

03. 컬렉션 자료구조 퀴즈

Total points 8/9

셋과 덕셔너리는 이지하니까 이지하게 풀어주시길 기대합니당~~ >_<

✓ 컬렉션 자료구조가 가지는 속성이 아닌 것은? *

1/1

- ☐ 멤버십(membership) 연산
- ☐ 크기(size) 함수
- ☒ 슬라이싱(slicing) 속성
- ☐ 반복성(iterability)
- ☐ 모르겠다.



✓ 다음은 set과 frozenset에 대한 코드이다. 이 중 에러가 발생하는 코드는? *

1/1

```
s1 = set([1, 2, 3, 4, 5])  
s1
```

☐ 1

```
s2 = {1, 2, 3}  
s2.add(5)  
s2
```

☐ 2

```
s3 = frozenset({1, 2, 3, 4, 5})  
s3.update({4, 5, 6})
```

☒ 3



```
s4 = frozenset({1, 2, 3, 4, 5})  
s4.intersection({4, 5, 6})
```

☐ 4

☐ 모르겠다.



✓ 그림은 정의된 함수와 리스트를 나타낸 것이다. 다음 중 결괏값이 False인 것은? * 1/1

```
def remove_dup(ls):  
    return list(set(ls))  
  
def intersection(ls1, ls2):  
    return list(set(ls1).intersection(set(ls2)))  
  
def union(ls1, ls2):  
    return list(set(ls1).union(set(ls2)))  
  
def difference(ls1, ls2):  
    return list(set(ls1).difference(set(ls2)))  
  
ls1 = [1, 2, 3, 4, 5, 5, 9, 11, 11, 15]  
ls2 = [4, 5, 6, 7, 8]
```

- ☐ remove_dup(ls1) == [1, 2, 3, 4, 5, 9, 11, 15]
- ☐ intersection(ls1, ls2) == [4, 5]
- ☐ union(ls1, ls2) == [1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 15]
- ☐ sorted(union([], ls2)) == sorted(ls2)
- ☒ difference(ls1, ls2) == difference(ls2, ls1) ✓
- ☐ 모르겠다.



✓ 다음 중 에러가 발생하는 코드는? *

1/1

```
pairs = [("a", 1), ("b", 2), ("c", 3)]  
d1 = dict(pairs)  
d2 = {"a": 1, "c": 2, "d": 3, "e": 4}
```

- ☐ d1.keys() & d2.keys()
- ☐ d1.items() - d2.items()
- ☒ d1.values() | d2.values() ✓
- ☐ {key: d2[key] for key in d2.keys() - {"c", "d"}}
- ☐ 모르겠다.



✓ 다음은 딕셔너리를 만드는 코드이다. 이 중 동작하지 않는 코드는? *

1/1

```
tarantino = {}
tarantino['name'] = '쿠엔틴 타란티노'
tarantino['job'] = '감독'
tarantino
```

☐ 1

```
sunnydale = dict({"name": "비피", "age": 16, "hobby": "게임"})
sunnydale
```

☐ 2

```
deginepad = dict(name="자일스", age=45, hobby="영화감상")
deginepad
```

☐ 3

```
algorithm = dict([{"name": "윌로"}, {"age": 15}, {"hobby": "게임"}])
algorithm
```

☒ 4

☐ 모르겠다.

```
goodboy = dict([(k, v) for k, v in enumerate(range(1, 5))])
goodboy
```

☐ 5


✕ 다음 코드의 출력 결과로 옳은 것은? *

0/1

```
def setdefault_dict(dict_data):
    newdata = {}
    for k, v in dict_data:
        newdata.setdefault(k, {"월로"}).add(v)
    return newdata

def test_setdef():
    dict_data = (("key1", "버피"),
                  ("key1", "자일스"),
                  ("key1", "자일스"),
                  ("key2", "버피"),
                  ("key2", "버피"),
                  ("key2", "월로"),)
    print(setdefault_dict(dict_data))

if name == 'main':
    test_setdef()
```

- ☐ {'key1': {'버피', '자일스', '월로'}, 'key2': {'버피', '월로'}}
- ☐ {'key1': ['월로', '버피', '자일스', '자일스'], 'key2': ['월로', '버피', '버피', '월로']}
- ☒ {'key1': ['월로', '버', '피', '자', '일', '스', '자', '일', '스'], 'key2': ['월로', '버', '피', '버', '피', '월', '로']} ✕
- ☐ {'key1': {'월로': ['버피', '자일스', '자일스'], 'key2': {'월로': ['버피', '버피', '월로']}}
- ☐ 모르겠다.

Correct answer

- ☒ {'key1': {'버피', '자일스', '월로'}, 'key2': {'버피', '월로'}}



✓ 다음 코드 중 에러가 발생하는 코드는? *

1/1

```
sunnydale = dict(name="진더", age=17, hobby="게임")
```

- ☐ sunnydale.update({"friend": "월로"})
- ☐ sunnydale.get("grade")
- ☒ sunnydale.pop()
- ☐ sunnydale.popitem()
- ☐ sunnydale.clear()
- ☐ 모르겠다.



✓ 다음 설명 중 틀린 것은? *

1/1

- ☐ defaultdict: 누락된 키에 데이터를 넣을 때 기본 값의 형태를 설정해준다.
- ☐ OrderedDict: 삽입 순서대로 항목을 저장한다.
- ☐ Counter: 해시 가능한 객체를 카운팅한다. 셋 연산을 사용할 수 있다.
- ☒ 표준 딕셔너리: 메모리의 최적화를 위해 Python 3.7 이상 버전에서 항목의 삽입 순서를 보존하지 않는다.
- ☐ 모르겠다.



✓ 애너그램을 확인하는 데 가장 좋은 해결책은? *

1/1

- ☐ set()
- ☐ list()
- ☒ dict()
- ☐ ord()
- ☐ 모르겠다.



This content is neither created nor endorsed by Google. - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

