08字串 (String)

我們在課程開始初期提過，字串可以用雙引號 “ ”，單引號 ‘ ’來匡列字串，例如 ”TVDI” 或 ’TVDI’ 都可以，請注意在其他幾個主流語言中 C、C++或是Java都只能使用雙引號匡列字串，單引號匡列單一字元。

由於字串經常使用，相關功能也不少，在這邊介紹較常運用的功能

1. 建立空字串

我們可以使用以下兩種方式建立空字串，一是以str()，二是以引號 “ “表示

s1 = str()

s2 = ""

print('s1=',s1,'s2=',s2)

輸出結果

s1= s2=

1. 字串運作

字串與一般變數相同，定義時直接給予初始值。

s3 = 'Lerning Python now!'

s4 = str(356981)

print('s3 = ',s3,'\ns4 = ',s4)

輸出結果

s3 = Lerning Python now!

s4 = 356981

使用與串列相同的方式取得，字串長度、取得字串內字元最大值與最小值

a = len(s3)

b = max(s3)

c = min(s3)

print('s3字串長度：',a,'\n字串內最大值：',b,'\n字串內最小值：',c)

輸出結果

s3字串長度： 19

字串內最大值： y

字串內最小值：

前面串列與數組等，在索引時所提到的中括號 [ ]，稱為索引運算子，括號內為索引值，在字串中，使用方式相同。

a = s3[3]

b = s3[-1]

c = s3[-6]

print ('字串中索引值3的字元：',a,'\n索引值-1的字元：',b,'\n索引值-6的字元：',c)

輸出結果

字串中索引值3的字元： n

索引值-1的字元： !

索引值-6的字元： n

與串列相同，索引值可以由後向前計算，同樣要注意的是，最後一個字元索引值為 -1 。

需要索引特定字串中特定區域，[ start:end ]

a = s3[8:14]

b = s3[:7]

c = s3[15:]

print('索引值[8:14]：',a)

print('索引值[:7]：',b)

print('索引值[8:]：',c)

輸出結果

索引值[8:14]： Python

索引值[:7]： Lerning

索引值[8:]： now!

字串連結使用 + 號 ，字串複製使用 \*號。

d = b +' '+ a + " " + c

print (d)

d = d \*2

print (d)

輸出結果

Lerning Python now!

Lerning Python now!Lerning Python now!

使用join 連結容器內容資料

A = ('633','665','998','772','443')

print('+'.join(A))        #將"+"與tuple內容串接

print(eval('+'.join(A)))    #將"+"與tuple內容串接並計算

輸出結果：

633+665+998+772+443

3511

檢視一段字串或一個字元，是否存在目標的字串中，使用 in ，not in

L1 = a in s3

L2 = b not in s3

print('L1:',L1, 'L2:',L2)

輸出結果

L1: True L2: False

檢查字元ASCII碼，注意輸出結果為整數數值型態

a = ord(s3[4])

print(f"{s3[4]} 的ASCII碼為 {a}")

輸出結果

n 的ASCII碼為 110

m=chr(a)

print(f'ASCII {a}是 字母 {m}')

輸出結果

ASCII 110是 字母 n

字串中字元也是可以逐一列印：

for i in s4:

    print(i)

for i in range (len(s4)):

    print(s4[i])

兩種方式輸出結果相同

3

5

6

9

8

1

1. 測試字串

python字串中的str類別提供很多字串運作的方法。包括測試字串、子字串處理，轉換字串、從字串除掉空白，以及將字串加以格式化。

測試字串的目的，在於測試字串是否屬於英文字母數字、字母、數字、以及其他種類，可參照下面的表格：

|  |  |
| --- | --- |
| 方法 | 說明 |
| isalpha() | 字串字元全由字母所組成，傳回True |
| isdigit() | 字串字元全由數字所組成，傳回True |
| isalnum() | 字串字元由字母和數字組成，傳回True |
| isidentifier() | 字串符合Python規範的識別字，傳回True |
| islower() | 字串英文字元由小寫組成，傳回True |
| isupper() | 字串英文字元由大寫組成，傳回True |
| isspace() | 字串字元由空白組成，傳回True |

範例：

s5 = 'John0123'

a = s3.isalnum()

b = s5.isalnum()

print(f'英數字元組合 s3:{a} s5:{b}')

輸出結果：

英數字元組合 s3:False s5:True

a = s3.isalpha()

b = s5.isalpha()

print(f'英文字母組合 s3:{a} s5:{b}')

輸出結果：

英文字母組合 s3:False s5:False

a = s3.isdigit()

b = s4.isdigit()

print(f'數字組合 s3:{a} s4:{b}')

輸出結果：

數字組合 s3:False s4:True

s1 = 'abcde'

s2 = 'ABCDE'

a = s1.islower()

b = s2.islower()

c = s1.isupper()

d = s2.isupper()

print(f'{a}, {b}, {c}, {d}')

輸出結果：

True, False, False, True

s2 = "      "

c = s2.isspace()

print (f"空白字元 s2:{c}")

輸出結果：

空白字元 s2:True

a = input("識別字：")

print(a.isidentifier()) #判斷字串是否可作為合法的識別字

輸出結果：

識別字：!ppt

False

1. 子字串

|  |  |
| --- | --- |
| 方法 | 說明 |
| endswith(s1) | 字串尾端是否為s1的內容，是則傳回True |
| startswith(s1) | 字串前端是否為s1的內容，是則傳回True |
| find(s1) | 字串中出現s1時，傳回最小索引碼 |
| rfind(s1) | 字串中出現s1時，傳回最大索引碼 |
| count(s1) | 計算字串中出現s1個數 |

