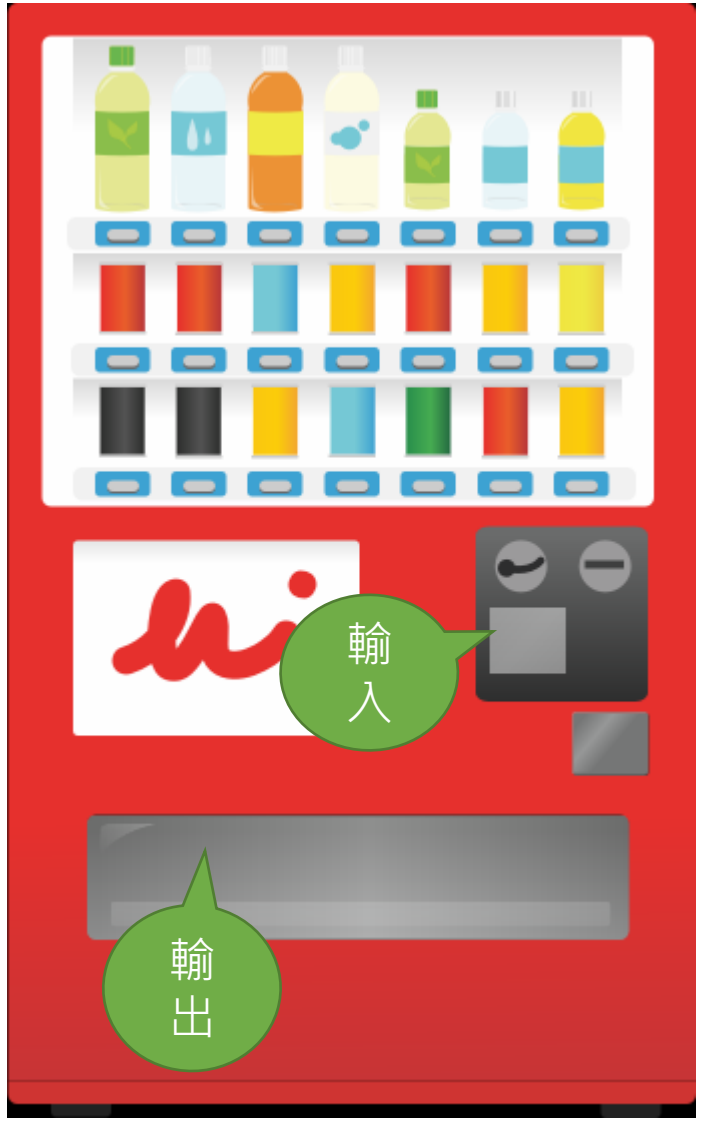
Python

函式呼叫

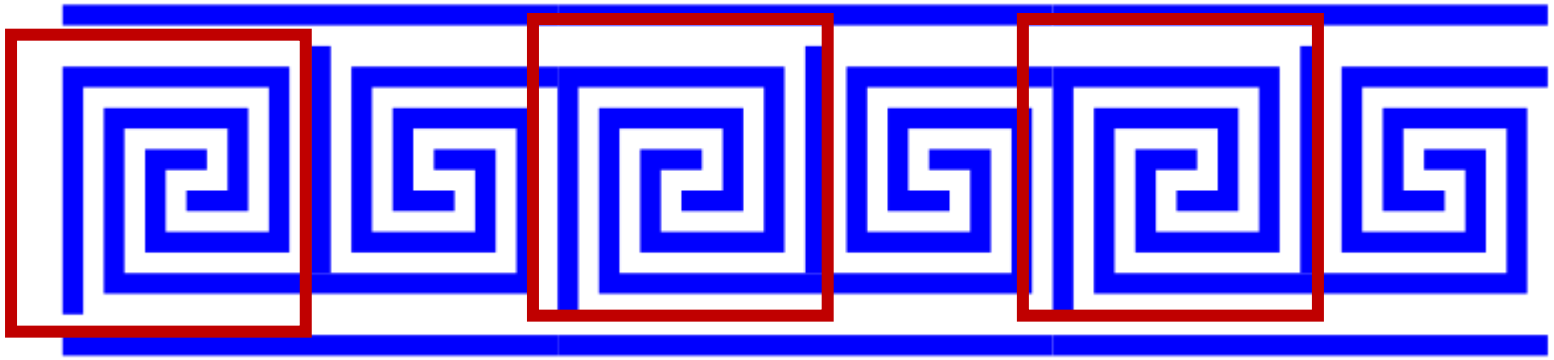
甚麼是函式?

