

# Report

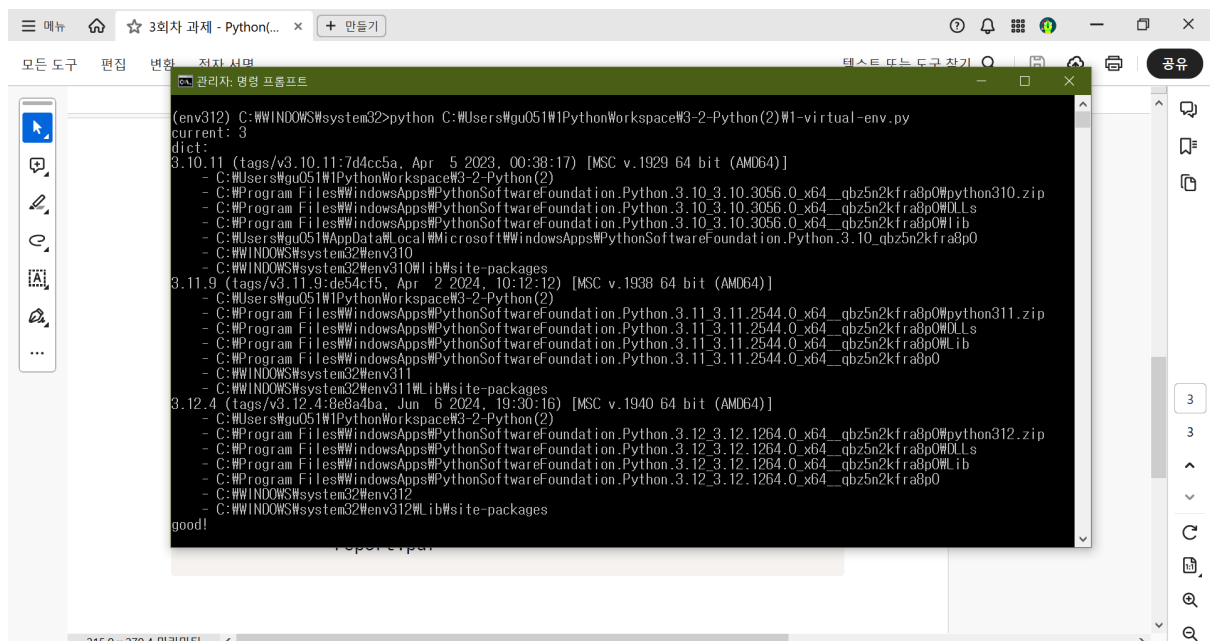
김준호

## 1. 1-virtual-env.py

Jupyter notebook과 cmd만 쓰다가 vscode로 하려니 자꾸 설정이 꼬이고 뭐가 잘 안돼서 cmd 창으로 다시 처음부터 시작했다...

가상환경 Python 버전은 3.10, 3.11, 3.12로 진행했다.

가상환경 생성은 cmd창에서 `python -m venv env310(311,312)`로 생성했다.



```
(env312) C:\WINDOWS\system32>python C:\Users\gu051\PythonWorkspace\3-2-Python(2)\1-virtual-env.py
current: 3
dict:
3.10.11 (tags/v3.10.11:7d4cc5a, Apr 5 2023, 00:38:17) [MSC v.1929 64 bit (AMD64)]
- C:\Users\gu051\PythonWorkspace\3-2-Python(2)
- C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.10.3.10.3056.0_x64__qbz5n2kfra8p0\python310.zip
- C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.10.3.10.3056.0_x64__qbz5n2kfra8p0\DLLs
- C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.10.3.10.3056.0_x64__qbz5n2kfra8p0\lib
- C:\Users\gu051\AppData\Local\Microsoft\WindowsApps\pythonSoftwareFoundation.Python.3.10.3056.0_x64__qbz5n2kfra8p0
- C:\WINDOWS\system32\env310
- C:\WINDOWS\system32\env310\lib\site-packages
3.11.9 (tags/v3.11.9:de54cf5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)]
- C:\Users\gu051\PythonWorkspace\3-2-Python(2)
- C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.11.3.11.2544.0_x64__qbz5n2kfra8p0\python311.zip
- C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.11.3.11.2544.0_x64__qbz5n2kfra8p0\DLLs
- C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.11.3.11.2544.0_x64__qbz5n2kfra8p0\lib
- C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.11.3.11.2544.0_x64__qbz5n2kfra8p0
- C:\WINDOWS\system32\env311
- C:\WINDOWS\system32\env311\lib\site-packages
3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)]
- C:\Users\gu051\PythonWorkspace\3-2-Python(2)
- C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.12.3.12.1264.0_x64__qbz5n2kfra8p0\python312.zip
- C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.12.3.12.1264.0_x64__qbz5n2kfra8p0\DLLs
- C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.12.3.12.1264.0_x64__qbz5n2kfra8p0\lib
- C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.12.3.12.1264.0_x64__qbz5n2kfra8p0
- C:\WINDOWS\system32\env312
- C:\WINDOWS\system32\env312\lib\site-packages
good!
```

## 2. 2-argparse.py

"—start"와 "—end"는 integer로 type 지정을 해주었고, "—verbose"는 boolean flag이다.

처음엔 "—start"의 default를 0으로 하는 코드를 넣지 않았다가, solution을 보고 수정했다.

## 3. 3-collection.py

`__init__`: 딕셔너리를 초기화하고 기본값을 생성하는 함수 `default_factory`를 설정한다.

`__getitem__`: 처음에는 다음과 같이 구성했다.

```
def __getitem__(self, key: K) -> V:
```

```
if key not in self:

    if self.default_factory is None:

        raise KeyError(key)

    self[key] = self.default_factory()

return super().__getitem__(key)
```

위 코드에 대한 설명: 키가 딕셔너리에 있는지 확인하여 키가 없고 default\_factory가 제공된 경우, 기본 값을 생성하여 딕셔너리에 추가하고 반환한다. default\_factory가 제공되지 않고 키가 없는 경우 KeyError를 발생시킨다.

그러나 solution을 확인해보니 solution의 코드가 훨씬 간단하고 효율적인 듯하여, solution에 있는 코드를 채택했다.

\_\_setitem\_\_이 필요 없는 이유:

\_\_setitem\_\_은 기본 딕셔너리 클래스에서 이미 제공하는 기능을 그대로 사용하기 때문에 별도로 구현할 필요가 없다. defaultdict 클래스는 딕셔너리의 항목을 설정하는 기능을 변경할 필요가 없으며, 항목에 접근할 때만 특별한 처리가 필요하기 때문에 \_\_getitem\_\_만 구현하면 된다.