While loop, Decimal 2 Binary
Fabonacci series
Prime factor

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)
```

10.7

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？

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

為什麼需要使用while迴圈?
難道不能只使用for迴圈，就好了呢？

```
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	T	1	2	“11”
迴圈3	T	0	1	“011”
迴圈4	T	1	0	“1011”
迴圈5	F			

```
b = ""
n = 11
while n > 0:
    r =  n % 2
    b = str(r) + b
    n = (n - r) // 2
    print(n>0, " r:",r," n:",n," b:",b)
print("0b"+b)
```

```
True  r: 1  n: 5  b: 1
True  r: 1  n: 2  b: 11
True  r: 0  n: 1  b: 011
False  r: 1  n: 0  b: 1011
0b1011
```

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	T	1	2	“11”
迴圈3	T	0	1	“011”
迴圈4	T	1	0	“1011”
迴圈5	F			

```
b = ""
n = 23
while n > 0:
    r =  n % 2
    b = str(r) + b
    n = (n - r) // 2
    print(n>0, " r:",r," n:",n," b:",b)
print("0b"+b)
```

```
True  r: 1  n: 11  b: 1
True  r: 1  n: 5  b: 11
True  r: 1  n: 2  b: 111
True  r: 0  n: 1  b: 0111
False  r: 1  n: 0  b: 10111
0b10111
```

迴圈	進入條件 n>0	r	n	b
初始值			23	" "
迴圈1	T	1	11	"1"
迴圈2	T	1	5	"11"
迴圈3	T	1	2	"111"
迴圈4	T	0	1	"0111"
迴圈5	T	1	0	"10111"
迴圈6	F			

列印小於n的Fabonacci數列

Fabonacci數列

```
26
0 1 1 2 3 5 8 13 21
```

a_4 + a_5 = a_6

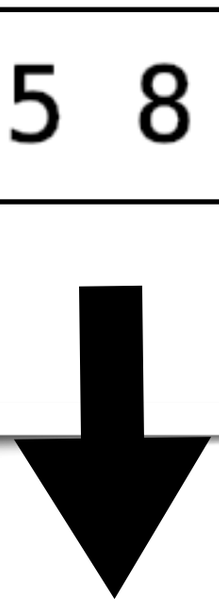
Fabonacci數列

```
26
0 1 1 2 3 5 8 13 21
```

n	a	b
4	3	5

Fabonacci數列

```
26
0 1 1 2 3 5 8 13 21
```



n	a	b
4	3	5
5	5	8

Fabonacci數列

```
26
0 1 1 2 3 5 8 13 21
```

n	a	b
4	3	5
5	5	8
6	8	13

如何在迴圈中更新a與b？

設定a, b的初始值

while 進入條件

print(a, end = ' ')
使用現有的a與b值
更新a與b值

```
import random
Dice = list(range(1,53))
n = random.choice(Dice)
print(n)
```

```
a, b = 0, 1
```

設定a, b的初始值

```
while a < n:
```

while

進入條件

```
print(a, end = ' ')
a, b = b, a+b
```

print(a, end = ‘ ’)
使用現有的a與b值
更新a與b值

Fabonacci數列

```
26
0 1 1 2 3 5 8 13 21
```

```
while a < n:
```

```
print(a, end = ' ')
a, b = b, a+b
```

n	a	b
4	3	5
5	5	8
6	8	13
7	13	21

如何在迴圈中更新a與b？

```
import random
Dice = list(range(1,53))
n = random.choice(Dice)
print(n)
a, b = 0, 1
while a < n:
    print(a, end = ' ')
    a, b = b, a+b
print()
```

```
23
0 1 1 2 3 5 8 13 21
```

產生小於n的Fabonacci數列，並儲存在串列中

```
n = 35
result = []
a, b = 0, 1
while a < n:
    result.append(a)
    a, b = b, a+b
print(result)
```

An empty array

Add a Fab number to array result

```
[0, 1, 1, 2, 3, 5, 8, 13, 21]
```

找n的所有質因數

找n的最小質因數

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

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

步驟

1. 輸入n，設定prime為空串列，max_p = 0
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, 且將max_p設為a
4. 如果停止條件成立，停止，否則跳到步驟2執行

goto

步驟

1. 輸入n，設定prime為空串列，max_p = 0
2. while n > max_p:
   A. 使用上述方法，找到n的第一個質因數a，並將n分解為$a * m$
   B. 將n設定為m，如果a不在串列prime中，將a附加在串列prime中, 且將max_p設為a

輸入n，設定prime為空串列，max_p = 1

```
import random
S = list(range(1,100000))
n = random.choice(S)
prime = []
max_p = 1
while n > max_p:
    for a in range(2,n):
        if n % a == 0:
            m = n//a
            print(n, 'equals', a, '*', m)
            break
    else:
        print(n, 'is a prime number')
        a = n
        m = 1
    n = m
    if not a in prime:
        prime.append(a)
        max_p = a
print(prime)
```

找n的最小質因數

```python
import random
S = list(range(1,100000))
n = random.choice(S)
prime = []
max_p = 1
while n > max_p:
    for a in range(2,n):
        if n % a == 0:
            m = n//a
            print(n, 'equals', a, '*', m)
            break
    else:
        print(n, 'is a prime number')
        a = n
        m = 1
    n = m
    if not a in prime:
        prime.append(a)
        max_p = a
print(prime)
```

找n的最小質因數，將n分解為 $a * m$

B. 將n設定為m，如果a不在串列prime中，將a附加在串列prime中, 且將max_p設為a

```python
import random
import numpy as np
S = list(range(1,100000))
n = random.choice(S)
prime = []
max_p = 1
while n > max_p:
    for x in range(2, int(np.ceil(n / 2))):
        if n % x == 0:
            print('n = ', n, ' a = ', x)
            a = x
            m = n // a
            break
    else:
        print('n is a prime ', n)
        a = n
        m = n // a

    n = m
    if not a in prime:
        prime.append(a)
        max_p = a
print(prime)
```

兩個陣列相加

陣列

```
import numpy as np
a = np.array([1,2])
print(a)
b = np.array([3,4])
print(b)
print(a+b)
```

```
[1 2]
[3 4]
[4 6]
```

串列

```
p=[1,2]
q=[3,4]
p+q
```

```
>>> p=[1,2]
>>> q=[3,4]
>>> p+q
[1, 2, 3, 4]
```

陣列size, max,min, mean

```
a = np.array([1,2, 3 ,5, 7])
print(a.size, a.max(),a.min(),a.mean())
```

```
>>> a = np.array([1,2, 3 ,5, 7])
... print(a.size,a.max(),a.min(),a.mean())
5 7 1 3.6
```

陣列元素加總

```
a = np.array([1,2, 3 ,5, 7])
print(np.sum(a))
```

```
18
```

陣列相乘

```
a = np.array([1,2, 3 ,5, 7])
print(np.multiply(a,a))
print(a*a)
```

```
>>> a = np.array([1,2, 3 ,5, 7])
... print(np.multiply(a,a))
... print(a*a)
[ 1  4  9 25 49]
[ 1  4  9 25 49]
```

二維陣列，矩陣

```
A = np.array([[1,2,3],[4,5,6],[7,8,9]])
print(A)
print(A.size)
```

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
9
```

```
A = np.array([[1,2,3],[4,5,6],[7,8,9]])
print(A)
print(np.transpose(A))
```

```
[[1 2 3]
 [4 5 6]
 [7 8 9]]
[[1 4 7]
 [2 5 8]
 [3 6 9]]
```