BOW 직접 구현

```
[ ]  1 docs = ['오늘 동물원에서 코끼리 원숭이를 보고 코끼리 원숭이에게 먹이를 줬어',
     2         '오늘 동물원에서 원숭이에게 사과를 줬어']
```

1. 토큰화 : 공백으로 토큰화 진행

```
[ ]  1 doc_ls = []
     2 for doc in docs:
     3     doc_ls.append(doc.split(' '))
     4 doc_ls
```

```
[['오늘', '동물원에서', '코끼리', '원숭이를', '보고', '코끼리', '원숭이에게', '먹이를', '줬어'],
 ['오늘', '동물원에서', '원숭이에게', '사과를', '줬어']]
```

2. 유니크한 토큰 사전 구하기

2-1. 빈 딕셔너리 생성

```
[ ]  1 from collections import defaultdict
     2
     3 word2id = defaultdict(lambda : len(word2id))
     4 word2id
```

```
defaultdict(<function __main__.<lambda>()>, {})
```

2-2. 위에서 생성한 빈 딕셔너리에 유니크한 토큰 넣기

```
[ ]  1 for doc in doc_ls:
     2   print(doc)
     3   for token in doc:
     4     word2id[token]
     5     print(token)
     6     print('\t', word2id)
```

```
[ ] ['오늘', '동물원에서', '코끼리', '원숭이를', '보고', '코끼리', '원숭이에게', '먹이를', '줬어']
오늘
        defaultdict(<function <lambda> at 0x79a9aa5d9750>, {'오늘': 0})
동물원에서
        defaultdict(<function <lambda> at 0x79a9aa5d9750>, {'오늘': 0, '동물원에서': 1})
코끼리
        defaultdict(<function <lambda> at 0x79a9aa5d9750>, {'오늘': 0, '동물원에서': 1, '코끼리': 2})
원숭이를
        defaultdict(<function <lambda> at 0x79a9aa5d9750>, {'오늘': 0, '동물원에서': 1, '코끼리': 2, '원숭이를': 3})
보고
        defaultdict(<function <lambda> at 0x79a9aa5d9750>, {'오늘': 0, '동물원에서': 1, '코끼리': 2, '원숭이를': 3, '보고': 4})
코끼리
        defaultdict(<function <lambda> at 0x79a9aa5d9750>, {'오늘': 0, '동물원에서': 1, '코끼리': 2, '원숭이를': 3, '보고': 4})
원숭이에게
        defaultdict(<function <lambda> at 0x79a9aa5d9750>, {'오늘': 0, '동물원에서': 1, '코끼리': 2, '원숭이를': 3, '보고': 4, '원숭이에게': 5})
먹이를
        defaultdict(<function <lambda> at 0x79a9aa5d9750>, {'오늘': 0, '동물원에서': 1, '코끼리': 2, '원숭이를': 3, '보고': 4, '원숭이에게': 5, '먹이를': 6})
줬어
        defaultdict(<function <lambda> at 0x79a9aa5d9750>, {'오늘': 0, '동물원에서': 1, '코끼리': 2, '원숭이를': 3, '보고': 4, '원숭이에게': 5, '먹이를': 6, '
['오늘', '동물원에서', '원숭이에게', '사과를', '줬어']
오늘
        defaultdict(<function <lambda> at 0x79a9aa5d9750>, {'오늘': 0, '동물원에서': 1, '코끼리': 2, '원숭이를': 3, '보고': 4, '원숭이에게': 5, '먹이를': 6, '
동물원에서
        defaultdict(<function <lambda> at 0x79a9aa5d9750>, {'오늘': 0, '동물원에서': 1, '코끼리': 2, '원숭이를': 3, '보고': 4, '원숭이에게': 5, '먹이를': 6, '
원숭이에게
        defaultdict(<function <lambda> at 0x79a9aa5d9750>, {'오늘': 0, '동물원에서': 1, '코끼리': 2, '원숭이를': 3, '보고': 4, '원숭이에게': 5, '먹이를': 6, '
사과를
        defaultdict(<function <lambda> at 0x79a9aa5d9750>, {'오늘': 0, '동물원에서': 1, '코끼리': 2, '원숭이를': 3, '보고': 4, '원숭이에게': 5, '먹이를': 6, '
줬어
        defaultdict(<function <lambda> at 0x79a9aa5d9750>, {'오늘': 0, '동물원에서': 1, '코끼리': 2, '원숭이를': 3, '보고': 4, '원숭이에게': 5, '먹이를': 6, '
```

```
1 word2id
```

```
defaultdict(<function __main__.<lambda>()>,
            {'오늘': 0,
             '동물원에서': 1,
             '코끼리': 2,
             '원숭이를': 3,
             '보고': 4,
             '원숭이에게': 5,
             '먹이를': 6,
             '줬어': 7,
             '사과를': 8})
```

