

가상환경 (Virtual Environment)

프리젠테이션 사용 불가

가상환경(Virtual Environment)

- 파이썬의 가상환경은 하나의 컴퓨터에서 프로젝트 별로 독립된 파이썬 실행 환경(Runtime)을 구성하는 것을 말한다.
- 하나의 시스템에서 여러 프로젝트를 전역환경(Global Environment) 하에서 진행하다 보면 각 각의 프로젝트에서 사용되는 패키지나 패키지 버전 충돌 등의 문제가 발생할 수 있다.
- 또한 공동 작업을 위해 프로젝트 소스 공유 시나 배포시에는 프로젝트에 필수적인 패키지 목록을 함께 제공해줘야 하는데 이때 다른 프로젝트에서 설치한 패키지 목록까지 함께 포함되는 문제도 있다.
- 위와 같은 문제들을 해결할 수 있는 방법이 가상환경(Virtual Environment)이다.
- 가상환경을 지원하는 대표적인 모듈은 다음과 같다.
 - 표준 라이브러리 : venv
 - 비 표준 라이브러리 : virtualenv
- 위 라이브러리 이 외에도 다양한 라이브러리가 있다.

venv 라이브러리를 사용한 가상환경 구축

- venv 모듈은 파이썬에서 지원하는 표준 라이브러리로 내장 모듈이다. 그러므로 아래의 명령을 실행하여 가상환경을 구성한다.

```
$ python -m venv .venv
```

- 위의 명령은 가상환경을 구성할 프로젝트 내에서 실행하며 가상환경 구성 정보를 저장하는 디렉토리는 일반적으로 .venv를 사용한다.
- 가상환경 구성 파일은 git에 올릴 필요가 없으므로 .venv 디렉토리를 .gitignore 파일에 추가해준다. .venv 디렉토리에는 가상환경하에서 파이썬 실행을 위한 실행기를 비롯하여 필수 파일들이 포함되어 있다.
- 가상환경 구성을 위한 디렉토리를 생성하였다면 가상환경을 활성화 하여야 한다.

```
$ ./venv/Scripts/activate # Windows 시스템이라면 .venv\Scripts\Activate.bat
```

- 가상환경을 비활성화 하는 방법은 다음과 같다.

```
(.venv) $ exit
```

가상 환경 하에서 패키지 설치하기

- 가상환경을 활성화 하면 아래와 같이 프롬프트 앞에 (.venv) 가 표시된다.
- `$../venv/Scripts/activate`
- `(.venv) $`
- 가상환경이 활성화 된 상태에서 아래와 같은 패키지를 설치해보도록 한다.
- `python -m pip install -U pip` # 필요하다면 pip 버전을 업그레이드 한다.
- `(.venv) $ python -m pip install pymysql` # pymysql 패키지 설치
- `(.venv) $ pip install django` # django 패키지 설치
- pymysql 패키지와 django 패키지를 설치한 후 설치 패키지 목록을 검색해보면 pymysql 패키지와 django 패키지 이 외에도 많은 패키지들이 설치되어 있을 것이다. 이것은 django 패키지가 의존하는 패키지들로서 django 패키지에 필수적인 패키지들이 함께 설치되었기 때문이다.

가상환경과 전역환경 비교

- 가상환경에서의 설치 패키지 목록 검색

```
$ pip list
Package      Version
-----
asgiref      3.7.2
Django       4.2.3
pip          23.2
PyMySQL      1.1.0
setuptools   65.5.0
sqlparse     0.4.4
tzdata       2023.3
```

- 전역환경에서의 설치 패키지 목록 검색

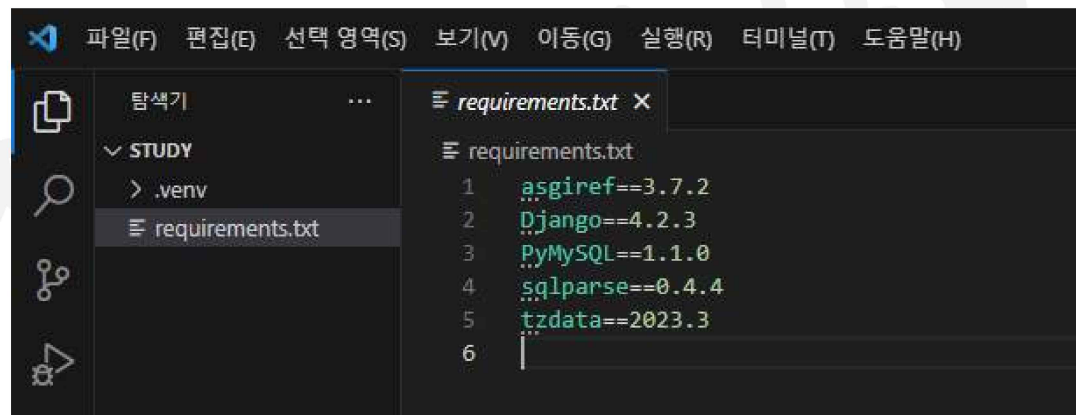
```
$ pip list
Package      Version
-----
altgraph     0.17.3
certifi      2023.5.7
charset-normalizer 3.2.0
idna         3.4
pefile       2023.2.7
pip          23.2
pyinstaller  5.13.0
pyinstaller-hooks-contrib 2023.5
pywin32-ctypes 0.2.2
requests     2.31.0
setuptools   65.5.0
urllib3      2.0.3
wheel        0.40.0
```

패키지 목록 파일 생성하기

- 가상환경을 활성화한 상태에서 아래의 명령을 사용하여 패키지 목록을 생성한다.

```
(.venv) $ pip freeze > requirements.txt
```

- 위의 명령을 실행하고 나면 아래와 같이 프로젝트 폴더에 requirement.txt 파일이 생성된다.



패키지 목록 파일을 사용하여 프로젝트 환경 구성하기

- 프로젝트 환경을 구성할 폴더에서 가상환경을 구성한 후 활성화 한다.

```
$ python -m pip install -U pip      # 필요소 pip 를 업그레이드 한다.  
$ python -m venv .venv  
$ ../.venv/Scripts/active
```

- 가상환경을 활성화 하였다면 아래의 명령을 사용하여 패키지를 설치한다.

```
$ pip install -r requirements.txt
```

- 가상환경 하에 패키지의 설치 여부를 확인한다.

```
$ pip list
```