

資料科學概論期末報告

第八章 一字串

系級：經濟三乙

組員：A106260018張鈞茹A106260048楊于萱
A106260052劉映妤A106260082姜品君

首先，如何建立空字串？

```
[6] 1 s1=str()  
    2 s1
```

```
☐→ ''
```

```
[7] 1 s2=''  
    2 s2
```

```
☐→ ''
```

```
[8] 1 s3=""  
    2 s3
```

```
☐→ ''
```

由於Python沒有字串和字元之分，所以可以使用單引號或雙引號刮起來。
但在其他程式語言中就不同了，字串是以雙引號刮起來的，而字元是以單引號刮起來的。

首先，先建立一個字串:Learning Python now!

```
[9] 1 s3='Learning Python now!'  
    2 s3
```

```
↳ 'Learning Python now!'
```

```
[10] 1 s4=str('Learnig Python now!')  
    2 s4
```

```
↳ 'Learnig Python now!'
```

計算字串的長度：

```
[11] 1 len(s3)
```

```
↳ 20
```

計算字串的最大值：

```
[12] 1 max(s3)
```

```
↳ 'y'
```

計算字串的最小值：

```
[13] 1 min(s3)
```

```
↳ ''
```


可利用索引運算子[]
用來擷取字串的某一字元

```
[11]  1 s3 = 'Learning Python now!'
      2 s3[3]

↳ 'r'
```

若索引值是負值，則須將
此值加上字串長度

```
[12]  1 s4 = 'Python'
      2 s4[-1]

↳ 'n'
```

由於s4的長度為6，
所以s4[-1]的真正索引值為
5。亦擷取s4[5]。

```
[13]  1 s4 = 'Python'
      2 s4[-3]

↳ 'h'
```

以此類推，s4[-3]即為s4[3]

也可以使用分割運算子
[start:end]表示擷取從
start 到 end-1

其中，s4[1:-1]
表示s4[1:-1+(len(s4))]
亦即s4[1:5]

```
[18] 1 s4[1:4]
```

```
↳ 'yth'
```

```
[19] 1 s4[:4]
```

```
↳ 'Pyth'
```

```
[20] 1 s4[1:]
```

```
↳ 'ython'
```

```
[21] 1 s4[1:-1]
```

```
↳ 'ytho'
```

```
[22] 1 s4[:4]
```

```
↳ 'Pyth'
```

前面提到，`s4[1:-1]`表示`s4[1:-1+(len(s4))]`，亦即`s4[1:5]`。
和串列一樣，`+`表示連結，而`*`表示複製。

```
[25]  1 s5 = 'Bright'  
      2 s6 = '_Tsai'  
      3 s5+s6
```

```
☞ 'Bright_Tsai'
```

```
[26]  1 s5*2
```

```
☞ 'BrightBright'
```

若要檢視某一字串是否在另一字串，
可使用 `in` 或 `not in`。

判斷字串是否出現某一字串中，
所以必須要加上引號。

同理，也可以利用 `for` 敘述列印字串
的所有元素值。

```
[27] 1 s5 = 'Bright'  
      2 s6 = '_Tsai'
```

```
[28] 1 'B' in s5
```

```
↳ True
```

```
[29] 1 'T' not in s6
```

```
↳ False
```

```
[31] 1 for i in s5:  
      2 |     print(i, end=' ')
```

```
↳ B r i g h t
```


字串是字母和數字組成：

```
1 s8='Linda'  
2 s8.isalnum()
```

```
True
```

字串是字母所組成：

```
[22] 1 s8.isalpha()
```

```
True
```

字串是數字所組成：

```
[23] 1 s8.isdigit()
```

```
False
```

字串是符合識別字的名稱：

```
[24] 1 s8.isidentifier()
```

```
True
```

字串的英文字元皆是由
大寫字母所組成：

```
[25] 1 s8.isupper()
```

```
False
```

字串的英文字元皆是由
小寫字母所組成：

```
[26] 1 s8.islower()
```

```
False
```

字串的字元皆是由白色
空白所組成：

```
[27] 1 s8.isspace()
```

```
False
```

子字串的運作

若字串的尾端是da子字串時，則回傳True。

```
1 s8 = 'Linda'  
2 s8.endswith('da')
```

True

若字串的开頭是Li子字串時，則回傳True。

```
1 s8.startswith('Li')
```

True

找尋字串中出現d子字串的最小索引值，並加以回傳。

```
s8.find('d')
```

```
3
```

找尋字串中出現B子字串的最小索引值，並加以回傳。

```
1 s8.find('B')
```

```
-1
```


找尋字串中出現e子字串的最大索引值，並加以回傳。

```
s10 = 'abcdeabcde'  
s10.rfind('e')  
9
```

計算字串中出現e的個數。

```
s10.count('e')  
2
```

轉換字串

capitalize()

