

Guide: Installing Anaconda, Python and Jupyter Notebooks in Visual Studio Code

This guide walks new students through installing **Anaconda** (which includes Python), verifying the installation, and setting up **Jupyter Notebooks** inside **Visual Studio Code (VS Code)**. Sections are provided for **Windows** and **macOS**, and a summary table at the end highlights essential commands. The instructions follow recommendations from the official Anaconda and Visual Studio Code documentation.

Introduction

Anaconda is a free distribution of Python and R packaged with hundreds of data-science libraries. Installing Anaconda provides the Python interpreter, the `conda` package manager and a graphical interface (Anaconda Navigator) for managing environments and launching tools such as Jupyter Notebook. **Jupyter Notebooks** are interactive documents that mix code and explanatory text. Visual Studio Code is a lightweight code editor that supports Python and Jupyter via extensions. The steps below show how to install and integrate these tools.

Windows setup

1. Download the Anaconda installer

1. Visit the official download page (`anaconda.com/download`) and click **Download** under "Distribution Installers" ¹. Choose the **Windows** installer that matches your system (64-bit is standard). Registration is optional.
2. Save the `.exe` file to your Downloads folder. If you wish, verify the installer's integrity by opening **PowerShell** and running `Get-FileHash <installer> -Algorithm SHA256` ². Compare the hash with the official value listed in Anaconda's archive ³.

2. Run the installer

1. In File Explorer, double-click the downloaded installer. Do **not** run it from the "Favorites" folder because this can cause permission problems ⁴.
2. On the first screen click **Next**, then accept Anaconda's Terms of Service ⁵.
3. Choose **Just Me** (recommended) or **All Users**. "Just Me" installs Anaconda for the current account, while "All Users" requires administrator privileges ⁶.
4. Select an installation directory (avoid spaces or special characters) ⁷. Installing inside a path with spaces may cause issues with open-source tools.
5. In the **Advanced Installation Options** screen:
6. Leave **Create shortcuts** checked to add Start Menu entries ⁸.
7. **Do not** select "Add Anaconda3 to my PATH environment variable" unless you understand the security implications ⁹. Adding Anaconda to `PATH` exposes internal binaries to other programs ¹⁰.
8. Leave "Register Anaconda3 as my default Python 3.x" checked so that VS Code and other IDEs use Anaconda's Python interpreter ¹¹.

- Click **Install** and wait a few minutes ¹². When finished, click **Next**, then **Finish**. Launching **Anaconda Prompt** from the Start Menu will open a command line configured for `conda` ¹³.

3. Verify the installation

- Open **Anaconda Navigator** from the Start Menu. If the graphical interface launches, Anaconda has been installed correctly ¹⁴.
- From the **Anaconda Prompt**, check that `conda` is available by running `conda --version` or `conda help`. The `conda help` command prints a list of available commands (e.g. `create`, `install`, `list`) ¹⁵. Seeing this output confirms that the package manager is functioning.
- Verify Python by typing `python --version` in the command prompt. The installed version number (e.g. "Python 3.12") should appear ¹⁶. If the command is not recognised, you may need to reopen the Anaconda Prompt.

4. Install Visual Studio Code

- Download Visual Studio Code from the official site (`code.visualstudio.com`) and run the installer. Accept the defaults. VS Code is free and cross-platform ¹⁷.
- Launch VS Code and open the **Extensions** view (Ctrl+Shift+X). Search for **Python** (published by Microsoft) and click **Install**. This extension provides Python language support and integrates the Jupyter Notebook experience inside VS Code ¹⁸. The Python extension includes the Jupyter functionality, so a separate Jupyter extension is no longer required.
- When prompted, allow VS Code to install recommended tools such as `pylance` for code completion.

5. Create a conda environment in VS Code (optional but recommended)

Creating project-specific environments ensures that dependencies are isolated. You can create environments either from within VS Code or via Anaconda Prompt.

Create an environment using VS Code

- Press **Ctrl+Shift+P** to open the Command Palette ¹⁹.
- Type **Conda** and select **Conda: Create...** (VS Code may show "Conda: Creates a `.conda` environment in the current workspace") ²⁰.
- Choose a Python version. VS Code will create a new environment in the `.conda` folder of your workspace.

