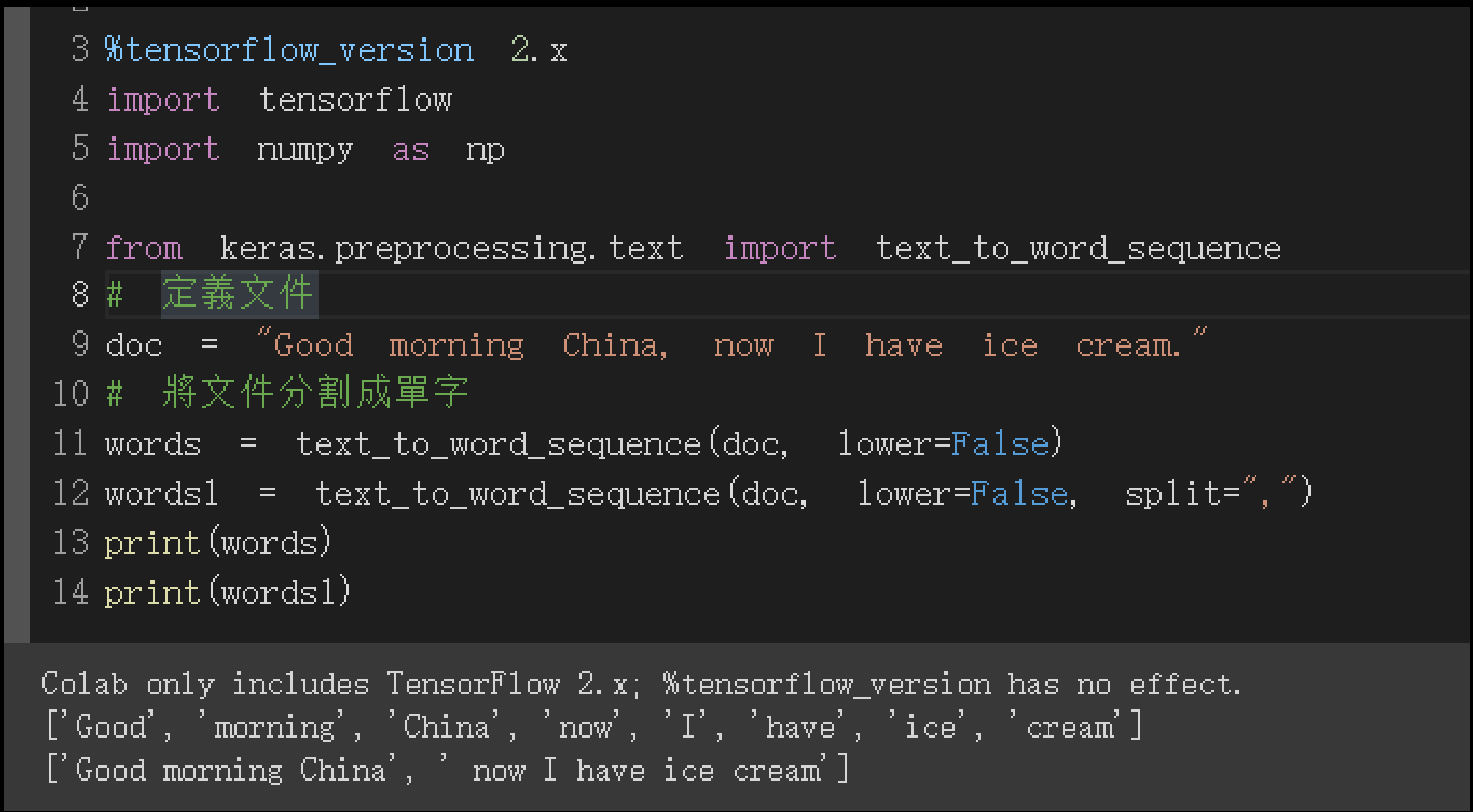
p.13-2~13-9

1.文字資料預處理
使用keras.preprocessing.text模組

1-1.斷詞：
text_to_word_sequence(text參數, filters參數, lower參數, split參數)

