KoNLPy 한국어 처리 패키지

KoNLPy(코엔엘파이라고 읽는다)는 한국어 정보처리를 위한 파이썬 패키지이다.

In [1]:

```
import warnings
warnings.simplefilter("ignore")

import konlpy
konlpy.__version__
```

Out[1]:

```
'0.5.1'
```

한국어 말뭉치

KoNLPy에서는 대한민국 헌법 말뭉치인 `kolaw` 와 국회법안 말뭉치인 `kobill` 을 제공한다. 각 말뭉치가 포함하는 파일의 이름은 `fields` 메서드로 알 수 있고 `open` 메서드로 해당 파일의 텍스트를 읽어들인다.

In [2]:

```
from konlpy.corpus import kolaw
kolaw.fileids()
```

Out[2]:

```
['constitution.txt']
```

In [3]:

```
c = kolaw.open('constitution.txt').read()
print(c[:40])
```

```
대한민국헌법

유구한 역사와 전통에 빛나는 우리 대한국민은 3·1운동으로
```

In [4]:

```
from konlpy.corpus import kobill
kobill.fileids()
```

Out[4]:

```
['1809895.txt',
 '1809890.txt',
 '1809899.txt',
 '1809898.txt',
 '1809891.txt',
 '1809892.txt',
 '1809894.txt',
 '1809893.txt',
 '1809896.txt',
 '1809897.txt']
```

In [5]:

```
d = kobill.open('1809890.txt').read()
print(d[:40])
```

```
지방공무원법 일부개정법률안

(정의화의원 대표발의 )

 의 안
 번 호
```

형태소 분석

KoNLPy는 다음과 같은 다양한 형태소 분석, 태깅 라이브러리를 파이썬에서 쉽게 사용할 수 있도록 모아놓았다.

- Hannanum: 한나눔. KAIST Semantic Web Research Center 개발.
  - http://semanticweb.kaist.ac.kr/hannanum/ (http://semanticweb.kaist.ac.kr/hannanum/)
- Kkma: 꼬꼬마. 서울대학교 IDS(Intelligent Data Systems) 연구실 개발.
  - http://kkma.snu.ac.kr/ (http://kkma.snu.ac.kr/)
- Komoran: 코모란. Shineware에서 개발.
  - https://github.com/shin285/KOMORAN (https://github.com/shin285/KOMORAN)
- Mecab: 메카브. 일본어용 형태소 분석기를 한국어를 사용할 수 있도록 수정.
  - https://bitbucket.org/eunjeon/mecab-ko (https://bitbucket.org/eunjeon/mecab-ko)
- Open Korean Text: 오픈 소스 한국어 분석기. 과거 트위터 형태소 분석기.
  - https://github.com/open-korean-text/open-korean-text (https://github.com/open-korean-text/open-korean-text)

여기에서는 한나눔, 꼬꼬마, 오픈코리안텍스트 형태소만 예제로 포함하였다.

In [6]:

```
from konlpy.tag import *

hannanum = Hannanum()
kkma = Kkma()
komoran = Komoran()
mecab = Mecab()
okt = Okt()
```

이 클래스들은 다음과 같은 메서드를 공통적으로 제공한다.

- nouns : 명사 추출
- morphs : 형태소 추출
- pos : 품사 부착

명사 추출

문자열에서 명사만 추출하려면 noun 명령을 사용한다.

