective:** Modify the script to be compatible with Python script execution.
- **Remove Jupyter-specific Commands:**
  - **Remove %matplotlib inline** (used in Jupyter notebooks).
- **Add Plot Display Command:**
  - **Use plt.show()** to display plots.

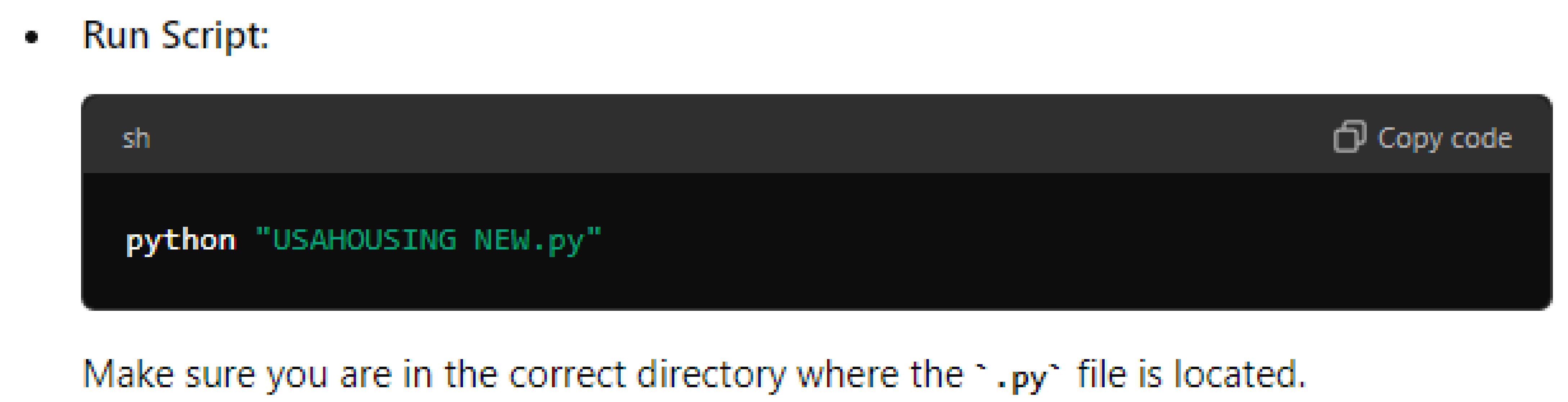
ADJUST THE PYTHON SCRIPT FOR NON-JUPYTER ENVIRONMENT



Example:

```python
import matplotlib.pyplot as plt

# Example plot
plt.plot([1, 2, 3], [4, 5, 6])
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.title('Sample Plot')

# Display the plot
plt.show()
```

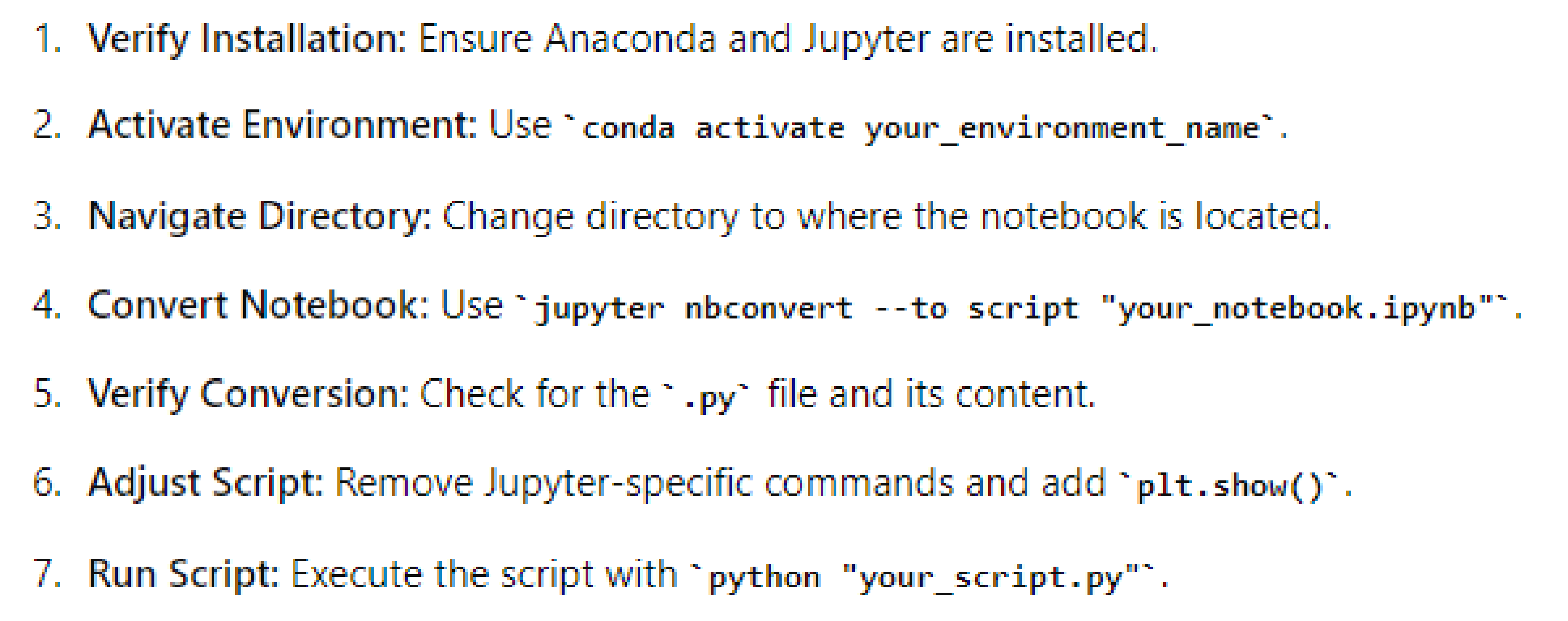
- **Objective:** Execute the converted Python script to ensure it works as expected.



