

**Range, Conditional statements,
for looping and score grading**

Python的range()其實是Python內建的range()型別的建構式(constructor)，用來表示不可變(immutable)的數字序列

```
r = range(5)
print(r)
print(list(r))
```

```
range(0, 5)
[0, 1, 2, 3, 4]
```

```
r = range(0,5)
print(r)
print(list(r))
```

從0到5-1，每次
增加1

已轉換為
串列

```
range(0, 5)
[0, 1, 2, 3, 4]
```

數字序列

等差級數遞增

```
r = range(0,10,3)  
print(r)  
print(list(r))
```

從0到10-1，每
次增加3

```
range(0, 10, 3)  
[0, 3, 6, 9]
```

等差級數遞減

```
r = range(-10,-100,-30)
print(r)
print(list(r))
```

從-10
到-100+1，每
次-30

```
range(-10, -100, -30)
[-10, -40, -70]
```

```
list(range(-10,-15,-1))
[-10, -11, -12, -13, -14]
```

從-10到-15+1，
每次-1

```
a = ['Mary', 'had', 'a', 'little', 'lamb']  
for i in range(len(a)):  
    print(i,a[i])
```

```
0 Mary  
1 had  
2 a  
3 little  
4 lamb
```

以串列
長度設
定for圈
的執行
範圍

以字串構成
的串列

if elif elif else

```
x = int(input("Please enter an integer: "))
if x < 0:
    x = 0
    print('Negative changed to zero')
elif x == 0:
    print('Zero')
elif x == 1:
    print('Single')
else:
    print('More')
```

