제 5 강 Dictionary

학습 목차

- Dictionary 정의
- Dictionary 연산
- Set
- Complex data type 정리

Dictionary

- Key 와 Value 짝 들의 집합
- 파이썬의 내장 자료구조

```
>>> a_student = {'name': 'daehyun', 'grade': 'A+'}
>>> a_student['name']
'daehyun'
>>> a_student['grade']
'A+'
>>> a_student['age']
Traceback (most recent call last):
  File "C:\Program Files\JetBrains\PyCharm Community Edition 2022.3.2\plugi
    coro = func()
           ^^^^^^
  File "<input>", line 1, in <module>
KeyError: 'age'
>>> a_student['age'] = 51
>>> a_student['age']
51
```

Dictionary vs. List

- List : 순서와 값이 다 같아야, 같음.
- Dictionary: 순서에 상관없이 key-value 짝이 모두 같으면, 같음.

```
>>> spam = ['cats', 'dogs', 'moose']
>>> bacon = ['dogs', 'moose', 'cats']
>>> spam == bacon
False
>>> eggs = {'name': 'Zophie', 'species': 'cat', 'age': '8'}
>>> ham = {'species': 'cat', 'age': '8', 'name': 'Zophie'}
>>> eggs == ham
True
```

keys(), values(), items()

```
data = {'color':'red', 'age':42}

for k in data.keys():
    print(k)

for v in data.values():
    print(v)

for k, v in data.items():
    print(f'{k=}, {v=}')
```

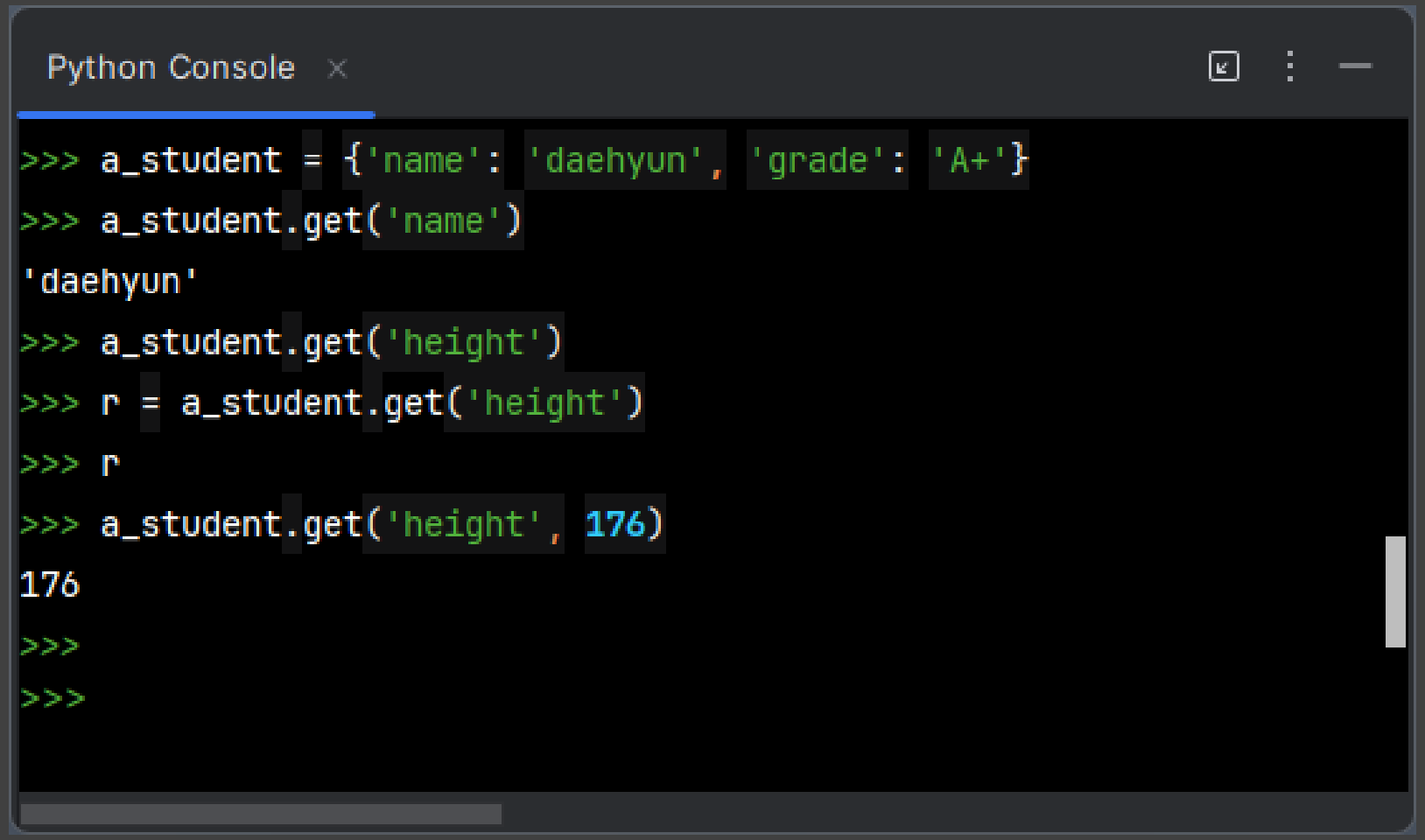
```
>>> data.keys()
dict_keys(['color', 'age’])

>>> list(data.keys())
['color', 'age']
>>>
```

Key와 value의 존재 확인

```
>>> spam = {'name': 'Zophie', 'age': 7}
>>> 'name' in spam
True
>>> 'name' in spam.keys()
True
>>> 'Zophie' in spam.values()
True
>>> 'color' in spam.keys()
False
>>> 'color' not in spam.keys()
True
>>> 'color' in spam
False
```

get()

