

Managing Python Packages and Virtual Environments

MANAGING PYTHON PACKAGES WITH PIP



Reindert-Jan Ekker

@rjekker <http://nl.linkedin.com/in/rjekker>



Overview



Dependencies

Packages

- Find, download and install
- Manage

Virtual environments

- Project dependencies
- Create, use and manage

Best practices

Recommended tools

NOT: creating and shipping



Modules in this Course

pip

Package management

Virtual environments

Projects and dependencies

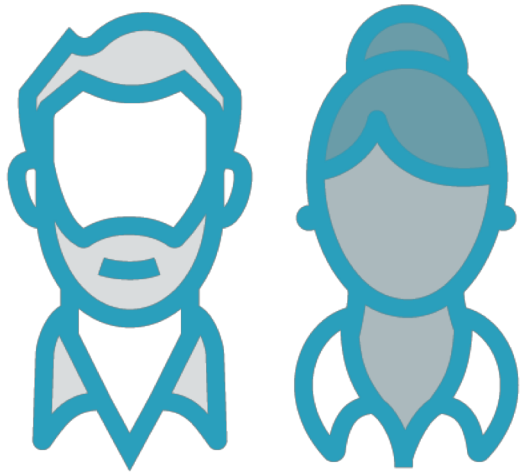
virtualenvwrapper

Making venv more convenient

Other tools



Audience

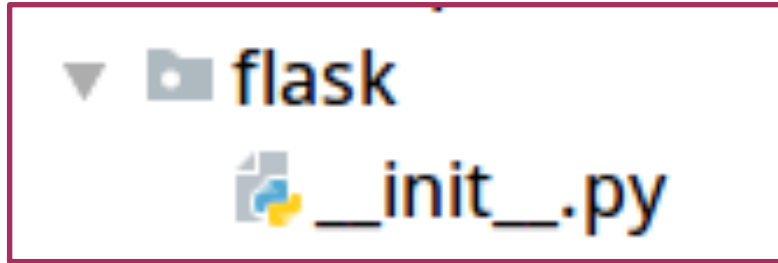


New Python developers

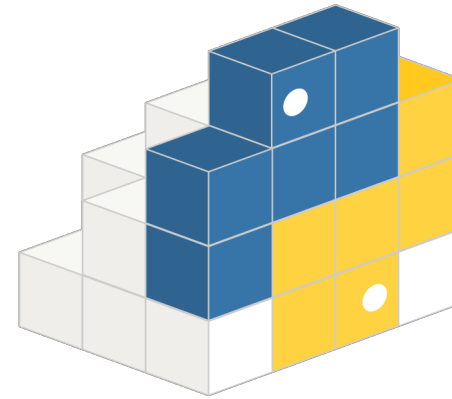
- Go from simple scripts to larger projects
- Use third-party modules
- Basic Python skills
- Python installed



A Package Is Not a Package



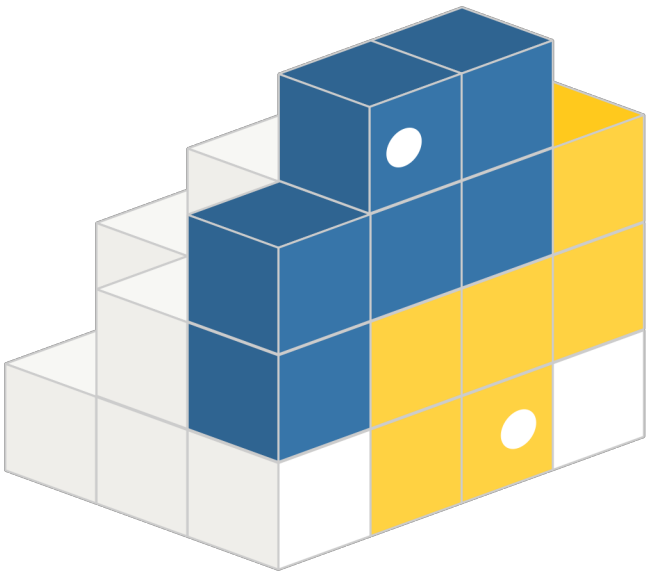
A directory containing
an `__init__.py` file
Used with `import`



A Python
distribution package



Python Distribution Package



Versioned archive file

Contains Python packages, modules, and other resource files

Used to distribute a Release

What an end-user will download and install



Demo



Getting started

- Project with a third-party dependency
- Check for presence of pip



Do I Need to Install pip?



pip comes pre-installed with

- Python 2 $\geq 2.7.9$
- Python 3 ≥ 3.4
- If you downloaded from python.org

Older versions

- Go to <http://pip-installer.org>
- Follow instructions under "Installation"
- If you downloaded from python.org

Demo



Windows

- Configuring PATH
- Run pip





Windows

- No need to install pip
- Configure your PATH





Mac

- Don't use default Python
- It's not meant for development
- Install Python using Homebrew:
<https://brew.sh>
- Or download from <https://python.org>



Linux

- Comes with python (probably 2)
- May need to install python 3 and/or pip
- Use package manager
- Might not install latest pip (but that's ok)

Demo



Linux

- Install pip
- Run pip
- Install pip for python3
- Run that



Demo



Managing packages with pip

- Installing packages
- Removing packages
- Listing packages
- Inspecting packages
- Getting help
- Searching packages





Best Practice

Always work inside a virtual environment
Don't use pip with `sudo`



Managing Packages with pip

Installing a package

(will also install its dependencies)

```
pip install requests
```

```
pip install six Django Keras          # multiple installs
```

Removing a package (but NOT its dependencies)

```
pip uninstall requests
```

```
pip uninstall six Django Keras        # multiple uninstalls
```



Finding Package Information

List all installed packages

```
pip list
```

Get info on a specific package

```
pip show requests
```



Getting Help



Pip built-in help: `pip help`

For specific command: `pip help install`

Or visit <http://pip-installer.org>

The cheese shop: <https://pypi.org>



Demo



Where are packages installed?

- The `sys.path` variable
- Installing for other python versions



Demo



A better way to call pip



Summary



Installing pip

Managing packages

- pip install
- pip uninstall
- pip list
- pip show
- The cheese shop

Package locations

Calling pip

- python -m pip