- Run Script:

```sh
python "USAHOUSING NEW.py"
```

Make sure you are in the correct directory where the `.py` file is located.



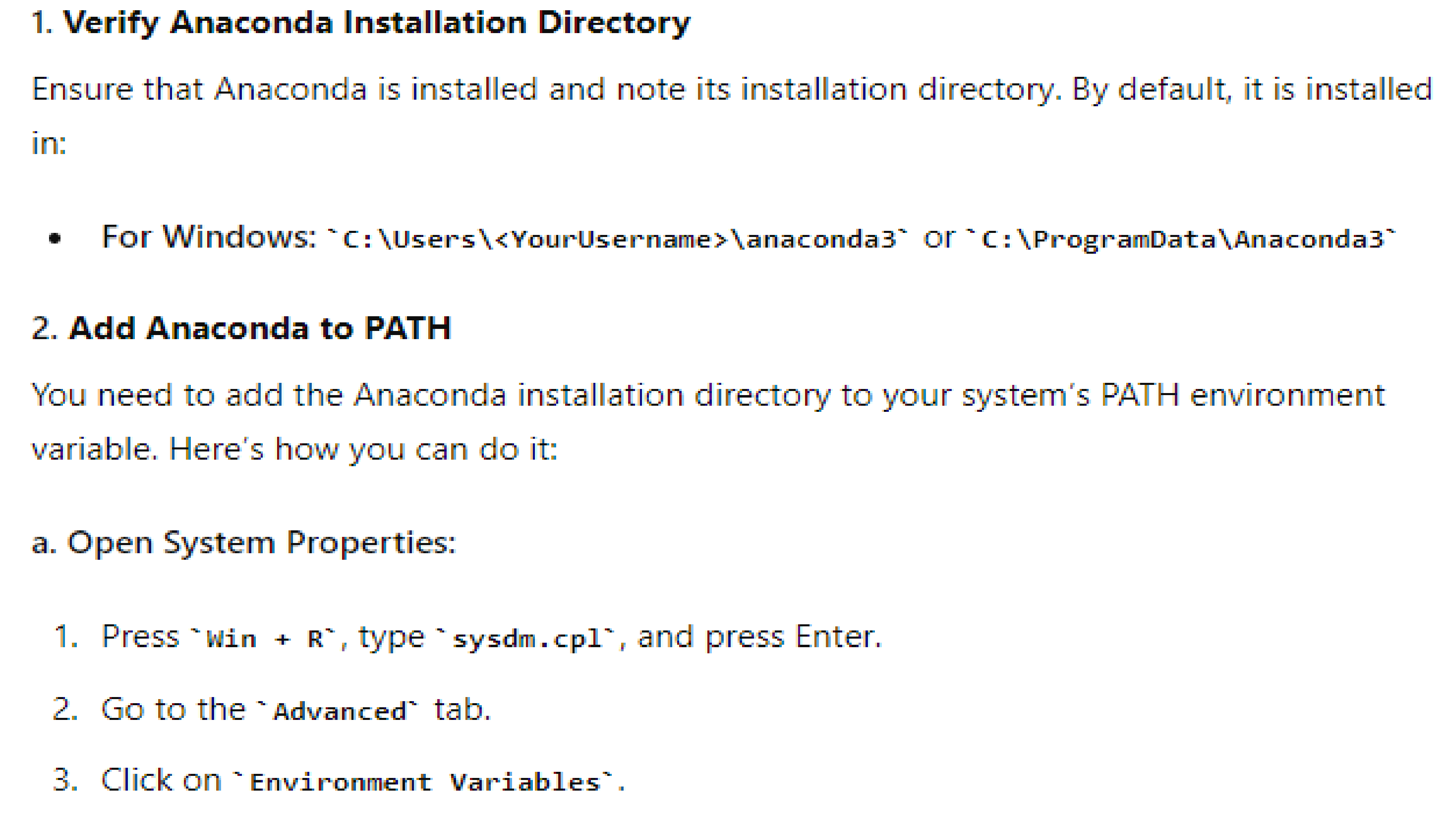
1. **Verify Installation:** Ensure Anaconda and Jupyter are installed.
2. **Activate Environment:** Use `conda activate your_environment_name`.
3. **Navigate Directory:** Change directory to where the notebook is located.
4. **Convert Notebook:** Use `jupyter nbconvert --to script "your_notebook.ipynb"`.
5. **Verify Conversion:** Check for the `.py` file and its content.
6. **Adjust Script:** Remove Jupyter-specific commands and add `plt.show()`.
7. **Run Script:** Execute the script with `python "your_script.py"`.

SUMMARY

- If you encounter an error indicating that conda is not recognized:
- **Check Anaconda Installation:** Make sure Anaconda is properly installed.
- **Check PATH Environment Variable:** Ensure the Anaconda installation directory is added to your system PATH.