將字串中第一個字元轉換為大寫，其餘字元轉為小寫後加已回傳。

```
[ ] 1 s11="welcome to Taipei"
```

```
[ ] 1 s11.capitalize()
```

```
➞ 'Welcome to taipei'
```

lower()

將字串中的所有字元轉換為小寫後加以回傳。

```
[ ] 1 s11.lower()
```

```
➞ 'welcome to taipei'
```

upper()

將字串中的所有字元的換為大系後加以區傳。

```
[ ] 1 s11.upper()
```

```
➞ 'WELCOME TO TAIPEI'
```

swapcase()

將字串中每一單字的第一個字元轉換為大寫，其餘字元轉換為小寫後加以回傳。

title()

將字串中大寫字元轉換為小寫字元，將小寫字元轉換為大寫字元後加以回傳。

replace(old, new)

將old字串以new字串取代之。

```
[ ] 1 s11="welcome to Taipei"
```

```
[ ] 1 s11.swapcase()
```

```
↳ 'WELCOME TO TAIPEI'
```

```
[8] 1 s11.title()
```

```
↳ 'Welcome To Taipei'
```

```
[9] 1 s11.replace("Taipei", "Tainan")
```

```
↳ 'welcome to Tainan'
```

如何從字串中去掉頭尾空白？

```
s12 = ' Learning Python Now! '  
s12  
' Learning Python Now! '
```

```
' Learning Python Now! '
```

```
s12.lstrip()
```

```
'Learning Python Now! '
```

lstrip() 刪除字串左側的空白後加以回傳。

`rstrip()` 刪除字串右側的空白後加以回傳。

```
s12.rstrip()
```

```
' Learning Python Now! '
```

`strip()` 刪除字串兩側的空白後加以回傳。

```
s12.strip()
```

```
'Learning Python Now! '
```

如何將字串加以格式化？

在給予20的欄位寬下向
中靠齊：

```
[30] 1 s13='Bright Tsai'  
     2 s13.center(20)
```

```
☞ '    Bright Tsai    '
```

在給予20的欄位寬下向
左靠齊：

```
[31] 1 s13.ljust(20)
```

```
☞ 'Bright Tsai      '
```

在給予20的欄位寬下向
右靠齊：

```
[32] 1 s13.rjust(20)
```

```
☞ '          Bright Tsai'
```

還有一個方法是 `split`，將字串解析到串列中。

```
lst = s100.split()
```

表示將字串 `s100` 以空白為分隔字符，
將字串 `s100` 加以分割，
然後存放於串列 `lst` 中。

從其輸出結果可得知它是存放於串列中。

```
[33] 1 s100='Apple Orange Banana Kiwi'
      2 lst=s100.split( )
      3 lst
```

```
↳ ['Apple', 'Orange', 'Banana', 'Kiwi']
```

`lst2 = s200.split('-')` 中，
是以 `'-'` (dash 字符) 為分隔字符，
將字串 `s200` 分割後存放於串列 `lst2`。

```
[34] 1 s200='01-13-2018'
      2 lst2=s200.split('-')
      3 lst2
```

```
↳ ['01', '13', '2018']
```

習題1:

試撰寫一程式，以不定數迴圈輸入以：時:分:秒表示的時間數字，隨後將它拆解存放於串列。最後再將此串列輸出。當輸入為end則結束輸入資料。

輸入與輸出樣本1:

20:12:56

hour: 22, min: 12, second:56

輸入與輸出樣本2:

22:10:01

hour: 22, min: 10, second:01

end


```
[ ] 1 #参考程式
    2
    3 lst = []
    4 while True:
    5     str = input()
    6     if str != 'end':
    7         lst = str.split(':')
    8         print('hour: %s, min: %s, second: %s' %(lst[0],lst[1], lst[2]))
    9     else:
   10         break
```



22:12:56

hour: 22, min: 12, second: 56

22:10:01

hour: 22, min: 10, second: 01

10:49:25

hour: 10, min: 49, second: 25

end

習題2:

試撰寫一程式，輸入一變數名稱，然後判斷它是否為合法的變數名稱。

假設取變數名稱的準則如下：

- A. 第一個字元需要英文字母
- B. 接下的字元可為英文字母或是數字
- C. 不可以為其他符號

輸入與輸出樣本 1:

輸入:

Lo

輸出:

Valid variable name

輸入與輸出樣本 2:

輸入:

7sdf ja

輸出:

Invalid variable name

輸入輸出樣本 3:

輸入:

abc123

輸出:

Valid variable name

輸入輸出樣本 4:

輸入:

abc123%

輸出:

Invalid variable name

```
[ ] 1 #参考程式
    2
    3 varName = input()
    4 validVar = True
    5
    6 if not varName[0].isalpha():
    7 |   validVar = False
    8 else:
    9 |   for i in range(0, len(varName)):
   10 |   |   if not varName[i].isalpha() and \
   11 |   |   |   not varName[i].isdigit():
   12 |   |   |   validVar = False
   13 |   |   |   break
   14 if validVar:
   15 |   print('Valid variable name')
   16 else:
   17 |   print('Invalid variable name')
```



abc123

Valid variable name

習題3:

撰寫一程式，輸入九個字串置放於一名為1st的字串，其長度不常於10個字元。接下來，每一列映出三個字串，並向左靠齊。

提示:每一個字串輸出欄位寬為15


```

[2] 1 #参考程式
    2
    3 lst=[]
    4 for i in range(1,10):
    5     str = input()
    6     lst.append(str)
    7
    8 for k in range(1,10):
    9     if k % 3 != 0:
10 #center
11         print("|"+lst[k-1].ljust(15)+"|",end=" ")
12     else:
13         print("|"+lst[k-1].ljust(15)+"|")

```



```

apple
orange
kiwi
banana
grape
pineapple
guava
cherry
blueberry
|apple      | |orange      | |kiwi      |
|banana     | |grape       | |pineapple  |
|guava      | |cherry      | |blueberry  |

```

習題4:

撰寫一程式，已依不一定迴圈要求使用者輸入字串，檢視若字串e字元尾端，則將此字串加入lst串列中，最後將其印出若字串e字元尾端，則將此字串加入lst串列中，最後將其印出。當使用者輸入end時將結束輸入的動作。

```
[ ] 1 #參考程式
    2
    3 lst=[]
    4 while True:
    5     str = input()
    6     if str != "end":
    7         if str.endswith("e"):
    8             lst.append(str)
    9     else:
   10         break
   11
   12 print(lst)
```

```
☞ apple
   pineapple
   banana
   orange
   kiwi
   grape
   end
['apple', 'pineapple', 'orange', 'grape']
```

習題5:

撰寫一程式，輸入一含有20字元以上的字串，請將字串中的字元屬性印出，如他是英文字母、數字、空白，還是其他屬性。


```
[4] 1 #參考程式
    2
    3 str=input()
    4 for i in range(len(str)):
    5     if str[i].isdigit():
    6         print(str[i]+":is a digit.")
    7     elif str[i].isalpha() and str[i].isupper():
    8         print(str[i]+":is upper alpha.")
    9     elif str[i].isalpha() and str[i].islower():
   10         print(str[i]+":is lower alpha.")
   11     elif str[i].isspace():
   12         print(str[i]+":is a space.")
   13     else:
   14         print(str[i]+":is a symbol character.")
```

☞ I am a student.
I:is upper alpha.
 :is a space.
a:is lower alpha.
m:is lower alpha.
 :is a space.
a:is lower alpha.
 :is a space.
s:is lower alpha.
t:is lower alpha.
u:is lower alpha.
d:is lower alpha.
e:is lower alpha.
n:is lower alpha.
t:is lower alpha.
.:is a symbol character.



The End