Create an environment using Anaconda Prompt

Alternatively, run the following command in the Anaconda Prompt:

```
conda create --name myenv python=3.12
```

Replace `myenv` with a meaningful environment name and select the Python version you need. Activate the environment using `conda activate myenv` before installing packages. After creating the environment, you can select it in VS Code using **Python: Select Interpreter** (see below).

6. Select the Anaconda interpreter in VS Code

- Open any Python or Jupyter file.

2. Press **Ctrl+Shift+P** and run **Python: Select Interpreter**. VS Code will display a list of available interpreters ²¹.
3. Choose the interpreter associated with your conda environment (it appears as `Python 3.x (<env-name>: conda)`) ²². If the environment does not appear, click **Refresh** ²².
4. The selected interpreter is shown in the VS Code status bar. From now on, scripts run in this environment.

7. Create and run Jupyter Notebooks in VS Code

1. Ensure that the environment is active and that the `jupyter` package is installed (`conda install jupyter`) if necessary.
2. Press **Ctrl+Shift+P** and execute **Create: New Jupyter Notebook** or create a new file with the `.ipynb` extension ²³.
3. Select a kernel from the top-right kernel picker. The kernel corresponds to your Python interpreter and environment ²⁴.
4. Add code to the cell and click the **Run** icon to execute it. You can run a selected cell with `Ctrl+Enter` or `Shift+Enter` and run all cells from the toolbar ²⁵.
5. Save your notebook with **Ctrl+S** and export it as a Python script, HTML or PDF via **... > Export** ²⁶.

8. Troubleshooting notes

- **Workspaces and trust:** If VS Code opens a notebook in an untrusted workspace, execution will be disabled. Mark the folder as trusted via the VS Code prompt to enable cell execution ²⁷.
- **PATH issues:** If the `python` command is not recognised outside Anaconda Prompt, you may need to add the Anaconda `Scripts` directory to your PATH. However, this is not recommended during installation because of potential security issues ¹⁰.

macOS setup

1. Download the installer

1. Visit `anaconda.com/download`, register if desired and click **Download for Mac** under “Distribution Installers” ²⁸. Choose the installer for your processor (Apple Silicon or Intel). Note that after 15 August 2025 Anaconda no longer produces new Intel Mac installers ²⁹, but existing installers remain available.
2. Optionally verify the installer’s integrity. Open **Terminal** (found in Applications → Utilities) and run:

```
shasum -a 256 ~/Downloads/<installer.pkg>
```

Compare the computed hash with the official hash in the Anaconda archive ³⁰. Matching hashes confirm that the file is not corrupted.

2. Run the macOS graphical installer

1. Double-click the `.pkg` file in your Downloads folder and click **Continue** ³¹.
2. Read the **Read Me** and click **Continue**. Review Anaconda’s Terms of Service and click **Agree** ³².

3. Choose an install location. "Install for all users of this computer" installs Anaconda in `/opt/anaconda3` (recommended). Selecting a specific disk allows installation in a different folder ³³. If the installer skips the destination selection page (a bug in some versions), click **Change Install Location...** to specify a location ³⁴.
4. Click **Install**. The process may take several minutes ³⁵. When complete, click **Continue**, then **Close** to exit the installer ³⁶.

3. Verify the installation

1. After installation, **Anaconda Navigator** should open automatically ³⁷. If it does not, open it from the Applications folder.
2. Open **Terminal** and run `conda --version` or `conda help` ¹⁵. The help output lists available commands, confirming that `conda` is functioning.
3. Check the Python version with `python --version` or `python3 --version` ³⁸ ¹⁶. macOS often includes a system version of Python; to avoid confusion, always run the Anaconda version from a shell that has `conda` activated (`(base)` will appear in the prompt after installation ³⁹).

4. Install Visual Studio Code

1. Download VS Code for macOS from `code.visualstudio.com` and drag the application to your **Applications** folder.
2. Launch VS Code. Open the **Extensions** view (Cmd+Shift+X), search for **Python** and click **Install** ¹⁸.
3. The Python extension includes Jupyter support, so no separate Jupyter extension is required.