In [7]:

```
hannanum.nouns(c[:40])
```

Out[7]:

```
['대한민국헌법', '유구', '역사', '전통', '빛', '우리', '대한국민', '3·1운동']
```

In [8]:

```
kkma.nouns(c[:40])
```

Out[8]:

```
['대한',
 '대한민국',
 '대한민국헌법',
 '민국',
 '헌법',
 '유구',
 '역사',
 '전통',
 '우리',
 '국민',
 '3',
 '1',
 '1운동',
 '운동']
```

In [9]:

```
# komoran은 빈줄이 있으면 에러가 남
komoran.nouns("₩n".join([s for s in c[:40].split("₩n") if s]))
```

Out[9]:

['대한민국', '헌법', '역사', '전통', '국민', '운동']

In [10]:

```
mecab.nouns(c[:40])
```

Out[10]:

['대한민국', '헌법', '역사', '전통', '우리', '국민', '운동']

In [11]:

```
okt.nouns(c[:40])
```

Out[11]:

['대한민국', '헌법', '유구', '역사', '전통', '우리', '국민', '운동']

형태소 추출

명사 뿐 아니라 모든 품사의 형태소를 알아내려면 `morphs` 라는 명령을 사용한다.

In [12]:

```
hannanum.morphs(c[:40])
```

Out[12]:

```
['대한민국헌법',
 '유구',
 '하',
 'ㄴ',
 '역사',
 '와',
 '전통',
 '에',
 '빛',
 '나는',
 '우리',
 '대한국민',
 '은',
 '3·1운동',
 '으로']
```

In [13]:

```
kkma.morphs(c[:40])
```

Out[13]:

```
['대한민국',
 '헌법',
 '유구',
 '하',
 'ㄴ',
 '역사',
 '와',
 '전통',
 '에',
 '빛나',
 '는',
 '우리',
 '대하',
 'ㄴ',
 '국민',
 '은',
 '3',
 '.',
 '1',
 '운동',
 '으로']
```

In [14]:

```
# komoran은 빈줄이 있으면 에러가 남
komoran.morphs("₩n".join([s for s in c[:40].split("₩n") if s]))
```

Out[14]:

```
['대한민국',
 '헌법',
 '유구',
 '하',
 'ㄴ',
 '역사',
 '와',
 '전통',
 '에',
 '빛나',
 '는',
 '우리',
 '대하',
 'ㄴ',
 '국민',
 '은',
 '3',
 '.',
 '1',
 '운동',
 '으로']
```

In [15]:

```
mecab.morphs(c[:40])
```

Out[15]:

```
['대한민국',
 '헌법',
 '유구',
 '한',
 '역사',
 '와',
 '전통',
 '에',
 '빛나',
 '는',
 '우리',
 '대한',
 '국민',
 '은',
 '3',
 '.',
 '1',
 '운동',
 '으로']
```

In [16]:

```
okt.morphs(c[:40])
```

Out[16]:

```
['대한민국',
 '헌법',
 'WnWn',
 '유구',
 '한',
 '역사',
 '와',
 '전통',
 '에',
 '빛나는',
 '우리',
 '대',
 '한',
 '국민',
 '은',
 '3',
 '.',
 '1',
 '운동',
 '으로']
```

품사 부착

pos 명령을 사용하면 품사 부착을 한다.

한국어 품사 태그세트로는 "21세기 세종계획 품사 태그세트"를 비롯하여 다양한 품사 태그세트가 있다. 형태소 분석기마다 사용하는 품사 태그가 다르므로 각 형태소 분석기에 대한 문서를 참조한다.

In [17]:

```
hannanum.pos(c[:40])
```

Out[17]:

```
[('대한민국헌법', 'N'),
 ('유구', 'N'),
 ('하', 'X'),
 ('ㄴ', 'E'),
 ('역사', 'N'),
 ('와', 'J'),
 ('전통', 'N'),
 ('에', 'J'),
 ('빛', 'N'),
 ('나는', 'J'),
 ('우리', 'N'),
 ('대한국민', 'N'),
 ('은', 'J'),
 ('3·1운동', 'N'),
 ('으로', 'J')]
```