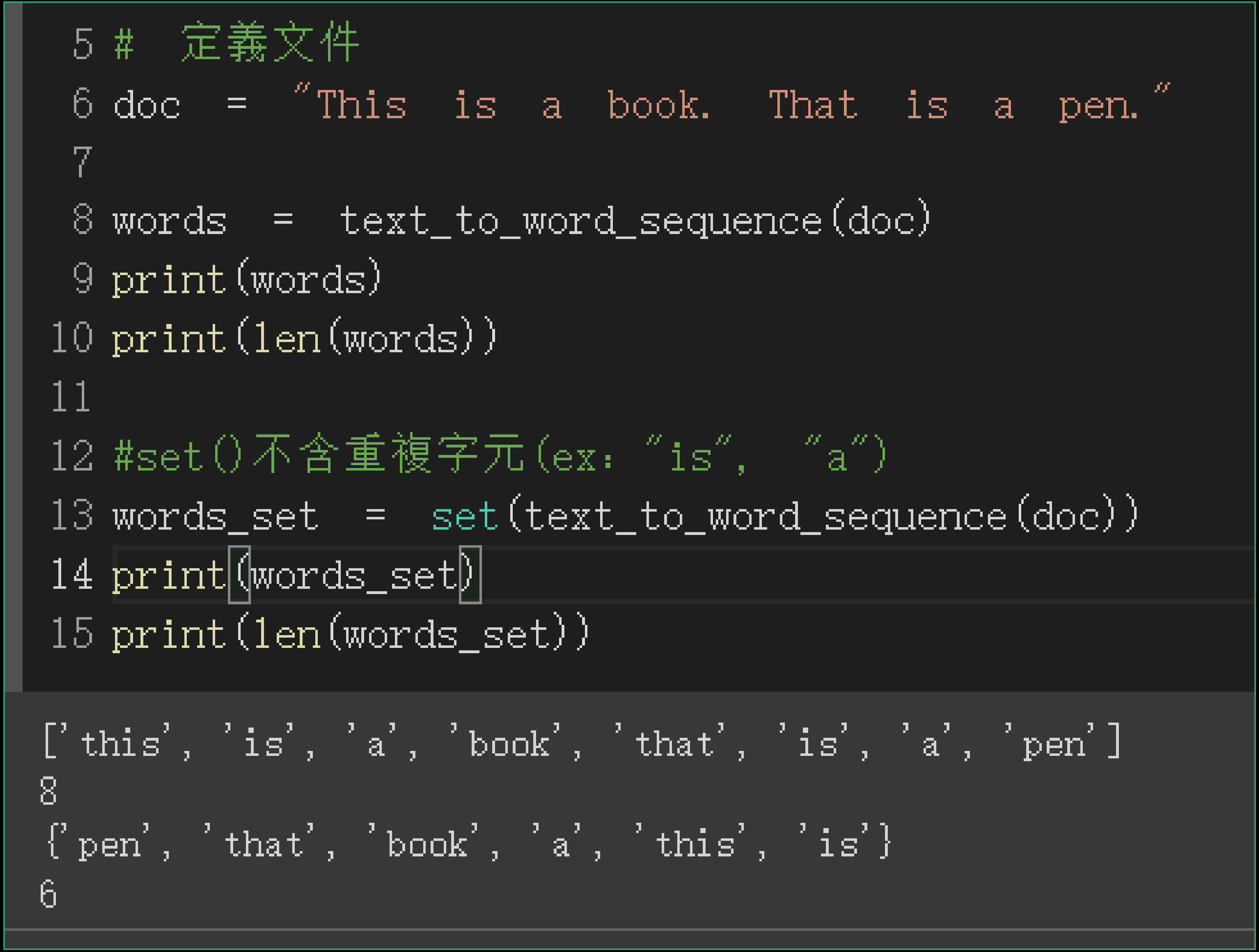
- filters參數：過濾字元，預設值!"# $ % & ( ) * + - , . / ; :<=>?@ [ ]/^_{ }|~等符號
- lower參數：預設True(轉小寫)，False(不轉)
- split參數：指定分割的字串，預設空白值(看到空白就做分割)



```python
%tensorflow_version 2.x
import tensorflow
import numpy as np

from keras.preprocessing.text import text_to_word_sequence
# 定義文件
doc = "Good morning China, now I have ice cream."
# 將文件分割成單字
words = text_to_word_sequence(doc, lower=False)
words1 = text_to_word_sequence(doc, lower=False, split=",")
print(words)
print(words1)
```

```
Colab only includes TensorFlow 2.x; %tensorflow_version has no effect.
['Good', 'morning', 'China', 'now', 'I', 'have', 'ice', 'cream']
['Good morning China', ' now I have ice cream']
```



