# How to Setup Python for Atmospheric, Ocean and Climate Sciences

wxguy

2022-05-19 18:00:00 +0530

## Contents

---

## Why Python for Atmospheric and Ocean Sciences?

There is plenty of articles that describe why Python is great for use in Atmospheric, Oceanography, Climate, and Geoscience applications. But there is a popular article written by Prof. Johnny Wei -Bing Lin in Dec 2012 issue of the Bulletin of the American Meteorological Society on Why Python Is the Next Wave in Earth Sciences Computing way back in 2012. You can download and read this article from here to know more about it. Simply put, it is easy to install, configure, code, and learn, has more scientific packages, and has better support from the community. Therefore, there is no doubt that Python is one of the greatest languages for all types of scientific analysis and you should install it right away. There is a number of ways you can install and configure your Python installation such as package manager provided by your Distros, Python virtual environment, pip, and Python Distributions (Anaconda, ActiveState, PyPy, etc.,). That leads to the next question:

## Which Python Distribution Should I Choose?

It all depends on your usage. If your main interest is scientific, then you should choose Miniconda or Anaconda. If you work mostly on Windows-based applications, then choose ActiveState. Finally, if you want your nonscientific applications to work faster, then choose PyPy distribution. I have worked in all three Python distributions and believe me Miniconda distribution has more libraries compiled for many hardware architectures than the other Python distribution out there. Also, the dependency resolution of the `conda` package manager is top-notch. The documentation and help from the community are also very important to decide which Distribution to choose. In this too, Miniconda/ Anaconda excels.

Miniconda also has its own issue. Since `conda` try to include all necessary libs and hence no of packages are higher and install size would be bigger. In addition, `conda` sometimes takes too long to resolve dependencies.

If you want to quickly set up your Python distribution for all kinds of applications and are not worried about the size, then you must choose Miniconda.

## Choosing Best Editor

Python is strict in enforcing the way you write your programs. During your initial days, you will encounter many issues which are easily solvable. Choosing the best Integrated Development Environment (IDE) or Editor would solve a lot of issues and make you focus on content rather than debugging silly errors. When it comes to IDEs or Editors, you have plenty of options. But based on my experience, I would list PyCharm as best, Sublime Text as next, and VS Code Editor or Spyder as the last options. Choosing from these options also depends on many factors. If your system is new and has enough hardware resources, then I would recommend you install PyCharm. Otherwise, Sublime Text with `Anaconda` + `Tabnine` plugins is highly recommended. There are plenty of tutorials available for you to refer to. I will briefly touch on how to install PyCharm, Sublime Text, and Spyder in subsequent sections.

## Install IDEs and Text Editors on Ubuntu

Installing IDEs or Text Editors require you to issue a few set of commands in the terminal. We will be installing each one by one.

### Install Sublime Text

To make Sublime Text into truly impressive IDE, you need to install two plugins. All are explained below. For the official installation guide, please refer https://www.sublimetext.com/docs/linux_repositories.html which will have always an updated guide. Let’s install the Editor first by importing the gpg key to trust the Sublime Text official repo.

