8~10번째

‘nltk’폴더에서 ‘nltk.corpus’파일 안에 있는 ‘gutenberg’를 연다.

이 안에 있는 txt파일들을 사용할 것이다.

15~19번째

‘alice’의 값을 gutenberg에 있는 파일 중 ‘carroll-alice.txt’로 설정

‘sample\_text’의 값은 ‘We will discuss~’ 이하에 있는 문장으로 설정된다.

22번째 줄

print len(alice) : ‘alice.txt’에 있는 글자 수를 구해 출력한다.

24번째 줄

print alice[0:100] : ‘alice.txt’에 있는 텍스트 중, 처음부터 100번째 글자까지 출력. (기호, 띄어쓰기 등등 다 포함)

28~30번째

‘default\_st’의 값을 ‘nltk.sent\_tokenize’로 설정

‘alice\_sentences’의 값은 default\_st의 값 안에 있는 텍스트 ‘alice’로 설정

‘sample\_sentences’의 값은 default\_st의 값 안에 있는 텍스트 ‘sample\_text’로 설정

32~37번째줄

‘Total sentences in sample\_text : ’를 프린트하고 ‘sample\_text’의 글자 수를 출력

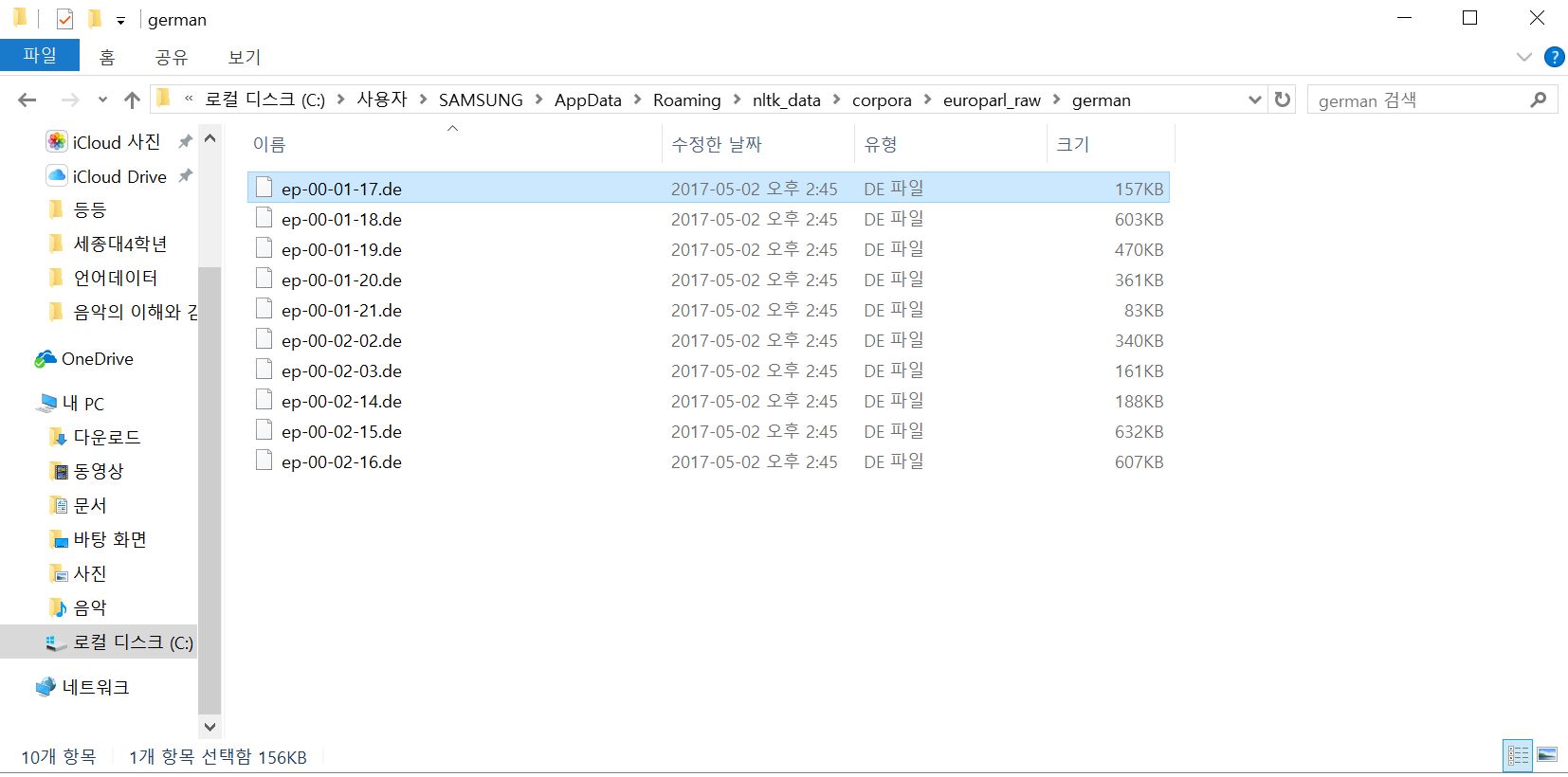
‘Sample text sentences : -‘를 프린트하고 위의 ‘sample\_text’를 줄에 맞춰 정렬해서 출력

‘\n’으로 한 줄 띄고, ‘Total sentences in alice : ‘와 alice.txt의 글자 수를 출력한다.