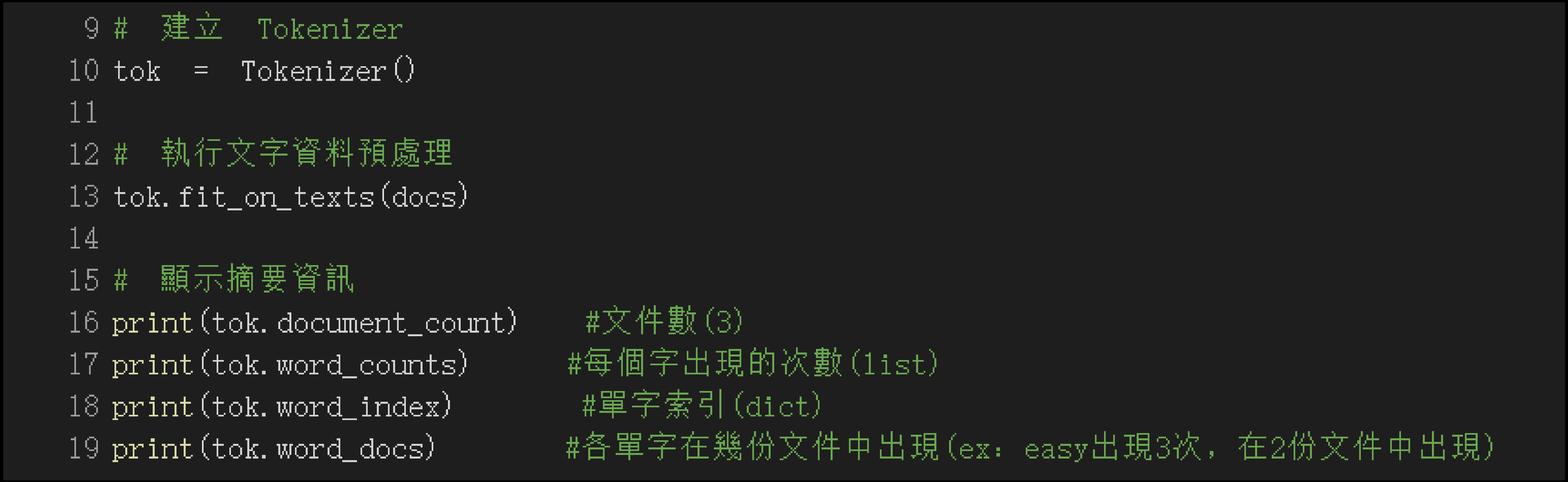
```python
#  定義文件
doc = "This is a book. That is a pen."

words = text_to_word_sequence(doc)
print(words)
print(len(words))

#set()不含重複字元(ex: "is", "a")
words_set = set(text_to_word_sequence(doc))
print(words_set)
print(len(words_set))
```

```
['this', 'is', 'a', 'book', 'that', 'is', 'a', 'pen']
8
{'pen', 'that', 'book', 'a', 'this', 'is'}
6
```

