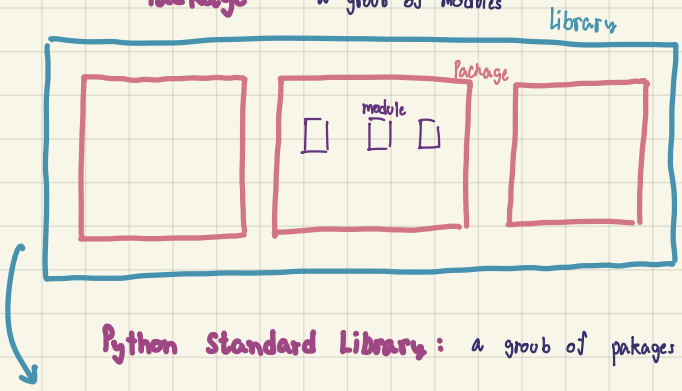
Key Terms

- **Package** - Python code bundled together for distribution and installation
- **Module** - Python code saved to a file for reuse
- **venv** - create a virtual environment
- **activate** - activate/enter the virtual environment
- **pip install** - install packages into the active virtual environment
- **pip list** - list packages installed in the current environment
- **pip freeze** - output installed packages to a requirements file
- **deactivate** - exit the active virtual environment

```
# Create virtual environment
python3 -m venv my_env

# Activate virtual environment
source my_env/bin/activate

# Install packages to virtual environment
pip install pandas

# List packages installed in environment
pip list

# Save installed packages to requirements file
pip freeze > requirements.txt

# Deactivate virtual environment
deactivate
```

Installing Packages with pip in Python

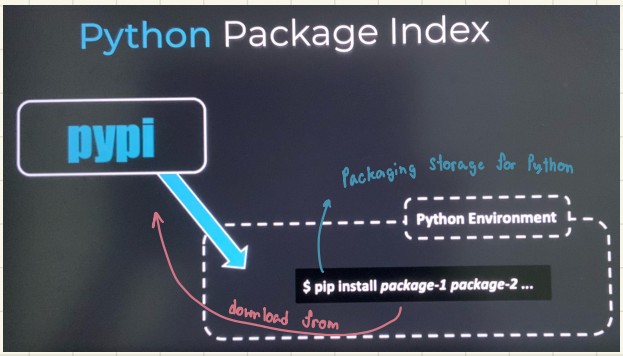
Package : a group of modules



Module : a code saved as a file (.py)

Python Standard Library : a group of pakages → can be used in any python Programe

Python Package Index :
- Third party
- AKA pypi



1. Type:
`source setup-env.sh`
To set up an empty environment.

2. Type:
`pip help`
To see the documentation for pip.

3. Type:
`pip help install`
To see the help for the `pip install` command.

4. Install the requests package:
`pip install requests`

5. See the packages installed in your environment:
`pip list`

6. To see the available versions of the `requests` package:
`pip install requests==`
Note: this will give an error message, but in that message all the available versions are displayed.

7. Choose an earlier version of requests and install that version:
`pip install requests==2.25.1`

8. Check the installed version:
`pip list`

9. Upgrade to the latest version:
`pip install requests --upgrade`

10. Check the installed version:
`pip list`

Saving Requirements File in Python

- **ls** - List the contents of a directory.

```
$ ls /path/to/directory

autoload bundle      colors      doc      ftplugin plugin
spell                syntax
```

- **cat** - Print the contents of a file to the screen

```
$ cat some_file.txt

The text of some_file.txt will be displayed here.
```

```
$ pip freeze
attrs==21.2.0
iniconfig==1.1.1
packaging==21.3
pluggy==0.13.1
py==1.11.0
pyparsing==3.0.6
pytest==6.2.5
toml==0.10.2
$ pip freeze > requirements.txt
```

redirect them to file

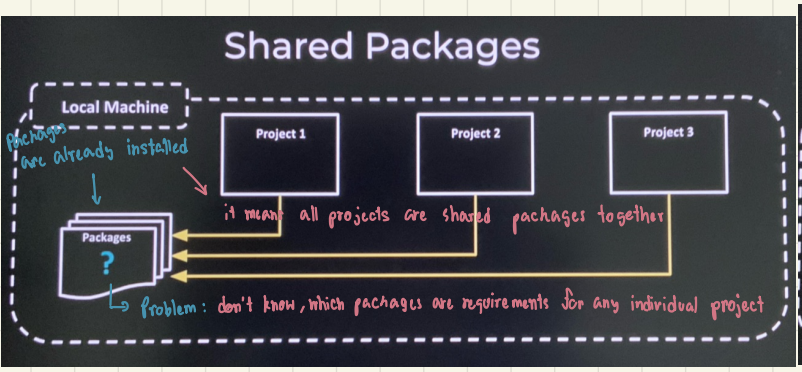
- pip freeze: it's like pip list but it in format can be used with the installed/uninstalled commands