TROUBLESHOOTING

Steps to Resolve conda Not Recognized Issue



1. Verify Anaconda Installation Directory

Ensure that Anaconda is installed and note its installation directory. By default, it is installed in:

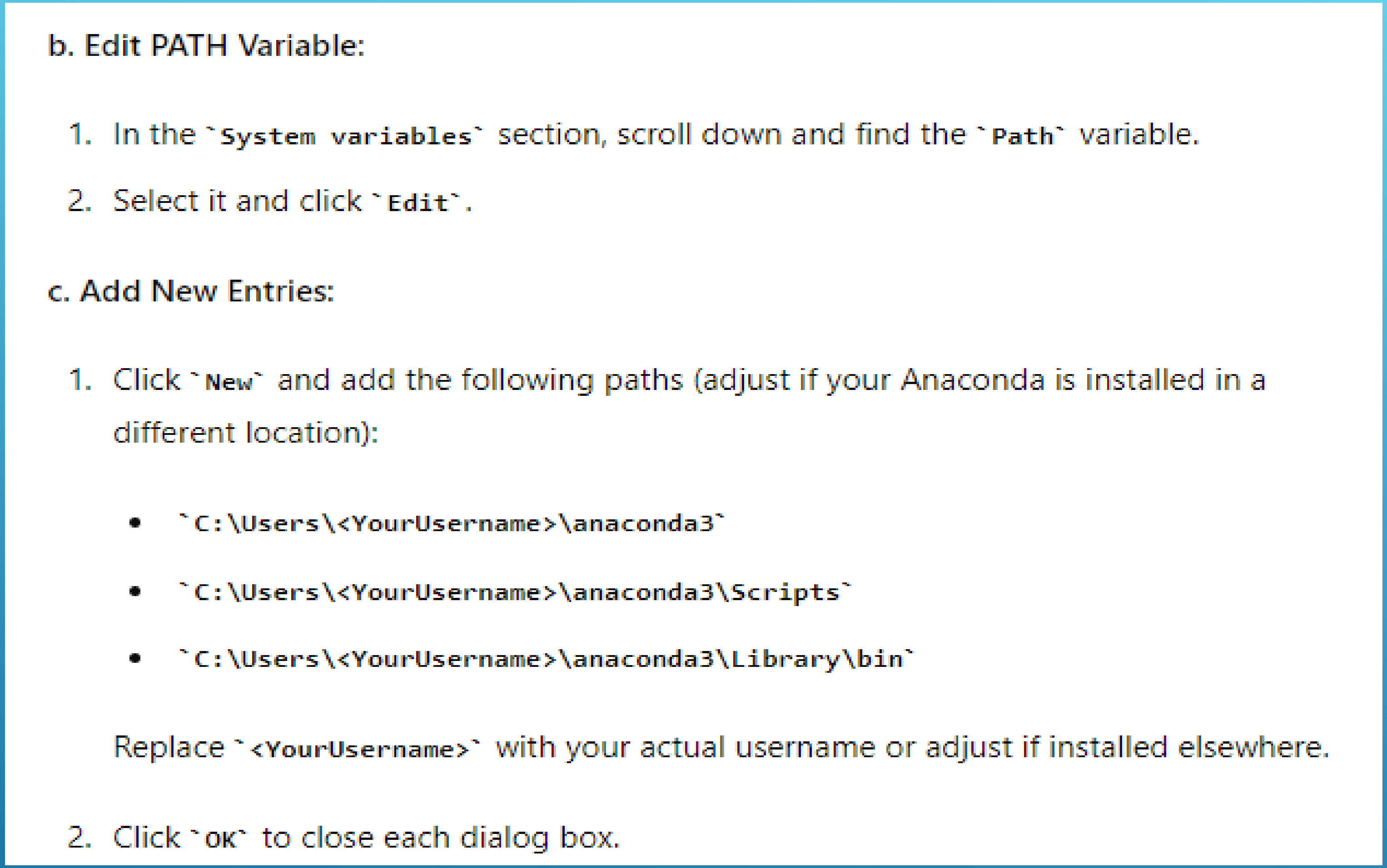
- For Windows: `C:\Users\<YourUsername>\anaconda3` Or `C:\ProgramData\Anaconda3`

2. Add Anaconda to PATH

You need to add the Anaconda installation directory to your system's PATH environment variable. Here's how you can do it:

a. Open System Properties:

1. Press `Win + R`, type `sysdm.cpl`, and press Enter.
2. Go to the `Advanced` tab.
3. Click on `Environment Variables`.



b. Edit PATH Variable:

1. In the `System variables` section, scroll down and find the `Path` variable.
2. Select it and click `Edit`.

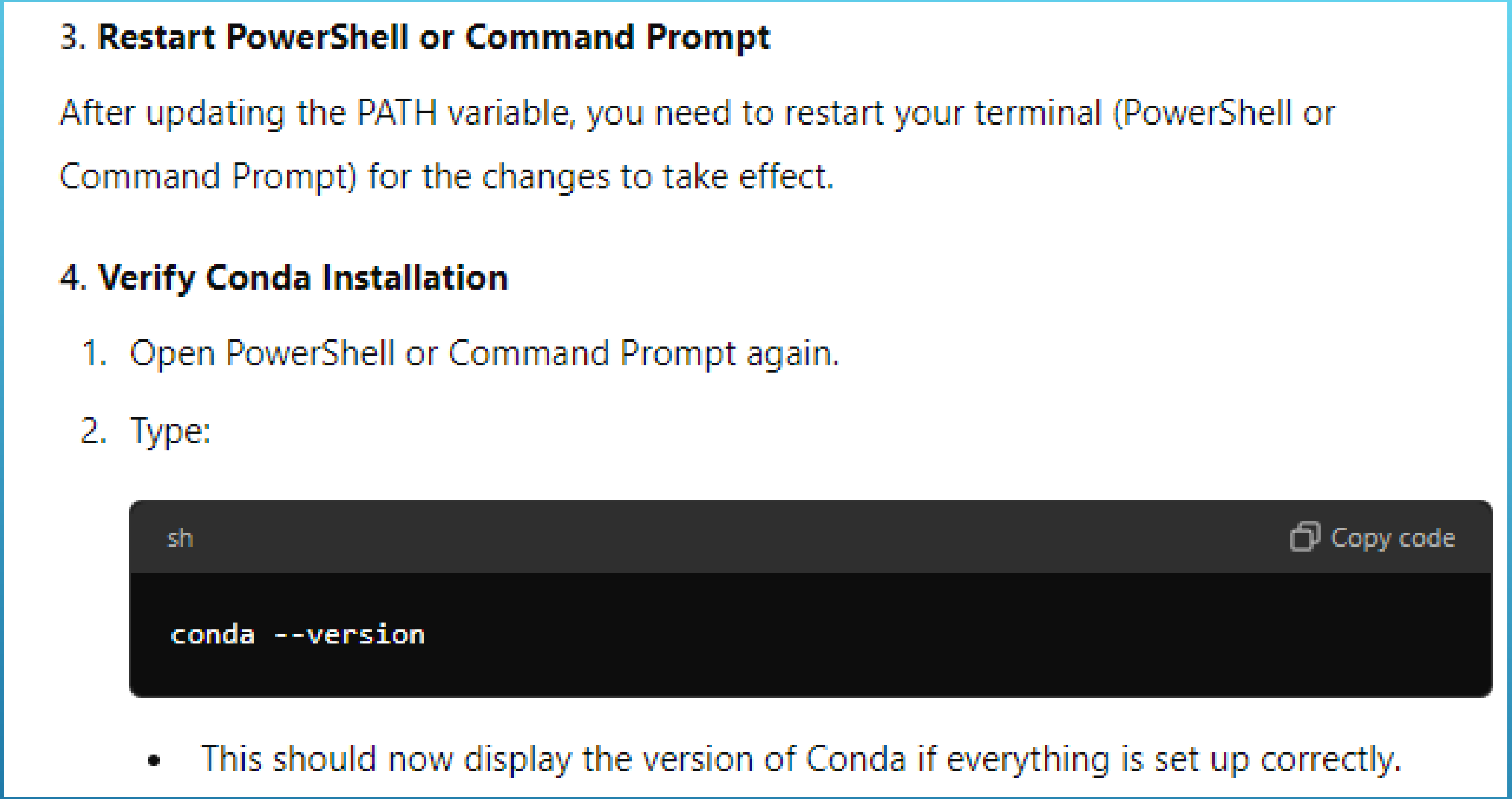
c. Add New Entries:

1. Click `New` and add the following paths (adjust if your Anaconda is installed in a different location):

   - `C:\Users\<YourUsername>\anaconda3`
   - `C:\Users\<YourUsername>\anaconda3\Scripts`
   - `C:\Users\<YourUsername>\anaconda3\Library\bin`

   Replace `<YourUsername>` with your actual username or adjust if installed elsewhere.