1-2.斷詞(處理大量資料)：
Tokenizer

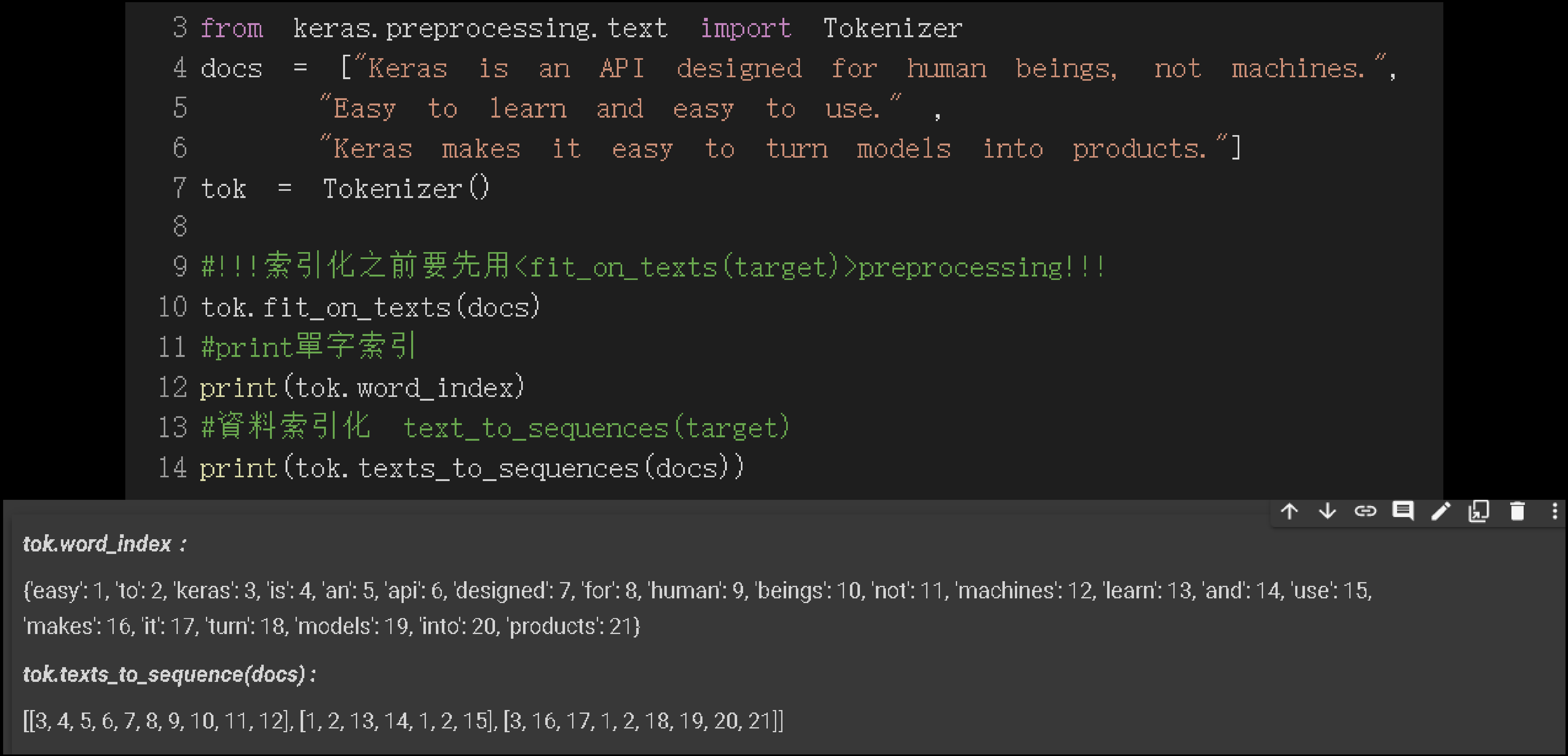


```
#  建立  Tokenizer
tok  =  Tokenizer()

#  執行文字資料預處理
tok.fit_on_texts(docs)

#  顯示摘要資訊
print(tok.document_count)    #文件數(3)
print(tok.word_counts)      #每個字出現的次數(list)
print(tok.word_index)        #單字索引(dict)
print(tok.word_docs)        #各單字在幾份文件中出現(ex： easy出現3次， 在2份文件中出現)
```

1-3. 文字資料索引化：



```
from keras.preprocessing.text import Tokenizer
docs = ["Keras is an API designed for human beings, not machines.",
        "Easy to learn and easy to use." ,
        "Keras makes it easy to turn models into products."]
tok = Tokenizer()

#!!!索引化之前要先用<fit_on_texts(target)>preprocessing!!!
tok.fit_on_texts(docs)
#print單字索引
print(tok.word_index)
#資料索引化  text_to_sequences(target)
print(tok.texts_to_sequences(docs))
```

tok.word_index :

