

Q : Python-RUN-2

基本上第二回合是第一回合的重複，只不過在增加了一些程式指令，並且做更多的應用。在程式的語法上新增加的指令如下：

1. 算術運算子：求商//；求餘數%；次方**
2. 科學記號: 3E8, 1.6E-19
3. 註解: 用井字號(#), 或''' '''
4. 判斷式：if ... elif ... else
5. : while迴圈
6. for迴圈+else

Q1: Python總共有7個算術運算子

1	加法	+	3.29 + 5	value1 + 100	value1 + value2 + 1.23e5
2	減法	-	3.29 - 5	value1 - 100	value1 - value2 - 1.23e5
3	乘法	*	3.29 * 5	value1 * 100	value1 * value2 * 1.23e5
4	除法	/	3.29 / 5	value1 / 100	value1 / value2 / 1.23e5
5	求商	//	3.29 //5	value1 // 100	value1 // value2 // 1.23e5
6	求餘數	%	3.29%5	value1 % 100	value1 % value2 % 1.23e5
7	次方	**	3.29 **2	value1 ** 0.5	value1 ** value2 ** 3

指定運算子(=)的優先順序最低，因此會先完成等號右邊的算術運算式，最後再執行指定運算子。依照算術運算規則，乘和除會先計算，而且從左邊開始，「//」和「%」這二個運算子的優先順序和乘除相同。至於次方運算子「**」，它的優先順序最高，而且是從右邊開始運算。如下之 $3 ** 2$ 要最先處理：

$$5 + 6 * 4 ** 3 ** 2$$

$$5 + 6 \times 4^{3^2}$$

圓括弧強制先執行某一部分的運算式。

註解: 用井字號(#), 或''' '''

```
1 a=3.  
2 b=5  
3 c=a+b  
4 s='This is YC Chen'  
5 print ('type of a=', type(a))  
6 print ('type of b=', type(b))  
7 print ('type of c=', type(c))  
8 print ('type of s=', type(s))  
9 #print ('a,b,c=', a,b,c)  
10 #print (s, 'Hello, 這是第一個PYTHON程式')
```

```
type of a= <class 'float'>  
type of b= <class 'int'>  
type of c= <class 'float'>  
type of s= <class 'str'>  
>>> □
```

```
1 '''這3行是註解  
2 Learning Python is fun.  
3 I hope you enjoy it.'''  
4 print ('Welcome to Tunghai Physics.')
```

```
Welcome to Tunghai Physics.  
>>> □
```

利用type打出變數的資料型態

Python有五個標準的數據類型：
Numbers（數字）
String（字符串）
List（列表）
Tuple（元組）
Dictionary（字典）

```
1 L=[1,2,56.]  
2 t='台中', 400  
3 s='Tunghai'  
4 print (type(L))  
5 print (type(t))  
6 print (type(s))
```

```
<class 'list'>  
<class 'tuple'>  
<class 'str'>  
>>> □
```

Q2: if ... elif ... else

if 關係運算式1:

如果關係運算式1為True，
就會執行這段程式碼
這段程式碼必須內縮至少
1個空格，建議內縮4個空格

...

elif 關係運算式2:

如果關係運算式2為True，
就會執行這段程式碼
這段程式碼必須內縮至少
1個空格，建議內縮4個空格

...

. # 其他elif程式碼

else:

如果前面的關係運算式全部
都是False就會執行這段程式碼
這段程式碼必須內縮至少
1個空格，建議內縮4個空格

...

```
          整數
          ↓
1  x=int(input('輸入年齡(1~24):'))
2  print ('x=',x)
3  if(x <= 3): → 整數比較大小
4      print ('托兒所')
5  elif(3 < x < 6):
6      print ('幼稚園')
7  elif(6 <= x < 12):
8      print ('小學')
9  elif(12 <= x < 15):
10     print ('國中')
11  elif(15 <= x < 18):
12     print ('高中')
13  else:
14     print ('大學')
```

```
          字符
          ↓
輸入年齡(1~24):14
x= 14
國中
>>>
```