2. Click `OK` to close each dialog box.



3. **Restart PowerShell or Command Prompt**

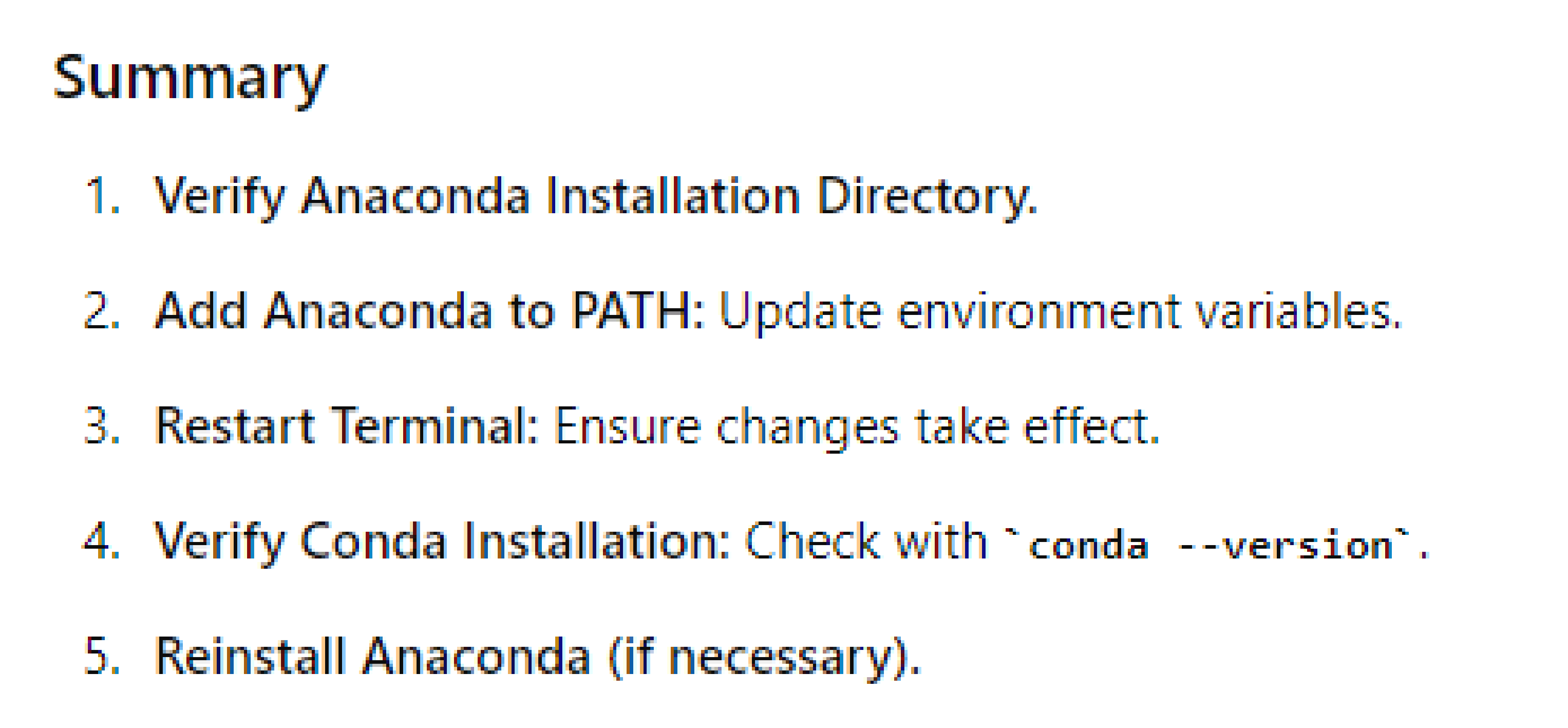
After updating the PATH variable, you need to restart your terminal (PowerShell or Command Prompt) for the changes to take effect.

4. **Verify Conda Installation**

1. Open PowerShell or Command Prompt again.
2. Type:

```sh
conda --version
```

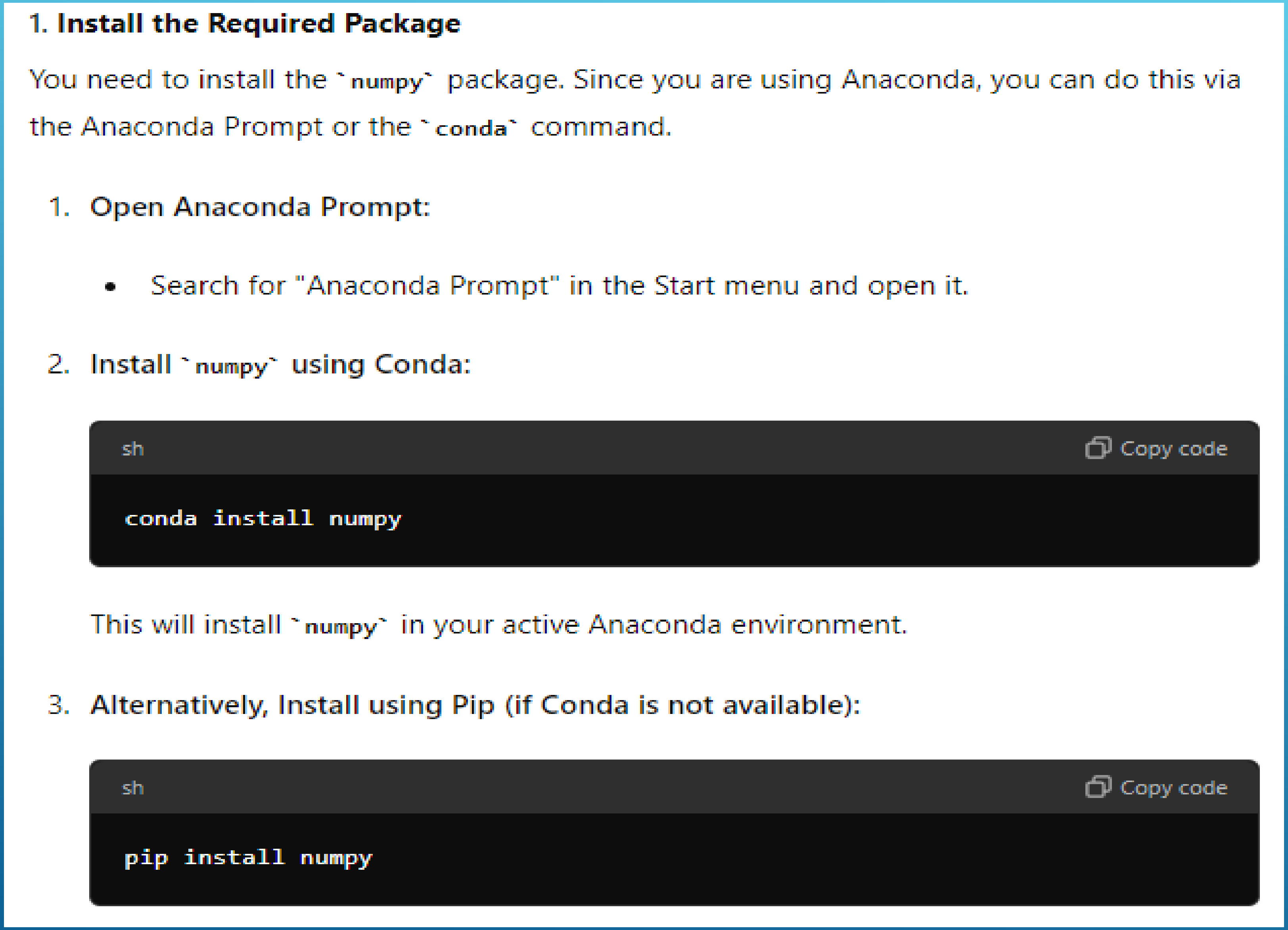
- This should now display the version of Conda if everything is set up correctly.



