

Base Initialization

```
import spacy

nlp = spacy.blank("en")
nlp = spacy.load("en_core_web_sm")
nlp = spacy.load("en_core_web_md")
nlp = spacy.load("en_core_web_lg")
nlp = spacy.load("en_core_web_lg")

doc = nlp("Insert your text here. This is a sentence.")
```

```
print("tokens: ", [token.text for token in doc])

output:
tokens: ['Insert', 'your', 'text', 'here', '.', 'This', 'string', 'will', 'be', 'used', 'to', 'create', 'a', 'doc', 'object', 'for', 'natural', 'language', 'processing']
```

Token Attributes applied

Remove punctuations for tokens

```
words = [token.text for token in doc if token.is_stop != True and token.is_punct != True]

print(words)
```

```
output:
['Insert', 'text', 'string', 'create', 'doc', 'object', 'natural', 'language', 'processing']
```

<https://spacy.io/api/token>

Lemma

```
for token in doc:
    print("token: ", token, " lemma: ", token.lemma_)

token: token object within object
lemma: 'base' form of token
(ex. going --> go; was --> be)
```

displaCy Visualizer

```
import spacy
from spacy import displacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("This is a sentence.")
displacy.render(doc, style='dep', jupyter=True)
```

Sentence

```
list(doc.sents)
print(list(text.sents))

output:
[Insert your text here., This string will be used to create a doc object., For natural language processing.]
```

Word Frequency

```
from collections import Counter
tokens = [token.text for token in doc]
word_freq = Counter(tokens)
common_words = word_freq.most_common(5)
print(common_words)
```

```
output:
[('Insert', 1), ('text', 1), ('string', 1), ('create', 1), ('doc', 1), ('object', 1), ('natural', 1), ('language', 1), ('processing', 1)]

Using texttrank component ("custom")
```

```
nlp.add_pipe("texttrank")
doc = nlp("Insert your text here. This is a sentence.")

for index, token in enumerate(doc):
    print("index: ", index, " \ntext: ", token.text)
    print("rank:", token.rank, " count: ", token.count)
    print("chunks: ", token.chunks)
```

```
output:
index: 0
text: natural language processing
rank: 0.22805281953000428 count: 1
chunks: [natural language processing]
index: 1
text: a doc object
rank: 0.1481208925234081 count: 1
chunks: [a doc object]
index: 2
text: your text
rank: 0.073534568107334 count: 1
chunks: [your text]
index: 3
text: This string
rank: 0.054063724281900864 count: 1
chunks: [This string]
```

Textacy

```
pip install textacy

import textacy

metadata = {
    "title": "Natural language processing",
    "url": "https://en.wikipedia.org/wiki/Natural_language_processing",
    "source": "https://en.wikipedia.org/wiki/Natural_language_processing"
}
doc = textacy.make_spacy_doc(metadata)
print(doc._meta["title"])

output: Natural language processing

Textacy is a Python library for performing a variety of natural language processing (NLP) tasks, built on the highperformance spaCy library"

https://github.com/chartbeat-labs/textacy
https://textacy.readthedocs.io/en/latest/
```

latest version documentation:
This string will be used to create a
<https://buildmedia.readthedocs.org/media/pdf/textacy/stable/textacy.pdf>
https://textacy.readthedocs.io/en/0.12.0/api-reference/datasets_resources.html