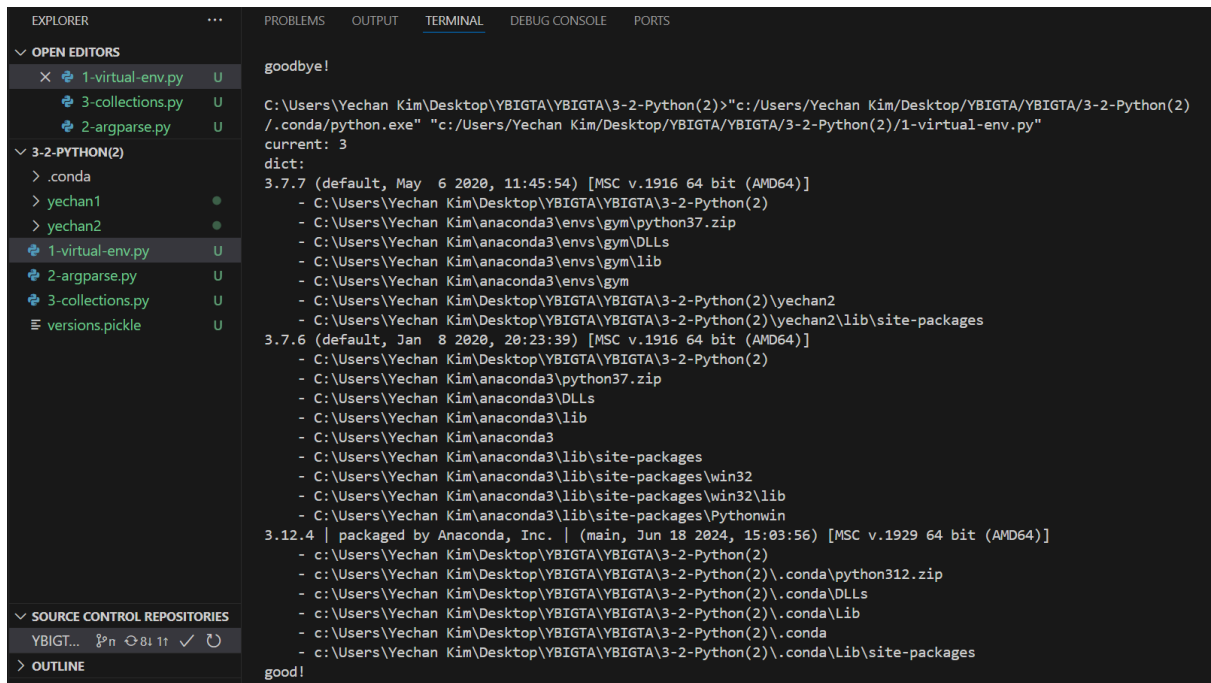


report

1-virtual-env.py_



```
goodbye!

C:\Users\Yechan Kim\Desktop\YBIGTA\YBIGTA\3-2-Python(2)>"c:/Users/Yechan Kim/Desktop/YBIGTA/YBIGTA/3-2-Python(2)/.conda/python.exe" "c:/Users/Yechan Kim/Desktop/YBIGTA/YBIGTA/3-2-Python(2)/1-virtual-env.py"
current: 3
dict:
3.7.7 (default, May 6 2020, 11:45:54) [MSC v.1916 64 bit (AMD64)]
- C:\Users\Yechan Kim\Desktop\YBIGTA\YBIGTA\3-2-Python(2)
- C:\Users\Yechan Kim\anaconda3\envs\gym\python37.zip
- C:\Users\Yechan Kim\anaconda3\envs\gym\DLLs
- C:\Users\Yechan Kim\anaconda3\envs\gym\lib
- C:\Users\Yechan Kim\anaconda3\envs\gym
- C:\Users\Yechan Kim\Desktop\YBIGTA\YBIGTA\3-2-Python(2)\yechan2
- C:\Users\Yechan Kim\Desktop\YBIGTA\YBIGTA\3-2-Python(2)\yechan2\lib\site-packages
3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
- C:\Users\Yechan Kim\Desktop\YBIGTA\YBIGTA\3-2-Python(2)
- C:\Users\Yechan Kim\anaconda3\python37.zip
- C:\Users\Yechan Kim\anaconda3\DLLs
- C:\Users\Yechan Kim\anaconda3\lib
- C:\Users\Yechan Kim\anaconda3
- C:\Users\Yechan Kim\anaconda3\lib\site-packages
- C:\Users\Yechan Kim\anaconda3\lib\site-packages\win32
- C:\Users\Yechan Kim\anaconda3\lib\site-packages\win32\lib
- C:\Users\Yechan Kim\anaconda3\lib\site-packages\Pythonwin
3.12.4 | packaged by Anaconda, Inc. | (main, Jun 18 2024, 15:03:56) [MSC v.1929 64 bit (AMD64)]
- c:\Users\Yechan Kim\Desktop\YBIGTA\YBIGTA\3-2-Python(2)
- c:\Users\Yechan Kim\Desktop\YBIGTA\YBIGTA\3-2-Python(2)\.conda\python312.zip
- c:\Users\Yechan Kim\Desktop\YBIGTA\YBIGTA\3-2-Python(2)\.conda\DLLs