- 函式(function)
  是一個建構程式時的小區塊
- 就像是一台自動販賣機，可以指定它的功能並且規範，例如：
  輸入：硬幣、商品
  輸出：所選擇的商品
- 輸入、輸出和功能已照規定設計好，要使用的時候，也只能依照著這個規定來使用它。
- function是容易被複製的



函式的目的

- 提高程式可讀性與可重用性
  - 將常用的或者重複的程式碼，分離出來成為一小段獨立程式區塊，稱為模組或函式
- 可以利用函式的技巧將大問題切割成小問題，以便於各個擊破...這個需要好好寫作，才能體會了



其實，我們已經用過許多函式

- 這些函式，如上述自動販賣機，都有固定的功能，例如：
  - 取得使用者輸入的 input()
  - 輸出 資料的print()
  - 將字串變數，拆成整數變數的split()
  - for的range()
- 這些函式，稱為內置函式
- 內置函式由主程式呼喚使用

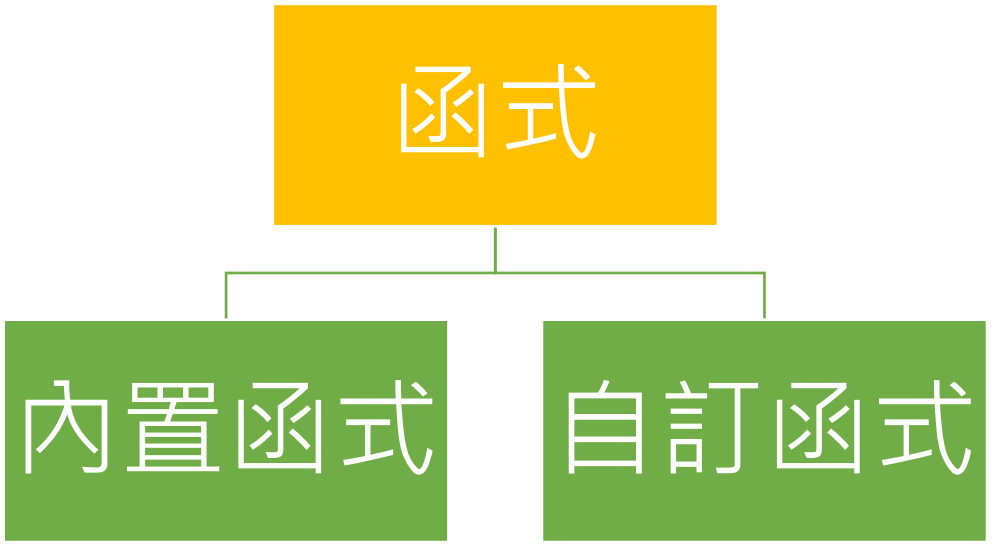
```
sum = 0
str=input()
start = int(str.split(" ")[0])
end = int(str.split(" ")[1])
for i in range(start, end+1, 1):
        sum = sum + i
        print(sum)
```

主程式內含許多內置程式

內置函式
(Built-in Function)

程式設計師可以製作函式，稱為自訂函式

- 程式設計師可以製作一個小機器，
  具有獨立的功能，例如：
  - 兩數相加
  - 變數交換
  - 1+2+3+...n的和
  - 最大公因數
  - 階乘等
- 這類的函數稱為自訂函式
  是由寫作者自行發展的一種函式
- 內置函式，則是python原有的函式