Summary

1. **Verify Anaconda Installation Directory.**
2. **Add Anaconda to PATH:** Update environment variables.
3. **Restart Terminal:** Ensure changes take effect.
4. **Verify Conda Installation:** Check with `conda --version`.
5. **Reinstall Anaconda (if necessary).**

Steps to Resolve ModuleNotFound Error



1. Install the Required Package

You need to install the `numpy` package. Since you are using Anaconda, you can do this via the Anaconda Prompt or the `conda` command.

1. Open Anaconda Prompt:
    - Search for "Anaconda Prompt" in the Start menu and open it.
2. Install `numpy` using Conda:

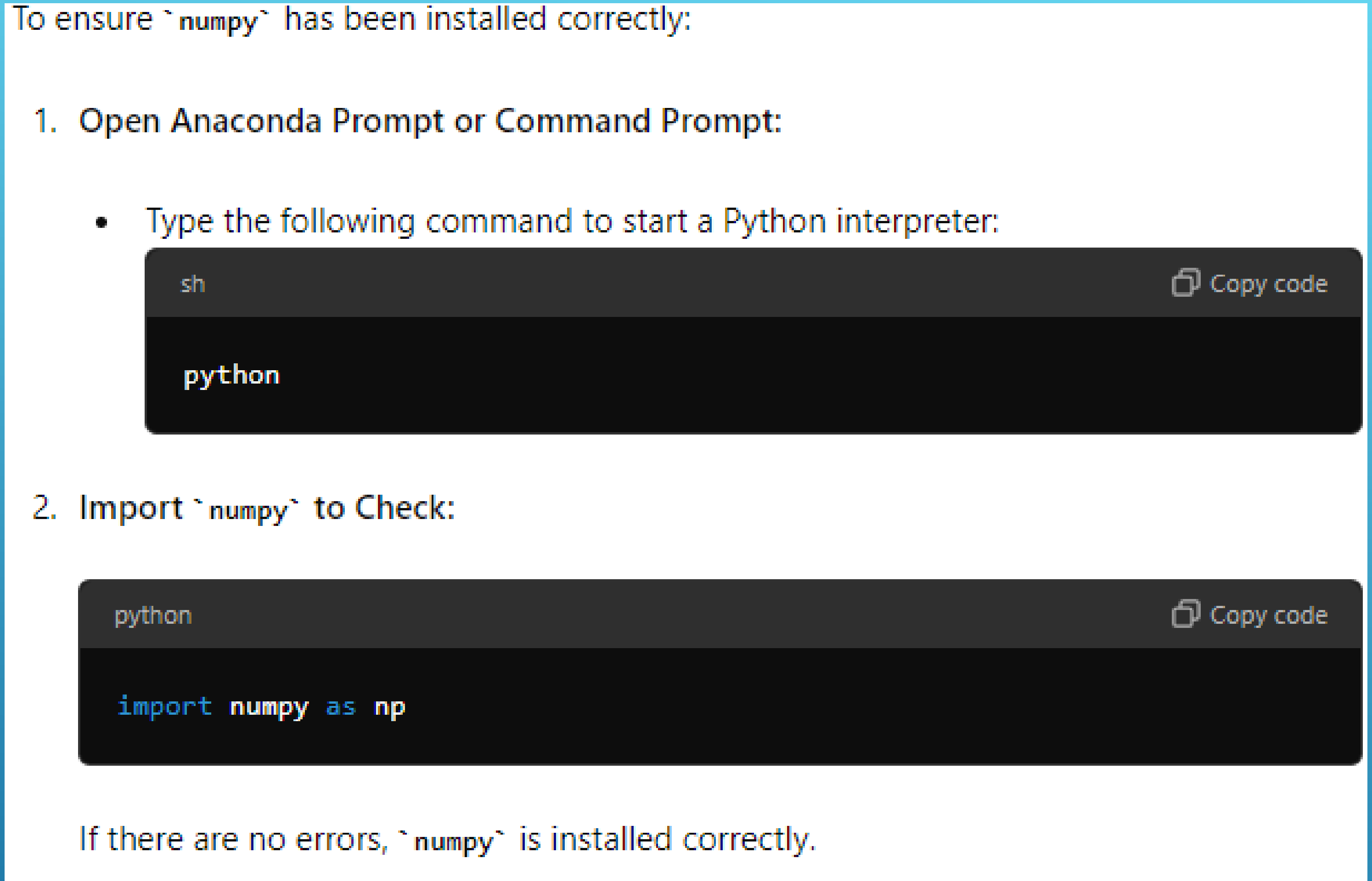
```sh
conda install numpy
```

This will install `numpy` in your active Anaconda environment.

