

3

# PYTHON程式設計快速上手

資訊社會必修的  
12堂Python通識課

## 本堂課重點

- ▶ 副程式及模組的觀念
- ▶ 內建函數及模組套件
- ▶ 階乘函數及費氏函數
- ▶ 各進制的數字轉換
- ▶ 簡易撲克牌遊戲

在大於1的自然數中，  
除了1和該數自己之外，  
無法再被其它的自然數整除的數，稱為質數。

質數的定義

```
n = int(input("請輸入一個數："))
is_prime = True
for i in range(2, n):
    if n % i == 0:
        print("{}不是質數！".format(n))
        is_prime = False
        break
if is_prime:
    print("{}是質數！".format(n))
```

## 使用PYTHON程式來判斷質數

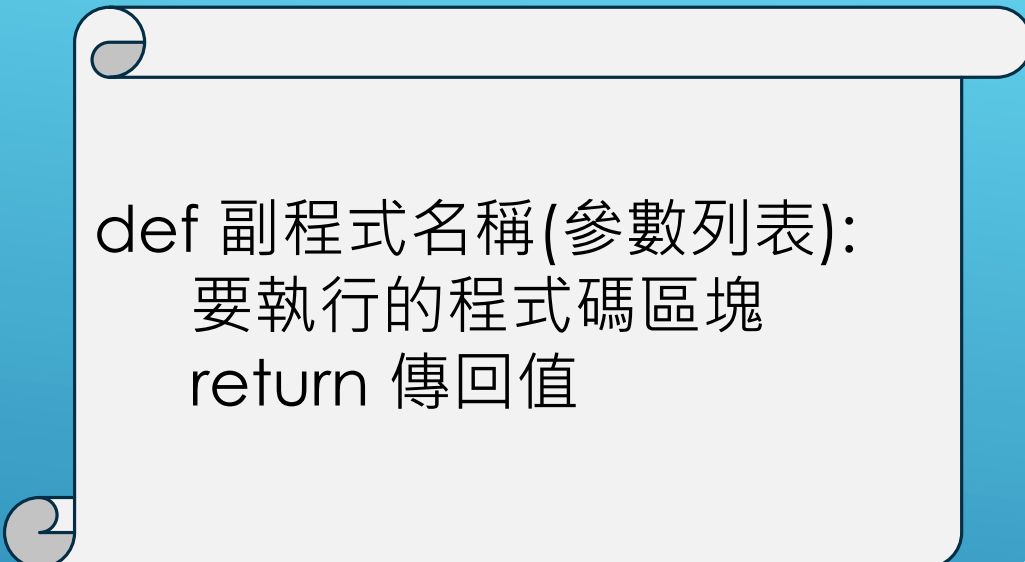
```
for n in range(100,201):  
    is_prime = True  
    for i in range(2, n):  
        if n % i == 0:  
            print("{}不是質數!".format(n))  
            is_prime = False  
            break  
    if is_prime:  
        print("{}是質數!".format(n))
```



100不是質數！  
101是質數！  
102不是質數！  
( 略 )  
198不是質數！  
199是質數！  
200不是質數！

找出100~200之間所有的質數

```
def sub_a():  
    print("我是副程式A")  
  
def sub_b():  
    print("我是副程式B，我有一個傳回值")  
    return 5  
  
sub_a()  
print(sub_b())  
sub_a()  
print(sub_b())
```



```
def 副程式名稱(參數列表):  
    要執行的程式碼區塊  
    return 傳回值
```

利用副程式機制來簡化重複使用的部份

```
def draw_bar(n):  
    print("*" * n)  
  
for i in range(1, 11):  
    draw_bar(i)
```



```
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
*****
```

輸出簡易圖形

```
def is_prime(x):  
    ret = True  
    for i in range(2, x):  
        if x % i == 0:  
            ret = False  
            break  
    return ret
```