如何自訂函式？

以下是定義函式的語法

1. 先將指定功能的程式碼定義在同一區塊，這個區塊起始於def
2. 函式名稱(為函式命一個英文名)
3. 在函式名稱() 的括號內寫入要傳入哪些變數(稱為參數)，這些變數來自主程式
4. 自訂函式的程式碼必須內縮
5. return
   將自訂函式執行結果回傳主程式的自訂函式呼叫處

```
def 函式名稱(參數1, 參數2, ...參數n):
    內部程式碼區塊
    return 回傳值
```

```
主程式
函式名稱( )
可以叫用自訂函式
```

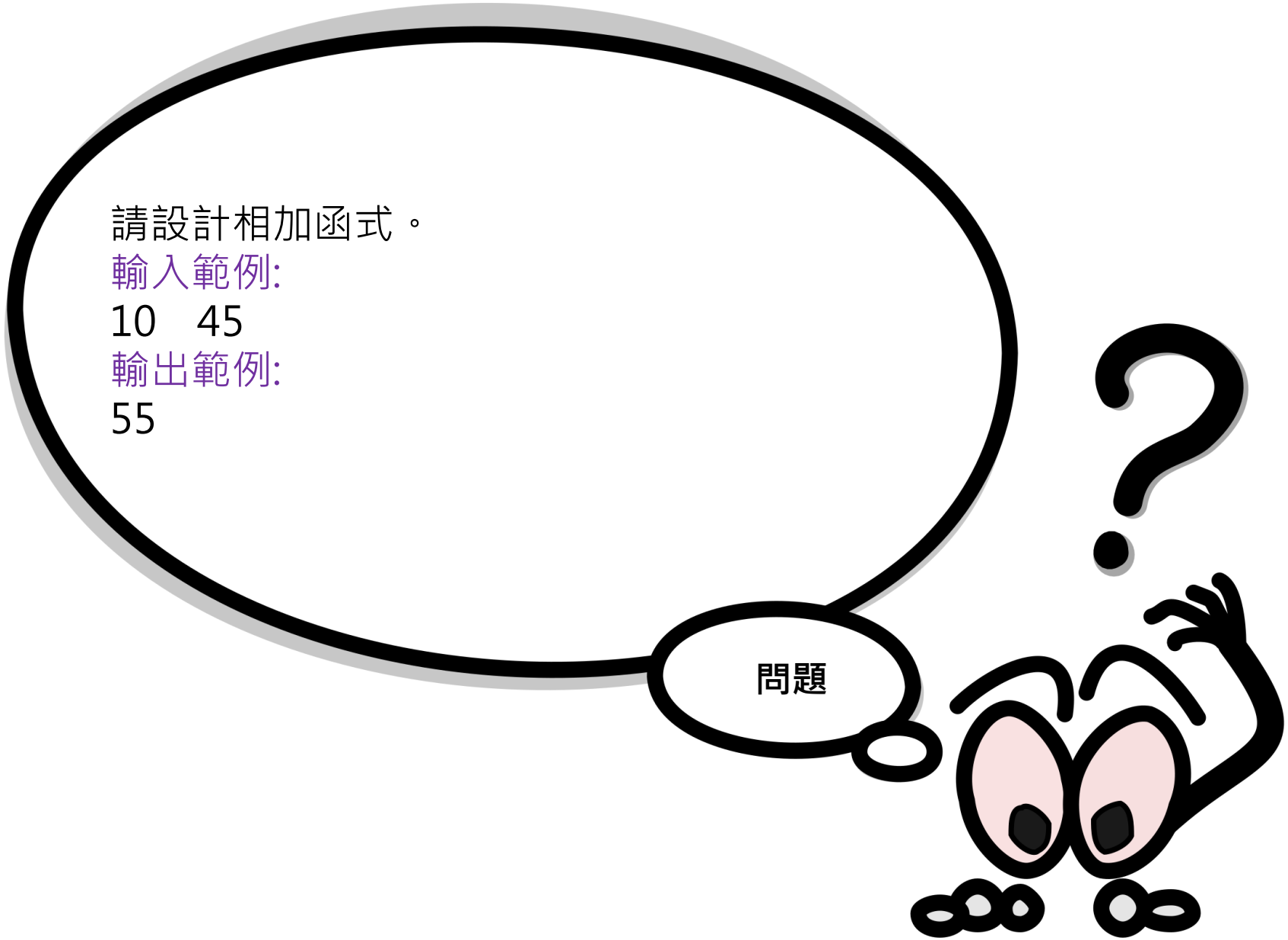
請設計相加函式。

輸入範例:

10　45

輸出範例:

55



自訂函式與叫用函式

- 第5,6與7行式主程式，輸入二個變數，然後呼叫add()，並且傳入 2 個變數到add
- 第1行到第3行是add函式
  - 1：定義一個名為add的副程式，傳入的參數來自主程式的x與y
  - 2：第 2 行執行加法
  - 第 3 行將計算結果sum回傳。
    當主程式呼叫slave，會回傳sum給主程式

```
1 def add(x,y):
2     sum = x+y
3     return sum
4
5 x=int(input())
6 y=int(input())
7 print(add(x,y))
```

def 函式名稱(參數1, 參數2, ...參數n):
    內部程式碼區塊
    return 回傳值

主程式
可以叫用自訂函式

我猜，你一定會有疑惑...

- 上一張的問題不需要自訂函式就可以解決
- 為什麼要函式解呢？
  - 本問題的確不需要函式解
  - 只是企圖使用簡單的例子，引導函式的解法
- 若以目的論，本單元的每一題不需要函式皆可完成，但是希望透過本單元學習函式的解題方式，以增進邏輯思維與解題能力

```
1 def add(x,y):
2     sum = x+y
3     return sum
4
5 x=int(input())
6 y=int(input())
7 print(add(x,y))
```

請問高 10，半徑 3 的圓柱體體積(圓周率=3.14)是多少？
以及當半徑多加 2 後，圓柱體體積又會變成是多少？

問題

沒有參數，沒有回傳值的自訂函式

```
def hello():
    print(   'Hello function')

hello() # 只要一直加 hello() 就好
hello()
hello()
hello()
hello()
```

函式的跨號內，沒有任何參數，表示沒有傳入參數，本題依題意，無須傳入參數，副程式hello即可工作

有參數，沒有回傳值的自訂函式

```
def hello(n):
    for i in range(1,1+n,1):
        print("Hello function")
n=int(input())
hello(n)
```

函式的跨號內，參數n，表示有傳入參數，本題依題意，傳入參數，副程式hello即可工作

沒有參數，有回傳值的自訂函式

```
def add():
    sum=0
    for i in range(1,101,1):
        sum=sum+i
    return sum
print(add())
```

函式的跨號內，沒有任何參數，表示沒有傳入參數，本題依題意，無須傳入參數，副程式hello工作後傳值回主程式

再看有參數有傳回值：
計算半徑 3 與3+2的圓柱體體積

```
r = 3
area = r * r * 3.14
volume = area * 10
print( '半徑{0}時的圓柱體體積:{1}'.format(r, volume))

r = 3 + 2
area = r * r * 3.14
volume = area * 10
print( '半徑{0}時的圓柱體體積:{1}'.format(r, volume))
```

圓柱體體積公式是固定的，一定要每次計算時都重複寫那幾行嗎？

執行結果

```
半徑 3 時的圓柱體體積: 282.6
半徑 5 時的圓柱體體積: 785.0
```

用函式計算半徑 3 與3+2的圓柱體體積

```
r = 3
area = r * r * 3.14
volume = area * 10
print('半徑 {0} 時的圓柱體體積: {1}'   .format(r, volume))

r = 3 + 2
area = r * r * 3.14
volume = area * 10
print('半徑 {0} 時的圓柱體體積: {1}'.format(r, volume))
```

```
# area3 函式計算體積
def area3(r, h):
  area2 = r * r * 3.14
  return area2 * h

r = 3
volume = area3(r, 10)
print(   '半徑 {0} 時的圓柱體體積: {1}'.format(r, volume))
r = 3+2
volume = area3(r, 10)
print('半徑 {0} 時的圓柱體體積: {1}'.format(r, volume))
```