2. BOW 구하기

```
[ ]  1 import numpy as np
     2 BoW_ls = []
     3 for i, doc in enumerate(doc_ls):
     4   bow = np.zeros(len(word2id),dtype = int)
     5   print(bow)
     6   for token in doc:
     7     bow[word2id[token]] +=1
     8     print(token,' => ',bow)
     9   BoW_ls.append(bow.tolist())
```

```
[0 0 0 0 0 0 0 0 0]
오늘  =>  [1 0 0 0 0 0 0 0 0]
동물원에서  =>  [1 1 0 0 0 0 0 0 0]
코끼리  =>  [1 1 1 0 0 0 0 0 0]
원숭이를  =>  [1 1 1 1 0 0 0 0 0]
보고  =>  [1 1 1 1 1 0 0 0 0]
코끼리  =>  [1 1 2 1 1 0 0 0 0]
원숭이에게  =>  [1 1 2 1 1 1 0 0 0]
먹이를  =>  [1 1 2 1 1 1 1 0 0]
줬어  =>  [1 1 2 1 1 1 1 1 0]
[0 0 0 0 0 0 0 0 0]
오늘  =>  [1 0 0 0 0 0 0 0 0]
동물원에서  =>  [1 1 0 0 0 0 0 0 0]
원숭이에게  =>  [1 1 0 0 0 1 0 0 0]
사과를  =>  [1 1 0 0 0 1 0 0 1]
줬어  =>  [1 1 0 0 0 1 0 1 1]
```

```
[ ]  1 BoW_ls
```

```
[[1, 1, 2, 1, 1, 1, 1, 1, 0], [1, 1, 0, 0, 0, 1, 0, 1, 1]]
```

```
[ ]  1 from IPython.core import display as ICD
     2 import pandas as pd
     3 sorted_vocab = sorted((value, key) for key, value in word2id.items())
     4 print('sorted_vocab',sorted_vocab)
     5
     6 vocab = []
     7 for v in sorted_vocab:
     8     vocab.append(v[1])
     9 print('vocab',vocab)
    10 for i in range(len(docs)) :
    11     print("문서{} : {}".format(i, docs[i]))
    12     ICD.display(pd.DataFrame([BoW_ls[i]], columns=vocab))
    13     print("\n\n")
```

```
sorted_vocab [(0, '오늘'), (1, '동물원에서'), (2, '코끼리'), (3, '원숭이를'), (4, '보고'), (5, '원숭이에게'), (6, '먹이를'), (7, '줬어'), (8, '사과를')]
vocab ['오늘', '동물원에서', '코끼리', '원숭이를', '보고', '원숭이에게', '먹이를', '줬어', '사과를']
문서0 : 오늘 동물원에서 코끼리 원숭이를 보고 코끼리 원숭이에게 먹이를 줬어
```

	오늘	동물원에서	코끼리	원숭이를	보고	원숭이에게	먹이를	줬어	사과를
0	1	1	2	1	1	1	1	1	0

```
문서1 : 오늘 동물원에서 원숭이에게 사과를 줬어
```