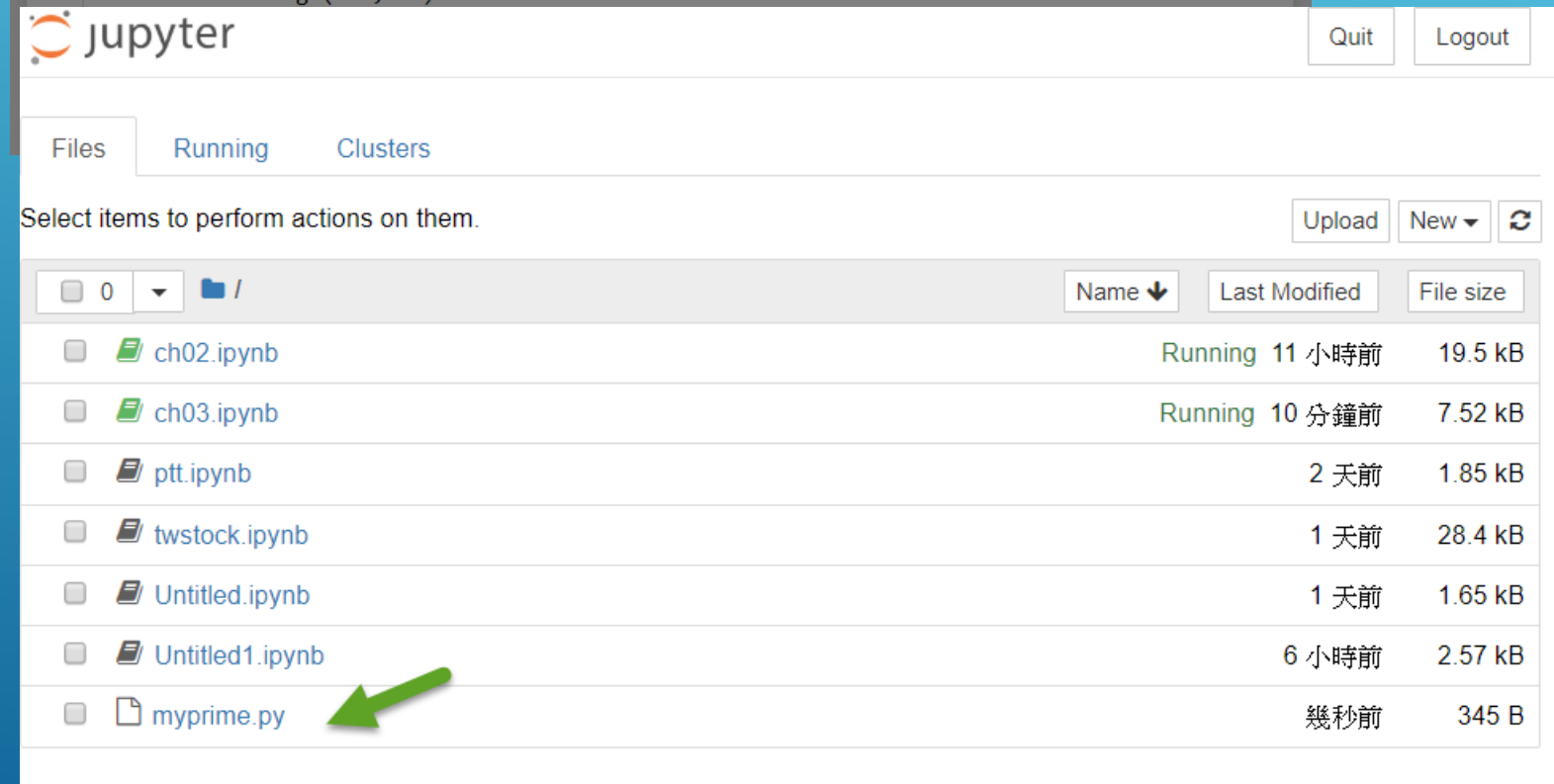
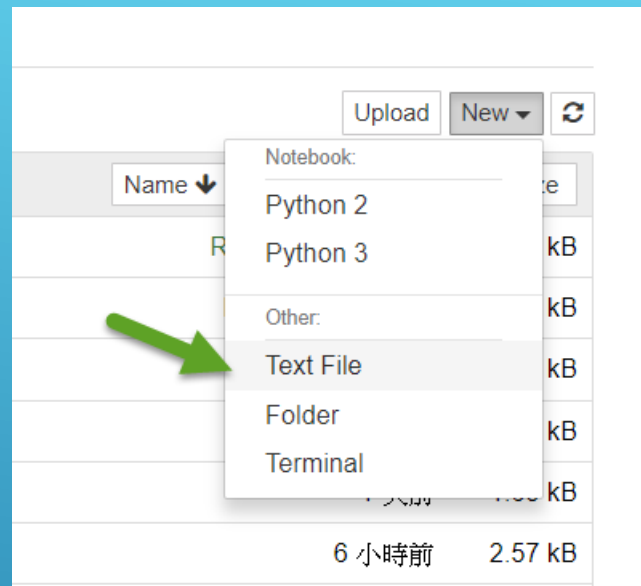
```
print("以下是100~200之間的所有質數列表：")  
for n in range(100,201):  
    if is_prime(n):  
        print("{} ".format(n), end="")
```

