**miniforge** is the community (conda-forge) driven minimalistic conda installer. Subsequent package installations come thus from conda-forge channel.

**miniconda** is the Anaconda (company) driven minimalistic conda installer. Subsequent package installations come from the anaconda channels (default or otherwise).

Uses of Virtual Env

* Virtual env is used **to manage Python packages for different projects**.
* Using virtual env allows you to avoid installing Python packages globally which could break system tools(let us say our windows is using some GUI package of python of certain older version and let us say we installed the latest version of that package globally which don’t have one of the function which was present in the older version then our whole windows can be disrupted) or other projects.

Globally installing Packages

Installing packages and their dependencies to the same global environment **can cause**

* **Version conflicts**
* **Break dependencies that the operating system has on Python packages**

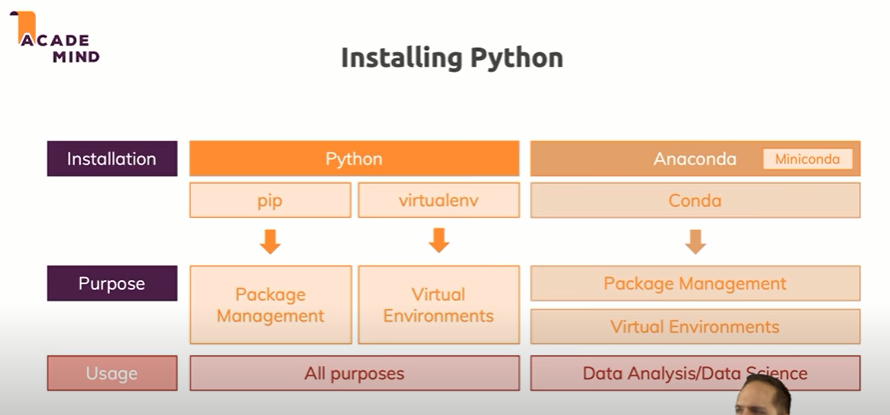
Dependencies pip package have on OS Python

I want to create a pip package which dependent on some OS specific files:

Let's say there are:

* **dependency\_Windows\_x86\_64.zip**
* **dependency\_Linux\_x86\_64.zip**
* **dependency\_MAC\_OS\_X.zip**

I do not want to include all three archives in a package project, but download them dynamically during the pip install my-package based on user's OS.



CONDA VS PIP 🡺 <https://towardsdatascience.com/environments-conda-pip-aaaaah-d2503877884c>

* Conda can make it pretty easy to install a particular python version.
* Whereas pip can create an environment with python versions you have currently installed .
* Many of the data science packages are compiled with intel MKL libraries as a result when using an intel CPU they can operate significantly faster without you as the user having to deal with painful compilation of large code bases for every package you install and then
* Having to install all of them are compatible
* We are all aware of pip, easy install, and virtualenv, but these tools did not meet all of our specific requirements. The main problem is that they are focused around Python, neglecting non-Python library dependencies, such as HDF5, MKL, LLVM, etc., which do not have a setup.py in their source code
* Also do not install files into Python’s site-packages directory.

CAN WE USE CONDA AND SUMULTANEOUSLY🡪

<https://stackoverflow.com/questions/56134588/is-that-a-bad-idea-to-use-conda-and-pip-install-on-the-same-environment>

<https://www.anaconda.com/blog/using-pip-in-a-conda-environment#:~:text=In%20summary%2C%20when%20combining%20conda,to%20install%20any%20remaining%20software>.

Setting virtual env using conda in vscode 🡪 Development…..

<https://code.visualstudio.com/docs/python/environments>

# [Using Pip to install packages to Anaconda Environment](https://stackoverflow.com/questions/41060382/using-pip-to-install-packages-to-anaconda-environment)

<https://stackoverflow.com/questions/41060382/using-pip-to-install-packages-to-anaconda-environment>

You can either use conda all the time or pip in the environment created by pip