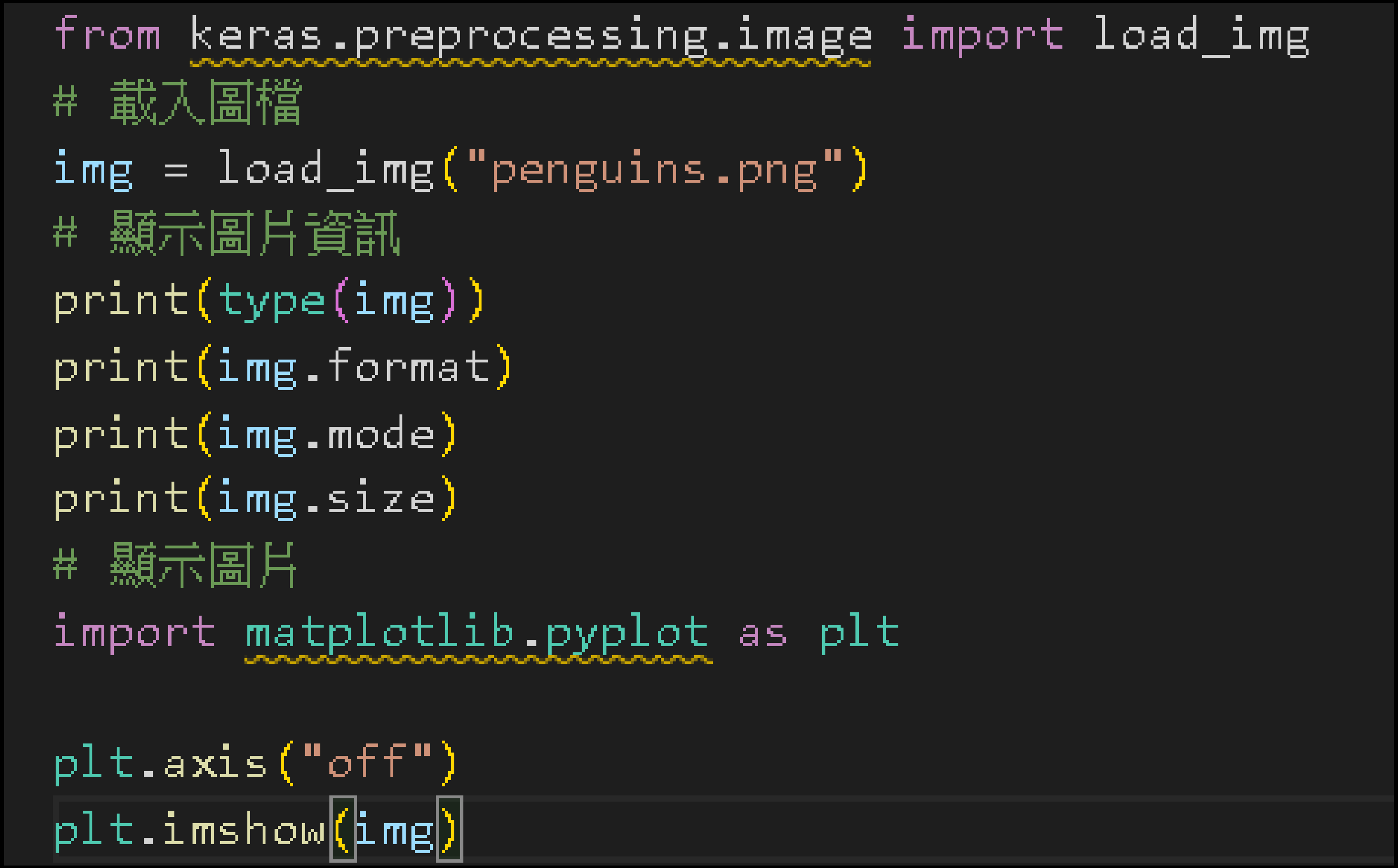
```
{'easy': 1, 'to': 2, 'keras': 3, 'is': 4, 'an': 5, 'api': 6, 'designed': 7, 'for': 8, 'human': 9, 'beings': 10, 'not': 11, 'machines': 12, 'learn': 13, 'and': 14, 'use': 15,
'makes': 16, 'it': 17, 'turn': 18, 'models': 19, 'into': 20, 'products': 21}
```

tok.texts_to_sequence(docs) :

```
[[3, 4, 5, 6, 7, 8, 9, 10, 11, 12], [1, 2, 13, 14, 1, 2, 15], [3, 16, 17, 1, 2, 18, 19, 20, 21]]
```