檢測質數的副程式（函數）



```
def is_prime(x):  
    ret = True  
    for i in range(2, x):  
        if x % i == 0:  
            ret = False  
            break  
    return ret  
  
def main():  
    print("以下是100~200之間的所有質數列表：")  
    for n in range(100,201):  
        if is_prime(n):  
            print("{} ".format(n), end="")  
  
if __name__ == "__main__":  
    main()
```

把程式模組化，  
設定MAIN進入點



建立模組的方法

```
import myprime
```

```
n1 = int(input("n1="))
```

```
n2 = int(input("n2="))
```

```
print("介於{}和{}之間的質數分別是 : ".format(n1, n2))
```

```
for n in range(n1, n2+1):
```

```
    if myprime.is_prime(n):
```

```
        print("{} ".format(n), end="")
```

n1=500

n2=600

介於500和600之間的質數分別是：  
503 509 521 523 541 547 557 563  
569 571 577 587 593 599

## 自訂模組的使用方法

```
a_list = [45, 34, 87, 43, 90, 45, 55, 98, 67, 94]
b_list = ["Tom", "Richard", "Judy", "Mary", "Lisa", "Gina"]
print(sorted(a_list))
print(sorted(b_list, reverse=True))
```

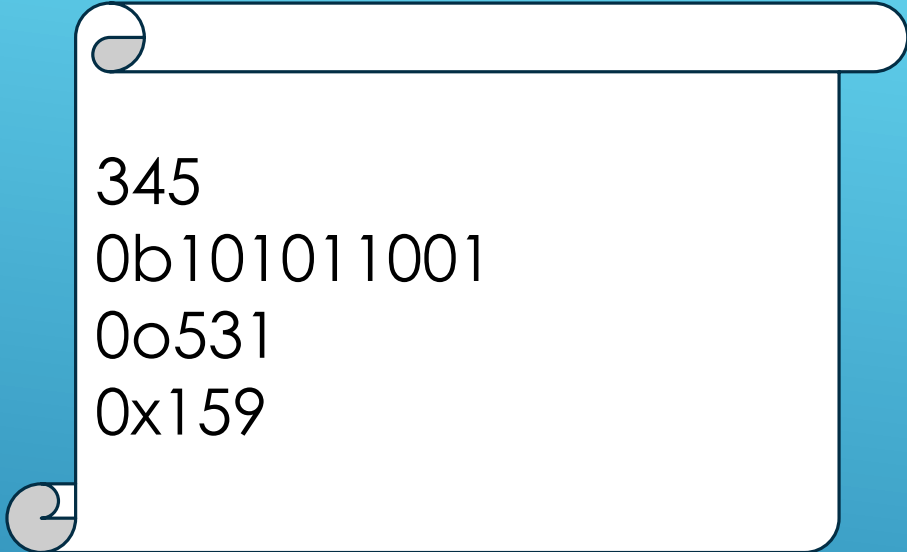
排列串列中元素的順序

內建函數名稱	說明
abs	傳回絕對值。
all	參數中的序列資料中所有的值都是True才會傳回True
any	參數中的序列值只要有一個是True就會傳回True
bin	轉換成二進位數字
dir	傳回某一物件的所有可用屬性及方法函數
divmod	在參數中放入被除數以及除數，然後一次傳回商以及餘數。
enumerate	把序列資料逐一系列，並為每一個項目加上索引數字
help	查詢某一物件或函數的用法
hex	轉換成十六進位數字
max	傳回序列資料中的最大值
min	傳回序列資料中的最小值
oct	轉換成八進位數字
reversed	傳回一個序列值的反序值
round	取小數以下指定位數的最接近值，如果不指定位數的話，就是取最接近的整數
sum	把指定的序列資料進行加總
type	傳回指定物件的型態

```
lst = [1, 0, 0, 0, 1]  
print(all(lst))  