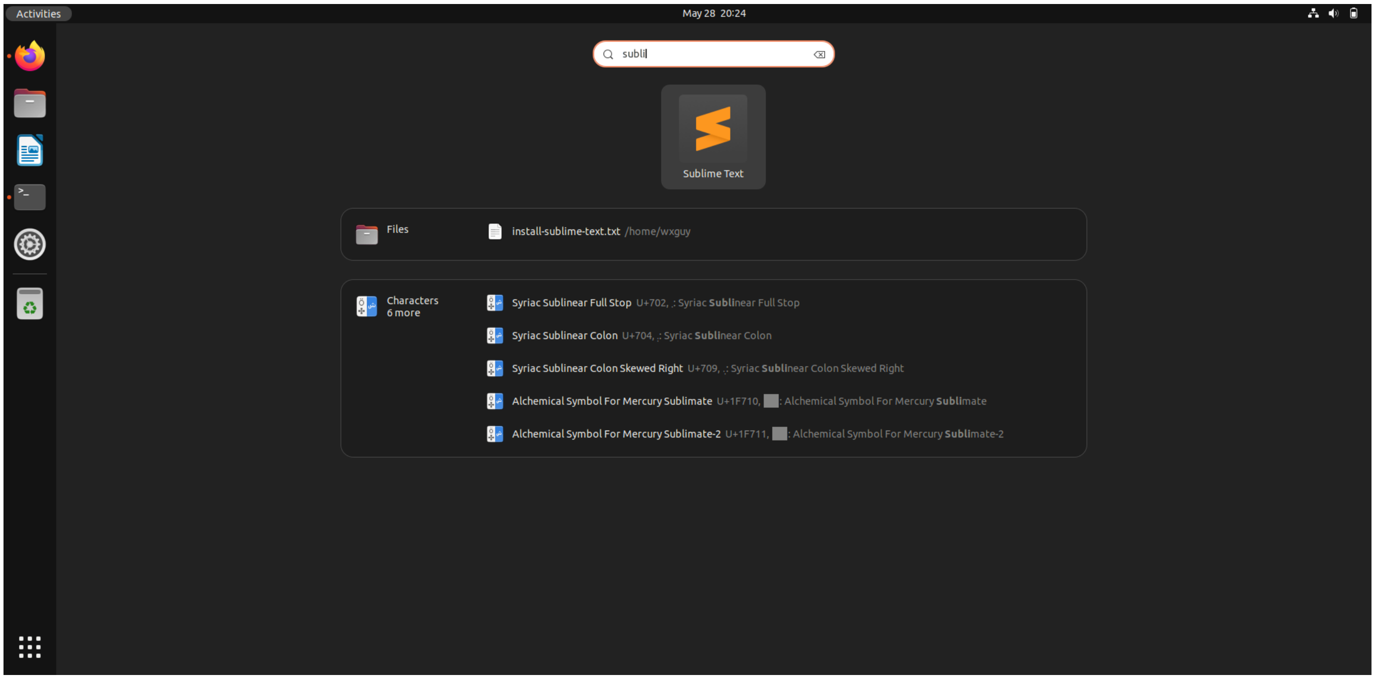
```
wget -qO - https://download.sublimetext.com/sublimehq-pub.gpg | gpg
 ↪  --dearmor | sudo tee /etc/apt/trusted.gpg.d/sublimehq-archive.gpg
```

Register Sublime Text official repo into the local system so that you can get a latest stable version of Editor whenever it is made available.

```
echo deb https://download.sublimetext.com/ apt/stable/ | sudo tee
 ↪  /etc/apt/sources.list.d/sublime-text.list
```

Now update the repo and install `sublime-text` :

```
sudo apt update && sudo apt install sublime-text
```


Activities
May 28 20:24
subli
Sublime Text
Files
install-sublime-text.txt /home/wxguy
Characters
6 more
Syriac Sublinear Full Stop U+702, .: Syriac Sublinear Full Stop
Syriac Sublinear Colon U+704, .: Syriac Sublinear Colon
Syriac Sublinear Colon Skewed Right U+709, .: Syriac Sublinear Colon Skewed Right
Alchemical Symbol For Mercury Sublimate U+1F710, : Alchemical Symbol For Mercury Sublimate
Alchemical Symbol For Mercury Sublimate-2 U+1F711, : Alchemical Symbol For Mercury Sublimate-2

*Open Sublime Text Editor*

You need to install **Package Control** before installing plugins on Sublime Text. Once opening the editor, go to `Tools` --> `Command Palette` --> search `install` --> select `Install Package Control` . This will take a few seconds and install Package Control on your system.

To install the plugin, go to `Tools` --> `Command Palette` --> search `anaconda` --> select the first listed plugin from *damnwidget* developer.

