

Package Management

Package what?

A package is a collection of Python code that usually serves a specific purpose. This is often similar to libraries in Java. Similar to Ubuntu's `apt-get`, Python includes a package manager called `pip`, which we recommend using. To install a package using `pip`, simply run

```
pip install <package name>
```

virtualenv

Good version control practice recommends keeping packages separate for two reasons

- Packages take up space. No point in bogging down a project with packages it doesn't need
- Different projects might need different versions of same package

To manage this, we create virtual environments using the `virtualenv` package:

1. install `virtualenv`: `$ pip install virtualenv`
2. create a new virtual environment: `$ virtualenv -p /usr/local/bin/python3 <new_env_name>`
3. start an environment: `$ source <my_env>/bin/activate`

Now anytime you install a package using `pip`, it will be installed only to your active environment. To exit, run:

```
$ deactivate
```

No root? No problem.

If you don't have root access (eg in attu), you can still use `virtualenv` by downloading `virtualenv-15.1.0.tar.gz` from <https://pypi.python.org/pypi/virtualenv#downloads>, then running:

```
$ tar xvfz virtualenv-15.1.0.tar.gz
```

A new virtual environment can be created from the directory where you unzipped the file by running:

```
$ python virtualenv-15.1.0/virtualenv.py <new_env_name>
```

MacOS Specific Tools

Since a good number of students and researchers use macOS, it will be valuable to go over some macOS specific tools for Python. This is a brief introduction to those tools and we encourage you to search online for additional documentation.

Homebrew

Homebrew is a package manager for macOS. It allows you to easily install most packages/programs that are needed on macOS and keeps their installation files organized.

1. Install Homebrew — visit <https://brew.sh/>.
2. Install a package with homebrew (e.g. the command line utility `wget`) with `brew install wget`.

Pyenv

Pyenv is a Python version manager for macOS. Some packages and pieces of code are only compatible with specific versions of Python and it is very useful to be able to easily change the version of python that your computer is running.

1. Install Pyenv with `$ brew install pyenv` and follow the additional instructions from Homebrew.
2. See all versions of Python that can be installed by pyenv with `$ pyenv install --list`.
3. Install a Python version with `$ pyenv install <version>`.
4. See all currently installed Python versions with `$ pyenv versions`.
5. Set the global (the default python interpreter version) with `$ pyenv global <version>`.

Pyenv-virtualenv

Pyenv-virtualenv is a pyenv plugin to allow pyenv to also manage Python virtual environments.

1. After you have already installed pyenv, install pyenv-virtualenv with `$ brew install pyenv-virtualenv`.
2. After installing the Python version that you would like to create a Python virtual environment from, create the virtual environment with `$ pyenv virtualenv <version> <virtualenv-name>`.
3. Activate the virtual environment with `$ pyenv activate <virtualenv-name>`. Once the environment is activated, you can use pip normally.