

第十四單元

用Python學英文

字串與串列

Python 並沒有細分字元或字串型態，用單引號 `b='Horng'` 或雙引號 `a="Horng"` 圍起的字元，通通是字串，例如：

```
a="Horng"
print(a)
```

要從字串截取子字串，可以同串列的截取子串列。例如：

```
b="Horng"
print(b)
print(b[0])#H
print(b[0:2])#Ho 0到2不含2
print(b[2:])#rng 2開始
print(b[:3])#Hor 0到3，不含3
```

`split()`

幾乎所有程式語言都是使用 `split()` 分割字串，Python 也不例外。例如：

```
a="There are four people in my family."
b=a.split(" ")#以空白為分割依據。
print(b)
print(b[0])
print(b[1])
```

以上 b 是一個串列，輸出如下圖：

```
['There', 'are', 'four', 'people', 'in', 'my', 'family.']
There
are
```

測資含有空白

有時候測試資料含有多筆資料，中間以空白隔開，例如：

```
2 3 4
```

那就要用 `split()` 分解資料，例如：

```
d=input("input a b c:")#請輸入 2 3 4
print(d)#2 3 4
e=d.split(' ')#以空白分割資料
print(e)# ['2','3','4']
print(e[0])#2
print(e[1])#3
print(e[2])#4
a=e[0]
b=e[1]
c=e[2]
print(a+b+c)# 234
print(int(a)+int(b)+int(c))
```

輸出結果如下圖：

```
input a b c:2 3 4
2 3 4
['2', '3', '4']
2
3
4
234
9
```

請留意以上變數 e 稱為串列 (list)。

範例14a

主客場籃球賽制。(108年APCS第二梯次試題)

1. 連打兩場，主場球隊兩場皆贏輸出 Win。
2. 兩場皆輸，輸出 Loss。
3. 其餘輸出 Tie。
4. 輸入格式：每場 4 節得分，主隊先輸入。以下分別是主隊、客隊兩場比賽的四小節得分

15 20 25 30 (主隊第一場得分)

15 16 17 18 (客隊第一場得分)

5 10 15 20 (主隊第二場得分)

5 6 7 8 (客隊第二場得分)

5. 參考解答如下：本例 4 次輸入資料與計算總分都一樣，所以將輸入資料寫成函式 inputdata()。

```
def inputdata():
    p=input('input a b c d:')
    q=p.split(' ')#本例測資是以空白分隔，所以要以split分解，請看第6單元
    print(q)
    r=[int(q1) for q1 in q]
    print(r)
    return sum(r)
a1=inputdata()
b1=inputdata()
a2=inputdata()
b2=inputdata()
num=0
if a1>b1:
    num=num+1
if a2> b2:
    num=num+1
if num==2:
    print('Win')
else:
    if num==0:
        print('Loss')
```

```
else:  
    print('Tie')
```

輸出結果

以下僅輸出『輸入第一筆資料』的結果，其餘請自行執行。

```
input a b c d:15 20 25 30  
['15', '20', '25', '30']  
[15, 20, 25, 30]
```

自我練習

1. 請寫一程式，讓使用者可以輸入任意數量的整數，整數與整數之間以半形逗號(,)隔開，輸入後可計算其和、平均、極大值與極小值。例如，輸入3,2,4，輸出9。輸入1,3,2,4，輸出10。
2. 假設a='03:20'代表3點20分，請寫程式輸出『3點20分』。
3. 假設a='2020/02/18'代表2020年2月18日，請寫程式輸出『2020年2月18日』
4. 請寫一程式，可以計算所輸入句子的字數。例如：'There are four people in my family.'表示共有7個字。
5. 請寫一程式，可以計算所輸入文章的句數。例如：' This is a book. There are four people in my family.'表示共有2句。

replace(ch1,ch2)

將ch1子字串由ch2取代。例如：

```
a="There are four people in my family."  
b=a.replace("are","is")  
print (b)#There is four people in my family.
```

startswith()、endswith()

