Pip pipenv cmds

Tuesday, 6 May 2025 1:24 PM

To create venv:

- pipenv install
- Pipenv shell
- Which python --> to know from which env its running
- Deactivate -> to stop the env
- pipenv --rm ->removes the environment
- source .venv/bin/activate -> to activate it again

Create venv using uv:

- Go to base directory -> uv sync
- uv run server.py

Useful commands:

- List down all dependencies with version : pip3 list
- To get same dependencies suited for requirements.txt file : pip3 freeze
- Save all requirements to requirements.txt file : pip3 freeze > requirements.txt
- Install dependencies from requirements.txt file : pip3 install -r requirements.txt
- To create a virtual environment with a specific Python version : python3.9 -m venv venv39
- You can manage Python versions with pyenv if needed:
  pyenv install 3.10.12
  pyenv virtualenv 3.10.12 myenv
  pyenv activate myenv
- Venv listing / creating / activating / removing / uninstall dependencies:
  - ls -d */ | grep -E "venv|env|\.venv"
  - python3 -m venv venv
  - source venv/bin/activate
  - Deactivate
  - pip uninstall <package>
- Get inside existing pipenv
  - Uv env
  - export PATH="$HOME/Library/Python/3.12/bin:$PATH"
  - pipenv shell
  - Run an application:
    - Pipenv lock
    - Pipenv sync
    - pipenv run python -m cspmcp.api_server
      - pipenv run gunicorn --workers 4 --bind 0.0.0.0:5000 cspmcp.api_server:app
    - pipenv run python main.py --all
  - List tools:
    - pipenv run python -m cspmcp.tools list
  - Run MCP tool:
    - pipenv run python -m cspmcp.tools incident-cli-tool

https://paypal.atlassian.net/wiki/spaces/PLATFORM/pages/1221493658/Managing+Python+requirements#1.-Configure-pip

Pipenv:

- Install pipenv :: pip install pipenv
- Create pip file :: pipenv install
- Create specific python version :: pipenv --python 3.10
- Activate pipenv environment :: pipenv shell
- Deactivate environment :: exit
- Check current environment info :: pipenv --venv
- Listing installed packages with version :: pipenv requirements
- Same like pip freeze :: pipenv requirements --dev

- Same like pip freeze :: pipenv requirements --dev
- To generate a requirements.txt style list :: pipenv lock -r > requirements.txt
- To generate a requirements.txt style list for dev package :: pipenv lock -r --dev > dev-requirements.txt
- Installing in pipenv:
    - Regular :: pipenv install requests
    - Dev dependencies :: pipenv install pytest --dev
    - Specific version :: pipenv install "flask==2.1.0"
- Uninstall package :: pipenv uninstall requests
- Generate and using pipfile.lock
    - To generate :: pipenv lock
    - Install all from Pipfile.lock :: pipenv sync
- Running script inside :: pipenv run python script.py
- Run any CLI tool installed in the environment :: pipenv run pytest
- Removing venv :: pipenv --rm
- To see where is the pipenv created :: pipenv --venv