

Python

呼叫數學函式



將計算的事
交給數學專家

三角函數

取log值

次方問題

問題



呼叫數學專家

`import math`

取絕對值

請輸入一個整數 x ，
使用 `math` 模組的 `fabs` 函式求絕對值，
重複處理直至 x 為0。

問題



使用數學函式

<叫上數學專家>
引入數學函式庫

```
import math
```

```
x = float(input())  
while x != 0:  
    print(math.fabs(x))  
    x = float(input())
```

<呼叫絕對值專家>
呼叫 `math.fabs()`
來算絕對值

呃，如果不用`math.fabs()`好像也不難呀！

```
x = float(input())  
while x!=0:  
    if x < 0:  
        x = x * -1.0  
    print(x)  
x = float(input())
```



沒錯，簡單！
可以自己來

在看一個例子，
你就知道，
數學專家有多好

次方問題

請設計一個 $\text{pow}(x,y)$ 函式，
計算 x 的 y 次方，
直到 $x=0$ 為止。

問題



叫上數學專家 math.pow

```
import math

line = input()
while line != '0':
    str1, str2 = line.split(' ')
    num1 = int(str1)
    num2 = int(str2)
    print(format(int(math.pow(num1, num2))))
    line = input()
```

<呼叫次方專家>
呼叫 `math.pow()`
來算x的y次方

這是強迫轉型，將結
果由 float 轉 int

不要叫專家，自己來試試吧!

```
line = input()
while line != '0':
    str1, str2 = line.split(' ')
    num1 = int(str1)
    num2 = int(str2)
    ans = 1
    for i in range(0, num2):
        ans = ans * num1
    print(ans)
    line = input()
```

專家來 vs. 自己來

感覺到
math.pow() 的方便了吧！

```
import math

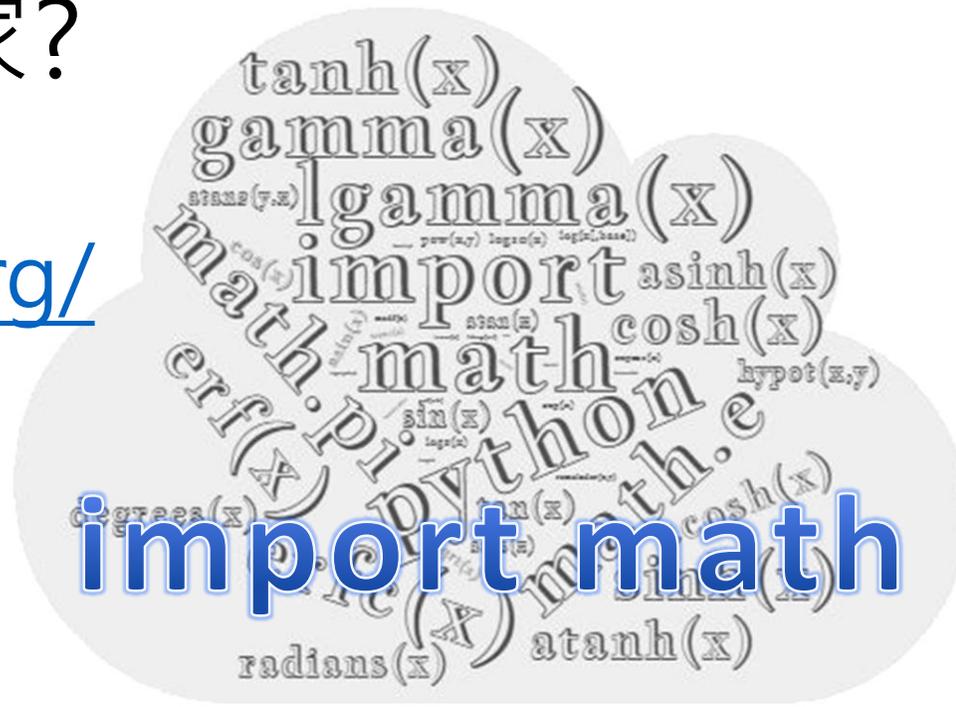
line = input()
while line != '0':
    str1, str2 = line.split(' ')
    num1 = int(str1)
    num2 = int(str2)
    print(format(int(math.pow(num1, num2))))
    line = input()
```

```
line = input()
while line != '0':
    str1, str2 = line.split(' ')
    num1 = int(str1)
    num2 = int(str2)
    ans = 1
    for i in range(0, num2):
        ans = ans * num1
    print(ans)
    line = input()
```