傳回字串是否由某一子字串開頭或結束。

```
a="洪國勝"
print(a.startswith("洪"))#True
print(a.endswith("勝"))#True
```

且字串也可用串列處理，使用部分的串列方法。例如：請鍵入以下程式，並觀察執行結果。

```
a='abcdef'
print(a[0])#a
print(a[-1])#f
print(a[-2])#e
print(a[0:3])#abc
print(a[2:])#cdef
print(a[-2:])#ef 倒數兩個
print(len(a))#6
print(max(a))#f 最大元素
b=list(a) #轉為串列
print(b)
```

範例14b

統一發票對獎。請寫一程式，可以幫使用者核對統一發票中獎情形。

程式列印

```
a1='59647042'#特別獎
a2='01260528'#特獎
a3=['01616970','69921388','53451508']#頭獎
a4=['710','585','633']#增開六獎
b='31616970' #測試資料
c=''
if b==a1 :
    c='特別獎'
elif b==a2 :
```

```
    c='特獎'  
else:  
    for i in range(len(a3)): #本例頭獎有三個  
        d=a3[i]  
        if b==d:  
            c='頭獎'  
        elif b[1:]==d[1:]:#大家都取末七位數  
            c='二獎'  
        elif b[2:]==d[2:]:  
            c='參獎'  
        elif b[3:]==d[3:]:  
            c='四獎'  
        elif b[4:]==d[4:]:  
            c='五獎'  
        elif b[5:]==d[5:]:  
            c='六獎'  
    for i in range(len(a4)): #增開六獎  
        if b[5:]==a4[i]:  
            c='六獎'  
  
print (c)
```

自我練習

1. 以上範例，我們僅核對一張發票，但我們通常很多發票，請將使用者的 6 張發票以串列儲存，並由電腦核對中獎情形。
2. 範例 14b 的得獎號碼是使用指派，請您自己寫程式開獎。
3. 請寫一程式，可以判斷一個字串是否全為大寫字母。例如：'AABDF' 傳回『是』，'AcFgk' 傳回『否』。
4. 請寫一程式，可以找出個字串所含的大寫字元、小寫字元、數字的個數。例如：'AcFgk7' 傳回大寫 2，小寫 3，數字 1
5. 中華民國的身分證字號有其特定的編碼原則。第一個字是大寫的英文字母，其餘 9 個字必須為數字，但在套用編碼原則時，第一個英文字母將會先依下表被轉換為數字：

字母	A	B	C	D	E	F	G	H	J	K	L	M	N
數字	10	11	12	13	14	15	16	17	18	19	20	21	22
字母	P	Q	R	S	T	U	V	X	Y	W	Z	I	O
數字	23	24	25	26	27	28	29	30	31	32	33	34	35

轉換後的身分證字號(共 11 位數字)每一位數均有固定的權重 (Weight)，由左往右依序為『1 9 8 7 6 5 4 3 2 1 1』。判斷身分證字號是否正確的方法為：每一位數字與其相對應的權重相乘後再加總，加總後的結果若為 10 的倍數則身分證字號即屬正確。請根據以上說明，寫一程式，驗證身份證號碼是否正確。

例如，『A157387560』將會轉為『10157387560』，然後權重計算如下：

1	0	1	5	7	3	8	7	5	6	0
1	9	8	7	6	5	4	3	2	1	1
1	0	8	35	42	15	32	21	10	6	0

$sum=1+0+8+35+42+15+32+21+10+6+0=170$ ，所以此身份證號碼是正確的。

範例14c

猜單字遊戲。

本例我寫一個猜單字遊戲，遊戲過程如下：

1. 輸出底線的個數提示單字的字母數，例如：單字是 apple，那就輸出 5 個底線『_ _ _ _ _』。
2. 使用者每次僅能輸入一個 a 到 z 的字母。本例假設輸入『p』。
3. 電腦將單字中合乎此字母的通通現形，所以本例將會輸出『_ p _ _ _』
4. 重複步驟 2 與 3，直到全部字母猜對為止。

