ILE3-008 모듈, 패키지

코드의 재활용을 위한 방안인 모듈과 모듈을 모아둔 패키지

모듈

코드의 재활용을 위해 `함수` + `변수` + `클래스`를 모아둔 파일

모듈 만들기

기본적으로 Python 파일 1개가 모듈
그러므로 모듈을 만들고 싶으면 Python 파일을 하나 더 만들어야 함

mod1.py

```
def add(a, b):
    return a + b

def sub(a, b):
    return a-b
```

모듈 사용

`import` 문

모듈을 읽어와 사용하기 위한 구문

```
import [모듈 이름]
```

```
>>> import mod1
>>> mod1.add(1, 2)
3
```

as

`import`문을 이용하여 가져온 모듈의 이름을 변경하여 사용하고 싶은 경우 사용

```
import [모듈 이름] as [별칭]
```

```
>> import mod1 as m
>> m.sub(1, 2)
-1
>> mod1.sin(1)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'mod1' is not defined
```

from

모듈의 일부만 가져오고 싶을 때 사용

```
from [모듈 이름] import [함수 or 클래스 이름]
```

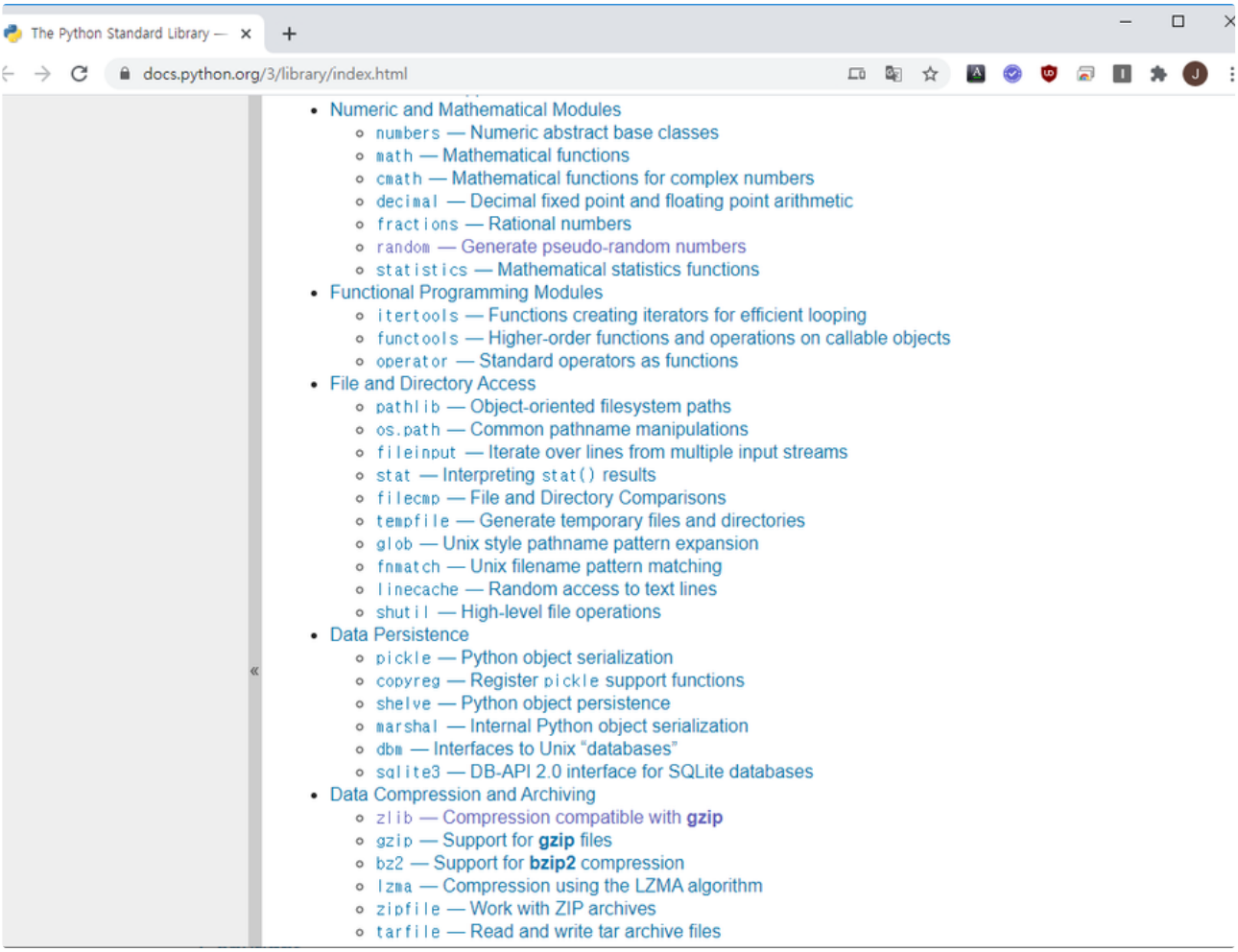
```
>> from mod1 import add
>> add(1, 3)
```

```
4
>> mod1.add(1, 3)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'mod1' is not defined
```