在If判斷式中，每一個關係運算式的下一行開始，到下一個關係運算式之前，稱為一個「**程式碼區塊**」。如果關係運算式的結果是True，它裏頭的程式碼區塊就會被執行，而且執行完該程式碼區塊之後，就會離開If判斷式，不會再檢查其他關係運算式。

If判斷式是由if區塊、elif區塊和else區塊組成，而且必須符合下列條件：

- 一定要有一個if區塊，而且只有一個。
- elif區塊可以沒有，也可以有一個或多個。
- else區塊可以沒有，也可以有一個，不可以多個。

if、elif和else中的**程式碼區塊**必須內縮（稱為Indent），因為Python語言是利用內縮來決定哪些程式碼是屬於同一個區塊。

◆ 邏輯判斷與關係運算子

```
1 a=3
2 b=5
3 print ('a,b=',a,b)
4 if(a == b): print ('a=b')
5 if(a != b): print ('a!=b')
6 if(a > b): print ('a > b')
7 if(a < b): print ('a < b')
8 |
```

```
a,b= 3 5
a!=b
a < b
>>> □
```

if 關係運算式:
運算式A
else:
運算式B

```
1 x=500.
2 print ('wavelength x=',x,' nm ')
3 if(400. < x < 700.):
4     print ('visible light')
5 else:
6     print ('invisible')
7
```

```
wavelength x= 500.0 nm
visible light
>>> □
```

Q3: while迴圈+邏輯判斷

```
1 n=0
2 while n < 5:
3     n += 1
4     print (n)
```

```
1
2
3
4
5
>>>
```

```
1 n=0
2 while True:
3     if(n > 5): break
4     n += 1
5     print (n)
```

```
1
2
3
4
5
6
>>>
```


Q4: For迴圈+邏輯判斷

巧妙地運用邏輯判斷
和迴圈，算出兩個整
數之間的所有質數

一個整數除了1之外，
沒有任何比他小的
整數可以整除這個
整數，滿足此條件
的整數就是質數。

```
1 lower = 900
2 upper = 1000
3 print("在",lower,"與",upper,"之間的質數:")
4 for num in range(lower,upper + 1):
5     K=0
6     for i in range(2,num):
7         if (num % i) == 0:
8             K=1
9             break
10    if(K==0): print(num)
```

```
在 900 與 1000 之間的質數：
907
911
919
929
937
941
947
953
967
971
977
983
991
997
>>> □
```

Q5: For迴圈+邏輯判斷，example-1

閏年就是公元年可以被4整除但不能被100整除，或者公元年可以被400整除。

```
1 while True:
2     year = int(input("Enter a year:(Enter 0 to stop) "))
3     if(year == 0): break
4     if (year % 4) == 0:
5         if (year % 100) == 0:
6             if (year % 400) == 0:
7                 print (year, '是閏年(2)')
8             else:
9                 print (year, '是平年(2)')
10        else:
11            print (year, '是閏年(1)')
12    else:
13        print (year, '是平年(1)')
```

語法重點while迴圈

用input輸入

三層嵌套的if ... else

```
Enter a year:(Enter 0 to stop) 1991
1991 是平年(1)
Enter a year:(Enter 0 to stop) 1996
1996 是閏年(1)
Enter a year:(Enter 0 to stop) 1900
1900 是平年(2)
Enter a year:(Enter 0 to stop) 2000
2000 是閏年(2)
Enter a year:(Enter 0 to stop) 0
>>>
```

Q6: 邏輯判斷if...elif...else，example-2

從外部輸入三個整數，使用if ... elif ... else來查找三個中最大的一個。

```
1 num1 = int(input("Enter num1 "))
2 num2 = int(input("Enter num2 "))
3 num3 = int(input("Enter num3 "))
4 print ('3 integers:', num1,num2,num3)
5 if (num1 >= num2) and (num1 >= num3):
6     largest = num1
7     print ('case 1: num1 is the largest')
8 elif (num2 >= num1) and (num2 >= num3):
9     largest = num2
10    print ('case 2: num2 is the largest')
11 else:
12    largest = num3
13    print ('case 3: num3 is the largest')
14 print (num1,num2,num3, ' The largest number=',largest)
```

```
Enter num1 4
Enter num2 34
Enter num3 109
3 integers: 4 34 109
case 3: num3 is the largest
4 34 109 The largest number= 109
>>>
```

Q7: For迴圈+else，example-3

for 自訂變數 in range(起始值, 結束值, 遞增值):

處理自訂變數中的資料

...

if 關係運算式:

 continue # 跳過以下程式碼，

立刻進入range數列的下一項

...