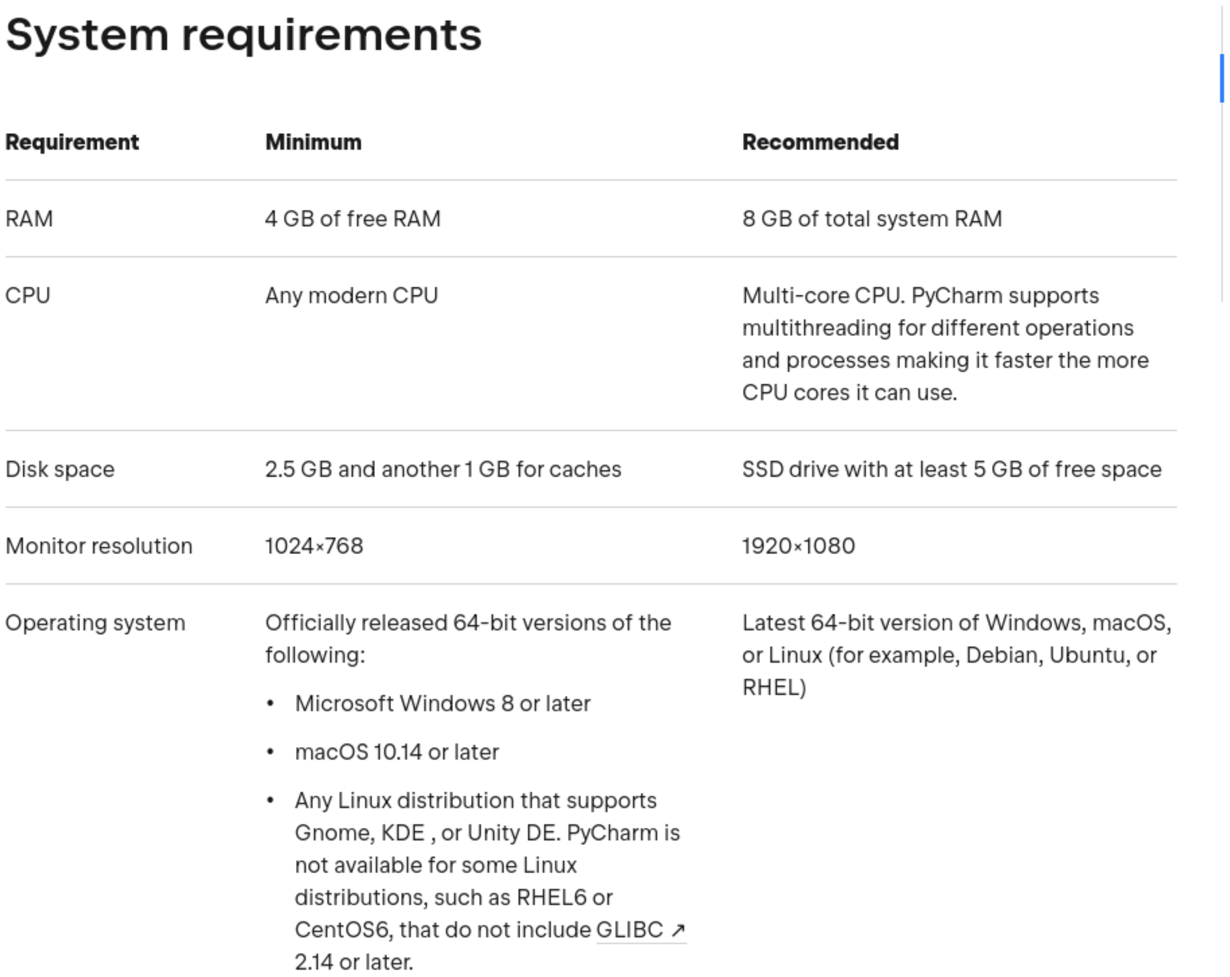
Next, we will install the `Tabnine` plugin. As per their official statement, this is an AI-powered auto-completion plugin to make faster code. In my experience, it provided a lot of good suggestions and I felt that I was working very fast. It is highly recommended for any Python developer. Let's install that too.