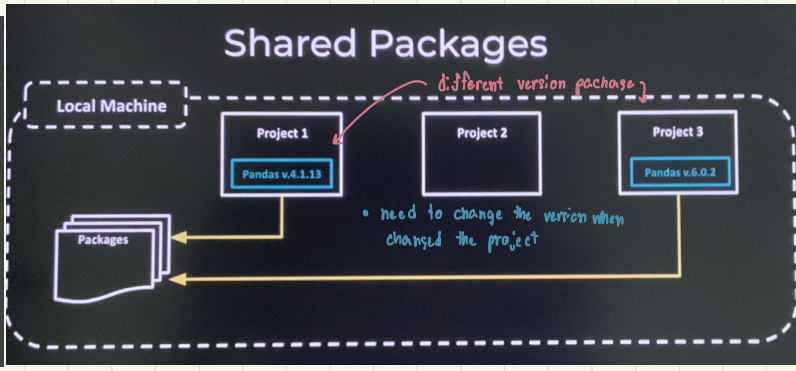
```
$ cat requirements.txt
attrs==21.2.0
iniconfig==1.1.1
packaging==21.3
pluggy==0.13.1
py==1.11.0
pyparsing==3.0.6
pytest==6.2.5
toml==0.10.2
$ pip uninstall -r requirements.txt
```

it also can be installed in .txt file

uninstall the packages in

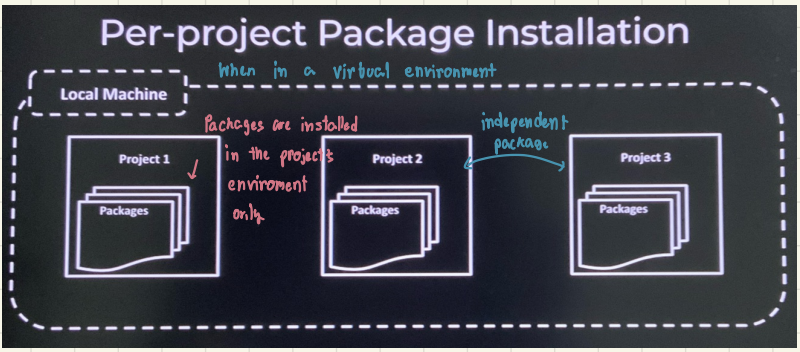
Creating and Using a Python Virtual Environment





specify a location to store packages when we're already created Virtual Environment

- 2 ways
  - store in a central Location (sub dir of home dir)
  - store in their project dir



introduce shell now

Shell Commands Used

- **mkdir** - Make directory

```
$ mkdir directory-name
```

- **pwd** - Print working directory

```
$ pwd
```

- **cd** - Change directory

```
$ cd directory-name
```

- **source** - Read and execute contents of a file

```
$ source file-name
```

```
$ mkdir my_first_project
$ cd my_first_project
$ pwd
/Users/kennedyrobertbehrman/Google Drive/projects/python.and.pandas.for.data.engineering.Duke.Coursera/my_first_project
$ python -m venv env   ← to create virtual environment
$ ls                   (module; environment's name)
env
$ ls env
bin        include    lib        pyvenv.cfg   (configuration)
$ source env/bin/activate
(env) $                activate new environment
```

```
(env) $ pip list
Package    Version
---------- -------
pip        21.3.1
setuptools 57.4.0
(env) $ pip install requests   this package will only reflect in this project
Collecting requests
  Using cached requests-2.26.0-py2.py3-none-any.whl (62 kB)
Collecting certifi>=2017.4.17
  Using cached certifi-2021.10.8-py2.py3-none-any.whl (149 kB)
Collecting idna<4,>=2.5
  Using cached idna-3.3-py3-none-any.whl (61 kB)
Collecting urllib3<1.27,>=1.21.1
  Using cached urllib3-1.26.7-py2.py3-none-any.whl (138 kB)
Collecting charset-normalizer~=2.0.0
  Using cached charset_normalizer-2.0.9-py3-none-any.whl (39 kB)
Installing collected packages: urllib3, idna, charset-normalizer, certifi, requests
Successfully installed certifi-2021.10.8 charset-normalizer-2.0.9 idna-3.3 requests-2.26.0 urllib3-1.26.7
(env) $
```

to end this env, and go back to share user environment

Instructions

1. Type:
`python -m venv ~/envs/my_first_env`
To create a virtual environment named 'my_first_env'.

2. Type:
`source ~/envs/my_first_env/bin/activate`
To activate your new environment.

3. Type:
`pip list`
To see the packages installed in your environment.

4. Type:
`pip install -r requirements.txt`
To install the packages saved to your requirements file from the previous lab.

3. Type:
`pip list`
To see the packages now installed in your environment.

3. Type:
`deactivate`
To deactivate your environment.