...

if 關係運算式:

 break # 離開迴圈

...

else:

當沒有用break指令跳出迴圈時

，會執行這個程式區塊

```
1 import random as rand # 載入亂數套件
2 for i in range(0, 5, 1): # 設定執行5次
3     # 產生一個介於1和100 (含)之間的亂數
4     rand_num = rand.randint(1, 100)
5     # 根據亂數決定檢測結果
6     if rand_num < 90:
7         print(i, rand_num, "通過", i+1, "次檢測")
8     else:
9         print(i, rand_num, "檢測失敗")
10        break
11 else:
12     # 當沒有用break指令跳出迴圈時執行
13     print("通過品管檢測")
```

```
0 5 通過 1 次檢測
1 78 通過 2 次檢測
2 2 通過 3 次檢測
3 53 通過 4 次檢測
4 47 通過 5 次檢測
通過品管檢測
>>> █
```

for迴圈的程式碼會處理自訂變數中的值，我們可以在迴圈中利用If判斷式檢查執行的情況。需要的話，可以利用continue指令跳過後面的程式碼，進入range數列的下一項，或是利用break指令直接離開迴圈。else區塊中的程式碼，只有在迴圈正常結束(執行到range數列的最後一項)的情況下才會執行。如果用break指令跳出迴圈，就不會執行。

Q8 : For迴圈+else , example-4

算出兩個整數之間的所有質數

```
1 lower=900
2 upper=1000
3 for num in range(lower,upper+1):
4     K=0
5     for i in range(2,num):
6         if(num%i == 0):
7             K=1
8             break
9     if(K == 0): print (num, '是質數')
```

```
907 是質數
911 是質數
919 是質數
929 是質數
937 是質數
941 是質數
947 是質數
953 是質數
967 是質數
971 是質數
977 是質數
983 是質數
991 是質數
997 是質數
>>>
```

```
1 lower=900
2 upper=1000
3 for num in range(lower,upper+1):
4     for i in range(2,num):
5         if(num%i == 0): break
6     else:
7         print (num, '是質數')
```

else區塊中的程式碼，只有在迴圈正常結束(執行到range數列的最後一項)的情況下才會執行。

Q9 : For迴圈+else , example-5

for迴圈 + 邏輯判斷(if ... else) + 求餘數(%)。縮排。

```
1 SUM=0
2 for i in range(1,11):
3     if(i%2==0):
4         n=i/2
5         print (i,'是偶數可被2整除, i/2=',n)
6     else:
7         SUM=SUM+i
8         print (i,'是奇數, 求和, SUM=',SUM)
9
```

```
1 是奇數, 求和, SUM= 1
2 是偶數可被2整除, i/2= 1.0
3 是奇數, 求和, SUM= 4
4 是偶數可被2整除, i/2= 2.0
5 是奇數, 求和, SUM= 9
6 是偶數可被2整除, i/2= 3.0
7 是奇數, 求和, SUM= 16
8 是偶數可被2整除, i/2= 4.0
9 是奇數, 求和, SUM= 25
10 是偶數可被2整除, i/2= 5.0
>>> 
```

Q10 : For迴圈流程控制

在迴圈的流程當中我們會經常使用到一些流程控制的指令，包括

- ◆ break
- ◆ Continue
- ◆ pass

```
1 for i in range(8):
2     print(i,end=' ')
3     if(i > 2): break
4     print (i,' i<2 non-break',i**2)
5 print ('.....')
6 for i in range(8):
7     print(i,end='/')
8     if(i > 2): continue
9     print (i,' i的平方=',i**2)
10 print ('-----')
11 for i in range(8):
12     print ('i =',i, end='//')
13     if(i < 3):
14         print ('P'+str(i))
15     else:
16         pass
```

```
0 0 i<2 non-break 0
1 1 i<2 non-break 1
2 2 i<2 non-break 4
3
.....
0/0 i的平方= 0
1/1 i的平方= 1
2/2 i的平方= 4
3/4/5/6/7/-----
i = 0//P0
i = 1//P1
i = 2//P2
i = 3//i = 4//i = 5//i = 6//i = 7//>>> □
```

