

Setting up Python on a Mac Computer

The following procedures describe one of many ways to create a setup for coding with Python on a Mac computer. The procedures include:

1. Installing Python, that is, installing the basic tools of the Python coding language
2. Installing Visual Studio (VS) Code, a free integrated development environment for coding from Microsoft
3. Creating a virtual environment for Python coding using conda, a virtual environment and coding package management tool
4. Setting up Jupyter notebooks in VS Code, user-friendly interface for writing code that can support multiple coding languages but is especially common for Python code development

1. Install Python and conda

Go to <https://www.anaconda.com/download>

Register or click on Skip registration.

Scroll down to the Miniconda installers (not Anaconda installers). If you have a newer computer with an Apple silicon processor, click on the Apple silicon graphical installer. If you have an older computer with an Intel processor, click on the Intel chip graphical installer. If you need to verify which one you have, click on the Apple menu, select "About This Mac," and look at the processor name.

Go to <https://docs.anaconda.com/miniconda/install/>

Open the macOS/Linux installation dropdown.

Open the dropdown under step 2 and follow the instructions to verify installer. You can use the site <https://text-compare.com/> to compare the hash values.

Once you verify the installer you downloaded, you can double-click on it to run it and follow the prompts to install.

2. Install VS Code

Go to <https://code.visualstudio.com/Download>. Download and run the .zip installer for Mac, again, selecting the installer for your processor type.

Double-click the .zip file to expose the VS Code app.

Double-click the app to run it.

3. Create / customize a virtual environment

Open VS Code.

Open the Extensions sidebar and install the "Python" extension.

Open a terminal in VS Code (*View > Terminal* or the keyboard shortcut `ctrl+``).

Confirm VS Code can see your miniconda installation by showing your miniconda version:

```
conda -V
```

Confirm your terminal is utilizing your base (i.e., default) conda environment.

```
conda env list
```

You should see your base environment listed, and an asterisk will indicate it is currently active.

You can then install several useful Python libraries and their dependencies to your base environment with the following commands. You can just press the enter key when the Y/N prompts come up. Each command takes a little time to complete. Installing packages via conda is slower than installing packages via pip. Using conda is ideal because it has more robust management of package dependencies to try to ensure all your code libraries are compatible. However, the latest versions of packages are not always available through conda, so when a recent package version is required, pip may be needed.

```
conda install -c conda-forge pandas
```

```
conda install -c conda-forge scipy
```

```
conda install -c conda-forge seaborn
```

```
conda install -c conda-forge scikit-learn
```

```
conda install -c conda-forge jupyterlab
```

At any point, you can use the following command to see all installed libraries in the current/active environment and their versions.

```
conda list
```

4. Set up Jupyter notebooks in VS Code

Open the Extensions sidebar in VS Code and install the “Jupyter” extension.

Open the Command Palette (*View > Command Palette* or keyboard shortcut shift+cmd+p). Start typing “create jupyter” to find and run the command *Create: New Jupyter Notebook*.

In the top right corner of the notebook, click on Select Kernel, select Python Environments..., then select your base conda environment. Your notebook can now use that base environment to execute Python code.