D01 Python 文字處理函數介紹

簡報閱讀 範例與作業 問題討論

重要知識點
字串長度
取出特定區間文字
合併字串
常用特殊字符 (escape c...

重要知識點

- 了解如何使用 Python 做基礎的文字處理(String)
- 本章節為實作課程，需搭配程式碼練習

今天會介紹的Functions

- 計算字串長度
- 取出字串內特定區間字符
- 合併字串
- 特殊字符運用
- 判斷字符存在與否、打小寫、數字
- 移除、取代字符
- 在字串中尋找指定字符

字串長度

如果想要知道字串的長度，我們可以直接使用內建的 len( ) function 來獲得答案

計算字串長度

```
In [3]: pratice_sentence = all_review[0]
In [4]: ## format 用法之後會有詳細講解，這裡先了解會將後方字串放入{}即可
        print('原始字串: {}'.format(pratice_sentence))
        ## 運用len()得到字串長度
        print('字串長度: {}'.format(len(pratice_sentence)))
原始字串: Wow... Loved this place.
字串長度: 24
```

```
pratice_sentence = all_review[0]
## format 用法之後會有詳細講解，這裡先了解會將後方字串放入{}即可
print('原始字串: {}'.format(pratice_sentence))
## 運用len()得到字串長度
print('字串長度: {}'.format(len(pratice_sentence)))
```

取出特定區間文字

String 如同 list 可透過 [:] 來擷取特定區間

取出特定字符

```
In [5]: print('取出前五個字: {}'.format(pratice_sentence[:5]))
        print('取出後五個字: {}'.format(pratice_sentence[-5:]))
        print('取出後中間五個字: {}'.format(pratice_sentence[5:10]))
取出前五個字: Wow..
取出後五個字: lace.
取出後中間五個字: . Lov
```

```
print('取出前五個字: {}'.format(pratice_sentence[:5]))
print('取出後五個字: {}'.format(pratice_sentence[-5:]))
print('取出後中間五個字: {}'.format(pratice_sentence[5:10]))
```

合併字串

合併字串

```
In [6]: print('原始字串1: {}'.format(all_review[0]))
        print('原始字串2: {}'.format(all_review[1]))
原始字串1: Wow... Loved this place.
原始字串2: Crust is not good.
In [7]: ##運用+符號可以直接合併兩字串
        all_review[0]+all_review[1]
Out[7]: 'Wow... Loved this place.Crust is not good.'
In [8]: ## 運用join function
        ''.join((all_review[0],all_review[1]))
Out[8]: 'Wow... Loved this place.Crust is not good.'
In [9]: ## 加入separator
        '/'.join((all_review[0],all_review[1]),)
Out[9]: 'Wow... Loved this place./Crust is not good.'
```

```
print('原始字串1: {}'.format(all_review[0]))
print('原始字串2: {}'.format(all_review[1]))
##運用+符號可以直接合併兩字串
all_review[0]+all_review[1]
## 運用join function
''.join((all_review[0],all_review[1]))
## 加入separator
'/'.join((all_review[0],all_review[1]),)
```

常用特殊字符 (escape characters)

'\t'：8個空白格 (tab)
'\n'：換行

其他如：

Code	Result
\'	Single Quote
\\	Backslash
\n	New Line
\r	Carriage Return
\t	Tab
\b	Backspace
\f	Form Feed
\ooo	Octal value
\xhh	Hex value

圖表來源：w3schools.com

檢查字元是否在字串內

當我們想判定字元是否存在於字串內：

檢查字元是否在字串內

```
In [12]: pratice_sentence = all_review[1]
         print('原始字串:  {}'.format(pratice_sentence))
原始字串:  Crust is not good.
In [13]: 'is' in pratice_sentence
Out[13]: True
In [14]: 'I' in pratice_sentence
Out[14]: False
In [15]: 'I' not in pratice_sentence
Out[15]: True
```

```
pratice_sentence = all_review[1]
print('原始字串: {}'.format(pratice_sentence))
'is' in pratice_sentence
'I' in pratice_sentence
'I' not in pratice_sentence
```

