

The following text is CUT and PASTED from Ned's January 24, 2016 email: "Don't use Sudo with Pip". A healthy discussion ensued (pip/virtualenv versus Conda)

At the "Structuring Your Python Project" presentation, I said, "Don't ever use 'sudo pip'." I've gotten a few questions about it, so here's more detail:

A python kit includes a file called setup.py. That file is executed to install the kit. A simple setup.py can be almost purely declarative, just providing information needed to install the kit. But it really is code, and it really is executed. A malicious person could upload a kit to PyPI that does damage to your system in its setup.py. If you run "sudo pip install", then you are downloading code and running it as root without knowing what the code does. This is dangerous.

One alternative is to use "pip install --user". This installs the package into your own directory, not the system directories, and so does not need sudo.

Another alternative is to use OS packages: "sudo apt-get python-foobar". The OS packages (I assume) have been vetted, and are safe to use.

The best alternative is to use virtualenv. This has a number of advantages: it separates your installation from the system; it lets you have multiple environments for multiple projects; and it makes it easy to do the right thing for each installation.

To use virtualenv, you may have to install it. You can use "sudo pip install virtualenv" for this :) Then:

```
# Create a directory venv containing the environment:
$ virtualenv venv
# Make venv the current environment:
$ source venv/bin/activate
# Install whatever you like:
$ pip install foobar
```

Once you've activated the environment, "pip" and "python" refer to that environment, and you are isolated from the system installation of Python. You won't need to be root to get your work done.

--Ned.