

Python

遞迴呼叫



一些複雜圖騰，
其實是有規則的



自然界的遞分形

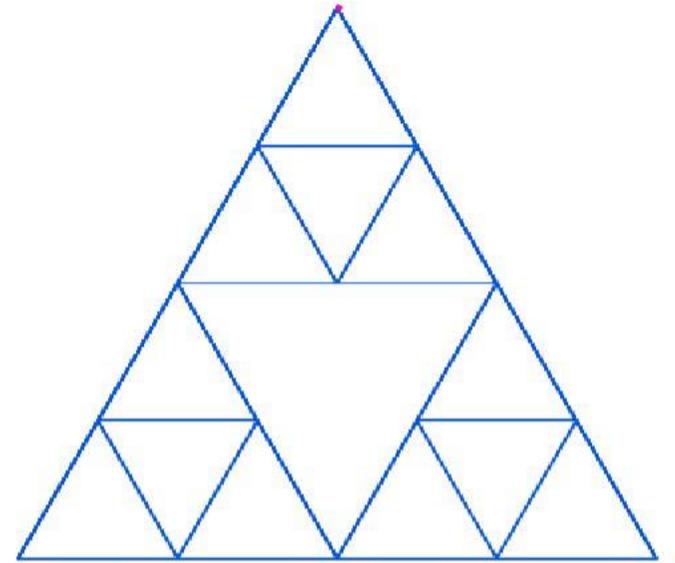
這些圖形，

- 不斷的重複同樣的東西，
- 只不過每次重複的是比較小的東西。



這個現象，稱為遞迴(recursion)

- 在電腦科學中，遞迴的本質
 - 重複地將問題分解為相同子問題的方式
 - 在程式語言支援下，可透過函式中呼叫自身函式來實現遞迴



用自己呼叫自己的方法， 計算階乘值

給一個N值，計算
 $1*2*3*...N$ 的結果

問題



N!數學上的遞迴函式

- 像是階梯一級一級的乘下一個數，故以階乘為名
 - $N! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 1$ ，其中 $n > 0$, $0! = 1$

例 1: $f_1(n) = n!$

由 $n!$ 定義可知

$$n! = \begin{cases} 1, & \text{當 } n=0, \\ n \cdot (n-1) \cdot \dots \cdot 1, & \text{當 } n > 0. \end{cases}$$

當 $n > 0$ 時，

$$\begin{aligned} f_1(n) &= n \cdot (n-1) \cdot \dots \cdot 1 \\ &= n \cdot [(n-1) \cdot \dots \cdot 1] \\ &= n \cdot f_1(n-1) \end{aligned}$$

因此， $f_1(n)$ 可重新定義為

$$f_1(n) = \begin{cases} 1, & \text{當 } n = 0, \\ n \cdot f_1(n-1), & \text{當 } n > 0. \end{cases}$$

於例 1 中，"當 $n = 0$ 時， $f_1(n) = 1$ " 稱為起始條件。

將n!數學遞迴轉為程式

```
def fac(num):  
    if num <= 0:  
        return 1  
    else  
        return num * fac(num-1)  
  
num = int(input())  
print(fac(num))
```

用遞迴解的名題

費氏數列

問題



費氏數列(1/2)

- 某人飼養一對新生兔子設兔子過了一個月後長大成熟，再過一個月即可產下一對兔寶寶。考慮兔子不死亡的條件下試問過了 n 個月後，會有幾對兔子？

月數	00	01	02	03	04	05	06	07	08	09
新生兔	0	1	0	1	1	2	3	5	8	13
成熟兔	0	0	1	1	2	3	5	8	13	21
總數	0	1	1	2	3	5	8	13	21	34

費波那西數列（Fibonacci Sequence），又譯費波拿契數、斐波那契數列、費氏數列、黃金分割數列。

在數學上，費波那西數列是以遞歸的方法來定義：

- $F_0 = 0$
- $F_1 = 1$
- $F_n = F_{n-1} + F_{n-2}$

費氏數列(2/2)

電腦程式解費氏數列

費氏函數

```
def fib(num):  
    if num <= 0:  
        return 0  
    elif num == 1:  
        return 1  
    else  
        return fib(num-1)+fib(num-2)
```

主程式

```
num = int(input())  
print(fib(num))
```