Function: strip( )

strip() 用來移除頭尾的字元

strip()

```
In [16]: pratice_sentence = all_review[1]
In [17]: print('原始字串:  {}'.format(pratice_sentence))
原始字串:  Crust is not good.
In [18]: ## 移除開頭
         pratice_sentence.strip('Crust')
Out[18]: ' is not good.'
In [19]: ## 移除尾部
         pratice_sentence.strip('good.')
Out[19]: 'Crust is not '
In [20]: ## 由於is並不是開頭或結尾字符，因此返回原字串
         pratice_sentence.strip('is')
Out[20]: 'Crust is not good.'
In [21]: ##移除開頭空格
         ' Crust is not good.'.strip()
Out[21]: 'Crust is not good.'
```

由於 is 不是開頭或結尾字元，所以並不會有任何變動

當我們什麼都沒指定時，strip 會移除頭尾空格

```
pratice_sentence = all_review[1]
print('原始字串: {}'.format(pratice_sentence))
## 移除開頭
pratice_sentence.strip('Crust')
## 移除尾部
pratice_sentence.strip('good.')
## 由於is並不是開頭或結尾字符，因此返回原字串
pratice_sentence.strip('is')
##移除開頭空格
' Crust is not good.'.strip()
```

Function: replace( )

replace( ) 用來替換字元

replace()

```
In [22]: pratice_sentence = all_review[2]
         print('原始字串:   {}'.format(pratice_sentence))
原始字串:   Not tasty and the texture was just nasty.
In [23]: ## 用disgusting取代nasty
         pratice_sentence.replace('nasty','disgusting')
Out[23]: 'Not tasty and the texture was just disgusting.'
In [24]: ## 最多取代兩次
         print(pratice_sentence.replace('t','T'))
         print(pratice_sentence.replace('t','T',2))
NoT TasTy and The TexTure was jusT nasTy.
NoT Tasty and the texture was just nasty.
```

```
pratice_sentence = all_review[2]
print('原始字串:   {}'.format(pratice_sentence))
## 用disgusting取代nasty
pratice_sentence.replace('nasty','disgusting')
## 最多取代兩次
print(pratice_sentence.replace('t','T'))
print(pratice_sentence.replace('t','T',2))
```

Function: split()

split( ) 用來切開字串

split()

```
In [25]: pratice_sentence = all_review[3]
         print('原始字串:   {}'.format(pratice_sentence))
原始字串:   Stopped by during the late May bank holiday off Rick Steve recommendation and loved it.
In [26]: ##用空格當分界隔開字串
         pratice_sentence.split(' ')
Out[26]: ['Stopped',
          'by',
          'during',
          'the',
          'late',
          'May',
          'bank',
          'holiday',
          'off',
          'Rick',
          'Steve',
          'recommendation',
          'and',
          'loved',
          'it.']
```

```
pratice_sentence = all_review[3]
print('原始字串:   {}'.format(pratice_sentence))
##用空格當分界隔開字串
pratice_sentence.split(' ')
```

Function: count( )

count() 用來計算字串內字元出現次數

count()

```
In [28]: pratice_sentence = all_review[4]
         print('原始字串:   {}'.format(pratice_sentence))
原始字串:   The selection on the menu was great and so were the prices.
In [29]: ## 計算a出現次數
         pratice_sentence.count('a') ## == pratice_sentence.count('a',0,len(pratice_sentence))
Out[29]: 3
In [30]: ##規定計算區間
         pratice_sentence.count('a',2,10)
Out[30]: 0
```

```
pratice_sentence = all_review[4]
print('原始字串: {}'.format(pratice_sentence))
## 計算a出現次數
pratice_sentence.count('a') ## == pratice_sentence.count('a',0,len(pratice_sentence))
##規定計算區間
pratice_sentence.count('a',2,10)
```