輸出結果

```

_ _ _ _ _
請輸入一個字母：p
_ p _ _ _
請輸入一個字母：a
a p _ _ _
請輸入一個字母：l
a p p l _
請輸入一個字母：e
a p p l e
BinGo

```

程式列印

```

a='apple'
s=len(a)
b=[]
c=[]
for i in range(s) :
    b.append(a[i])
    c.append('_')
for i in range(s) :
    print('%c '% c[i],end='')
s1=0
while (s1<s) :
    d=input('請輸入一個字母：')
    for i in range(s):
        if d==b[i]:
            c[i]=d
            s1=s1+1
    for i in range(s) :
        print('%c '% c[i],end='')
print()
print("BinGo")

```

自我練習

以上範例僅用一個單字，請擴大此題目，找來 50 個單字，每次以亂數出現一個單字。

範例14d

英文默寫。英文要能進步，就是要強迫自己能將看過的句子覆誦一次，您想想看，中文別人講一次，我們通常可以覆誦一次，其實英文也要這樣自我要求。以下是我寫的程式，只要將一篇文章先輸入或複製貼上，程式就會逐一出現每一個句子、每個句子的第一個單字、並將該句剩餘的單字以底線『_』標示，讓使用者填空輸入，輸入完電腦還能自動批改。

演算法

1. 將整篇文章複製，以字串儲存。
2. 將文章以點(.)分解為若干句子。例如，3個句字有3個點，那分解後有4句，最後一句是空白，所以本例程式撰寫就減1。
3. 本例將每一個句子的第1個單字輸出，且每一個單字以底線『_』標示，這樣使用者比較好作答，電腦也比較方便批改。
4. 逐一由使用者輸入整個句子。
5. 本例假設使用者所輸入單字數與題目一樣。例如：題目提示5個單字，使用者就輸入5個單字。
6. 逐一按順序比對兩個句子的單字，並統計正確單字的個數，也就是第1個單字與第1個單字比對，第2個單字與第2個單字比對。
7. 若全部單字都對，輸出『全對』，否則輸出正確單字的個數。
8. 輸出正確的句子。

輸出結果

```

Action _____
請輸入句子0: Action speak louder than words.
全對
正確句子是Action speak louder than words.
Wasting _____
請輸入句子1: Wasting time is rob onself.
錯了2 個單字
正確句子是Wasting time is robbing oneself.
Never _____
請輸入句子2: Never say di.
錯了1 個單字
正確句子是Never say die.

```

程式列印

```

a="Action speak louder than words.Wasting time is robbing
  oneself.Never say die."
b=a.split('.')#整篇文章分解
#print(len(b))
#print(b)#由輸出發現，b的長度是4，所以下就減1
for i in range(len(b)-1):
    c=b[i]+'.'
    #每個句字再以空白分解
    d=c.split(' ')
    dlen=len(d)
    #print(dlen)
    print(d[0],end=' ')
    for j in range(1,dlen) :
        print('_____',end=' ')
    #print('.')
    e=input("請輸入句子%d: "%i)
    f=e.split(' ')
    #print(len(f))
    s=0
    #計算答對幾個單字
    for j in range(dlen):
        if (d[j]==f[j]):
            s=s+1
    if s==dlen :

```

```

    print("全對")
else:
    print("錯了%d 單字" %(dlen-s))
print("正確句子是%s"%c)

```

自我練習

1. 本例假設使用者輸入的單字數與提示的單字數一致，若使用者輸入太長或太短都會造成執行錯誤，這就由您再加強了。想當程式設計師，就要靠自己。
2. 同範例，但改爲通通輸出，這樣就可讓使用者練習英打，並統計使用者 5 分鐘可完成的單字數。