到底有多少數學專家？

很多，[看這](#)

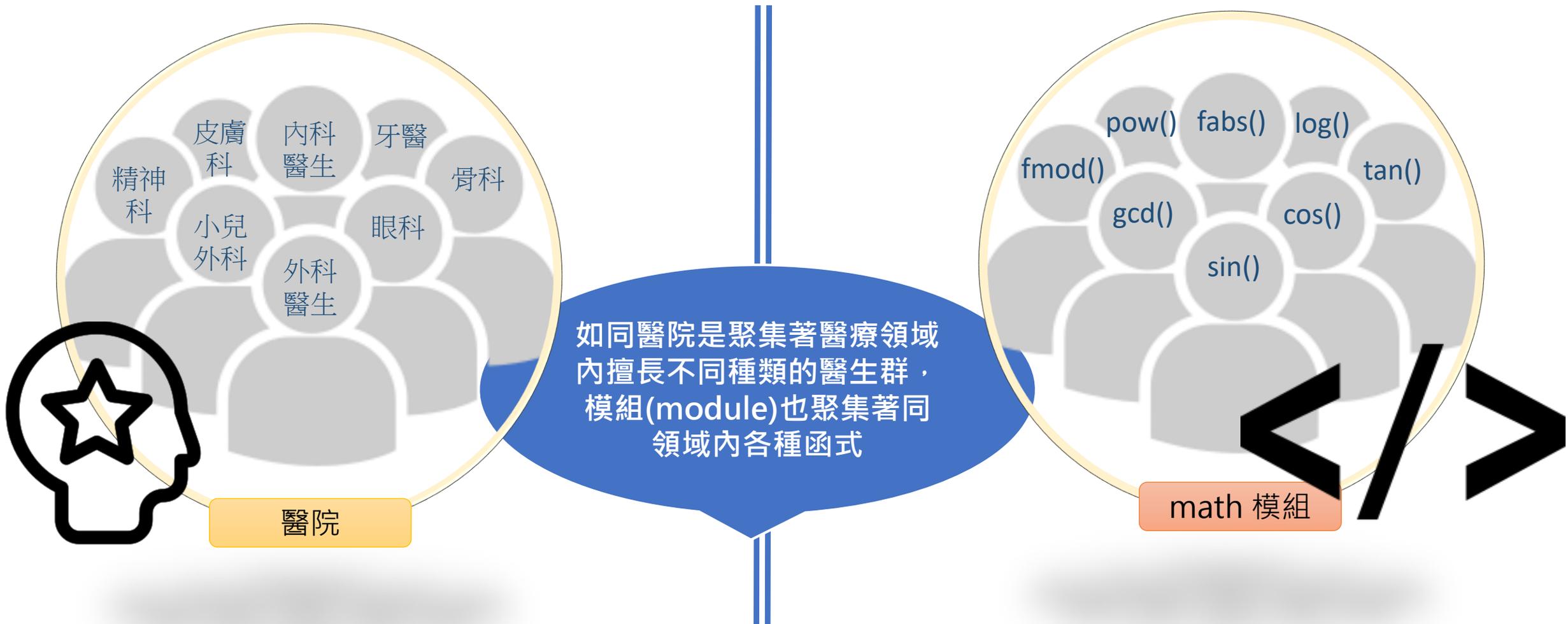
<https://docs.python.org/3/library/math.html>