2.圖片資料預處理
使用keras.preprocessing.image模組

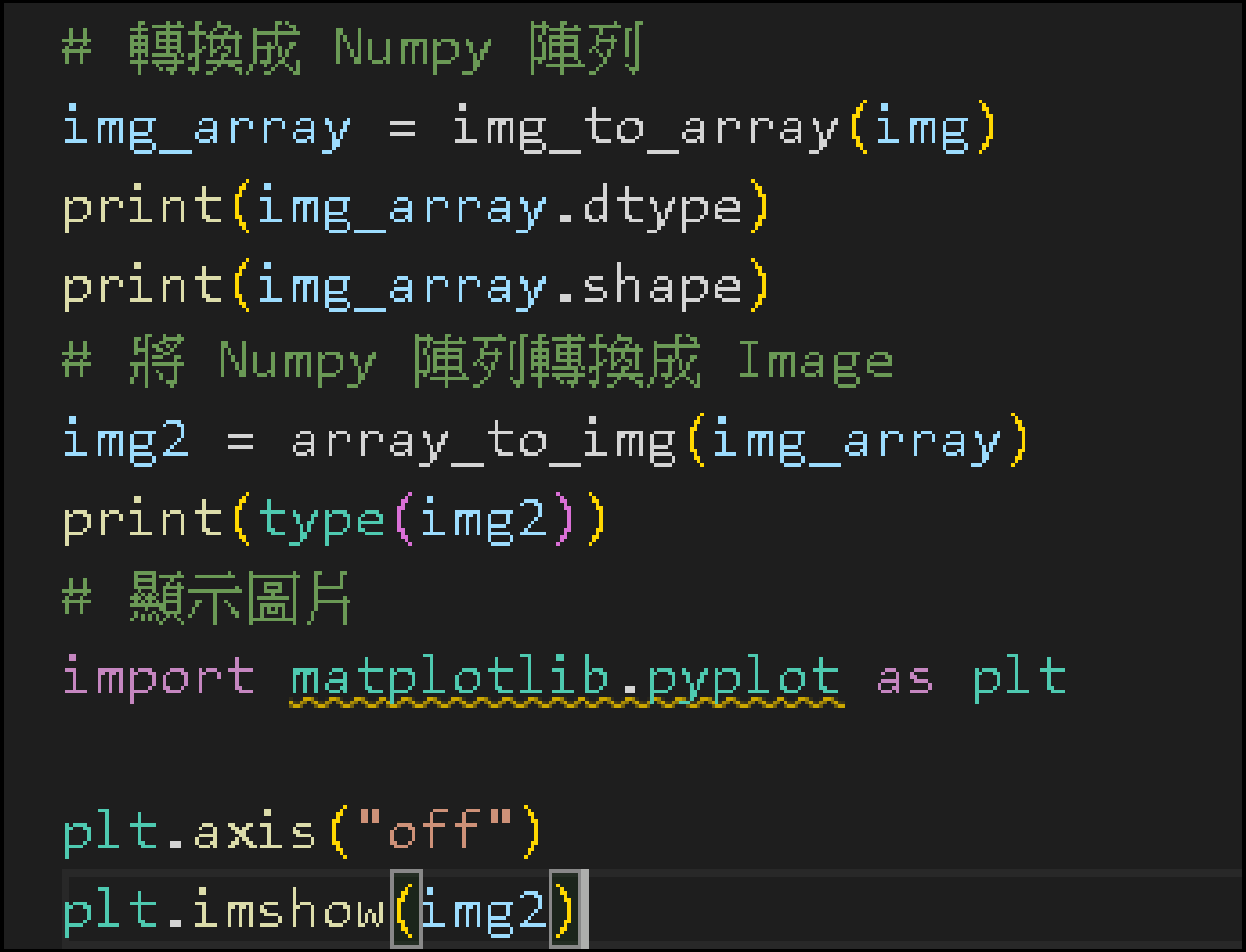
載入&顯示



```
from keras.preprocessing.image import load_img
# 載入圖檔
img = load_img("penguins.png")
# 顯示圖片資訊
print(type(img))
print(img.format)
print(img.mode)
print(img.size)
# 顯示圖片
import matplotlib.pyplot as plt

plt.axis("off")
plt.imshow(img)
```

圖片和NumPy陣列互相轉換



```
# 轉換成 Numpy 陣列
img_array = img_to_array(img)
print(img_array.dtype)
print(img_array.shape)
# 將 Numpy 陣列轉換成 Image
img2 = array_to_img(img_array)
print(type(img2))
# 顯示圖片
import matplotlib.pyplot as plt

plt.axis("off")
plt.imshow(img2)
```

3. 資料增強(Keras圖片增強)

目的：訓練資料不足，用圖片增強技術增加資料量

方法：用現有圖片，將其裁剪、旋轉、翻轉、縮放做變形，創造更多的圖片訓練資料

3. 資料增強(Keras圖片增強)

ImageDataGenerator物件的**參數**可以指定使用什麼操作處理影像

- 隨機旋轉

```
ImageDataGenerator(rotation_range=40)
```

- 隨機位移

```
ImageDataGenerator(width_shift_range=0.2,
                   height_shift_range=0.2)
```

- 隨機推移變換(垂直軸不動的推移)

```
ImageDataGenerator(shear_range=15,
                   fill_mode="constant")
#fill_mode=填滿方式
```

- 隨機縮放

```
ImageDataGenerator(zoom_range=0.2)
```

- 隨機翻轉

```
ImageDataGenerator(horizontal_flip=True,
                   vertical_flip=True)
```