Function: startswith() / endswith()

startswith() / endswith() 用來判定字串頭尾是否為該字元

startswith() / endswith()

```
In [31]: pratice_sentence = all_review[5]
         print('原始字串:   {}'.format(pratice_sentence))
原始字串:   Now I am getting angry and I want my damn pho.
In [32]: ##檢查字串開頭是否為Now
         pratice_sentence.startswith('Now')
Out[32]: True
In [33]: ##檢查字串結尾是否為.
         pratice_sentence.endswith('.')
Out[33]: True
In [34]: pratice_sentence.endswith('I')
Out[34]: False
```

```
pratice_sentence = all_review[5]
print('原始字串:  {}'.format(pratice_sentence))
##檢查字串開頭是否為Now
pratice_sentence.startswith('Now')
##檢查字串結尾是否為.
pratice_sentence.endswith('.')
pratice_sentence.endswith('I')
```

Function: capitalize

capitalize() 會將字串開頭轉為大寫

capitalize()

```
In [35]: ##將開頭轉換為大寫
         'we love python'.capitalize()
Out[35]: 'We love python'
```

```
##將開頭轉換為大寫
'we love python'.capitalize()
```

Function: find() / index()

find() / index() 用來尋找字串中字元所在位置，index()不同於find()，當字元不存在時，會報錯，find()則會返回-1

find() / index()

```
In [36]: pratice_sentence = all_review[6]
         print('原始字串:   {}'.format(pratice_sentence))
原始字串:   Honeslty it didn't taste THAT fresh.)
In [37]: ## 尋找字串中 it 在哪
         pratice_sentence.find('it') ## == pratice_sentence.index('it')
Out[37]: 9
In [38]: pratice_sentence[9:9+len('it')]
Out[38]: 'it'
In [39]: ## index與find功能相同，但找不到字元時會報錯，find會回報-1
         pratice_sentence.index('where')
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-39-9ef5681f06c7> in <module>
      1 ## index與find功能相同，但找不到字元時會報錯，find會回報-1
----> 2 pratice_sentence.index('where')

ValueError: substring not found
In [40]: pratice_sentence.find('where')
Out[40]: -1
```

```
pratice_sentence = all_review[6]
print('原始字串:  {}'.format(pratice_sentence))
## 尋找字串中 it 在哪
pratice_sentence.find('it') ## == pratice_sentence.index('it')
pratice_sentence[9:9+len('it')]
## index與find功能，但找不到字元時會報錯，find會回報-1
pratice_sentence.index('where')
pratice_sentence.find('where')
```

Function: upper() / lower()

upper() / lower() 會分別將整串字串轉換為大/小寫

upper() / lower()

```
In [41]: pratice_sentence = all_review[7]
         print('原始字串:   {}'.format(pratice_sentence))
原始字串:   The potatoes were like rubber and you could tell they had been made up ahead of time being kept under a wa
rmer.
In [42]: ##全部轉換為大寫
         pratice_sentence.upper()
Out[42]: 'THE POTATOES WERE LIKE RUBBER AND YOU COULD TELL THEY HAD BEEN MADE UP AHEAD OF TIME BEING KEPT UNDER A WARMER.'
In [43]: ##全部轉換為小寫
         pratice_sentence.lower()
Out[43]: 'the potatoes were like rubber and you could tell they had been made up ahead of time being kept under a warmer.'
```

```
pratice_sentence = all_review[7]
print('原始字串: {}'.format(pratice_sentence))
##全部轉換為大寫
pratice_sentence.upper()
##全部轉換為小寫
pratice_sentence.lower()
```

Function: Counter()

Counter() 是一個相當方便的 function，可以快速計算字串中所有字元出現過的次數
from collections import Counter

Counter()

```
In [44]: from collections import Counter
In [45]: ##快速計算所有字元出現次數
         count_ = Counter(all_review[8])
         print(count_)
Counter({'e': 5, ' ': 4, 'r': 3, 't': 2, 'o': 2, 'T': 1, 'h': 1, 'f': 1, 'i': 1, 's': 1, 'w': 1, 'g': 1, 'a': 1, '.':
1})
In [52]: ##出現最多的字元(前五名)
         count_.most_common(5)
Out[52]: [('e', 5), (' ', 4), ('r', 3), ('t', 2), ('o', 2)]
In [48]: ## 1 字元出現幾次
         count_.get('h')
Out[48]: 1
```

