

# defaultdict

- Dictionary의 value 에 초기값을 부여할 수 있는 객체
- 숫자, 리스트 등으로 초기화 가능

```
from collections import defaultdict
dd = defaultdict(int) #값의 디폴트 자료형을 지정(int)하여 디폴트딕셔너리 생성
dd['X'] = 1
dd['B'] #값을 설정하지 않으면 디폴트 자료형의 초기값으로 설정 (int형 0)
dd['C'] = 3
print(dd)
```

```
defaultdict(<class 'int'>, {'X': 1, 'B': 0, 'C': 3})
```

```
#list dictionary
s = [('yellow', 1), ('blue', 2), ('yellow', 3), ('blue', 4), ('red', 1)]
d = defaultdict(list) #list 자료형으로 디폴트딕셔너리 생성
for k, v in s:
    d[k].append(v) #리스트 형식의 값에 요소 추가
d['green'] #value로 리스트 초기화
print(d) #디셔너리 값 확인
print(sorted(d.items())) #키의 가나다순으로 정렬
```

```
defaultdict(<class 'list'>, {'yellow': [1, 3], 'blue': [2, 4], 'red': [1], 'green': []})
[('blue', [2, 4]), ('green', []), ('red', [1]), ('yellow', [1, 3])]
```

# defaultdict

- defaultdict 초기값 설정

```
#defaultdict로 초기값 설정
d = defaultdict(kim=25, min=22, jin=23)
print(d)
d = defaultdict(int, kim=25, min=22, jin=23)
print(d)
print(d['kim'])
d = defaultdict(list, kim=[100,50,70], min=[88,87,80], jin=[90,89,97])
print(d)
```

```
defaultdict(None, {'kim': 25, 'min': 22, 'jin': 23})
defaultdict(<class 'int'>, {'kim': 25, 'min': 22, 'jin': 23})
25
defaultdict(<class 'list'>, {'kim': [100, 50, 70], 'min': [88, 87, 80], 'jin': [90, 89, 97]})
```

```
#각 문자의 개수를 value로 저장하는 딕셔너리 생성
s = 'abcdaacc'
d = defaultdict(int)
for k in s:
    d[k] += 1 #key에 대한 값이 없을때 value가 0으로 초기화되어 카운팅 가능
print(d)
print(sorted(d.items()))
```

```
defaultdict(<class 'int'>, {'a': 3, 'b': 1, 'c': 3, 'd': 1})
[('a', 3), ('b', 1), ('c', 3), ('d', 1)]
```

# Counter

- 리스트, 문자열 등의 개수를 세는 객체
- 결과는 딕셔너리 형태로 반환

```
from collections import Counter
words = ['red', 'blue', 'red', 'green', 'blue', 'blue']
cnt = Counter(words)  #list counting
print(cnt)

st = "aabccadd"
cnt = Counter(st)     #string counting
print(cnt)

print(cnt.keys())
print(cnt.values())
for k, v in cnt.items():
    print(k,v)
```

```
Counter({'blue': 3, 'red': 2, 'green': 1})
Counter({'a': 3, 'd': 3, 'c': 2, 'b': 1})
dict_keys(['a', 'b', 'c', 'd'])
dict_values([3, 1, 2, 3])
a 3
b 1
c 2
d 3
```

# Counter

## • Counter 객체의 함수(method)

- elements() : Counter의 요소들 반환
- update() : Counter의 요소로 포함되면서 덧셈 처리
- subtract() : Counter에서 요소들의 값에 대한 뺄셈처리
- most\_common(n) : 가장 빈도수가 큰 요소에서부터 작은 순으로 n개 추출

```
#Count의 함수(메소드)
a1 = "aabccddadd"
a2 = "abde"
cnt = Counter(a1)           # a1으로 카운터 객체 생성
print(cnt)
print(sorted(cnt.elements())) # 요소들 확인
cnt.update(a2)              # a2를 추가하여 수정
print(cnt)
cnt.subtract(a2)            # a2의 값을 뺀 결과
print(cnt)
mc = cnt.most_common(3)     # 개수가 큰값에서 작은값순으로 3개만 추출
print(mc)
```

```
Counter({'a': 3, 'd': 3, 'c': 2, 'b': 1})
['a', 'a', 'a', 'b', 'c', 'c', 'd', 'd', 'd']
Counter({'a': 4, 'd': 4, 'b': 2, 'c': 2, 'e': 1})
Counter({'a': 3, 'd': 3, 'c': 2, 'b': 1, 'e': 0})
[('a', 3), ('d', 3), ('c', 2)]
```

# Counter

- Counter 연산
  - 덧셈(+), 뺄셈(-)
  - 교집합(&), 합집합(|)

```

a1 = "aabccdaddd"
a2 = "abd"
x = Counter(a1)
y = Counter(a2)
print("x = ", x)
print("y = ", y)
print("x + y = ", x + y)
print("x - y = ", x - y)
print("x & y = ", x & y)    #교집합
print("x | y = ", x | y)    #합집합

```

```

x = Counter({'a': 3, 'd': 3, 'c': 2, 'b': 1})
y = Counter({'a': 1, 'b': 1, 'd': 1})
x + y = Counter({'a': 4, 'd': 4, 'b': 2, 'c': 2})
x - y = Counter({'a': 2, 'c': 2, 'd': 2})
x & y = Counter({'a': 1, 'b': 1, 'd': 1})
x | y = Counter({'a': 3, 'd': 3, 'c': 2, 'b': 1})

```

# QUIZ!

## Report 8. 문장의 단어 카운팅 프로그램

- (1)두 개의 문장을 문자열로 입력
- (2)두 문장의 단어들을 카운팅하여 하나의 Counter 객체에 결합
- (3)단어 중 빈도수가 가장 큰 값 2개만 추출하여 확인

- 학습활동 결과는 e-루리에 제출바랍니다.
- 제목 : Report8. 단어 카운팅 프로그램
- 제출내용 : Report8\_count \_성명. ipynb 파일을 제출
- 제출기한 : 2021년 5월 18일 오후 11:58