範例14e

英文聽寫。

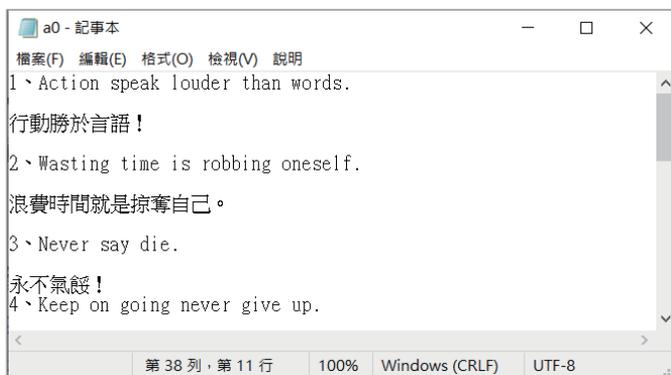
我們於第一單元曾經強調，Python 之所以強，就是有源源不絕的好用套件不斷發表，上網搜尋『Python 文字轉語音』，找到套件如下 (<https://codertw.com/%E7%A8%8B%E5%BC%8F%E8%A%9E%E8%A8%80/368251/>)：

```

import win32com.client
speaker = win32com.client.Dispatch("SAPI.SpVoice")
a="Hello, it works!"
speaker.Speak(a)
a="洪國勝，您好"
speaker.Speak(a)

```

真的可以發音，也就是視障者，可以將任何中英文文章複製，就可以自動播放。或是，您也可將英文課本複製下來，貼到變數 a，那電腦就會用語音輸出，讓您不斷練習聽力。以下是將文章，先放在記事本，以 a0.txt 存檔（請與 Python 程式放在同一資料夾），如下圖，然後寫程式幫您播放。



撰寫以下程式，就可讀檔，且整篇發音。

```
import win32com.client
speaker = win32com.client.Dispatch("SAPI.SpVoice")
f='a0.txt'
open(f,'r',encoding='utf-8') as f1:
    data=f1.read()
f1.close()
print(data)
speaker.Speak(data)
```

其次，根據以上程式，我將以上程式改寫如下，就可完成英文聽寫程式，希望可以幫助大家學習英文。

輸出結果

將英文課文複製，然後貼到以下變數 a，電腦就會逐一唸出每一句，且電腦提示每一句的長度，使用者就可開始輸入聽到的單字，然後電腦還會檢查這些單字是否正確。

```

請輸入句子0: Action speak louder than words.
全對
正確句子是Action speak louder than words.

請輸入句子1: Wasting time is rob oneself.
錯了1 個單字
正確句子是Wasting time is robbing oneself.

請輸入句子2: Never aa bb.
錯了2 個單字
正確句子是Never say die.

```

程式列印

```

import win32com.client
speaker = win32com.client.Dispatch("SAPI.SpVoice")
speaker.Speak("Hello, 各位同學大家好, 現在開始聽寫測驗")
#將課文複製, 然後貼來以下變數a, 電腦就會逐一唸出每一句。
a="Action speak louder than words.Wasting time is robbing
  oneself.Never say die."
b=a.split('.')#整篇文章分解
#print(len(b))
#print(b)#由輸出發現, b的長度是4, 所以下就減1
for i in range(len(b)-1):
    c=b[i]+'.'
    speaker.Speak(c)#您要念幾遍, 就寫幾次
    speaker.Speak(c)
    #每個句字再以空白分解
    d=c.split(' ')
    dlen=len(d)
    #print(dlen)
    #print(d[0],end=' ')
    for j in range(dlen) :
        print('____',end=' ')
    #print('.')
    e=input("請輸入句子%d: "%i)
    f=e.split(' ')
    #print(len(f))
    s=0
    #計算答對幾個單字

```

```
for j in range(dlen):
    if (d[j]==f[j]):
        s=s+1
if s==dlen :
    print("全對")
else:
    print("錯了%d 個單字" %(dlen-s))
print("正確句子是%s"%c)
```

自我練習

1. 範例 14e 有一個讀檔程式，請將以上程式套用範例 14e 讀檔程式，這樣就可以將程式與資料分開，每次只要改變資料就可以，不用改程式。