字串處理

蔡尚融

2018-04-23

字串（String）

- 字串（string）是一序列字元 (character)。
- 用索引值（index）取得所在位置字元，範圍從零開始。

```
>>> 'abcdefghijk'[4]
'e'
```

- 負索引值從尾端取得字元。

```
>>> 'abcdefghijk'[-3]
'i'
```

- 字串中的字元不能替換。
- 用 `len()` 取得字串長度。

```
>>> len('abcdefghijk')
11
```

迴圈掃過字串

- 用 while 迴圈：

```
>>> mytext = 'hello, world'
>>> ii = 0
>>> while ii < len(mytext):
...  print(mytext[ii])
...  ii = ii + 1
```

- 用 for 迴圈：

```
>>> mytext = 'hello, world'
>>> for ch in mytext:
...  print(ch)
```

切割字串

- 取得由索引 m 到 小於 n 範圍的子字串。

```
>>> mytext = 'hello, world'
>>> mytext[3:9]
```

- 取得由索引 m 到結尾的子字串。

```
>>> mytext[3:]
```

- 取得至索引 n 之前的子字串。

```
>>> mytext[:10]
```

- 指定索引間距 k ，取得子字串。

```
>>> mytext[::3]
```

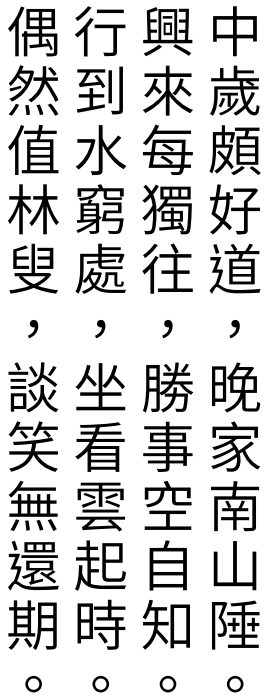
- 反轉字串。

```
>>> mytext[::-1]
```

隨堂練習：

以直式方式印出王維的詩《終南別業》

中歲頗好道，晚家南山陲。
興來每獨往，勝事空自知。
行到水窮處，坐看雲起時。
偶然值林叟，談笑無還期。



搜尋與取代子字串

- 用 `find()` 成員函式搜尋子字串，結果回傳為找到的索引位置，若回傳為 -1 則表示沒有此字串。
  - 從位置零開始：

```
>>> mytext = 'A test string for test str.find()
function.'
>>> mytext.find('test')
2
```

  - 從指定位開始搜尋：

```
>>> mytext.find('text', 3)
18
```

  - 從指定位與範圍搜尋：

```
>>> mytext.find('text', 3, 9)
-1
```

- `rfind()` 從尾端搜尋子字串。

- 用 in 運算子，檢查一字串是否為另一字串的子字串。

```
>>> 'text' in mytext
False
```

- replace() 函式，用以取代字串中指定的子字串，並回傳新字串。
  - 全部取代：

```
>>> mytext.replace('test', 'text')
'A text string for text str.find() function.'
```

  - 指定個數取代：

```
>>> mytext.replace('test', 'text', 1)
'A text string for test str.find() function.'
```

字串分割

- 將字串以空白（whitespace）字元作區分，分割成一串列 字串。

```
>>> septext = 'Whitespace is an esoteric programming
language'
>>> septext.split()
['Whitespace', 'is', 'an', 'esoteric', 'programming',
'language']
```

- 指定分割字元

```
>>> septext.split(sep = 's')
['White', 'pace i', ' an e', 'oteric programming
language']
```

- 指定最大分割數。

```
>>> septext.split(maxsplit = 3)
['Whitespace', 'is', 'an', 'esoteric programming
language']
```

- 從字串尾端分割字串。

```
>>> septext.rsplit(maxsplit = 3)
['Whitespace', 'is', 'an', 'esoteric programming
language']
```

隨堂練習：

計算下列字串中的單詞（word）個數。

Whales are a widely distributed and diverse group of fully aquatic placental marine mammals.

簡易文字檔案讀寫

- 寫入文字檔

```
#!/usr/bin/env python

testfile = open('testfile', 'w')
testfile.write("This is a test text for test file.")
testfile.close()
```

- 讀取文字檔

```
#!/usr/bin/env python

testfile = open('testfile', 'r')
filetext = testfile.read()
print(filetext)
```

範例：簡易替換密碼

藉由改變字母表上字母順序以特定方式排列，並以此對應關係將明文轉換成密文。例如原字母明文與密文對應關係為：

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
JFKBZTXRYCIVQUDPESAMLGHOWN
```

明文：MESSAGE
密文：QZAAJXZ

產生一組明文與密文對應關係並存成檔案，以此對應關係處理一文字檔案中字母（含大小寫）與數字加密。考慮 ASCII 碼，用 `ord()` 函式將字元轉成整數；用 `chr()` 函式將整數轉成字元。

數字與大小寫字母次序打亂。

字元	數字 0 到 9	大寫字母	小寫字母
ASCII	48 ～ 57	65 ～ 90	97 ～ 122

```
#!/usr/bin/env python import random

def keygen_text(filename):
    keylist = list(range(ord('0'), ord('9') + 1))
    keylist.extend(list(range(ord('A'), ord('Z') + 1)))
    keylist.extend(list(range(ord('a'), ord('z') + 1)))

    random.shuffle(keylist)
```

```
keytext = ''
for ii in range(len(keylist)):
    keytext = keytext + chr(keylist[ii])

keyfile = open(filename, 'w')
keyfile.write(keytext)
keyfile.close()

return keytext
```

產生對應關係串列，考慮在 ASCII 範圍（0 ～ 128）內，數字與大小寫字母作替換，其他不改變。

```
def keymap_setup(keytext):
    keymap = list(range(128))

    ii = 0
    ij = ord('0')
    while ii < 10:
        keymap[ij] = ord(keytext[ii])
        ii = ii + 1
        ij = ij + 1
```

```
ij = ord('A')
while ii < 36:
    keymap[ij] = ord(keytext[ii])
    ii = ii + 1
    ij = ij + 1

ij = ord('a')
while ii < 62:
    keymap[ij] = ord(keytext[ii])
    ii = ii + 1
    ij = ij + 1

return keymap
```

對輸入字串作對應轉換。

```
def simple_encrypt(orig):
    text = ''
    for ii in range(len(orig)):
        text = text + chr(keymap[ord(orig[ii])])
    return text
```

讀入測試檔案，並輸出結果檔案。

```
keytext = keygen_text('mykeytext')
keymap = keymap_setup(keytext)

oldtxtfile = open('myoldtext', 'r')
textold = oldtxtfile.read()
oldtxtfile.close()

textnew = simple_encrypt(textold)

newtxtfile = open('mynewtext', 'w')
newtxtfile.write(textnew)
newtxtfile.close()
```

字串對應轉換

- 使用字串（string）型態的 translate()函式，達成轉換。
- 需用用字串型態的產生對應（map）。

```
#!/usr/bin/env python

itab = 'abcdefghij'
otab = '1234567890'

tmap = str.maketrans(itab, otab)
oldtxt = 'This is a test string for example of abcdefghij'
newtxt = oldtxt.translate(tmap)
print(oldtxt)
print(newtxt)
```

隨堂練習：密文解譯

將前「簡易替換密碼」範例產生的密文，以範例產生的金鑰檔（mykeytext）還原成明文，並輸出至螢幕。

提示：

1. 將金鑰檔內容讀入成字串。
2. 用「簡易替換密碼」範例中部份函式實作，產生還原數字與字母字串。
3. 用字串對應方式轉換。