EX2-1 : 請編寫一個PYTHON程式可以實現下面列印在螢幕上的結果。在螢幕上打出五個'-'號可以用下面這個指令:

print ('-'*5)

```
*
-----
**
-----
***
-----
****
-----
*****
-----
>>>
```

```
*
**
***
****
*****
-----
>>>
```

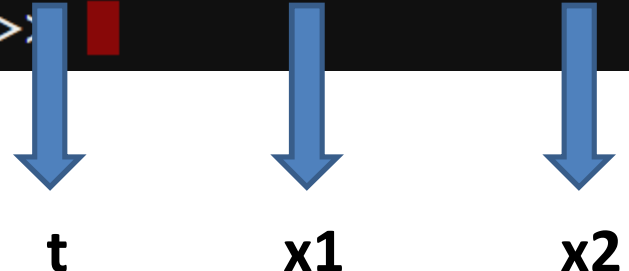
```
*
**
***
****
*****
-----
****
***
**
*
>>>
```

```
*
-----
**
-----
*
**
***
-----
*
**
***
****
-----
>>>
```


EX2-2：請編寫一個PYTHON程式可以計算兩個物體運動的座標隨時間增加的數據。x1是v=10的等速度運動，x2是初速度v0=0，加速度a=10的等加速度運動。以時間間隔dt=0.1追蹤他們的位置，直到x2追到x1。t=0，x1=5, x2=0。

0.0	2.000	0.000
0.1	2.300	0.050
0.2	2.900	0.250
0.3	3.800	0.700
0.4	5.000	1.500
0.5	6.500	2.750
0.6	8.300	4.550
0.7	10.400	7.000
0.8	12.800	10.200
0.9	15.500	14.250

>>>



t x1 x2

EX2-3 : 請執行下面的程式並且從輸出的結果來理解這個程式是如何將數據儲存在列表當中，再如何從列表當中取出數據來做比較。

```
a=[90,80,67,45,71,52,77,82,88,66] #國
b=[30,78,67,55,51,52,47,82,98,35] #數
c=[82,85,72,75,69,52,88,92,95,91] #英
N=len(a)
for i in range(N):
    print (i+1,a[i],b[i],c[i])
    if(b[i] > a[i]): print ('數學較優')
```

```
1 90 30 82
2 80 78 85
3 67 67 72
4 45 55 75
數學較優
5 71 51 69
6 52 52 52
7 77 47 88
8 82 82 92
9 88 98 95
數學較優
10 66 35 91
```

請修改這個程式先判斷是否三科都及格，再決定這個同學最優的科目，其輸出結果應如下所示。

```
2 號3科都及格: 80 78 85-->英文最優
3 號3科都及格: 67 67 72-->英文最優
8 號3科都及格: 82 82 92-->英文最優
9 號3科都及格: 88 98 95-->數學最優
>>>
```

EX2-4 :請寫出一個可以計算二維拋體運動的座標的程式。這個程式將能夠計算拋體的x座標和y座標隨時間的變化，在拋體落地時停止計算，並且輸出落地的時間和落地的座標如下所示。

在這個程式設計當中你必須用到數學模組 (import math),將角度換成弧度(radians)並且知道拋體運動的解:

$$x(t) = v_0 \cos(\theta)t$$

$$y(t) = v_0 \sin(\theta)t - \frac{1}{2}gt^2$$

<i>t</i>	<i>x</i>	<i>y</i>
0.00	0.000	0.000
0.20	2.000	3.268
0.40	4.000	6.144
0.60	6.000	8.628
0.80	8.000	10.720
1.00	10.000	12.421
1.20	12.000	13.729
1.40	14.000	14.645
1.60	16.000	15.169
1.80	18.000	15.301
2.00	20.000	15.041
2.20	22.000	14.389
2.40	24.000	13.345
2.60	26.000	11.909
2.80	28.000	10.081
3.00	30.000	7.862
3.20	32.000	5.250
3.40	34.000	2.246

>>>