

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 string will be used to create a doc object. For natural language processing.")
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

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

words = [token.text for token in doc if token.is_stop != True and token.is_punct != True]
word_freq = Counter(words)
common_words = word_freq.most_common(5)
print(common_words)
```

output:

```
[('Insert', 1), ('text', 1), ('string', 1), ('create', 1), ('doc', 1)]
```

Using textrank component ("custom")

```
nlp.add_pipe("textrank")
doc = nlp("Insert your text here. This string will be used to create a doc object. For natural language processing.")

for index, token in enumerate(doc._phrases):
    print("index: ", index, "\n\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
```

Textacy

```
pip install textacy
```

```
import textacy
```

```
metadata = {
    "title": "Natural language processing",
    "url": "https://en.wikipedia.org/wiki/Natural_language_processing",
    "source": "wikipedia",
}
doc = textacy.make_spacy_doc(text, metadata, lang="en_core_web_sm")
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:

<https://buildmedia.readthedocs.org/media/pdf/textacy/stable/textacy.pdf>
https://textacy.readthedocs.io/en/0.12.0/api_reference/datasets_resources.html

```
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)
```

```
rank: 0.054063724281900864 count: 1
chunks: [This string]
```



By **LoiVyen**
cheatography.com/loivyen/

Not published yet.
Last updated 2nd October, 2022.
Page 1 of 2.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>