print(any(lst))
```

ANY和ALL

```
number = 345  
print(number)  
print(bin(number))  
print(oct(number))  
print(hex(number))
```



```
345  
0b101011001  
0o531  
0x159
```

## 進制轉換函數

```
d1, d2 = 34, 3
```

```
q, r = divmod(d1, d2)
```

```
print("{} / {} = {}...{}".format(d1, d2, q, r))
```

34 / 3 = 11...1

取商數和餘數



```
ranking = ['Python', 'C/C++', 'Java', 'Javascript', 'PHP']  
for r, language in enumerate(ranking, 1):  
    print("Rank #{0} is {1}.".format(r, language))
```

好用的列舉函數ENUMERATE

```
Rank #1 is Python.  
Rank #2 is C/C++.  
Rank #3 is Java.  
Rank #4 is Javascript.  
Rank #5 is PHP.
```

```
lst = [3, 6, 1, 20, 12, 5]
r_lst = reversed(lst)
for i in r_lst:
    print(i, " ", end="")
print("\nOriginal list:{}".format(lst))
```

反序函數

random模組函數名稱	使用例	說明
random.seed(a, version)	random.seed(20)	設定隨機數種子值為20
random.randint(a, b)	random.randint(1, 6)	在1~6之間任取一個整數值
random.choice(seq)	fruit = ['Apple', 'Cherry', 'Banana'] random.choice(fruit)	在fruit串列中以隨機的方式任取其中一個項目傳回
random.shuffle(seq)	fruits = ['Apple', 'Cherry', 'Banana'] random.shuffle(fruit)	以隨機的方式打亂fruit串列中項目的順序
random.random()	random.random()	標準函數，會傳回0到1之間的任一浮點數
random.uniform(a, b)	random.uniform(10, 100)	以均勻分佈的方式在10到100之間任選整數傳回

```
import random
fruit = ['Apple', 'Cherry', 'Banana', 'Strawberry']
print("Before:", fruit)
random.shuffle(fruit)
print("After:", fruit)
print("Today's lucky fruit is:", random.choice(fruit))
```