- Key가 없을 경우, 기본 value 를 대신 선택.



setdefault()

- Key가 없을 경우 기본 value를 지정.
- 기존 key가 이미 있으면, 무시됨.

```
>>> spam = {'name': 'Pooka', 'age': 5}
>>> spam.setdefault('color', 'black')
'black'
>>> spam
{'color': 'black', 'age': 5, 'name': 'Pooka'}
>>> spam.setdefault('color', 'white')
'black'
>>> spam
{'color': 'black', 'age': 5, 'name': 'Pooka'}
```

characterCounter.py v.1

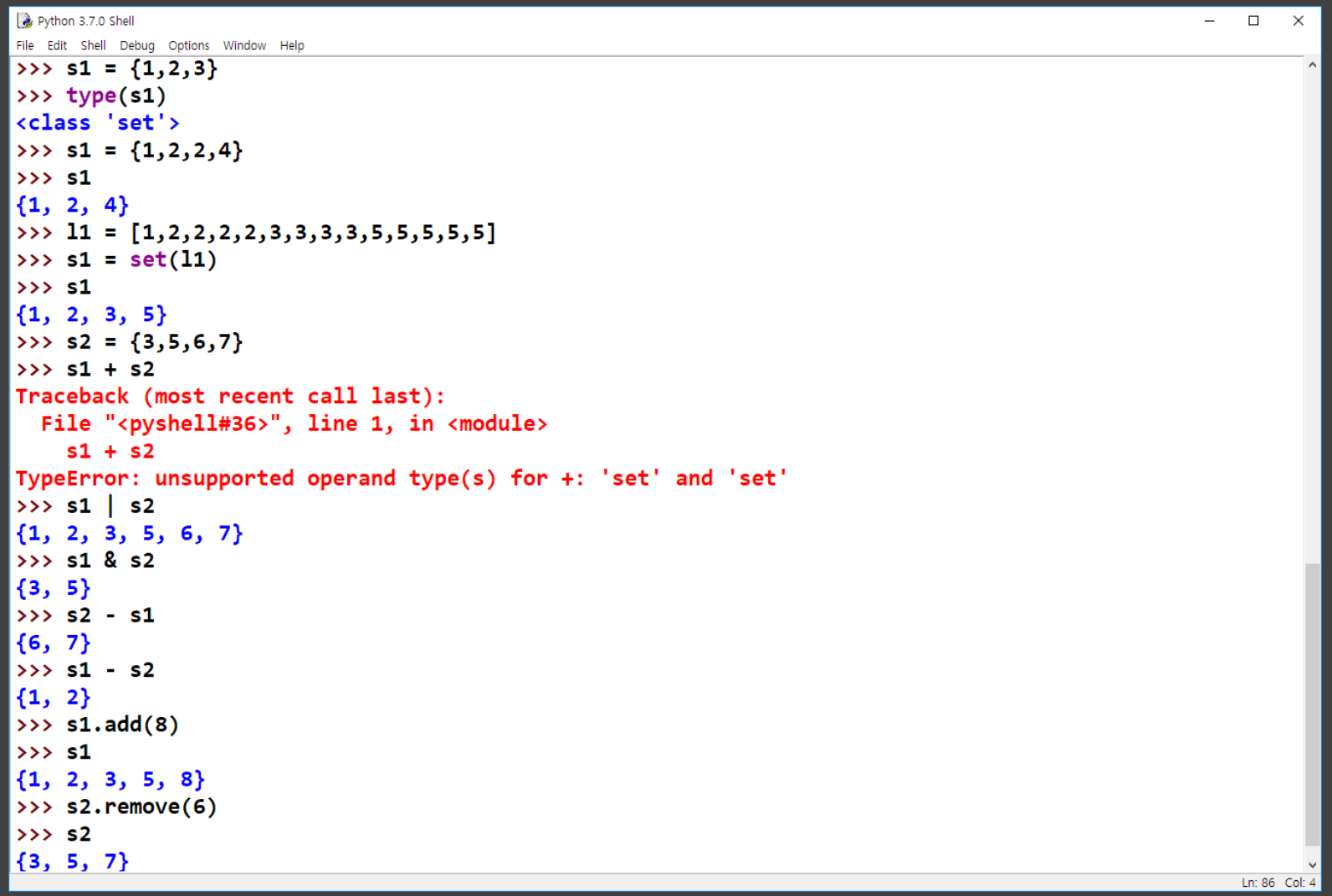
```
text = 'It was a bright cold day in April, and the clocks were striking thirteen.'
count = {}
for c in text:
    count.setdefault(c, 0) # c 가 있으면 초기화되지 않음.
    count[c] += 1

print(count)
```

characterCounter.py v.2

```
text = 'It was a bright cold day in April, and the clocks were
striking thirteen.'
from collections import defaultdict

count = defaultdict(lambda: 0)
for c in text:
    count[c] += 1

print(count)
print(type(count))
```

set

- 집합 자료형. 리스트와 달리, 중복을 허용하지 않고, 순서가 없음.



Complex Data Type

- **List – list**
  - 순서가 있는, 중복을 허용하는 데이타들의 집합.
  - 원하는 데이터를 찾기 위해, 순서 index 를 이용.

  [val1, val2, …]

- **Dictionary – dict**
  - 검색을 위한 키를 갖는 데이타들의 집합
  - key – value 쌍 들의 집합

  { key1: val1, key2: val2, … }

- **Tuple – tuple**
  - 순서가 있는, 중복을 허용하는 데이타들의 집합
  - 다만, 데이타값을 변경하는 것은 불가

  (val1, val2, …)

- **Set – set**
  - 중복을 허용하지 않는, 순서에 상관없는 데이타들의 집합

  { val1, val2, … }