重複寫的從兩行變一行

另一種函式寫法，再自訂一個計算面積的函式

```
# area3 函式計算體積
def area3(r, h):
    area2 = r * r * 3.14
    return area2 * h


r = 3
volume = area3(r, 10)
print('半徑 {r} 時的圓柱體體積: {v}'.format(r=r,
v=volume))

r = 3+2
volume = area3(r, 10)
print('半徑 {r} 時的圓柱體體積: {v}'.format(r=r,
v=volume))
```

```
# area2 函式計算面積
def area2(r):
    return r * r * 3.14

# area3 函式計算體積
def area3(r, h):
    return area2(r) * h

r = 3
volume = area3(r, 10)
print('半徑 {r} 時的圓柱體體積: {v}'.format(r=r, v=volume))

r = 3+2
volume = area3(r, 10)
print('半徑 {r} 時的圓柱體體積: {v}'.format(r=r, v=volume))
```

呼叫方式沒變，只是多定義了一個函式area2，在原有的 area3函式中又呼叫area2函式

函式的回傳值

```
# area3 函式計算體積
def area3(r, h):
  area2 = r * r * 3.14
  return area2 * h

#主程式
r = 3
volume = area3(r, 10)
print(  '半徑 {0} 時的圓柱體體積: {1}'.format(r, volume))
```

最後一行加上
return 可以決定
area函式區塊
要回傳什麼值，
本題：
傳回area2*h

area2*h該傳回哪裡？
傳回主程式中該函
式所指定的變數
(在此為volume)

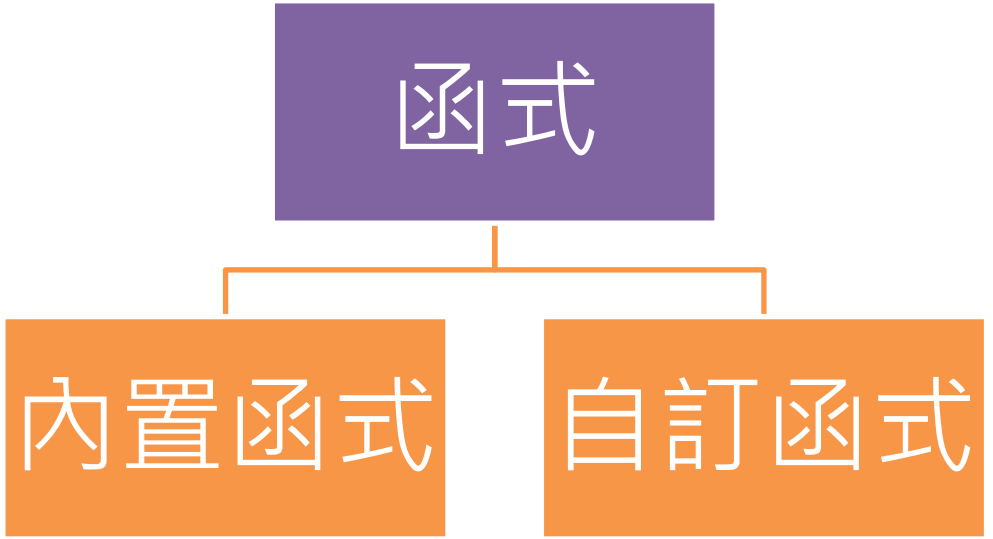
PYTHON

延伸的概念

函式的種類 (function)

- Python 本身提供的函式稱為內置函式(Built-in Function)
  - 程式語言預先定義的函式
  - 格式不可變動

- 使用者自己依照需求定義的函式稱為自訂函式，使得大問題可以切割成小問題
  - 程式編輯者自己定義的函式
  - 根據使用需求，程式編輯者隨時可以更動函式的定義