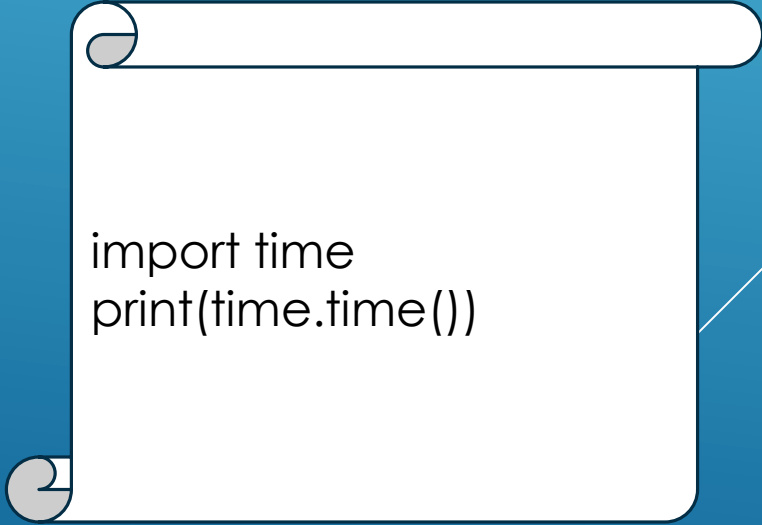
洗牌函數SHUFFLE

```
import random
card_type = ['Heart', 'Spade', 'Diamond', 'Club']
deck = [i for i in range(52)]
random.shuffle(deck)
print("你得到的5張牌是：")
for i in range(5):
    print(card_type[deck[i]//13], end="")
    print("\t", deck[i]%13+1)
```

## 撲克牌洗牌及發牌程式

- ▶ 從Unix作業系統時就流傳下來的一個記錄時間的格式叫做Epoch time，每一台電腦中都會有這樣的一個紀錄資料
- ▶ 它是一個從1970年1月1日午夜以來所經過的秒數
- ▶ 有些沒有內建Real Time Clock的嵌入式系統是以開機時間來開始算

## EPOCH TIME



```
import time  
print(time.time())
```

```
1: from datetime import datetime
2: now = datetime.now()
3: print("今天是{}".format(datetime.strftime(now, "%Y-%m-%d")))
4: date = input("請輸入一個日期 ( yyyy-mm-dd):")
5: target = datetime.strptime(date, "%Y-%m-%d")
6: diff = now-target
7: print("到今天共經過了{}天。".format(diff.days))
```

計算兩日期相差天數的程式

```
import calendar
print(calendar.month(2018,10))
print(calendar.calendar(2019))
```

產生日曆的程式

October 2018

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

2019

January

Mo	Tu	We	Th	Fr	Sa	Su
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

February

Mo	Tu	We	Th	Fr	Sa	Su
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28			

March

Mo	Tu	We	Th	Fr	Sa	Su
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

April

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

May

Mo	Tu	We	Th	Fr	Sa	Su
	1	2	3	4	5	
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

June

Mo	Tu	We	Th	Fr	Sa	Su
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

July

Mo	Tu	We	Th	Fr	Sa	Su
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

August

Mo	Tu	We	Th	Fr	Sa	Su
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

September

Mo	Tu	We	Th	Fr	Sa	Su
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

October

Mo	Tu	We	Th	Fr	Sa	Su
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

November

Mo	Tu	We	Th	Fr	Sa	Su
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

December

Mo	Tu	We	Th	Fr	Sa	Su
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					



內建函數名稱	說明
<code>math.ceil(x)</code>	傳回大於或等於X的最小整數
<code>math.fabs(x)</code>	傳回X的絕對值
<code>math.factorial(x)</code>	傳回X的階乘函數值
<code>math.floor(x)</code>	傳回小於或等於X的最大整數
<code>math.fsum(seq)</code>	計算正確的數列值的加總值
<code>math.gcd(x, y)</code>	傳回X和Y的最大公因數
<code>math.sqrt(x)</code>	傳回X的平方根
<code>math.cos(th)</code>	傳回三角函數COS值
<code>math.sin(th)</code>	傳回三角函數SIN值
<code>math.degrees(th)</code>	把弧度轉換成角度
<code>math.radians(th)</code>	把角度轉換成弧度
<code>math.pi</code>	圓周率常數
<code>math.e</code>	指數常數

```
import math
degrees = [i*math.pi/180 for i in range(181)]
for degree in degrees:
    print("#"*math.floor(math.sin(degree)*50))
```

利用SIN函數計算要顯示的#數量

```
n = int(input("n="))
total = 0
for i in range(1, n+1):
    total = total + i
print("1+2+...+{}={}".format(n, total))
```

連續加總程式第1版

```
n = int(input("n="))  
numbers = [i for i in range(1, n+1)]  
print("1+2+...+{}={}".format(n, sum(numbers)))
```