In [18]:

```
kkma.pos(c[:40])
```

Out[18]:

```
[('대한민국', 'NNG'),
 ('헌법', 'NNG'),
 ('유구', 'NNG'),
 ('하', 'XSV'),
 ('ㄴ', 'ETD'),
 ('역사', 'NNG'),
 ('와', 'JC'),
 ('전통', 'NNG'),
 ('에', 'JKM'),
 ('빛나', 'VV'),
 ('는', 'ETD'),
 ('우리', 'NNM'),
 ('대하', 'VV'),
 ('ㄴ', 'ETD'),
 ('국민', 'NNG'),
 ('은', 'JX'),
 ('3', 'NR'),
 ('·', 'SP'),
 ('1', 'NR'),
 ('운동', 'NNG'),
 ('으로', 'JKM')]
```

In [19]:

```
# komoran은 빈줄이 있으면 에러가 남
komoran.pos("₩n".join([s for s in c[:40].split("₩n") if s]))
```

Out[19]:

```
[('대한민국', 'NNP'),
 ('헌법', 'NNP'),
 ('유구', 'XR'),
 ('하', 'XSA'),
 ('ㄴ', 'ETM'),
 ('역사', 'NNG'),
 ('와', 'JC'),
 ('전통', 'NNG'),
 ('에', 'JKB'),
 ('빛나', 'VV'),
 ('는', 'ETM'),
 ('우리', 'NP'),
 ('대하', 'VV'),
 ('ㄴ', 'ETM'),
 ('국민', 'NNP'),
 ('은', 'JX'),
 ('3', 'SN'),
 ('·', 'SP'),
 ('1', 'SN'),
 ('운동', 'NNP'),
 ('으로', 'JKB')]
```

In [20]:

```
mecab.pos(c[:40])
```

Out[20]:

```
[('대한민국', 'NNP'),
 ('헌법', 'NNG'),
 ('유구', 'XR'),
 ('한', 'XSA+ETM'),
 ('역사', 'NNG'),
 ('와', 'JC'),
 ('전통', 'NNG'),
 ('에', 'JKB'),
 ('빛나', 'VV'),
 ('는', 'ETM'),
 ('우리', 'NP'),
 ('대한', 'VV+ETM'),
 ('국민', 'NNG'),
 ('은', 'JX'),
 ('3', 'SN'),
 ('·', 'SC'),
 ('1', 'SN'),
 ('운동', 'NNG'),
 ('으로', 'JKB')]
```

In [21]:

```
okt.pos(c[:40])
```

Out[21]:

```
[('대한민국', 'Noun'),
 ('헌법', 'Noun'),
 ('ㅜnㅜn', 'Foreign'),
 ('유구', 'Noun'),
 ('한', 'Josa'),
 ('역사', 'Noun'),
 ('와', 'Josa'),
 ('전통', 'Noun'),
 ('에', 'Josa'),
 ('빛나는', 'Verb'),
 ('우리', 'Noun'),
 ('대', 'Modifier'),
 ('한', 'Modifier'),
 ('국민', 'Noun'),
 ('은', 'Josa'),
 ('3', 'Number'),
 ('·', 'Punctuation'),
 ('1', 'Number'),
 ('운동', 'Noun'),
 ('으로', 'Josa')]
```

부착되는 품사 태그의 기호와 의미는 `tagset` 속성으로 확인할 수 있다.

In [22]:

```
okt.tagset
```

Out[22]:

```
{'Adjective': '형용사',
 'Adverb': '부사',
 'Alpha': '알파벳',
 'Conjunction': '접속사',
 'Determiner': '관형사',
 'Eomi': '어미',
 'Exclamation': '감탄사',
 'Foreign': '외국어, 한자 및 기타기호',
 'Hashtag': '트위터 해쉬태그',
 'Josa': '조사',
 'KoreanParticle': '(ex: ㅋㅋ)',
 'Noun': '명사',
 'Number': '숫자',
 'PreEomi': '선어말어미',
 'Punctuation': '구두점',
 'ScreenName': '트위터 아이디',
 'Suffix': '접미사',
 'Unknown': '미등록어',
 'Verb': '동사'}
```