- c:\Users\Yechan Kim\Desktop\YBIGTA\YBIGTA\3-2-Python(2)\.conda\Lib
- c:\Users\Yechan Kim\Desktop\YBIGTA\YBIGTA\3-2-Python(2)\.conda
- c:\Users\Yechan Kim\Desktop\YBIGTA\YBIGTA\3-2-Python(2)\.conda\Lib\site-packages

good!
```

conda에서 다양한 파이썬 버전을 선택하는 방법으로 가상환경을 생성해서 실행하였습니다.

2-argparse.py

```
C:\Users\Yechan Kim\Desktop\YBIGTA\YBIGTA\3-2-Python(2)>"c:/Users/Yechan Kim/Desktop/YBIGTA/YBIGTA/3-2-Python(2)/.conda/python.exe" "c:/Users/Yechan Kim/Desktop/YBIGTA/YBIGTA/3-2-Python(2)/2-argparse.py"
None None False
```

오픈소스 바탕으로 argparse에 제약조건을 추가하여 작성하였습니다.

3-collections.py

```
C:\Users\Yechan Kim\Desktop\YBIGTA\YBIGTA\3-2-Python(2)>"c:/Users/Yechan Kim/Desktop/YBIGTA/YBIGTA/3-2-Python(2)/.conda/python.exe" "c:/Users/Yechan Kim/Desktop/YBIGTA/YBIGTA/3-2-Python(2)/3-collections.py"
well done!
goodbye!
```

오픈소스 바탕으로 구성하였습니다.

- **상속:** `defaultdict` 는 내장 `dict` 를 상속하여 기본적인 딕셔너리 동작을 유지합니다.
- **생성자 (`__init__`):**
 - `__init__` 메서드에서는 `default_factory` 라는 호출 가능한 객체를 받습니다. 이 객체는 새로운 기본값을 생성하는 함수 역할을 합니다.
- **`__getitem__` 메서드:**
 - `__getitem__` 메서드는 딕셔너리에서 키를 가져올 때 호출됩니다 (`d[key]`).
 - 기존 딕셔너리에 키가 존재하지 않을 경우 (`KeyError` 가 발생할 경우), `__missing__` 메서드를 호출합니다.
- **`__missing__` 메서드:**
 - `__missing__` 메서드는 키가 딕셔너리에 존재하지 않을 때 호출되는 메서드입니다.
 - 여기서는 `default_factory` 를 사용하여 새 값을 생성하고, 해당 키에 할당한 후 그 값을 반환합니다.
- **`__setitem__` 메서드의 불필요성:**
 - `defaultdict` 에서는 기본값을 설정하는 주요 기능이 "읽기" 동작에 집중되어 있습니다.
 - 즉, `__getitem__` 에서 키를 접근할 때 기본값 생성이 필요한 것이지, 값을 설정할 때 추가적인 동작이 필요하지 않기 때문에 `__setitem__` 메서드를 구현할 필요가 없습니다.