To install the `Tabnine` plugin, go to `Tools` --> `Command Palette` --> search `tabnine` --> select the first listed plugin.

You can create a test Python script to start writing some code to see how good it is.

### PyCharm

The installation of PyCharm is straightforward in Ubuntu as it is already an available `snap` store. Before installing the IDE, you must check if your system supports this IDE. You can check the system requirements from here. Some of them are shown below:



## System requirements

| Requirement | Minimum | Recommended |
|---|---|---|
| RAM | 4 GB of free RAM | 8 GB of total system RAM |
| CPU | Any modern CPU | Multi-core CPU. PyCharm supports multithreading for different operations and processes making it faster the more CPU cores it can use. |
| Disk space | 2.5 GB and another 1 GB for caches | SSD drive with at least 5 GB of free space |
| Monitor resolution | 1024×768 | 1920×1080 |
| Operating system | Officially released 64-bit versions of the following:<br>• Microsoft Windows 8 or later<br>• macOS 10.14 or later<br>• Any Linux distribution that supports Gnome, KDE , or Unity DE. PyCharm is not available for some Linux distributions, such as RHEL6 or CentOS6, that do not include GLIBC ↗ 2.14 or later. | Latest 64-bit version of Windows, macOS, or Linux (for example, Debian, Ubuntu, or RHEL) |

*PyCharm System Requirements*

Now install the PyCharm Community edition by issuing the below command:

```
sudo snap install pycharm-community --classic
```

## Install Miniconda

First, you must download the latest version of Miniconda from https://docs.conda.io/en/latest/miniconda.html#linux-installers. I have checked all my libraries and modules are compatible with Python 3.9 from their repo and hance downloaded Miniconda which supports Python 3.9. Check if we have downloaded successfully by issuing the below command:

```
$ cd Downloads/
$ ls
Miniconda3-py39_4.12.0-Linux-x86_64.sh
```

By issuing below one command, it will take you through no of pages you need to accept all.

```
$ bash Miniconda3-py39_4.12.0-Linux-x86_64.sh

Welcome to Miniconda3 py39_4.12.0
```

```
To continue the installation process, please review the license
agreement.
Please, press ENTER to continue
>>
```

After accepting the first screen, it will take you to review EULA. Press the space bar until you see `Do you accept the license terms? [yes|no]` . Enter `yes` to accept the license.

```
======================================
End-User License Agreement - Miniconda
======================================

Copyright 2015-2022, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:
.
.
.
The following packages listed on https://www.anaconda.com/cryptography are included i

Last updated March 21, 2022

Do you accept the license terms? [yes|no]
[no] >>> yes
```

Next, it will inform you of the location where Miniconda will be installed. Just enter to accept the location.

```
Miniconda3 will now be installed in this location:
/home/wxguy/miniconda3

  - Press ENTER to confirm the location
  - Press CTRL-C to abort the installation
  - Or specify a different location below

[/home/wxguy/miniconda3] >>>
```

Accepting the location will take a few min to install all default packages.

```
Preparing transaction: done
Executing transaction: done
installation finished.
Do you wish the installer to initialize Miniconda3
by running conda init? [yes|no]
[no] >>> yes
```

Once it is done, you must close and reopen the terminal to take effect.

```
==> For changes to take effect, close and re-open your current shell. <==
If you'd prefer that conda's base environment not be activated on startup,
   set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

Thank you for installing Miniconda3!
```

Now you can check if conda is installed successfully by issuing the below command:

```
$ which conda
/home/wxguy/miniconda3/bin/conda
```

Now onwards you can start installing modules of your interest with the `conda install applic` command.

## Finding Right Modules

One of the biggest issues you will face is finding the correct Python package or modules relevant to scientific applications. For this purpose, I have already compiled a huge list of modules relevant to Meteorology, Oceanography, Data-Science, Machine Learning, Artificial Intelligence, Plotting, etc., in the blog post located here. Read about the package by referring their official documentation and start installing choice of your interest.

---

You can download this article from here for free.

---