In [23]:

```
tagsets = pd.DataFrame()
N = 67
tagsets["Hannanum-기호"] = list(hannanum.tagset.keys()) + list("*" * (N - len(hannanum.tagset)))
tagsets["Hannanum-품사"] = list(hannanum.tagset.values()) + list("*" * (N - len(hannanum.tagset)))
tagsets["Kkma-기호"] = list(kkma.tagset.keys()) + list("*" * (N - len(kkma.tagset)))
tagsets["Kkma-품사"] = list(kkma.tagset.values()) + list("*" * (N - len(kkma.tagset)))
tagsets["Komoran-기호"] = list(komoran.tagset.keys()) + list("*" * (N - len(komoran.tagset)))
tagsets["Komoran-품사"] = list(komoran.tagset.values()) + list("*" * (N - len(komoran.tagset)))
tagsets["Mecab-기호"] = list(mecab.tagset.keys()) + list("*" * (N - len(mecab.tagset)))
tagsets["Mecab-품사"] = list(mecab.tagset.values()) + list("*" * (N - len(mecab.tagset)))
tagsets["OKT-기호"] = list(okt.tagset.keys()) + list("*" * (N - len(okt.tagset)))
tagsets["OKT-품사"] = list(okt.tagset.values()) + list("*" * (N - len(okt.tagset)))
tagsets
```

Out[23]:

	Hannanum-기호	Hannanum-품사	Kkma-기호	Kkma-품사	Komoran-기호	Komoran-품사	Mecab-기호	Mecab-품사	OKT-기
0	E	어미	EC	연결 어미	EC	연결 어미	EC	연결 어미	Adjecti
1	EC	연결 어미	ECD	의존적 연결 어미	EF	종결 어미	EF	종결 어미	Adve
2	EF	종결 어미	ECE	대등 연결 어미	EP	선어말어미	EP	선어말어미	Alp
3	EP	선어말어미	ECS	보조적 연결 어미	ETM	관형형 전성 어미	ETM	관형형 전성 어미	Conjuncti
4	ET	전성 어미	EF	종결 어미	ETN	명사형 전성 어미	ETN	명사형 전성 어미	Determir
5	F	외국어	EFA	청유형 종결 어미	IC	감탄사	IC	감탄사	Eo
6	I	독립언	EFI	감탄형 종결 어미	JC	접속 조사	JC	접속 조사	Exclamati
7	II	감탄사	EFN	평서형 종결 어미	JKB	부사격 조사	JKB	부사격 조사	Forei
8	J	관계언	EFO	명령형 종결 어미	JKC	보격 조사	JKC	보격 조사	Hasht
9	JC	격조사	EFQ	의문형 종결 어미	JKG	관형격 조사	JKG	관형격 조사	Jo