> `from math import *`와 같이 `*`을 통해 모든 변수/함수들을 가져 올 수 있지만 여러 모듈을 쓰는 경우 충돌이 발생 할 수 있으므로 사용 금지

표준 모듈

Python에 기본적으로 내장되어 있는 모듈
별도로 설치를 안해도 사용 가능
파이썬 표준 라이브러리



패키지

여러개의 모듈들을 모아, 하나의 덩어리로 만들어 둔 것
일반적으로 `pip`를 이용하여 다운로드 받아 사용

> scipy TenserFlow, Flask 등등

설치된 패키지의 조회

```
> pip list
```

패키지의 설치

```
> pip install [패키지 이름]
```

해당 명령어를 사용하는 경우, 해당 Python 버전 모두에 해당 Package가 설치 됨

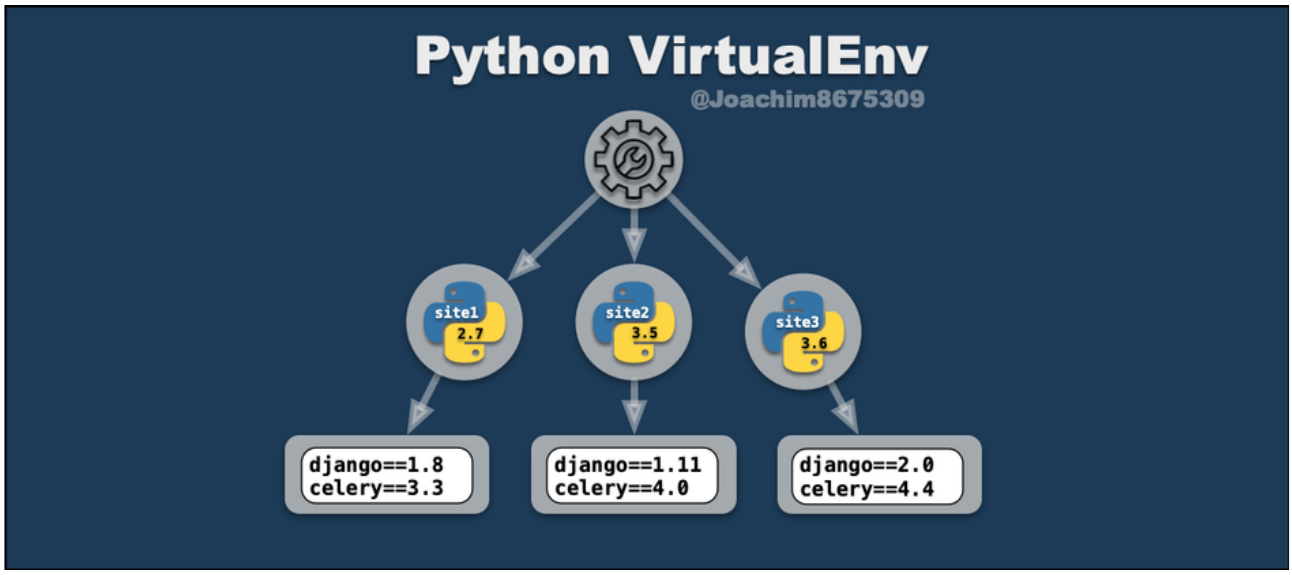
Vritual Environment

시스템의 Python Package와 개발 중인 프로젝트의 Python Package를 분리하여 사용

필요한 상황

Python 프로젝트를 2개 진행을 하고 있음
1번 프로젝트에서는 A 라이브러리의 **v2 버전만** 사용 가능
2번 프로젝트에서는 A 라이브러리의 **v1 버전만** 사용 가능

> Virtual Environment를 사용하지 않는 경우 프로젝트를 사용할 때 마다 패키지를 지우고 설치하고를 반복 하여야 함



설치

Windows 환경에서는 Python 홈페이지를 통해 설치한 경우에는 기본적으로 설치되어 있음

virtualenv 생성

virtualenv가 적용될 폴더를 만들 위치에서 아래의 명령어 사용 (`cmd` 추천)
아래 명령어 호출 시 [PATH] 경로가 생성되고, 해당 경로 내에 virtualenv가 구축 됨

```
> virtualenv [PATH]
```

Example

```
> virtualenv dev_1
created virtual environment CPython3.8.4.final.0-32 in 476ms
```

```
  creator CPython3Windows(dest=E:\venv\test\dev_1, clear=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=C:\Users
    added seed packages: pip==20.1.1, setuptools==49.2.0, wheel==0.34.2
  activators BashActivator,BatchActivator,FishActivator,PowerShellActivator,PythonActivator,XonshActivator
```

명령 프롬프트에서 virtualenv 사용

활성

> Powershell에서는 보안 문제로 에러가 발생하여 사용이 힘듬

virtualenv가 생성된 폴더에서 `Script\activate`를 실행
실행이 정상적으로 된 경우에는 명령 구문 앞에 `([VIRTUALENV_NAME])`이 표시 됨

Example

```
E:\\venv\\dev_1> Scripts\\activate

(dev_1) E:\\venv\\dev_1>
```

비활성

virtualenv가 생성된 폴더에서 `Script\\deactivate`를 실행
실행이 정상적으로 된 경우에는 명령 구문 앞에 `([VIRTUALENV_NAME])`가 제거 됨