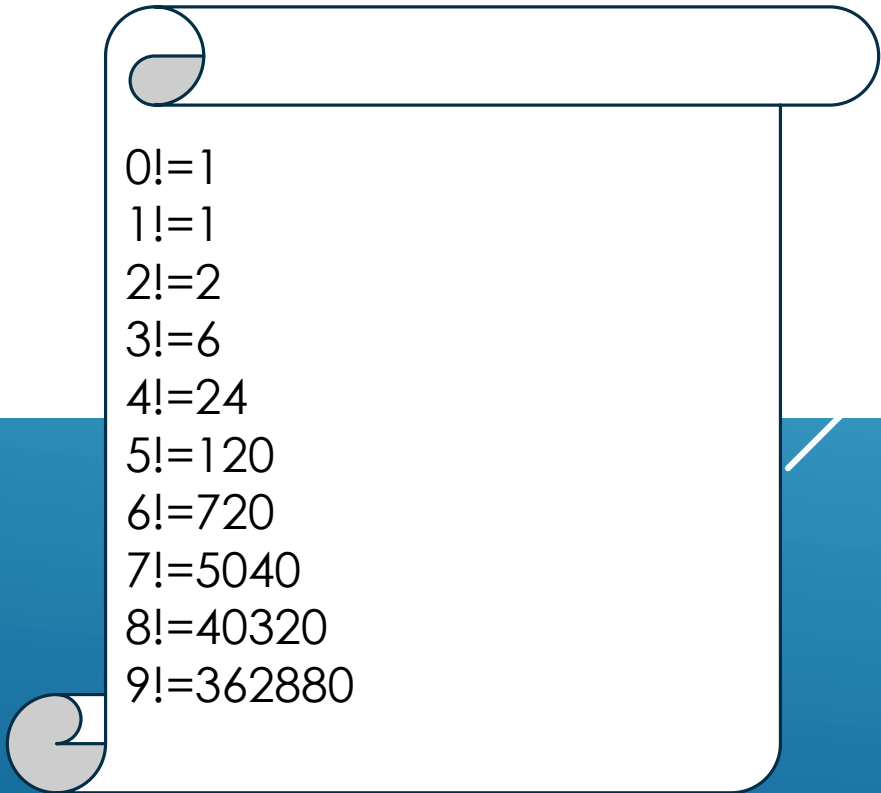
連續加總程式第2版

```
n = int(input("n="))  
factorial = 1  
for i in range(1, n+1):  
    factorial = factorial * i  
print("{}!={}".format(n, factorial))
```

階乘函數計算

```
def factorial(x):  
    ret = 1  
    for i in range(1, x+1):  
        ret *= i  
    return ret  
  
def main():  
    for i in range(10):  
        print("{}!={}".format(i, factorial(i)))  
  
if __name__ == "__main__":  
    main()
```

## 把階乘函數變成副程式



0!=1  
1!=1  
2!=2  
3!=6  
4!=24  
5!=120  
6!=720  
7!=5040  
8!=40320  
9!=362880

```
def factorial(x):  
    if x == 0 or x == 1:  
        return 1  
    else:  
        return x * factorial(x-1)
```

自己呼叫自己

```
def main():  
    for i in range(10):  
        print("{}!={}".format(i, factorial(i)))  
  
if __name__ == "__main__":  
    main()
```

階乘函數的遞迴版本

費氏數列，又名斐波那契數列 ( Successione di Fibonacci ) ，它的定義如下：

假設數列是  $F = \{f_0, f_1, f_2, \dots, f_n\}$  ，

則  $f_0 = 0$  ，  $f_1 = 1$  ，對於任一  $n \geq 2$  ，

$$f_n = f_{n-1} + f_{n-2} \circ$$

## 費氏數列的定義



```
1: def fibonacci(x):
2:     if x == 0 or x == 1:
3:         return 1
4:     else:
5:         return fibonacci(x-1) + fibonacci(x-2)
6:
7: def main():
8:     for i in range(20):
9:         print(fibonacci(i), " ", end="")
10:
11: if __name__ == "__main__":
12:     main()
```