3. Alternatively, Install using Pip (if Conda is not available):

```sh
pip install numpy
```

VERIFY THE INSTALLATION



To ensure `numpy` has been installed correctly:

1. Open Anaconda Prompt or Command Prompt:
   - Type the following command to start a Python interpreter:

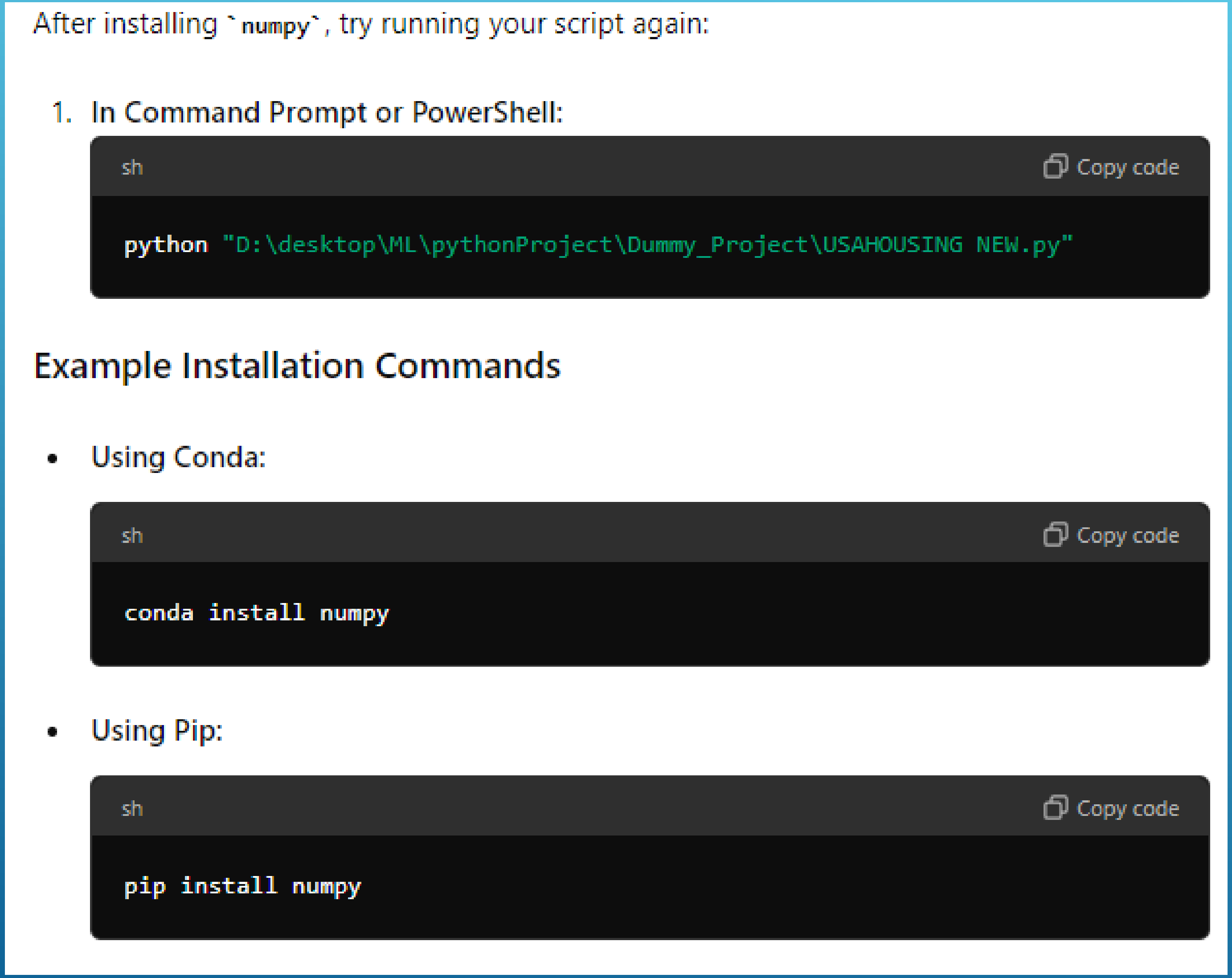
```sh
python
```

2. Import `numpy` to Check:

```python
import numpy as np
```

If there are no errors, `numpy` is installed correctly.

RUN YOUR SCRIPT AGAIN



After installing `numpy`, try running your script again:

1. In Command Prompt or PowerShell:

```sh
python "D:\desktop\ML\pythonProject\Dummy_Project\USAHOUSING NEW.py"
```

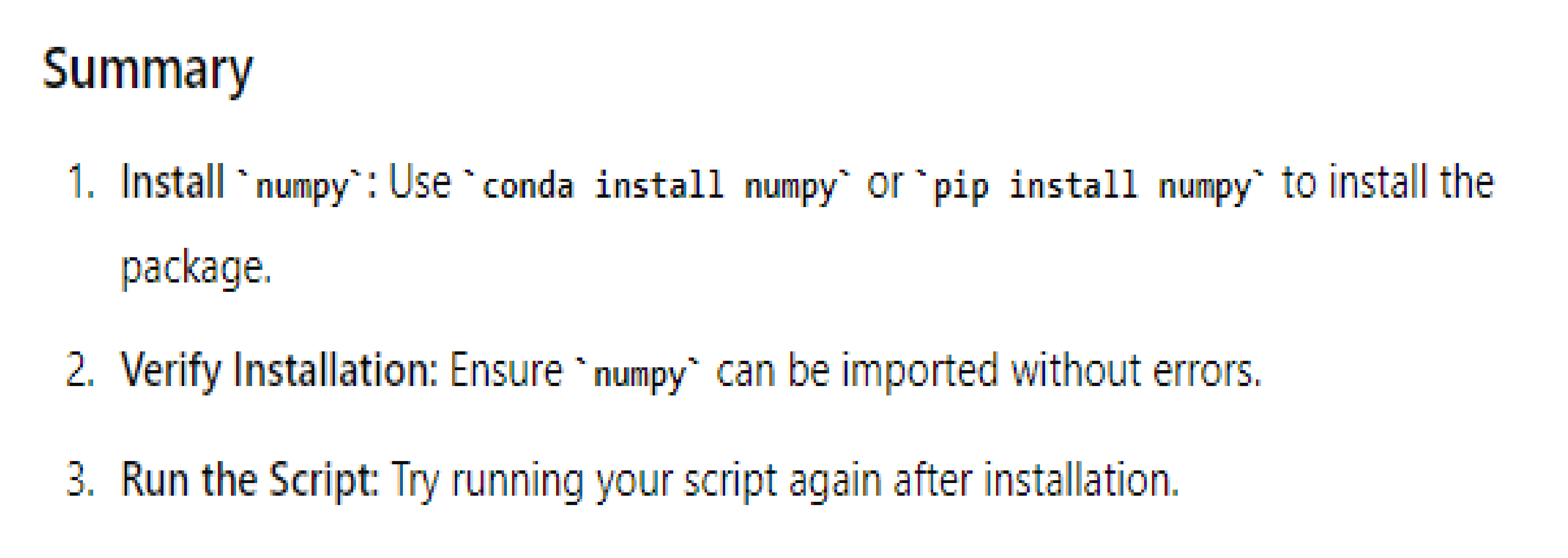
Example Installation Commands

- Using Conda:

```sh
conda install numpy
```

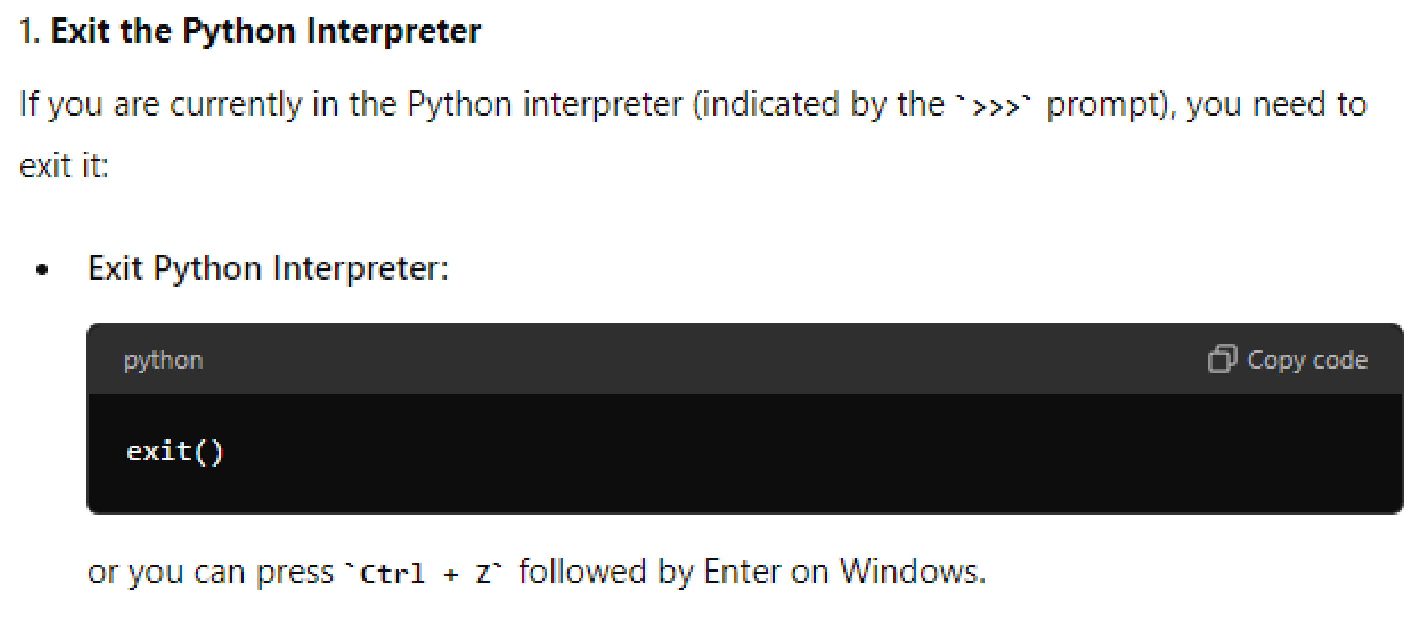
- Using Pip:

```sh
pip install numpy
```



Summary

1. **Install `numpy`:** Use `conda install numpy` or `pip install numpy` to install the package.
2. **Verify Installation:** Ensure `numpy` can be imported without errors.
3. **Run the Script:** Try running your script again after installation.



1. **Exit the Python Interpreter**

If you are currently in the Python interpreter (indicated by the `>>>` prompt), you need to exit it:

- Exit Python Interpreter:

```python
exit()
```

or you can press `Ctrl + Z` followed by Enter on Windows.