	Hannanum-기호	Hannanum-품사	Kkma-기호	Kkma-품사	Komoran-기호	Komoran-품사	Mecab-기호	Mecab-품사	OKT-기
10	JP	서술격 조사	EFR	존칭형 종결 어미	JKO	목적격 조사	JKO	목적격 조사	KoreanParti
11	JX	보조사	EP	선어말 어미	JKQ	인용격 조사	JKQ	인용격 조사	No
12	M	수식언	EPH	존칭 선어말 어미	JKS	주격 조사	JKS	주격 조사	Numb
13	MA	부사	EPP	공손 선어말 어미	JKV	호격 조사	JKV	호격 조사	PreEo
14	MM	관형사	EPT	시제 선어말 어미	JX	보조사	JX	보조사	Punctuati
15	N	체언	ET	전성 어미	MAG	일반 부사	MAG	일반 부사	ScreenNar
16	NB	의존명사	ETD	관형형 전성 어미	MAJ	접속 부사	MAJ	접속 부사	Sut
17	NC	보통명사	ETN	명사형 전성 어미	MM	관형사	MM	관형사	Unkno
18	NN	수사	IC	감탄사	NA	분석불능 범주	NNB	의존 명사	Ve
19	NP	대명사	JC	접속 조사	NF	명사추정 범주	NNBC	단위를 나타내는 명사	
20	NQ	고유명사	JK	조사	NNB	의존 명사	NNG	일반 명사	
21	P	용언	JKC	보격 조사	NNG	일반 명사	NNP	고유 명사	
22	PA	형용사	JKG	관형격 조사	NNP	고유 명사	NP	대명사	
23	PV	동사	JKI	호격 조사	NP	대명사	NR	수사	
24	PX	보조 용언	JKM	부사격 조사	NR	수사	SC	구분자 , · / :	
25	S	기호	JKO	목적격 조사	NV	용언추정 범주	SE	줄임표 …	
26	X	접사	JKQ	인용격 조사	SE	줄임표	SF	마침표, 물음표, 느낌표	
27	XP	접두사	JKS	주격 조사	SF	마침표, 물음표, 느낌표	SH	한자	
28	XS	접미사	JX	보조사	SH	한자	SL	외국어	

	Hannanum-기호	Hannanum-품사	Kkma-기호	Kkma-품사	Komoran-기호	Komoran-품사	Mecab-기호	Mecab-품사	OKT-기
29	*	*	MA	부사	SL	외국어	SN	숫자	
...	...	...	...	...	...	...	...	...	
37	*	*	NNG	보통명사	VCP	긍정 지정사	VX	보조 용언	
38	*	*	NNM	단위의존명사	VV	동사	XPN	체언 접두사	
39	*	*	NNP	고유명사	VX	보조 용언	XR	어근	
40	*	*	NP	대명사	XPN	체언 접두사	XSA	형용사 파생 접미사	
41	*	*	NR	수사	XR	어근	XSN	명사파생 접미사	
42	*	*	OH	한자	XSA	형용사 파생 접미사	XSV	동사 파생 접미사	
43	*	*	OL	외국어	XSN	명사파생 접미사	*	*	
44	*	*	ON	숫자	XSV	동사 파생 접미사	*	*	
45	*	*	SE	줄임표	*	*	*	*	
46	*	*	SF	마침표, 물음표, 느낌표	*	*	*	*	
47	*	*	SO	붙임표 (물결, 숨김, 빠짐)	*	*	*	*	
48	*	*	SP	쉼표, 가운뎃점,콜론,빗금	*	*	*	*	
49	*	*	SS	따옴표,괄호표, 줄표	*	*	*	*	
50	*	*	SW	기타기호 (논리수학기호, 화폐기호)	*	*	*	*	
51	*	*	UN	명사추정범주	*	*	*	*	
52	*	*	VA	형용사	*	*	*	*	

	Hannanum-기호	Hannanum-품사	Kkma-기호	Kkma-품사	Komoran-기호	Komoran-품사	Mecab-기호	Mecab-품사	OKT-기
53	*	*	VC	지정사	*	*	*	*	
54	*	*	VCN	부정 지정사, 형용사 '아니다'	*	*	*	*	
55	*	*	VCP	긍정 지정사, 서술격 조사 '이다'	*	*	*	*	
56	*	*	VV	동사	*	*	*	*	
57	*	*	VX	보조 용언	*	*	*	*	
58	*	*	VXA	보조 형용사	*	*	*	*	
59	*	*	VXV	보조 동사	*	*	*	*	
60	*	*	XP	접두사	*	*	*	*	
61	*	*	XPN	체언 접두사	*	*	*	*	
62	*	*	XPV	용언 접두사	*	*	*	*	
63	*	*	XR	어근	*	*	*	*	
64	*	*	XSA	형용사 파생 접미사	*	*	*	*	
65	*	*	XSN	명사파생 접미사	*	*	*	*	
66	*	*	XSV	동사 파생 접미사	*	*	*	*	