利用遞迴來計算費氏數列

十進位數字 254，以位數權重來看的話，會是如下所示的樣子：

$$2 * 10^2 + 5 * 10^1 + 4 * 10^0$$

同樣的，如果是八進位數字 254 的話，會是如下所示的樣子：

$$2 * 8^2 + 5 * 8^1 + 4 * 8^0$$

當然，如果是十六進位數字的話，則是如下所示的樣子：

$$2 * 16^2 + 5 * 16^1 + 4 * 16^0$$

```
bin_digits = []  
n = int(input("請輸入一個數字："))  
while n > 1:  
    bin_digits.append(str(n % 2))  
    n = n // 2  
bin_digits.append(str(n))  
print("二進位數字是：{}".format("".join(reversed(bin_digits))))
```

十進位轉二進位程式

```
oct_digits = []  
n = int(input("請輸入一個數字："))  
print("驗證用：", oct(n))  
while n > 7:  
    oct_digits.append(str(n % 8))  
    n = n // 8  
oct_digits.append(str(n))  
print("八進位數字是：{}".format("".join(reversed(oct_digits))))
```

十進位轉八進位程式

```
digit_mapping = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9',  
                 'A', 'B', 'C', 'D', 'E', 'F']  
  
hex_digits = []  
n = int(input("請輸入一個數字："))  
print("驗證用：", hex(n))  
while n > 15:  
    hex_digits.append(digit_mapping[n % 16])  
    n = n // 16  
hex_digits.append(digit_mapping[n])  
print("十六進位數字是：{}".format("".join(reversed(hex_digits))))
```

十進位轉十六進位程式

```

card_types = ['黑桃', '紅心', '梅花', '方塊']
card_numbers = ['A', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'J', 'Q', 'K']
deck = list()
for card_type in card_types:
    for card_number in card_numbers:
        deck.append((card_type, card_number))
print(deck)

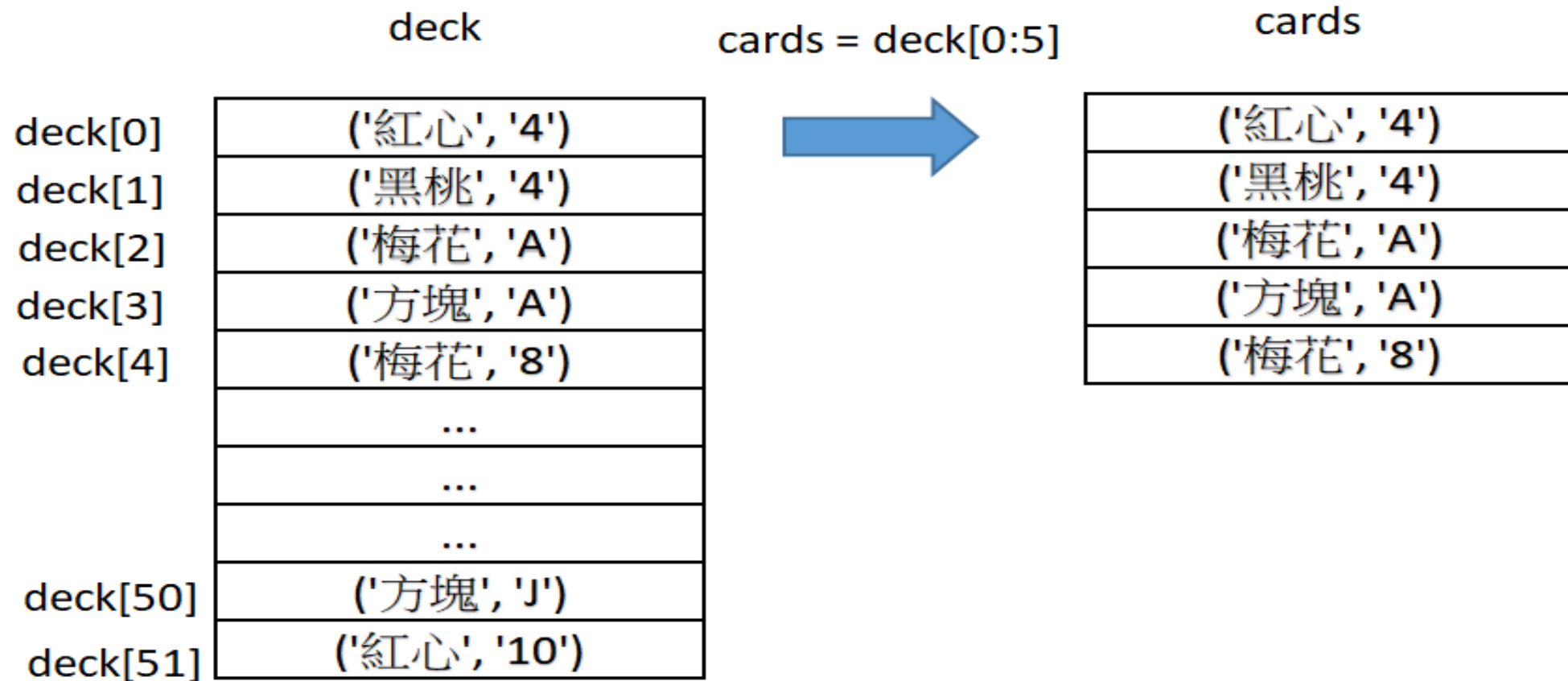
```

