

```
In [1]: import spacy
```

sprawdź wersję

```
In [2]: spacy.__version__
```

```
Out[2]: '2.3.2'
```

zaimportuj obiekt umożliwiający przetwarzanie języka polskiego

```
In [3]: from spacy.lang.pl import Polish
```

```
In [4]: nlp = Polish()
```

obiekt Doc

```
In [5]: doc = nlp("Adam kupił mieszkanie za 100000 $.")
```

```
In [6]: type(doc)
```

```
Out[6]: spacy.tokens.doc.Doc
```

obiekt Token - wyszukiwanie po indeksie

```
In [7]: token = doc[0]; token.text
```

```
Out[7]: 'Adam'
```

sprawdź indeksy

```
In [8]: "Index: ", [token.i for token in doc]
```

```
Out[8]: ('Index: ', [0, 1, 2, 3, 4, 5, 6])
```

wypisz wszystkie tokeny

```
In [9]: "Text: ", [token.text for token in doc]
```

```
Out[9]: ('Text: ', ['Adam', 'kupił', 'mieszkanie', 'za', '100000', '$', '.'])
```

sprawdź czy token jest słowem

```
In [10]: "is_alpha: ", [token.is_alpha for token in doc]
```

```
Out[10]: ('is_alpha: ', [True, True, True, True, False, False, False])
```

sprawdź czy token jest znakiem interpunkcyjnym

```
In [11]: "is_punct: ", [token.is_punct for token in doc]
```

```
Out[11]: ('is_punct: ', [False, False, False, False, False, False, True])
```

sprawdź czy token jest liczbą

```
In [12]: "like_num: ", [token.like_num for token in doc]
```

```
Out[12]: ('like_num: ', [False, False, False, False, True, False, False])
```

znajdź w tekście wybrany atrybut

```
In [14]: # Iteruje po wszystkich tokenach
```

```
for token in doc:
    # jeżeli token będzie: liczbą
    if token.like_num:
        # next_token zostanie utworzony na podstawie wyszukiwania po indeksie
        next_token = doc[token.i + 1]
        # jeżeli next_token będzie: $, to go wypisz
        if next_token.text == "$":
            print("kwota w dolarach:", token.text)
```

kwota w dolarach: 1000000

## obiekt Span

In [15]: `span = doc[0:2]; span.text`

Out[15]: 'Adam kupił'