

1 [HOL] Python virtualenv 설정하기

1. 가상환경이란?

- 1)Python을 사용할 때 기본으로 제공되는 library만 사용되지 않고 많은 기능을 제공하는 library들을 추가하여 사용하게 된다.
- 2)이러한 library가 특정 project에서만 사용되거나 project를 배포할 때 필요한 library만 포함시켜 배포하고 싶을 경우가 있을 것이다.
- 3)이럴때 사용할 수 있는 것이 가상환경(virtualenv) 이다.
- 4)가상환경을 만들고, 그 가상환경에서 library를 추가하면 추가된 library는 그 가상환경에서만 사용 되어진다.

2. 가상환경 설치

- 1)먼저 가상환경을 설치해보자.
- 2)OS가 Windows 인 경우는 관리자권한으로 Command 창으로 들어간다.
- 3)\$ python -m pip install --upgrade pip
- 4)\$ pip install virtualenv

3. 가상환경 설정

- 1)가상환경 file이 생성될 Python workspace로 이동하자.
- 2)본인은 현재 C:/PythonHome이다.
- 3)다음과 같이 명령을 실행한다.

```
C:\PythonHome>python -m venv PythonProject
```

- 4)위의 명령으로 ProjectEnv라는 새로운 가상환경을 생성했다.

```
C:\PythonHome>cd PythonProject
C:\PythonHome\PythonProject>
```

- 5)만든 새 환경을 사용하려면 먼저 활성화해야 한다.
- 6)활성화하기 위해 새환경 폴더 아래 Scripts 폴더 아래에 activate.bat file을 실행한다.

```
C:\PythonHome\PythonProject>cd Scripts
C:\PythonHome\PythonProject\Scripts>activate
```

```
(PythonProject) C:\PythonHome\PythonProject\Scripts>
```

- 7)환경을 활성화하면 환경의 이름이 명령 프롬프트에 표시되어 현재 가상 환경에 있음을 알 수 있다.
- 8)가상 환경에서 library를 설치하거나 스크립트를 실행하면 그 환경에만 영향이 있게 된다.

```
(PythonProject) C:\PythonHome\PythonProject\Scripts>python
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb 4 2025, 15:23:48) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

- 9)deactivate 명령으로 환경에서 떠날 수 있다.

```
(PythonProject) C:\PythonHome\PythonProject\Scripts>deactivate
```

- 10)다시 가상환경으로 들어가기 위해서는 activate 명령을 다시 실행하면 된다.
- 11)가상환경에서 VSCode를 사용하려면 다음과 같이 입력하면 된다.

```
(PythonProject) C:\PythonHome\PythonProject> code .
```

4. VSCode에서 가상환경 설정하기

- 1)가상환경 만들기 : VSCode의 Terminal에서(현재 PowerShell Base)
PS C:\PythonHome>python -m venv tutorial-env
- 2)VSCode Terminal에서 가상환경을 만들면 자동으로 VSCode의 우측 하단에 현재 가상 환경 폴더를 VSCode의 기본 폴더로 지정할 것인가라는 메시지 창이 나타난다.
- 3)여기서 Yes를 선택하면 자동으로 VSCode가 해당 가상폴더를 인식하게 된다.
- 4)수동으로 선택하려면 VSCode의 좌측 하단에 보면 현재 Interpreter(Python 3.13.2 64-bit)는 C:/Program Files/Python313로 되어 있는데, 여기를 클릭하면 VSCode 상단에서 [Select Interpreter]아래에 +Enter interpreter path..., .\Python 3.13.2 64-bit('tutorial-env':venv) .\tutorial-env\Scripts\python.exe, Python 3.13.2 64-bit C:\Program Files\Python313\python.exe 이렇게 목록이 나온다.
- 5)여기서 tutorial-env를 선택하거나 아니면 제일 위의 목록 +Enter interpreter path...를 선택해서 지정하면 된다.
- 6)만일 .\Python 3.13.2 64-bit('tutorial-env':venv)를 선택하면 VSCode 좌측 하단에 Python 3.13.2 64-bit ('tutorial-env':venv)로

설정된다.

7)새 터미널을 열어보자. 하지만 "이 시스템에서 스크립트를 실행할 수 없으므로 C:\pythonhome1\tutorial-env\Scripts\Activate.ps1 파일을 로드할 수 없습니다" 즉 보안 오류가 발생한다.

8)그 이유는 기본적으로 Windows 10은 PowerShell의 실행을 막고 있기 때문이다.

9)터미널에서 Get-ExecutionPolicy을 입력하면 Restricted라고 나온다. 이 옵션이 기본값이다.

10)이것을 수정하려면 Windows Powershell을 관리자 권한으로 실행한 후,

```
PS C:\WINDOWS\system32> Set-ExecutionPolicy RemoteSigned
```

실행 규칙 변경

실행 정책은 신뢰하지 않는 스크립트로부터 사용자를 보호합니다. 실행 정책을 변경하면 about_Execution_Policies 도움말 항목(<https://go.microsoft.com/fwlink/?LinkID=135170>)에 설명된 보안 위험에 노출될 수 있습니다. 실행 정책을 변경하시겠습니까?

[Y] 예(Y) [A] 모두 예(A) [N] 아니요(N) [L] 모두 아니요(L) [S] 일시 중단(S) [?] 도움말 (기본값은 "N"): A <==== A를 선택한다.

11)다시 VSCode 터미널에서

```
PS C:\PythonHome> & c:/pythonhome1/tutorial-env/Scripts/Activate.ps1를 실행하면 아래와 같이 제대로 activate된다.  
(tutorial-env) PS C:\pythonhome1>
```

12)터미널을 닫았다가 다시 열면 제대로 잘 설정된 것을 알 수 있다. 즉, Windows Powershell에서 한번만 보안 처리를 해주면 된다.

5. macOS에서 가상환경 구성하기

1)macOS에서도 Windows와 마찬가지로 venv library를 사용한다.

2)먼저 기본 디렉토리를 생성한다.

```
$ mkdir PythonHome
```

3)해당 디렉토리로 이동한다.

```
$ cd PythonHome
```

4)해당 디렉토리에서 가상 환경을 생성한다.

```
$ python3 -m venv ./PythonProject
```

5)source 명령을 통해 가상환경으로 들어간다.

```
$ source ./PythonProject/bin/activate
```

6)방금 생성한 가상환경의 pip list를 확인한다.

```
$ pip3 list
```

Package	Version
-----	-----
pip	25.0

7)Python3 버전 체크도 해본다.

```
$ python3 -V
```

```
python 3.13.2
```

8)VSCode에서 Python Interpreter 설정하기

-View > Command Palette 또는 Command + Shift + P를 눌러서 명령 팔레트를 연 후

-다음과 같이 입력한다.

>Python Inter...라고 입력하면 [Python:Select Interpreter] 메뉴를 선택한다.

-그 다음 목록에서 Python 3.13.2 ('PythonProject') ./bin/python - Recommended 를 선택하면 된다.