pipenv

>> Installation

pip install pipenv

>> Create a new Virtual Enviorment

pipenv shell

>> install package

pipenv install pandas

>> exit from Virtual Enviorment

exit

>> Run some thing in Virtual Enviorment

pipenv run python
pipenv run pip list
pipenv run python script.py

>> Install libraries by requirements.txt

pipenv run pip freeze > requirements.txt

>> Install a package just in Development Enviorement not in Production Enviorment

pip install mathplotlib --dev
pip install mathplotlib -d

>> Uninstall a package

pipenv uninstall scipy

>> Change python version to 3.6(You should install that version in your OS before)

pipenv python 3.6

>> Remove Virtual Enviorment

pipenv -rm

>> Path of Virtual Enviorment

pipenv --venv

>> Checking installed package safety

pipenv check

>> Graph

pipenv graph

>> pipenv lock

$ pipenv lock is used to create a Pipfile.lock, which declares all dependencies (and sub-dependencies) of your project, their latest available versions, and the current hashes for the downloaded files. This ensures repeatable, and most importantly deterministic, builds.

pipenv lock

>> Install dependencies in production by using pipfile.lock

pipenv install --ignore-pipfile

>> .env

If you want to load automatically some environment variables each time you start the project, you can set a .env file at the root folder of the project, next to the Pipfile

echo MY_TOKEN=SuperToKen >.env
echo MY_VAR=SuperVar >>.env