如果 $x < 0$
執行的命令

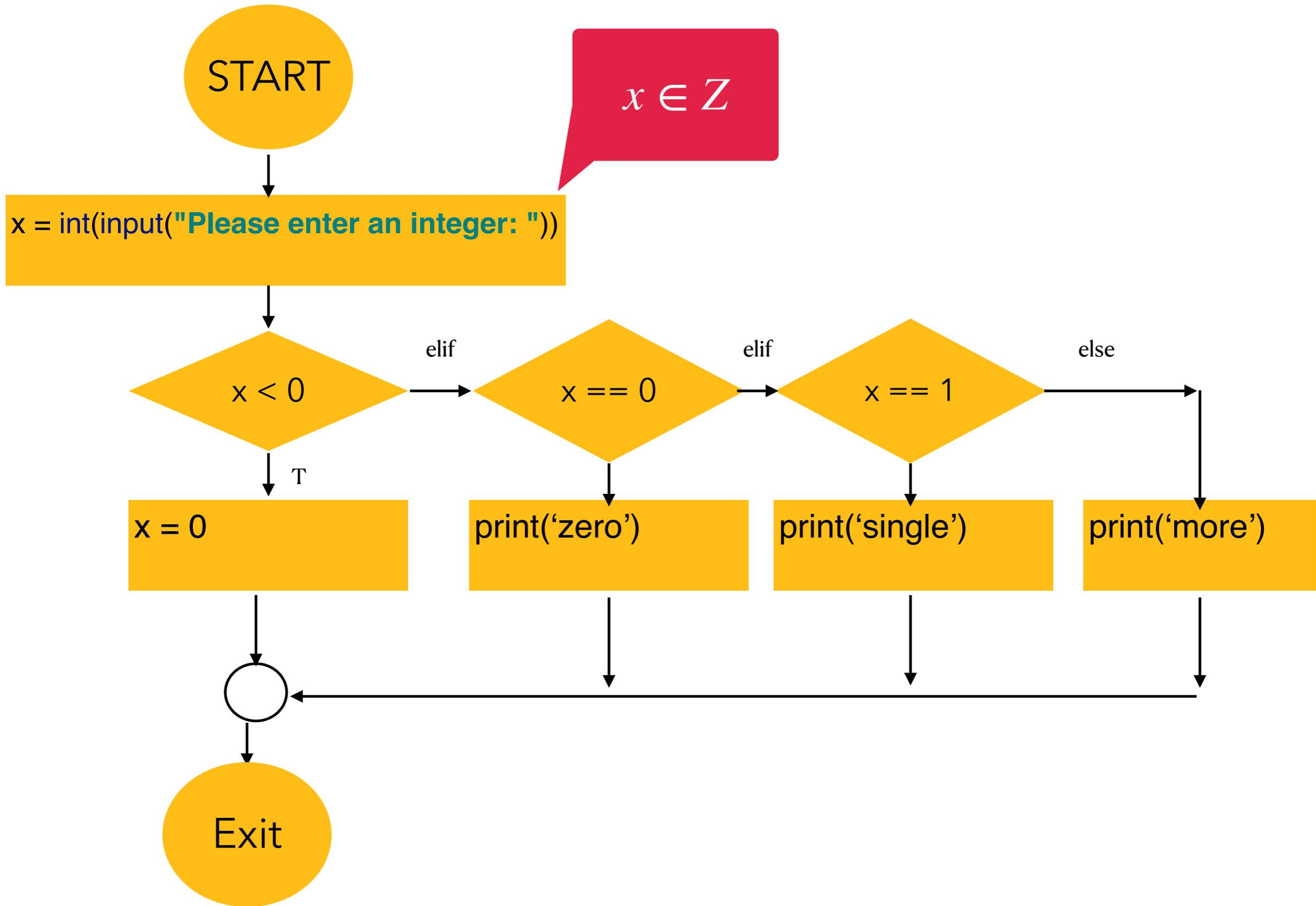
否則，如果 $x == 0$
執行的命令

否則，如果 $x == 1$
執行的命令

否則，執行的命令

```
Please enter an integer: -3
Negartive changed to zero
```

```
Please enter an integer: 1
Single
```



在集合a中增加
一個元素

```
p = [1, 2, 2]
a = set(p)
print(len(a))
a.add(4)
print(a)
```

集合a的元
素個數

```
2
{1, 2, 4}
```

```
words = ['cat', 'window', 'defenestrate']  
for w in words:  
    print(w, len(w))
```

依序將串列words中的元素代入w，執行迴圈命令

```
cat 3  
window 6  
defenestrate 12
```

len(w)代表w的長度，也就是字元個數

```
words = ['cat', 'window', 'defenestrate']
n = len(words)
for i in range(n):
    w = words[i]
    if len(w) > 6:
        words.append(w)
print(words)
```

使用串列的insert方法

將w加入串列words中，w在串列words中的索引值為1

```
['cat', 'defenestrate', 'window', 'defenestrate']
```

```
for i in range(5):  
    print(i)
```

```
0  
1  
2  
3  
4
```

從0到5-1依序代入i，
執行迴圈命令

範例：分數等地

隨機產生N個學生的分數，

95-100分為'A+'

90-94分為'A'

85-89為'A-'

80-84為'B+'

75-79為B

其他為'B-'

範例：隨機產生分數，依規定評定grade

步驟：

1. 匯入套件
2. 設定grades為空串列；設定N，隨機產生N個介於60到100間分數，並儲存在串列scores中
3. for score in scores:
 使用if elif elif ... elif else: 設定score的grade
 並儲存在串列grades中
4. 輸出分數與grade

```

import random
S = list(range(60,101))
N = 20
scores = []
for i in range(N):
    s = random.choice(S)
    scores.append(s)

grade = []
for score in scores:
    if score >= 95:
        g = 'A+'
    elif score >= 90:
        g = 'A'
    elif score >= 85:
        g = 'A-'
    elif score >= 80:
        g = 'B+'
    elif score >= 75:
        g = 'B'
    else:
        g = 'B-'
    grade.append(g)
for i in range(len(grade)):
    print(scores[i],grade[i])

```

68	B-
85	A-
69	B-
75	B
88	A-
83	B+
81	B+
75	B
98	A+
65	B-
96	A+
90	A
98	A+
83	B+
79	B
67	B-
86	A-
74	B-
88	A-
65	B-

95-100	分為 'A+'
90-94	分為 'A'
85-89	為 'A-'
80-84	為 'B+'
75-79	為 B
其他	為 'B-'

```
import random
S = list(range(60,101))
N = 20
scores = []
for i in range(N):
    s = random.choice(S)
    scores.append(s)

grade = []
for score in scores:
    if score >= 95:
        g = 'A+'
    elif score >= 90:
        g = 'A'
    elif score >= 85:
        g = 'A-'
    elif score >= 80:
        g = 'B+'
    elif score >= 75:
        g = 'B'
    else:
        g = 'B-'
    grade.append(g)
for i in range(len(grade)):
    print(scores[i],grade[i])
```

匯入套件random

串列S的元素包含
60-100

使用套件random的方法，
從串列S中隨機選擇一個元
素

```
import random
S = list(range(60,101))
N = 20
scores = []
for i in range(N):
    s = random.choice(S)
    scores.append(s)

grade = []
for score in scores:
    if score >= 95:
        g = 'A+'
    elif score >= 90:
        g = 'A'
    elif score >= 85:
        g = 'A-'
    elif score >= 80:
        g = 'B+'
    elif score >= 75:
        g = 'B'
    else:
        g = 'B-'
    grade.append(g)
for i in range(len(grade)):
    print(scores[i],grade[i])
```

