import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("This is the first sentence. Next we see the second sentence.")

for sent in doc.sents:
    print(len(sent))

6
7

import re

p = re.compile('[Ww]oodchuck')
m = p.match('Woodchucks ran after a woodchuck.')

m

<re.Match object; span=(0, 9), match='Woodchuck'>

m.group()

'Woodchuck'

len('Woodchuck'), 'Woodchuck ran ...'[8]

(9, 'k')

m.span(), m.start()

((0, 9), 0)

m = p.match('Three Woodchucks ran after a woodchuck.')
print(m)

None
```python
m = p.search('Three Woodchucks ran after a woodchuck.')
m.group(), m.span(), 'Three Woodchucks'.find('Woodchuck')

('Woodchuck', (6, 15), 6)

matches = p.findall('Three Woodchucks ran after a woodchuck.')
matches

['Woodchuck', 'woodchuck']

matches = p.finditer('Three Woodchucks ran after a woodchuck.')
for m in matches:
    print(m.span())

(6, 15)
(29, 38)

p = re.compile('[Ww]oodchuck|[Gg]roundhog')
matches = p.findall('The woodchuck appears at the beginning in the movie Groundhog Day')
matches

['woodchuck', 'Groundhog']

import re
p = re.compile('[Ww]oodchuck | [Gg]roundhog')
matches = p.findall('The woodchucks appears at the beginning in the movie Groundhog Day')
matches

[' Groundhog']

p = re.compile('[Ww]oodchuck | [Gg]roundhog', re.VERBOSE)
matches = p.findall('The woodchucks appears at the beginning in the movie Groundhog Day')
matches

['woodchuck', 'Groundhog']

p = re.compile(r'[Ww]oodchuck | [Gg]roundhog', re.VERBOSE)
matches = p.findall('The woodchuck appears at the beginning in the movie Groundhog Day')
matches

['woodchuck ', 'Groundhog']
```
p = re.compile('[Ww]oodchucks?|[Gg]roundhogs?')
p.findall('Woodchucks, by any other name, such as groundhog, '
  'would woodchuck the same.')

['Woodchucks', 'groundhog', 'woodchuck']

p = re.compile('^How')
p.findall('How do you do? I do how I always do.')

['How']

p = re.compile('[Hh]ow')
p.findall('How do you do? I do how I always do.')

['How', 'how']

#p = re.compile('[^a-zA-Z][tT]he[^a-zA-Z]')
p = re.compile('[tT]he')
p.findall('The cat ran after the dog, but the other dog intervened.')

['The', 'the', 'the', 'the']

p = re.compile('[tT]he')
matches = p.finditer('The cat ran after the dog, '
  'but the other dog intervened.')
for m in matches:
  print(m)
print()
matches = p.finditer('The cat ran after the dog, '
  'but the other dog intervened.')
for m in matches:
  print(m.group(), m.start(), m.end())

<re.Match object; span=(0, 3), match='The'>
<re.Match object; span=(18, 21), match='the'>
<re.Match object; span=(31, 34), match='the'>
<re.Match object; span=(36, 39), match='the'>

The 0 3
the 18 21
the 31 34
the 36 39

p = re.compile('[^a-zA-Z][tT]he[^a-zA-Z]')

#p = re.compile('[tT]he')
\[
p.\text{findall}(\text{'The cat ran after the dog, '},
\text{  'but the other dog intervened.'})
\]

\[56\]: \[ \text{' the ', ' the '} \]

\[57\]: \[ s = \text{\textquoteleft The cat ran after the dog, but the other dog intervened.\textquoteright} \]

\[ \]
\[
p1 = \text{re.compile}(\text{'[\-a-zA-Z] [tT]he [\-a-zA-Z]'}, \text{re.VERBOSE})
\]
\[
r1 = p1.\text{findall}(s)
\]
\[
\text{print}(r1)
\]
\[
p2 = \text{re.compile}(\text{'~([tT]he) [\-a-zA-Z]'}, \text{re.VERBOSE})
\]
\[
r2 = p2.\text{findall}(s)
\]
\[
\text{print}(r2)
\]
\[
\text{# Instead of trying to combine the two patterns (but try it as a homework \_ \_ exercise).}
\]
\[
r3 = p1.\text{findall}(\text{' ' + s})
\]
\[
\text{print}(r3)
\]
\[
\]
\[58\]: \[ ['the', 'the'] \]
\[58\]: \[ ['The'] \]
\[58\]: \[ ['The', 'the', 'the'] \]

\[59\]: \[ \]
\[
\text{import re}
\]
\[
p = \text{re.compile}(\text{r'[a+b+]')}
\]
\[
p.\text{findall}(\text{\textquoteleft aabb aaabbb abcba aba aaaabb\textquoteright})
\]

\[58\]: \[ ['aabb', 'aaabbb', 'ab', 'ab', 'aaaabb'] \]

\[59\]: \[ \]
\[
\text{import re}
\]
\[
p = \text{re.compile}(\text{r'[Pythons?')}
\]
\[
\text{matches} = p.\text{findall}(\text{\textquoteleft Python is a fun programming language. '},
\text{  'There are many pythons in the jungle. '},
\text{  'I like PYTHON!\textquoteright})
\]
\[
\text{print}(\text{matches})
\]
\[
\]
\[59\]: \[ ['Python', 'pythons'] \]

\[60\]: \[ \]
\[
p = \text{re.compile}(\text{r'\s(cats?|dogs?)\W'})
\]
\[
\text{matches} = p.\text{findall}(\text{\textquoteleft It is raining cats and dogs. '},
\text{  'Her cat likes catfish.'})
\]
\[
\text{print}(\text{matches})
\]
\[
\]
\[60\]: \[ ['cats', 'dogs', 'cat'] \]
Parantheses for capturing patterns.

```python
[31]: p = re.compile('colou?r')
p.sub('color', 'I would like to drive a blue coloured car. ')

[31]: 'I would like to drive a blue <color>ed car.'
```

```python
[62]: p = re.compile('([0-9]+)', re.VERBOSE)
p.sub(r'<\1> extra', 'the 35 boxes')

[62]: 'the <35> extra boxes'
```

```python
[33]: p = re.compile('([0-9]+)', re.VERBOSE)
p.sub(r'<\1> extra', '10 whiskey bottles and 35 boxes of gold')

[33]: '<10> extra whiskey bottles and <35> extra boxes of gold'
```

```python
[64]: p = re.compile(r'Isaac (?![A-Zimov])')
matches = p.finditer('I like reading Isaac Asimov '
                      'and listening to Isaac Perlman '
                      'and playing chess with Isaac .')
for m in matches:
    print(m.span(), m.group())

    (45, 51) Isaac
    (82, 88) Isaac
```

```python
[65]: p = re.compile(r'Isaac (?![A-Zimov]Perlman)')
matches = p.finditer('I like reading Isaac Asimov '
                      'and listening to Isaac Perlman '
                      'and playing chess with Isaac .')
for m in matches:
    print(m.span(), m.group())

    (82, 88) Isaac

[ ]:
```