PYTHON

延伸的概念

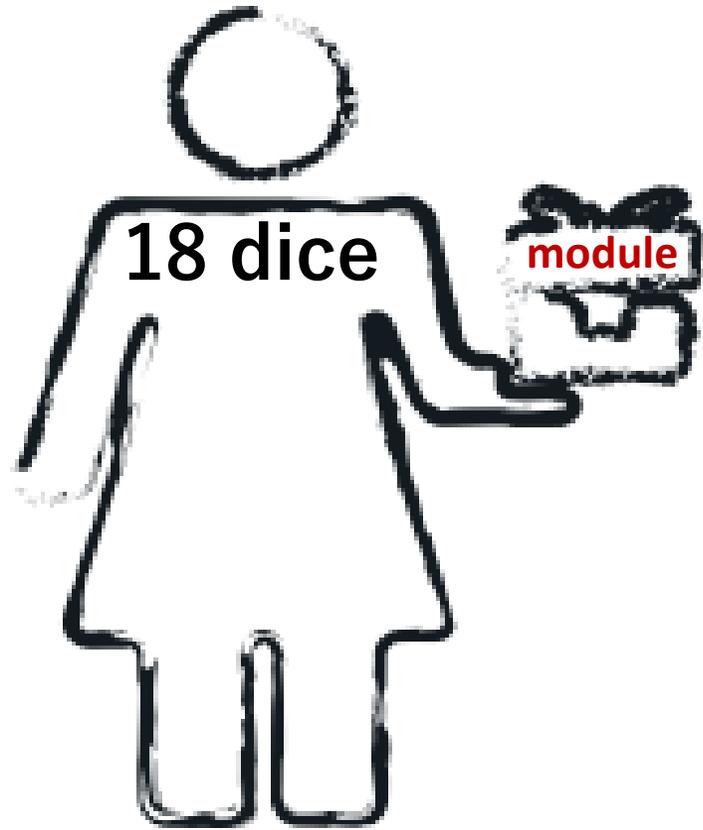
概念 1: 同領域的**函式群** => **模組** (module)



概念 2: 模組 (module)

- Python 本身提供的模組稱為內建模組(Built-in Module)
(<https://docs.python.org/3/py-modindex.html>)
- 使用者也可以依照自己的需求建立模組

概念 3: 載入模組 (import module)



- 如同去醫院才能找到看病的醫生
- 載入模組(專家群)才能呼叫需要的函數
 - import module

載入整個模組 import module

會將**整個模組**載入，下面的程式碼可以任意呼叫模組內的所有函式

```
import math
```

```
num1 = 10
```

```
num2 = 20
```

```
print(math.fabs(num1))
```

```
print(format(int(math.pow(num1, num2))))
```

module.function()

使用模組內的函式時
要加上模組名稱才能
呼叫

載入模組的某些函式 `from module import func1, func2`

```
from math import fabs, pow
```

```
num1 = 10
```

```
num2 = 20
```

```
print(fabs(num1))
```

```
print(format(int(pow(num1, num2))))
```

import 時就要明確指出要載入哪些**指定函式**，只有在這裡有寫出的函式才能在下面的程式呼叫

module.function()

使用模組內的函式時不要加上模組名稱叫

載入模組的全部函式 `from module import *`

```
from math import *
```

```
num1 = 10
```

```
num2 = 20
```

```
print(fabs(num1))
```

```
print(format(int(pow(num1, num2))))
```

會將**模組內的所有函式**載入，下面的程式碼可以任意呼叫已經載入的**模組內的函式**

`module.function()`

使用**模組內的函式**時不要加上**模組名稱**叫

概念 3: 載入模組 (import module)

- 載入整個模組

`import module`

- 載入模組的某些函式

`from module import function1, function2`

- 載入模組的全部函式

`from module import *`