為什麼要自訂函式-大問題切成小問題

- 當程式碼處理的問題越來越多時，也需要將大問題切成各個小問題，讓閱讀的人比較容易理解，當程式出錯時，也比較容易找問題！

- 至於大問題要切成多少個小問題，那就要靠多寫程式的經驗累積來判斷囉！請登入DICE練習

為什麼要自訂函式-切割分段

就像文章要分段落閱讀，沒分段落的文章很難閱讀！

從前有個可愛的小姑娘，誰見了都喜歡，但最喜歡她的是她的奶奶，簡直是她要什麼就給她什麼。一次，奶奶送給小姑娘一頂用絲絨做的小紅帽，戴在她的頭上正好合適。從此，姑娘再也不願意戴任何別的帽子，於是大家便叫她“小紅帽”。一天，媽媽對小紅帽說：“來，小紅帽，這裡有一塊蛋糕和一瓶葡萄酒，快給奶奶送去，奶奶生病了，身子很虛弱，吃了這些就會好一些的。趁著現在天還沒有熱，趕緊動身吧。在路上要好好走，不要跑，也不要離開大路，否則你會摔跤的，那樣奶奶就什麼也吃不上了。到奶奶家的時候，別忘了說‘早上好’，也不要一進屋就東瞧西瞅。”“我會小心的。”小紅帽對媽媽說，並且還和媽媽拉手作保證。奶奶住在村子外面的森林裡，離小紅帽家有很長一段路。小紅帽剛走進森林就碰到了一條狼。小紅帽不知道狼是壞傢伙，所以一點也不怕它。“你好，小紅帽，”狼說。“謝謝你，狼先生。”“小紅帽，這麼早要到哪裡去呀？”“我要到奶奶家去。”“你那圍裙下面有什麼呀？”“蛋糕和葡萄酒。昨天我們家烤了一些蛋糕，可憐的奶奶生了病，要吃一些好東西才能恢復過來。”“你奶奶住在哪裡呀，小紅帽？”

從前有個可愛的小姑娘，誰見了都喜歡，但最喜歡她的是她的奶奶，簡直是她要什麼就給她什麼。一次，奶奶送給小姑娘一頂用絲絨做的小紅帽，戴在她的頭上正好合適。從此，姑娘再也不願意戴任何別的帽子，於是大家便叫她“小紅帽”。

一天，媽媽對小紅帽說：“來，小紅帽，這裡有一塊蛋糕和一瓶葡萄酒，快給奶奶送去，奶奶生病了，身子很虛弱，吃了這些就會好一些的。趁著現在天還沒有熱，趕緊動身吧。在路上要好好走，不要跑，也不要離開大路，否則你會摔跤的，那樣奶奶就什麼也吃不上了。到奶奶家的時候，別忘了說‘早上好’，也不要一進屋就東瞧西瞅。”

“我會小心的。”小紅帽對媽媽說，並且還和媽媽拉手作保證。

奶奶住在村子外面的森林裡，離小紅帽家有很長一段路。小紅帽剛走進森林就碰到了一條狼。小紅帽不知道狼是壞傢伙，所以一點也不怕它。

“你好，小紅帽，”狼說。

“謝謝你，狼先生。”

“小紅帽，這麼早要到哪裡去呀？”

內置函式

完整的內置函式清單，請參考這裡(https://www.w3schools.com/python/python_ref_functions.asp)

參數

functionName(參數名=預設值)

若呼叫者沒有傳入參數時，函式內就會將該變數設為預設值

```
def myLove(fruit = "芒果"):
  print("我喜歡吃 {fruit}".format(fruit=fruit))


myLove("蘋果")
myLove("柳丁")
myLove()
myLove("西瓜")
```

參數

變數

在函式內要使用的變數值，若是需要由呼叫者提供，就需透過**參數**傳入。

參數個數依照函式需求而定，並沒有限制喔！

執行結果

```
我喜歡吃 蘋果
我喜歡吃 柳丁
我喜歡吃 芒果
我喜歡吃 西瓜
```