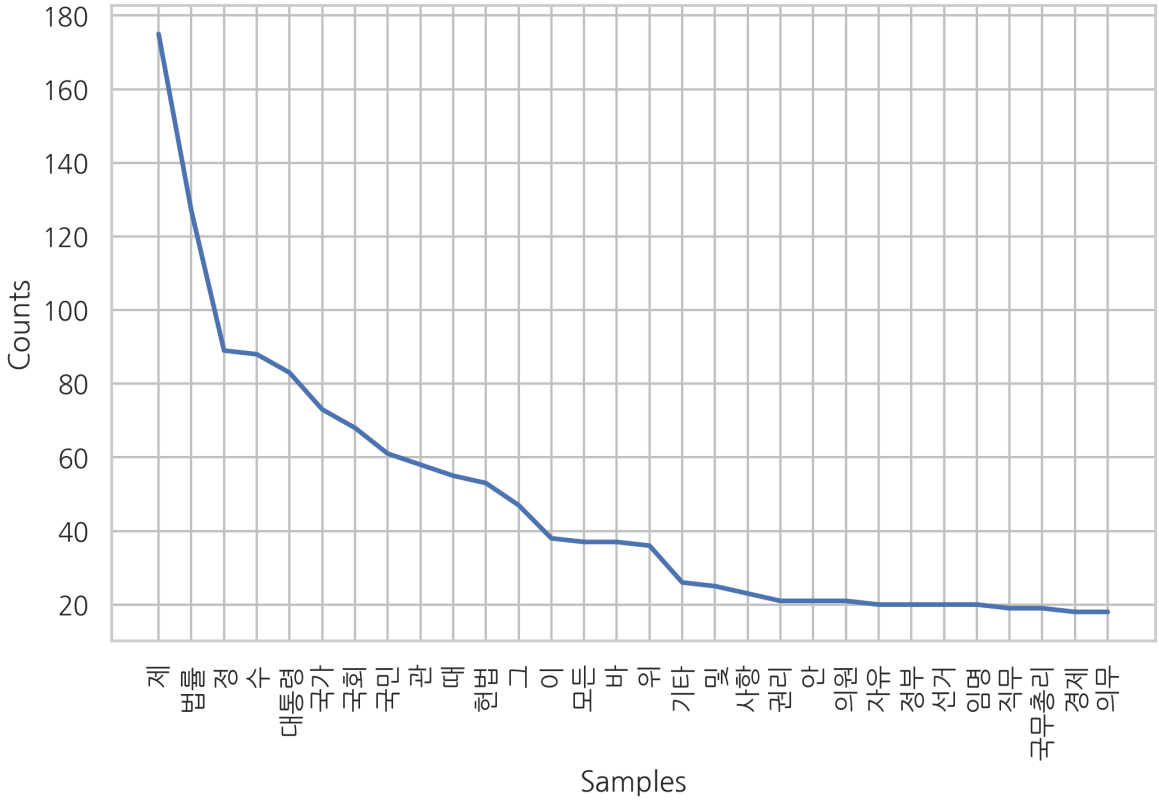
67 rows × 10 columns

koNLPy의 형태소 분석기와 NLTK의 Text 클래스를 결합하여 NLTK 기능을 사용할 수도 있다.

In [24]:

```
from nltk import Text

kolaw = Text(okt.nouns(c), name="kolaw")
kolaw.plot(30)
plt.show()
```



In [25]:

```
from wordcloud import WordCloud

# 자신의 컴퓨터 환경에 맞는 한글 폰트 경로를 설정
font_path = '/usr/share/fonts/truetype/nanum/NanumGothic.ttf'

wc = WordCloud(width = 1000, height = 600, background_color="white", font_path=font_path)
plt.imshow(wc.generate_from_frequencies(kolaw.vocab()))
plt.axis("off")
plt.show()
```