Managing Python Packages

Reindert-Jan Ekker
nl.linkedin.com/in/rjekker/
@rjekker

pluralsight
hardcore dev and IT training

Overview

- **Python Package**
  - Finding packages
  - Installing and removing packages

- **Available tools**
  - What to use
  - What not to use

- **This module does NOT tell the whole story**
  - To learn how to install packages correctly, view the next module as well!

How an Installed Package is Found

- **sys.path**
  - List of directories searched for packages
  - First item: current directory (‘.’)
  - Last item: site-packages

- **site-packages**
  - Third-party packages are installed here
  - Location depends on OS, Python version, Installation procedure
  - Will be somewhere *inside* your Python installation
  - May not even be called site-packages (Debian: dist-packages)

Packaging: Current State of Affairs

How to create, distribute and install packages?

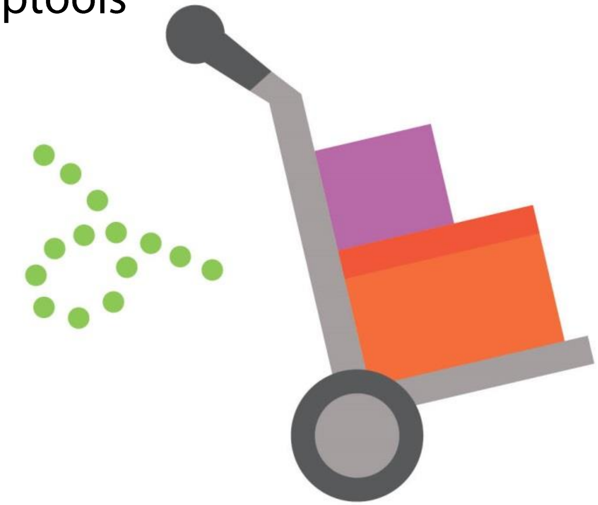
Lots of info on the web is outdated or contradictory

Multiple tools:
distutils, setuptools, easy_install, ez_setup, pip

Multiple formats:
egg, wheel, source dist

Preferred Tools

- **Python Packaging Authority (PyPA)**
- **Creating and distributing packages**
  - □ Use setuptools
  - □ For distutils: discontinued, merged into setuptools
  - □ Easy_install is included in setuptools
  - □ See last module of this course



- **Installing packages:**
  - □ Use pip
  - □ With virtualenv (see next module)

Installing Pip

- **Starting with Python 3.4, pip comes pre-installed**
  - No need to install anything
- **Python 2 or Python 3 < 3.4**
  - Follow instructions at http://www.pip-installer.org/
  - Download and run `get_pip.py`
  - Linux/Mac users may need to use `sudo`
- **Mac users using Homebrew (http://brew.sh):**
  - Homebrew installs pip with Python
- **Windows users:**
  - `pip` will be in the `Scripts` directory of the Python installation
  - Add `C:\PythonXX\Scripts\` to your `PATH`

Installing Packages With Pip

- **Installing a package**
  - pip install requests
  - pip install requests==1.0
  - pip install ‘requests>=2.1’
  - Use quotes to prevent redirection with ‘>’ and ‘<‘
- **Removing a package**
  - pip uninstall requests
- **Be careful!**
  - Don’t start installing lots of packages
  - First learn about virtualenv

Inspecting Packages With pip

- **List all installed packages**
  - pip list

- **Get info on a specific package**
  - pip show requests

- **Search for a package**
  - pip search query

- **Cheese Shop:** http://pypi.python.org/

Requirements Files

- **pip freeze > requirements.txt**
  - Save your project's dependencies in `requirements.txt`
  - Including correct versions

- **pip install –r requirements.txt**
  - Install all packages listed in requirements.txt

- **This is *not* the list of requirements downloaded by pip when installing**

Resources

- **Watch out!**
  - □ There is a *lot* of outdated info on the web.

- **Python Packaging User Guide (http://goo.gl/8o4L4m)**
  - □ https://python-packaging-user-guide.readthedocs.org/en/latest/index.html

- **Pip (http://goo.gl/K7suLs)**
  - □ http://pip.readthedocs.org/en/latest/index.html

- **Python Package Index (PyPI)**
  - □ http://pypi.python.org/

Summary

- **Pip**
  - Find, query, install and remove package
  - Installing pip
  - Freeze requirements