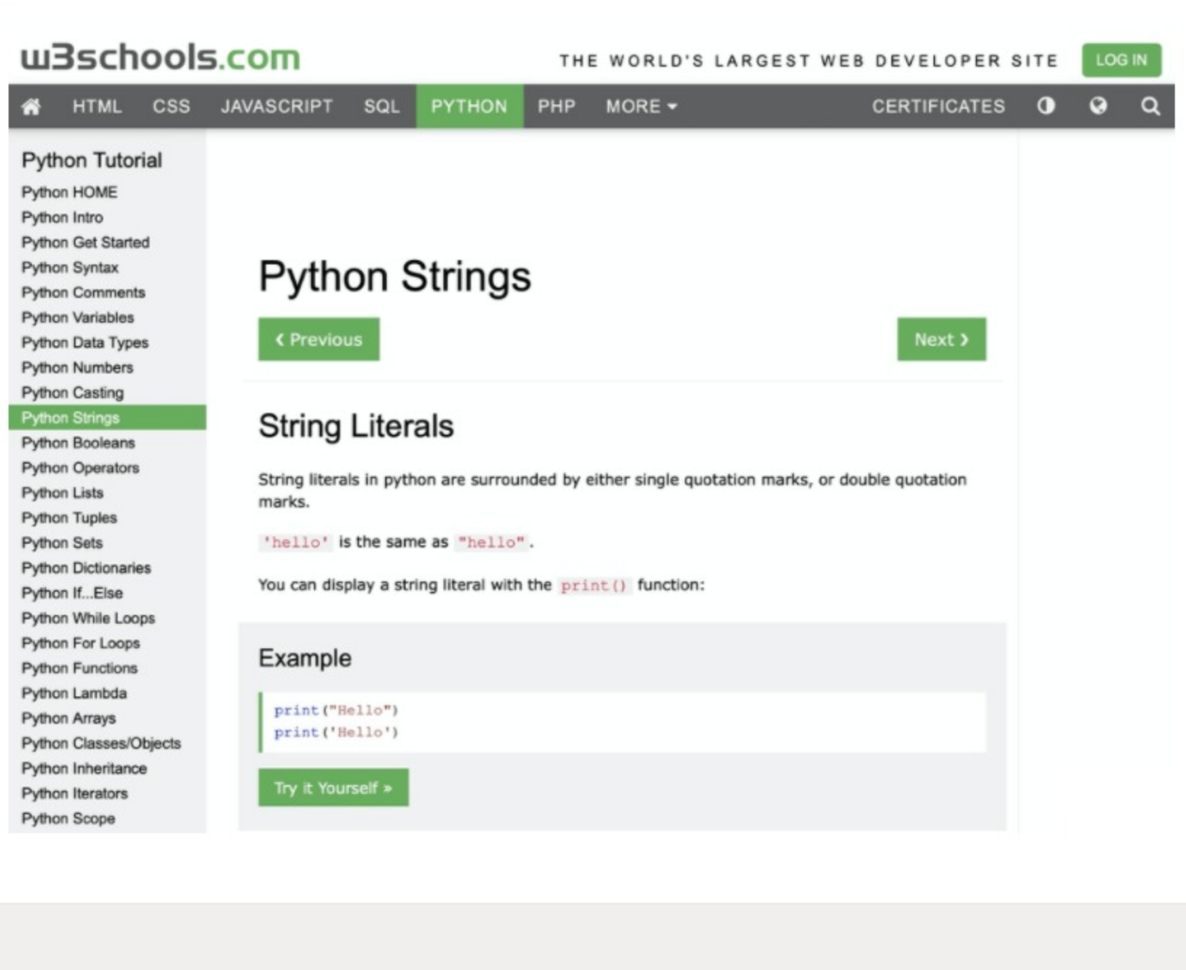
返回一個dictionary對應字串內所有的字元與出現過次數

```
from collections import Counter
##快速計算所有字元出現次數
count_ = Counter(all_review[8])
print(count_)
##出現最多的字元(前五名)
count_.most_common(5)
## 1 字元出現幾次
count_.get('h')
```

參考資料

W3schools：常用 Functions
網站：W3schools

課程中列舉了最常使用的一些 functions，然而 python 本身支援更多樣的操作，有興趣的學員們可以延伸閱讀並作練習。



下一步：閱讀範例與完成作業