Command Line Python Scripting

Syntax

- Install Python packages

```
pip install [package name]
```

- Upgrade pip

```
pip install --upgrade pip
```

- Install the virtualenv tool

```
pip install virtualenv
```

    - Note: Pyhon 3 should already have the venv module.

- Create a virtual evironment

```
virtualenv [name of environment]
```

- Change the Python interpreter

```
virtualenv -p [path to desired Python] [name of environment]
```

- Activate virtual environment

```
source [name of environment]/bin/activate
```

- Check Python version

```
python -V
```

- Check installed packages and version

```
pip freeze
```

- Switch a virtualenv off

```
deactivate
```

Concepts

- Command line python interpreter good for testing snippets of code quickly, as well as debugging.
- Command line python interpreter not good for developing Python programs.
- Common way to develop with Python: use an IDE or text editor to create Python files, and then run from command line.
- You can enter the default Python executable using `python`.
- We can access Python `3` using the `python3` executable.
- Packages are an important way to extend Python's functionality.
- Pip is the best way to install packages from the command line.
- Virtual Environments allows us to have a certain version of Python and other packages without worrying about how it will affect other projects on our system.
- By default, virtualenv will use the `python` executable.
- We can import functions from a package into a file as well as functions and classes into another file.

Resources

- Python Package Index
- Python Virtual Environments – A Primer