‘First 5 sentences in alice : ‘를 출력하고 alice.txt.에서 첫 5문장을 출력한다. 줄 바꿈이 있는 곳에는 ‘\n’으로 표시

41번째 줄

‘nltk.corpus’에서 ‘europar1\_raw’를 연다



43번째 줄  
‘german\_text’의 값을 ‘german’파일에 있는 ‘ep-00-01-17.de’로 설정

45~47번째 줄

‘german\_text’의 글자 수를 출력

‘german\_text’에서 100번째 글자까지 출력

51번째 줄

‘german\_sentences\_def’의 값을 위에서 정했던 default\_st의 값 중 text는 ‘german\_text’로 설정하고 언어는 ‘german’

54~55번째 줄

‘german\_tokenizer’의 값은 ‘nltk.data.load’에서 가져온다.

‘german\_sentences’는 위의 tokenizer에서 ‘german\_text’를 기본값으로 가진다.

59번째 줄

‘german\_tokenizer’의 변수형을 출력 class ‘german\_tokenizer’을 출력

63~66번째 줄

63번째 줄에 대한 것이 사실일 때, true라고 출력하고

그 밑에 있는 65번째 줄에 대한 ‘german\_sentences’의 문장 5줄을 출력한다.

70~72번째 줄

‘punkt\_st’의 값을 ‘nltk.tokenize.PunktSentenceTokenizer()’로 설정

위에 입력한 ‘sample\_sentences’의 값은 ‘punkt\_st.tokenize(sample\_text)로 설정

sample\_sentences 출력 > 이 때 문장이 끝나는 곳마다 줄을 바꿔서 정렬하여 출력

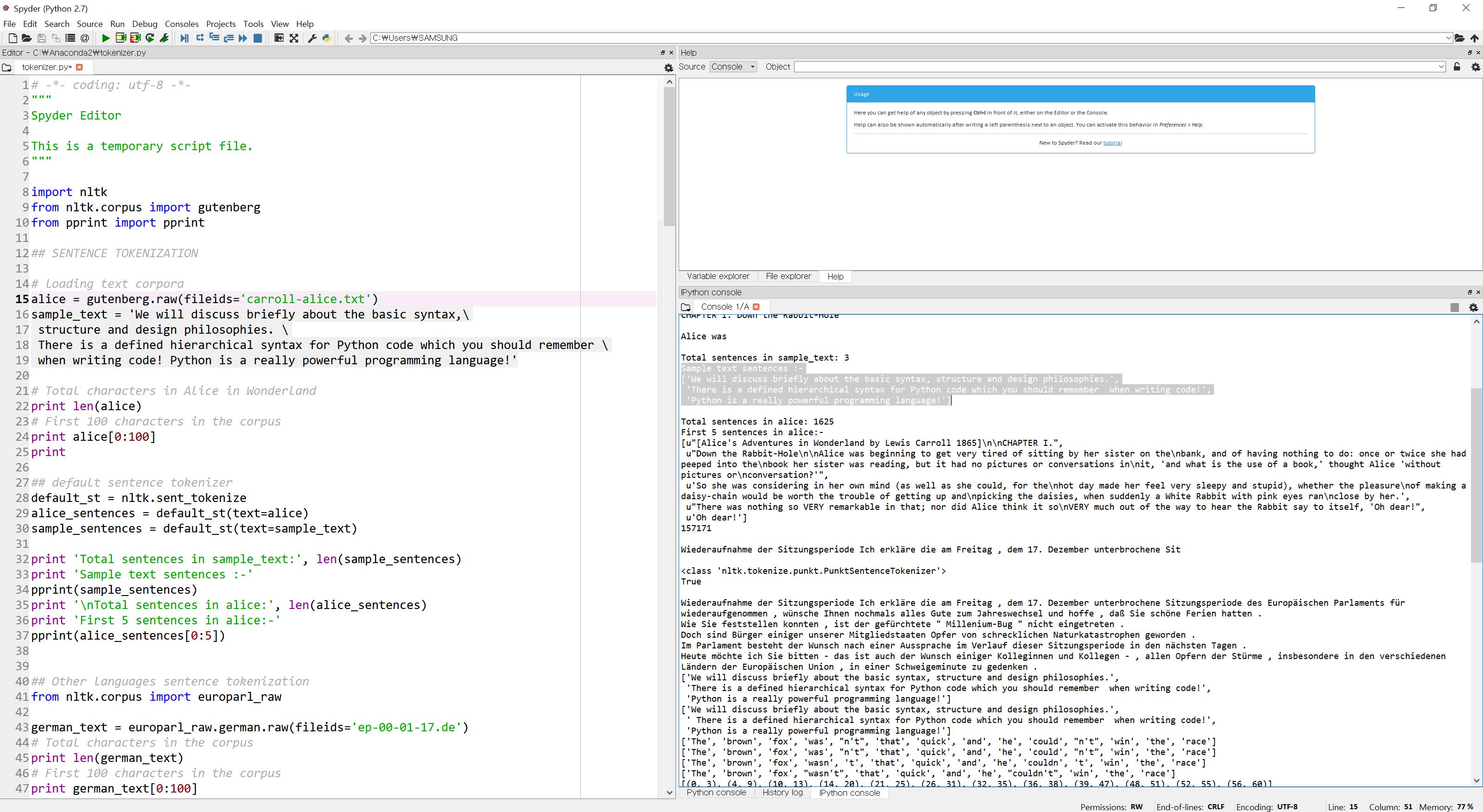
75~80번째 줄

‘SENTENCE\_TOKENS\_PATTERN’을 전체 모든 기호 및 문자로 설정

‘regex\_st’의 값은 ‘nltk.tokenize.RegexpTokenizer’의 pattern은 SENTENCE\_TOKENS\_PATTERN을 가진다.

sample\_sentences 의 값은 위의 regex\_st값을 가진다.

sample\_sentences 출력



84번째: sentence의 값을 "The brown fox wasn't that quick and he couldn't win the race" 로 정해준다.

87번째: default\_wt의 값을 nltk.word\_tokenize로 정해주고

88번쩨: words의 값을 default\_wt(sentence)로 정해준다

89번째: 따라서 word를 print했을 때 단어별로 문장을 나눠서 출력해준다.

['The', 'brown', 'fox', 'was', "n't", 'that', 'quick', 'and', 'he', 'could', "n't", 'win', 'the', 'race'] 이렇게 출력해준다.

92번째 treebank\_wt의 값은 nltk.TreebankWordTokenizer()로 정해준다.

93번째 word는 treebanck\_wt.tokenize(sentence)로 정해준후

94번째 word를 print 했을 경우 문장하나하나를 구문 분석하여 코퍼스로 구성해 단어별로 문장을 나눠서 출력해준다.

['The', 'brown', 'fox', 'was', "n't", 'that', 'quick', 'and', 'he', 'could', "n't", 'win', 'the', 'race'] 위와 같이 이렇게 출력해준다.

