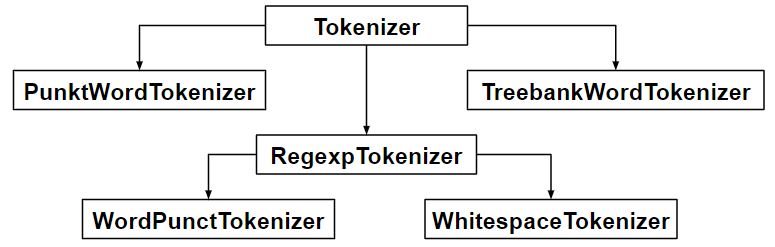
**Natural Language Processing (NLP)**

**Natural Language Processing (NLP)**is a subfield of computer science, artificial intelligence, information engineering, and human-computer interaction. This field focuses on how to program computers to process and analyze large amounts of natural language data. It is difficult to perform as the process of reading and understanding languages is far more complex than it seems at first glance.

**Tokenization**is the process of tokenizing or splitting a string, text into a list of tokens. One can think of token as parts like a word is a token in a sentence, and a sentence is a token in a paragraph.

**Key points of the article –**

* Text into sentences tokenization
* Sentences into words tokenization
* Sentences using regular expressions tokenization



**Code #1: Sentence Tokenization –**Splitting sentences in the paragraph

|  |
| --- |
| from nltk.tokenize import sent\_tokenize    text = "Hello everyone. Welcome to GeeksforGeeks. You are studying NLP article"  sent\_tokenize(text) |

**Output :**

['Hello everyone.',

'Welcome to GeeksforGeeks.',

'You are studying NLP article']

**How sent\_tokenize works ?**  
The sent\_tokenize function uses an instance of PunktSentenceTokenizer from the nltk.tokenize.punkt module, which is already been trained and thus very well knows to mark the end and beginning of sentence at what characters and punctuation.  
   
**Code #2: PunktSentenceTokenizer –**When we have huge chunks of data then it is efficient to use it.

|  |
| --- |
| import nltk.data    # Loading PunktSentenceTokenizer using English pickle file  tokenizer = nltk.data.load('tokenizers/punkt/PY3/english.pickle')    tokenizer.tokenize(text) |

**Output :**

['Hello everyone.',

'Welcome to GeeksforGeeks.',

'You are studying NLP article']

**Code #3: Tokenize sentence of different language –**One can also tokenize sentence from different languages using different pickle file other than English.

|  |
| --- |
| import nltk.data    spanish\_tokenizer = nltk.data.load('tokenizers/punkt/PY3/spanish.pickle')    text = 'Hola amigo. Estoy bien.'  spanish\_tokenizer.tokenize(text) |

**Output :**

['Hola amigo.',

'Estoy bien.']

**Code #4: Word Tokenization –**Splitting words in a sentence.

|  |
| --- |
| from nltk.tokenize import word\_tokenize    text = "Hello everyone. Welcome to GeeksforGeeks."  word\_tokenize(text) |

**Output :**

['Hello', 'everyone', '.', 'Welcome', 'to', 'GeeksforGeeks', '.']

**How *word\_tokenize*works?**  
word\_tokenize() function is a wrapper function that calls tokenize() on an instance of the TreebankWordTokenizer class.  
   
**Code #5: Using TreebankWordTokenizer**

|  |
| --- |
| from nltk.tokenize import TreebankWordTokenizer    tokenizer = TreebankWordTokenizer()  tokenizer.tokenize(text) |

**Output :**

['Hello', 'everyone.', 'Welcome', 'to', 'GeeksforGeeks', '.']

These tokenizers work by separating the words using punctuation and spaces. And as mentioned in the code outputs above, it does not discard the punctuation, allowing a user to decide what to do with the punctuations at the time of pre-processing.  
   
**Code #6: PunktWordTokenizer –**It doen’t seperates the punctuation from the words.

|  |
| --- |
| from nltk.tokenize import PunktWordTokenizer    tokenizer = PunktWordTokenizer()  tokenizer.tokenize("Let's see how it's working.") |

**Output :**

['Let', "'s", 'see', 'how', 'it', "'s", 'working', '.']

**Code #6: WordPunctTokenizer –**It seperates the punctuation from the words.

|  |
| --- |
| from nltk.tokenize import WordPunctTokenizer    tokenizer = WordPunctTokenizer()  tokenizer.tokenize("Let's see how it's working.") |

**Output :**

['Let', "'", 's', 'see', 'how', 'it', "'", 's', 'working', '.']

**Code #7: Using Regular Expression**

|  |
| --- |
| from nltk.tokenize import RegexpTokenizer    tokenizer = RegexpTokenizer("[\w']+")  text = "Let's see how it's working."  tokenizer.tokenize(text) |

**Output :**

["Let's", 'see', 'how', "it's", 'working']

**Code #7: Using Regular Expression**

filter\_none

brightness\_4

|  |
| --- |
| from nltk.tokenize import regexp\_tokenize    text = "Let's see how it's working."  regexp\_tokenize(text, "[\w']+") |

**Output :**

["Let's", 'see', 'how', "it's", 'working']