

▼ Tokenizer

```
import tensorflow as tf
from tensorflow import keras
from keras.preprocessing.text import Tokenizer
```

```
sentences = [
    "I love my dog",
    "I love my cat",
    "Do you think my dog is amazing?"
]
```

```
tokenizer = Tokenizer(num_words = 100, oov_token = "<OOV>") # Out of Vocab
tokenizer.fit_on_texts(sentences)
word_index = tokenizer.word_index
print(word_index)
```

```
{' <00V>': 1, 'my': 2, 'i': 3, 'love': 4, 'dog': 5, 'cat': 6, 'do': 7, 'you': 8, 'think': 9}
```



▼ Padding and Truncating


```
from tensorflow.keras.preprocessing.sequence import pad_sequences
sequences = tokenizer.texts_to_sequences(sentences)
```

```
padded = pad_sequences(sequences, padding="post", truncating="post", maxlen=5)
padded
```

```
array([[3, 4, 2, 5, 0],
       [3, 4, 2, 6, 0],
       [7, 8, 9, 2, 5]], dtype=int32)
```

```
test_data = [
    "I really love my dog",
    "my dog loves my manatee"
]
```

```
test_seq = tokenizer.texts_to_sequences(test_data)
test_seq
```

 `[[3, 1, 4, 2, 5], [2, 5, 1, 2, 1]]`

```
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

```
nadded1 = nadd_sequences(sequences)
```

```

padded1 = pad_sequences(sequences,
padded2 = pad_sequences(sequences, padding = "post")
padded3 = pad_sequences(sequences, padding = "post", maxlen = 5)
padded4 = pad_sequences(sequences, padding = "post", truncating = "post", maxlen = 5)

print("First Padded :\n",padded1)
print("Second Padded :\n",padded2)
print("Third Padded :\n",padded3)
print("Fourth Padded :\n",padded4)

```

```

First Padded :
[[ 0  0  0  3  4  2  5]
 [ 0  0  0  3  4  2  6]
 [ 7  8  9  2  5 10 11]]
Second Padded :
[[ 3  4  2  5  0  0  0]
 [ 3  4  2  6  0  0  0]
 [ 7  8  9  2  5 10 11]]
Third Padded :
[[ 3  4  2  5  0]
 [ 3  4  2  6  0]
 [ 9  2  5 10 11]]
Fourth Padded :
[[3 4 2 5 0]
 [3 4 2 6 0]
 [7 8 9 2 5]]

```