# regex word tokenizer

TOKEN\_PATTERN = r'\w+'

regex\_wt = nltk.RegexpTokenizer(pattern=TOKEN\_PATTERN,

gaps=False)

words = regex\_wt.tokenize(sentence)

print words

96-101번째는 토큰 자체를 식별하는 패턴을 줌으로 print word해줄 경우

['The', 'brown', 'fox', 'wasn', 't', 'that', 'quick', 'and', 'he', 'couldn', 't', 'win', 'the', 'race'] 이 값을 출력해준다. 위 2개와는 다르게 ‘wasn’,를 하나로 묶어 출력해준다.

GAP\_PATTERN = r'\s+'

regex\_wt = nltk.RegexpTokenizer(pattern=GAP\_PATTERN,

gaps=True)

words = regex\_wt.tokenize(sentence)

print words

103-107번째는 토큰의 간격을 식별하는 패턴을 사용한다.

따라서 출력했을 경우에 ['The', 'brown', 'fox', "wasn't", 'that', 'quick', 'and', 'he', "couldn't", 'win', 'the', 'race'] 문장에 있는 간격마다 나누어서 이렇게 출력해준다.

word\_indices = list(regex\_wt.span\_tokenize(sentence))

print word\_indices

print [sentence[start:end] for start, end in word\_indices]

109-111번째 줄은 각 토큰의 시작과 끝의 지수를 얻은 다음 출력하도록 해준다.

따라서 이 값은

[(0, 3), (4, 9), (10, 13), (14, 20), (21, 25), (26, 31), (32, 35), (36, 38), (39, 47), (48, 51), (52, 55), (56, 60)]

['The', 'brown', 'fox', "wasn't", 'that', 'quick', 'and', 'he', "couldn't", 'win', 'the', 'race'] 이렇게 출력이 된다.

# derived regex tokenizers

wordpunkt\_wt = nltk.WordPunctTokenizer()

words = wordpunkt\_wt.tokenize(sentence)

print words

113-116번째는 TheWordPunkTokenizer 은 r’\w+’ 토큰 패턴을 사용하여, 문장을 독립적으로 알파벳, 그리고 알파벳이 아닌 토큰으로 토큰화 해주게 한다. 따라서 출력을 했을 때에

['The', 'brown', 'fox', 'wasn', "'", 't', 'that', 'quick', 'and', 'he', 'couldn', "'", 't', 'win', 'the', 'race'] 이렇게 출력해준다.

whitespace\_wt = nltk.WhitespaceTokenizer()

words = whitespace\_wt.tokenize(sentence)

print words

마지막 118-120번째는 TheWhiteSpace Toknizer로 인해 문장을 탭, 줄바꿈 공백과 같은 공백에 기반한 당어로 토큰화 시켜주어 문장을 나눈다. 따라서 이 명령에 따라 출력할 경우

['The', 'brown', 'fox', "wasn't", 'that', 'quick', 'and', 'he', "couldn't", 'win', 'the', 'race'] 이렇게 출력해준다.