執行迴圈命令N次

將隨機產生的分數附加在串
列scores中

```
import random
S = list(range(60,101))
N = 20
scores = []
for i in range(N):
    s = random.choice(S)
    scores.append(s)
```

```
grade = []
for score in scores:
```

```
    if score >= 95:
        g = 'A+'
    elif score >= 90:
        g = 'A'
    elif score >= 85:
        g = 'A-'
    elif score >= 80:
        g = 'B+'
    elif score >= 75:
        g = 'B'
    else:
        g = 'B-'
    grade.append(g)
```

```
for i in range(len(grade)):
    print(scores[i],grade[i])
```

依序將串列scores中的元素代
入score，執行迴圈命令

如果分數大於等於95分，為'A+'
否則如果，分數大於等於90分，為'A'
否則如果，分數大於等於85，為'A-'
否則如果，分數大於等於80，為'B+'
否則如果，分數大於等於75，為B
其他為'B-'

將g附加在串列grade中

```
>>> for num in range(2, 10):
...     if num % 2 == 0:
...         print("Found an even number", num)
...         continue
...     print("Found a number", num)
Found an even number 2
```

```
Found a number 3
Found an even number 4
Found a number 5
Found an even number 6
Found a number 7
Found an even number 8
Found a number 9
```

執行到continue時
跳過迴圈中continue
之後的指令

繼續執行迴圈指令

範例：質數

從1-999999中隨機產生一個正整數，如果不是質數，印出一個最小質因數與其他數相乘的式子，否則印出'**PRIME**'

正整數 n 是不是質數？

1. 從2到 $\text{int}(n/2)$ ，依序檢查是不是 n 的因數，如果遇到是 n 的因數，印出訊息，迴圈結束
2. 如果從2到 $\text{int}(n/2)$ 的正整數都不是 n 的因數， n 就是質數

使用 for else 命令，實作步驟1與2

```
import random
S = list(range(1,100000))
n = random.choice(S)
for x in range(2,int(n/2)):
    if n % x == 0:
        print(n, 'equals', x, '*', n//x)
        break
    else:
        print(n, 'is a prime number')
```

執行for迴圈命令，當執行到break時，迴圈會中斷執行，跳開for迴圈，也不執行else:中的指令

如果執行for迴圈命令時，都沒有執行到break指令，迴圈執行結束後，會執行 else: 中的指令，印出質數訊息

```
import random
S = list(range(1,100000))
n = random.choice(S)
for x in range(2,int(n/2)):
    if n % x == 0:
        print(n, 'equals', x, '*', n//x)
        break
    else:
        print(n, 'is a prime number')
```

匯入套件random
將屬於範圍中的整數，1到
99999，儲存在串列S中
從串列S中隨機選擇整數n

```
import random
S = list(range(1,100000))
n = random.choice(S)
for x in range(2,int(n/2)):
    if n % x == 0:
        print(n, 'equals', x, '*', n//x)
        break
else:
    print(n, 'is a prime number')
```

執行for迴圈命令，依序從2到int(n/2)代入x

如果執行for迴圈命令時，都沒有執行到Break指令，迴圈執行結束後，會執行 else: 中的指令，印出質數訊息

```
import random
S = list(range(1,100000))
n = random.choice(S)
for x in range(2,int(n/2)):
    if n % x == 0:
        print(n, 'equals', x, '*', n//x)
        break
else:
    print(n, 'is a prime number')

```

執行for迴圈命令，依序從2到n-1代入x

如果整數n除以x的餘數為0
列印分解式
執行break指令，中斷迴圈執行

```
import random
S = list(range(1, 100000))
n = random.choice(S)
for x in range(2, int(n/2)):
    if n % x == 0:
        print(n, 'equals', x, '*', n//x)
        break
else:
    print(n, 'is a prime number')
```

40499 is a prime number

56221 equals 11 * 5111

36143 equals 47 * 769

```
import random
import numpy as np
S = list(range(1,100000))
n = random.choice(S)
squareRoot = int(np.sqrt(n))
print('floor of squareRoot :', squareRoot)
for x in range(2,squareRoot):
    if n % x == 0:
        print(n, 'equals', x, '*', n//x)
        break
else:
    print(n, 'is a prime number')
```

練習：上述的程式已經可以找到一個最小的質因數，想一想，如何找出一個正整數的所有質因數呢？

$$n = a_1 \times n_1$$

如果 a_1 代表第一個找到的 n 的質因數， a_2 代表繼續從 n_1 找到第一個的質因數，

$$n_1 = a_2 \times n_2$$

a_2 就是 n 的第二個質因數

$$n = a_1 \times n_1$$

$$n_1 = a_2 \times n_2$$

•

•

$$n_i = a_{i+1} \times n_{i+1}$$

•

•

$$n = a_1 \times n_1$$

$$210 = 2 \times 105$$

$$n_1 = a_2 \times n_2$$

$$105 = 3 \times 35$$

•

•

•

•

$$n_i = a_{i+1} \times n_{i+1}$$

$$35 = 5 \times 7$$

•

•

•

•

步驟

方法論

1. 輸入 n ，設定 $prime$ 為空串列
2. 使用上述方法，找到 n 的第一個質因數 a ，並將 n 分解為 $a * m$
3. 將 n 設定為 m ，如果 a 不在串列 $prime$ 中，將 a 附加在串列 $prime$ 中
4. 如果停止條件成立，停止，否則跳到步驟2執行