輸出結果：

a = s3.endswith('now!')

b = s3.startswith('Learning')

print(f'{a}, {b}')

True, False

c = s3.find('n')

d = s3.rfind('n')

e = s3.count('n')

print(f'第一個出現的 n：{c}, 最後一個出現的 n：{d}, 共 {e} 個 n')

輸出結果：

第一個出現的 n：3, 最後一個出現的 n：15, 共 4 個 n

1. 字串轉換

|  |  |
| --- | --- |
| 方法 | 說明 |
| capitalize() | 把字串第一個字元轉為大寫，其他轉為小寫後回傳 |
| lower() | 把字串中所有字元轉為小寫後回傳 |
| upper() | 把字串中所有字元轉為大寫後回傳 |
| title() | 將字串中每一個英文字第一個字元轉換為大寫，其餘字元轉換為小寫後回傳 |
| swapcase() | 將字串中大寫轉為小寫，小寫轉為大寫後回傳 |
| replace(old,new) | 將old字串以new字串取代 |

s3 = s3.replace('Python','Python\_Ver3.3')# 改換字串

print(s3)

輸出結果

Lerning Python\_Ver3.3 now!

s3 = s3.lower()  #字串全部改為小寫

print(s3)

輸出結果

lerning python\_ver3.3 now!

s3 = s3.upper()  #字串全部改為大寫

print(s3)

輸出結果

LERNING PYTHON\_VER3.3 NOW!

s3 = s3.capitalize()  #第一個字元改為大寫，其他改為小寫

print(s3)

輸出結果

Lerning python\_ver3.3 now!

s3 = s3.title() #字串內每個英文字第一個字母大寫

print(s3)

輸出結果

Lerning Python\_Ver3.3 Now!

s3 = s3.swapcase() #字串內所有大小寫交換

print(s3)

輸出結果

lERNING pYTHON\_vER3.3 nOW!

1. 去除字串頭尾空白

|  |  |
| --- | --- |
| 方法 | 說明 |
| lstrip() | 刪除字串左側空白後回傳 |
| rstrip() | 刪除字串右側空白後回傳 |
| strip() | 刪除字串兩側空白後回傳 |

s3 = " "+ s3 + " "

print(s3+'.')

s3 = s3.lstrip()

print(s3+'.')

輸出結果

Lerning Python now! .

Lerning Python now! .

s3 = " "+ s3 + " "

s3 = s3.rstrip()

print(s3+'.')

輸出結果

Lerning Python now!.

s3 = " "+ s3 + " "

s3 = s3.strip()

print(s3+'.')

輸出結果

Lerning Python now!.

1. 字串依照格式排列

|  |  |
| --- | --- |
| 方法 | 說明 |
| center(width) | 在width的欄位寬下向中靠齊，加以回傳 |
| ljust(width) | 在width的欄位寬下向左靠齊，加以回傳 |
| rjust(width) | 在width的欄位寬下向右靠齊，加以回傳 |

s4 = s4.center(10)

print(s4+'.')

輸出結果

356981 .

s4 = s4.ljust(10)

print(s4+'.')

輸出結果

356981 .

s4 = s4.rjust(10)

print(s4+'.')

輸出結果

356981.

分割字串

使用split()這個方法，將字串拆分，在split的括號()中，若沒有任何內容，會以空白字元作為字串切割點，若括號內有其他符號，例如('-')則會以 – 這個字元作為字串切割點，以下以範例說明

s1 = '2019/10/20'

s2 = '02-28721940-262'

s3 = '蘋果 香蕉 柿子 西瓜'

lst1 = s1.split('/')

lst2 = s2.split('-')

lst3 = s3.split()

print('lst1 = ',lst1)

print('lst2 = ',lst2)

print('lst3 = ',lst3)

輸出結果

lst1 = ['2019', '10', '20']

lst2 = ['02', '28721940', '262']

lst3 = ['蘋果', '香蕉', '柿子', '西瓜']

依照換行內容切割方式

s5 = "23\n12\n45\n56"

lst5 = s5.splitlines()

print('lst5 = ',lst5)

s6 = "23\n12\n45\n56"

lst6 = s6.split("\n")

print('lst6 = ',lst6)

輸出結果

lst5 = ['23', '12', '45', '56']

lst6 = ['23', '12', '45', '56']

練習題：

1. 使用者輸入一個字串，顯示該字串的每一個字元索引
2. 使用者輸入一個字串，顯示每個字元對應的ASCII碼，並將ASCII碼加總
3. 使用者輸入一個句子，(至少五個詞，以空白隔開，並輸出倒數三個詞)
4. 使用者輸入一個字串，將字串1.全部改為大寫。2.每個字第一個字母改為大寫。
5. 使用者輸入一個字串和一個字元，計算字串中，字元出現的次數。
6. 使用者輸入一個字串，字串為五個數字，以空白隔開，加總五個數字並計算平均
7. 使用者輸入一串序號，格式為ddd-dd-ddd ，d為數字。檢查輸入格式正確，則輸出"格式正確！"，否則輸出"格式錯誤！"
8. 使用者輸入一組字串，檢查是否符合規則：1.至少8個字元，2.只有英文字母和數字，3.至少有一個大寫英文字母。符合顯示"密碼已設定"，不合顯示"密碼設定不符規則"