	deck
deck[0]	('黑桃', 'A')
deck[1]	('黑桃', '2')
deck[2]	('黑桃', '3')
deck[3]	('黑桃', '4')
deck[4]	('黑桃', '5')
	...
	...
	...
deck[50]	('方塊', 'Q')
deck[51]	('方塊', 'K')

產生52張撲克牌程式

```
1: import random
2: card_types = ['黑桃', '紅心', '梅花', '方塊']
3: card_numbers =
['A', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'J', 'Q', 'K']
3: deck = list()
4: for card_type in card_types:
5:     for card_number in card_numbers:
6:         deck.append((card_type, card_number))
7: random.shuffle(deck)
8: cards = deck[0:5]
9: print(cards)
```

撲克牌發牌程式



發牌示意圖



```
10: print("你的牌是：")
11: for i in range(5):
12:     print("#{}:{}\t{}".format(i, cards[i][0], cards[i][1]))

13: changed = input("請輸入你想要換的牌（用空白隔開）：")
14: changed_index = changed.split()
15: card_top = 5
16: for i in range(len(changed_index)):
17:     cards[int(changed_index[i])] = deck[card_top]
18:     card_top += 1
19: for i in range(5):
20:     print("#{}:{}\t{}".format(i, cards[i][0], cards[i][1]))
```

## 抽換牌介面程式

changed\_index=[1, 2]

deck

deck[0]	('紅心', '4')
deck[1]	('黑桃', '4')
deck[2]	('梅花', 'A')
deck[3]	('方塊', 'A')
deck[4]	('梅花', '8')
deck[5]	('黑桃', '5')
deck[6]	('黑桃', '6')
	...
deck[50]	('方塊', 'J')
deck[51]	('紅心', '10')

cards

('紅心', '4')
('黑桃', '5')
('黑桃', '6')
('方塊', 'A')
('梅花', '8')

card\_top=6

抽換牌時變數的運用與變化情形

```
changed = input("請輸入你想要換的牌 ( 用空白隔開 ) : ")
changed_index = changed.split()
card_top = 5
for i in range(len(changed_index)):
    cards[int(changed_index[i])] = deck[card_top]
    card_top += 1
cards.sort(key = operator.itemgetter(1))
for i in range(5):
    print("#{}:{}\t{}".format(i, cards[i][0], cards[i][1]))
```

輸出時加上排序功能的程式碼