表格計算

最小質因數

變數名稱	n	a	m	prime
初始狀態	210			空串列
迴圈一	210	2	105	
迴圈二				
迴圈三				
迴圈四				

如何設定
停止條件
呢？

步驟

1. 輸入 n ，設定 $prime$ 為空串列
2. 使用上述方法，找到 n 的第一個質因數 a ，並將 n 分解為 $a * m$
3. 將 n 設定為 m ，如果 a 不在串列 $prime$ 中，將 a 附加在串列 $prime$ 中
4. 如果停止條件成立，停止，否則跳到步驟2執行

變數名稱	n	a	m	$prime$
初始狀態	210			空串列
迴圈一	210	2	105	2
迴圈二	105	3	35	2,3
迴圈三	35	5	7	2,3,5
迴圈四	7	7	1	2,3,5,7

步驟

1. 輸入 n ，設定 $prime$ 為空串列
2. 使用上述方法，找到 n 的第一個質因數 a ，並將 n 分解為 $a * m$
3. 將 n 設定為 m ，如果 a 不在串列 $prime$ 中，將 a 附加在串列 $prime$ 中
4. 如果停止條件成立，停止，否則跳到步驟2執行

變數名稱	n	a	m	$prime$
初始狀態	240			空串列
迴圈一	240	2	120	2
迴圈二	120	2	60	2
迴圈三	60	2	30	2
迴圈四	30	2	15	2
迴圈五	15	3	5	2,3
迴圈六	5	5	1	2,3,5
迴圈七	1			

**範例：使用for迴圈，每次從1-99999中隨機產生一個整數並印出該整數
產生10個數，求出最小整數**

```
import random
N = 100000
S = list(range(1,N))
min = N
for i in range(10):
    n = random.choice(S)
    print(n)
    if n < min:
        min = n
print('minimum is ', min)
```

```
71525
3155
449
4292
93110
57411
1609
71816
8451
13825
minimum is 449
```

```
import random
N = 100000
S = list(range(1,N))
min = N
for i in range(10):
    n = random.choice(S)
    print(n)
    if n < min:
        min = n
print('minimum is ', min)
```

串列S中的元素，
包含1-99999，都小於N

以min代表最小值，初始
值設定為N

```
import random
N = 100000
S = list(range(1,N))
min = N
for i in range(10):
    n = random.choice(S)
    print(n)
    if n < min:
        min = n
print('minimum is ', min)
```

串列S中的元素，
包含1-99999，都小於N

第一次產生的n，必定小
於N

if條件會成立
第一次產生的n會儲存在
min變數中

```
import random
N = 100000
S = list(range(1,N))
min = N
for i in range(10):
    n = random.choice(S)
    print(n)
    if n < min:
        min = n
print('minimum is ', min)
```

迴圈執行完畢後，min變數
中值代表十個整數的最小值

後續迴圈產生的n值，如果
小於min，會儲存在min變
數中

否則，min的內容不變

**練習：使用for迴圈，每次從1-99999中隨機產生一個整數並印出該整數
產生10個數，求出最大整數**

**練習：使用for迴圈，每次從1-99999中隨機產生一個整數並印出該整數
產生10個數，求出總和與平均值**

練習：使用for迴圈，每次從1-99999中隨機產生一個整數並印出該整數
產生10個數，求出偶數整數的總和與平均值

While looping Decimal to Binary

While迴圈的進入條件要如何設計呢？

```
import random
N = 100000
S = list(range(1,N))
n = random.choice(S)
print(bin(n))
print("converting",n," to")
b = ""
while n > 0:
    r = n % 2
    b = str(r) + b
    n = (n - r) // 2
print("0b"+b)
```

While迴圈和for迴圈有什麼不一樣呢？

While迴圈中的主體命令要如何設計呢？

為什麼需要使用while迴圈？難道不能只使用for迴圈，就好了呢？

```
0b10100100001000110
converting 84038 to
0b10100100001000110
```

```

while n > 0:
    r = n % 2
    b = str(r) + b
    n = (n - r) // 2

```

迴圈	進入條件 n>0	r	n	b
初始值			11	""
迴圈1	T	1	5	"1"
迴圈2				
迴圈3				
迴圈4				
迴圈5				