5. Create or use conda environments

Although the base environment works, creating separate environments for projects is considered best practice. Environments can be created from the terminal or within VS Code.

Create an environment via Terminal

Use the following commands:

```
# create a new environment with Python 3.12
conda create --name myenv python=3.12
# activate the environment
conda activate myenv
```

Create an environment in VS Code

1. Press **Cmd+Shift+P** and run **Conda: Create...** ²⁰.
2. Follow the prompts to choose a Python version and environment name. VS Code will create the environment in your workspace.

6. Select the interpreter

1. Open any Python file or a blank notebook.
2. Press **Cmd+Shift+P** and run **Python: Select Interpreter** ²¹.

3. Choose the interpreter associated with your desired conda environment. If your environment does not appear, click **Refresh** ²².

7. Create and run Jupyter Notebooks in VS Code

1. Ensure that your chosen environment has the `jupyter` package installed (`conda install jupyter`).
2. Press **Cmd+Shift+P** and run **Create: New Jupyter Notebook** or create a new `.ipynb` file ²³.
3. Choose a kernel from the top-right kernel picker ²⁴.
4. Write code in the first cell and click the **Run** icon. Use `Shift+Enter` to run a cell and move to the next one ⁴⁰. Run all cells via the toolbar if desired ⁴¹.
5. Save your notebook and export it as needed ²⁶.

8. Notes for macOS users

- After installation, the Terminal prompt may display `(base)` indicating that the base conda environment is active ³⁹. To temporarily disable automatic activation, run `conda config --set auto_activate_base false`.
- If macOS warns that the package is incompatible, consult Anaconda's troubleshooting page ⁴².
- When running notebooks in VS Code, ensure the workspace is trusted; untrusted folders disable code execution ²⁷.

Summary of key commands

Task	Windows command or action	macOS command or action
Verify <code>conda</code> installation	<code>conda --version</code> or <code>conda help</code> in Anaconda Prompt ¹⁵	<code>conda --version</code> or <code>conda help</code> in Terminal ¹⁵
Verify Python version	<code>python --version</code> in Anaconda Prompt ¹⁶	<code>python --version</code> or <code>python3 --version</code> ³⁸
Create conda environment (CLI)	<code>conda create --name myenv python=3.12</code>	<code>conda create --name myenv python=3.12</code>
Activate environment	<code>conda activate myenv</code>	<code>conda activate myenv</code>
Install Jupyter	<code>conda install jupyter</code>	<code>conda install jupyter</code>
Open Command Palette	Ctrl+Shift+P	Cmd+Shift+P
Create notebook in VS Code	Create: New Jupyter Notebook ²³	Same command
Run a cell	Click Run or use <code>Ctrl+Enter</code> / <code>Shift+Enter</code> ⁴⁰	Click Run or use <code>Shift+Enter</code> (macOS uses the same shortcuts)

Conclusion

By following the steps above, new students can set up a robust Python development environment on both Windows and macOS. Anaconda simplifies package management and includes Python out of the box. Visual Studio Code, combined with the Python extension, provides an intuitive editor for writing scripts and working with Jupyter Notebooks. Creating project-specific conda environments helps keep dependencies organised and reproducible, and verifying installations with the commands shown ensures that your tools are ready for coursework or research.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 28 29 30 31 32 33 34 35 36 37 42 Installing
Anaconda Distribution - Anaconda

<https://www.anaconda.com/docs/getting-started/anaconda/install>

15 39 Setup – Introduction to Conda for (Data) Scientists

<https://carpentries-incubator.github.io/introduction-to-conda-for-data-scientists/setup/>

16 38 Check Python Version – How to Check Py in Mac, Windows, and Linux

<https://www.freecodecamp.org/news/check-python-version-how-to-check-py-in-mac-windows-and-linux/>

17 18 19 20 21 22 Visual Studio Code (VS Code) - Anaconda

<https://www.anaconda.com/docs/tools/working-with-conda/ide-tutorials/vscode>

23 24 25 26 27 40 41 Jupyter Notebooks in VS Code

<https://code.visualstudio.com/docs/datascience/jupyter-notebooks>