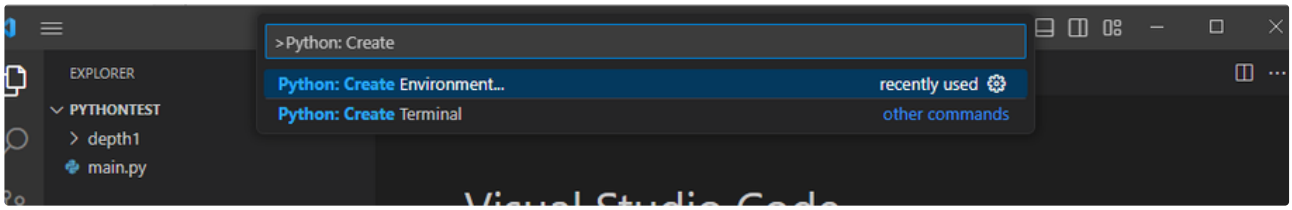
```
(dev_1) E:\venv\dev_1>Scripts\deactivate
E:\venv\dev_1>
```

VS Code에서 virtualenv 사용

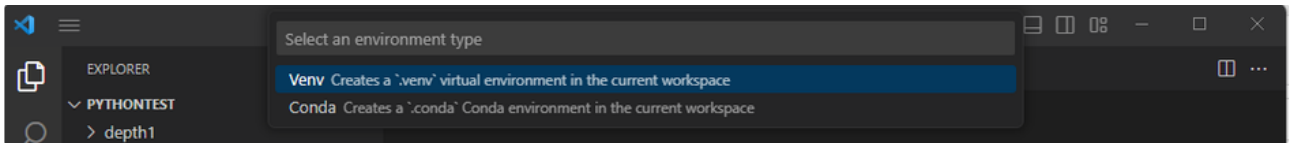
`launch.json`이 설정 완료 된 것으로 가정합니다.

virtualenv 생성

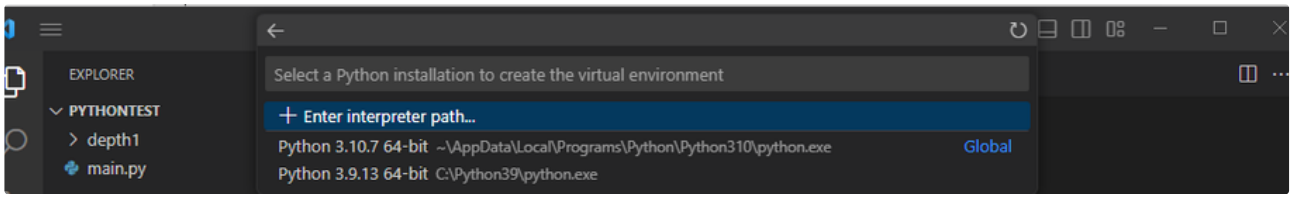
1. VS Code `ctrl` + `shift` + `p`를 입력 후 `Python: Create Enviroment...`를 입력 후 선택



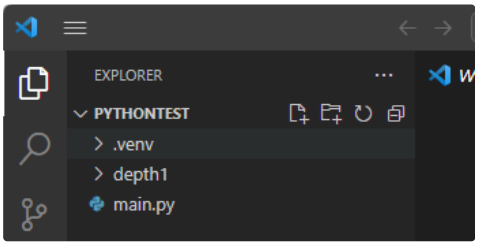
2. `Venv` 선택

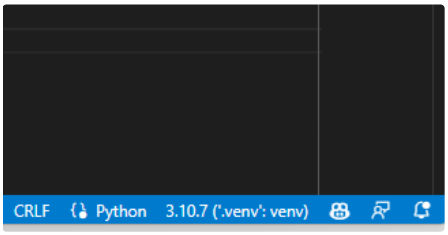


3. 사용할 Python Version 선택

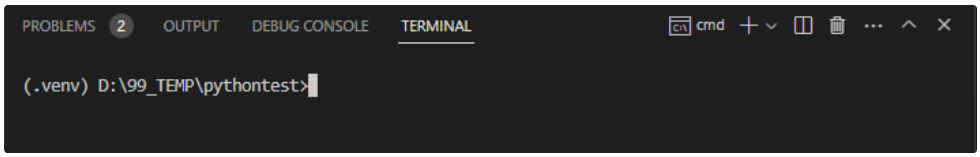


4. `.venv` 디렉토리가 생성되고, 우측 하단에 venv 설정 정보가 표시





5. Terminal을 실행 시 Virtual Environment가 적용되어 있는 것을 확인



과제

1. Virtual Env를 1개 생성
2. 1.에서 생성한 Virtual Env를 활성화
3. psycopg2 모듈을 최신 버전으로 설치
4. pip list 명령어로 설치된 모듈 내역 출력
5. Virtual Env 비활성화
6. pip list 명령어로 설치된 모듈 내역 출력

4, 6에서 출력된 내용을 메일로 첨부해서 제출 바랍니다.