	오늘	동물원에서	코끼리	원숭이를	보고	원숭이에게	먹이를	줬어	사과를
0	1	1	0	0	0	1	0	1	1

단어의 순서를 고려하지 않은 BOW

```
[1]  1 docs = ['나는 양념 치킨을 좋아해 하지만 후라이드 치킨을 싫어해',
     2         '나는 후라이드 치킨을 좋아해 하지만 양념 치킨을 싫어해']
     3 docs
```

```
['나는 양념 치킨을 좋아해 하지만 후라이드 치킨을 싫어해', '나는 후라이드 치킨을 좋아해 하지만 양념 치킨을 싫어해']
```

[4]
```
doc_ls = []
for doc in docs:
  # print(doc.split(' '))
  doc_ls.append(doc.split(' '))
  # break
doc_ls
```

```
[['나는', '양념', '치킨을', '좋아해', '하지만', '후라이드', '치킨을', '싫어해'],
 ['나는', '후라이드', '치킨을', '좋아해', '하지만', '양념', '치킨을', '싫어해']]
```

[11]
```
from collections import defaultdict

word2id = defaultdict(lambda : len(word2id))
for doc in doc_ls:
    for token in doc:
        word2id[token]
word2id
```

```
defaultdict(<function __main__.<lambda>()>,
            {'나는': 0,
             '양념': 1,
             '치킨을': 2,
             '좋아해': 3,
             '하지만': 4,
             '후라이드': 5,
             '싫어해': 6})
```

[22]
```
# dictionary 값 확인
print(word2id['양념'])
print(word2id['치킨을'])
```

```
1
2
```

```python
import numpy as np

BoW_ls = []
for i, doc in enumerate(doc_ls):
  bow = np.zeros(len(word2id), dtype=int)
  # print(bow)
  for token in doc:
    # print(token)
    bow[word2id[token]] += 1 # 해당 토큰의 위치(column)
    # print(bow)
    # break
  BoW_ls.append(bow.tolist())
```

0초

[24]
```python
BoW_ls
```

```
[[1, 1, 2, 1, 1, 1, 1], [1, 1, 2, 1, 1, 1, 1]]
```

[ ]
```python
import numpy as np

BoW_ls = []
for i, doc in enumerate(doc_ls):
    bow = np.zeros(len(word2id), dtype=int)
    for token in doc:
        bow[word2id[token]] += 1 # 해당 토큰의 위치(column)
    BoW_ls.append(bow.tolist())
BoW_ls
```

```
[[1, 1, 2, 1, 1, 1, 1], [1, 1, 2, 1, 1, 1, 1]]
```

sklearn

```
docs = ['오늘 동물원에서 코끼리 원숭이를 보고 코끼리 원숭이에게 먹이를 줬어',
        '오늘 동물원에서 코끼리 원숭이를 보고 코끼리 원숭이에게 먹이를 줬어 오늘 동물원에서 원숭이에게 사과를 줬어']
```

```
# 토큰빈도계산 : CountVectorizer
from sklearn.feature_extraction.text import CountVectorizer
#선언
count_vect = CountVectorizer()

BoW = count_vect.fit_transform(docs)
BoW.toarray()
```

```
array([[1, 1, 1, 0, 1, 1, 1, 1, 2],
       [2, 1, 1, 1, 2, 1, 2, 2, 2]])
```

```
from IPython.core import display as ICD

vocab = count_vect.get_feature_names_out()
for i in range(len(docs)) :
    print("문서{} : {}".format(i, docs[i]))
    ICD.display(pd.DataFrame([BoW.toarray()[i]], columns=vocab))
    print("\n\n")
```

문서0 : 오늘 동물원에서 코끼리 원숭이를 보고 코끼리 원숭이에게 먹이를 줬어

	동물원에서	먹이를	보고	사과를	오늘	원숭이를	원숭이에게	줬어	코끼리
0	1	1	1	0	1	1	1	1	2

문서1 : 오늘 동물원에서 코끼리 원숭이를 보고 코끼리 원숭이에게 먹이를 줬어 오늘 동물원에서 원숭이에게 사과를 줬어

	동물원에서	먹이를	보고	사과를	오늘	원숭이를	원숭이에게	줬어	코끼리
0	2	1	1	1	2	1	2	2	2