

slido:~\$



# 函式 & 遞迴

## 函式：~ 函式 / 前言 \$

程式語言裡的函式多半和數學函數一樣  
稍微複習一下數學函數的功能

1. 找到對應的數（用  $x$  找到  $f(x)$ ）
2. 遞迴
3. 讓你數學被當

函式 : ~ 函式 / 前言 \$

基本上，函式也擁有上述的功能

因此只要你學會了數學函數，這部份不會有太多的問題

函式：~ 函式 / 數學函數 \$

基本上，函式也擁有上述的功能

因此只要你學會了數學函數，這部份不會有太多的問題

## 函式：~ 函式 / 數學函數 \$

右邊是三個數學函數

我們可以來看看這些函數有什麼性質

$$f(x) = \frac{9}{5} (x-32)$$

$$g(x, y) = 3x + 2y$$

$$h(x) = h(x-1) + 3$$

$$(h(0)=5)$$

## 函式：~ 函式 / 數學函數 \$

1. 名子可以被隨意更改

$$f(x) = \frac{9}{5} (x-32)$$

$$g(x, y) = 3x + 2y$$

$$h(x) = h(x-1) + 3$$
$$(h(0) = 5)$$

## 函式：~ 函式 / 數學函數 \$

### 2. 可以有多個參數

$$f(x) = \frac{9}{5} (x-32)$$

$$g(x, y) = 3x + 2y$$

$$h(x) = h(x-1) + 3$$
$$(h(0) = 5)$$



## 函式：~ 函式 / 數學函數 \$

### 3. 函數裡面可以放其他函數 (遞迴)

$$f(x) = \frac{9}{5} (x-32)$$

$$g(x, y) = 3x + 2y$$

$$h(x) = h(x-1) + 3$$

$$(h(0)=5)$$

## 函式：~ 函式 / 數學函數 \$

### 3. 函數裡面可以放其他函數 (遞迴)

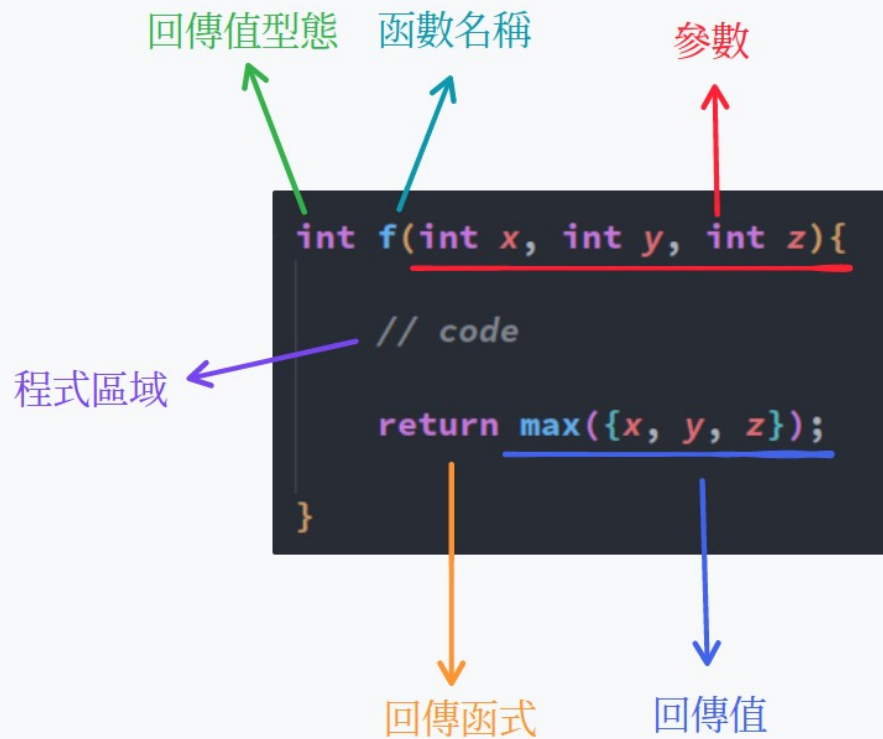
$$f(x) = \frac{9}{5} (x-32)$$

$$g(x, y) = 3x + 2y$$

$$h(x) = h(x-1) + 3$$

$$(h(0)=5)$$

# 函式：~ 函式簡介 / 語法 / 架構型態 \$



## 函式：~ 函式簡介 / 語法 / 回傳值型別 \$

因為函式有值，所以就有型別

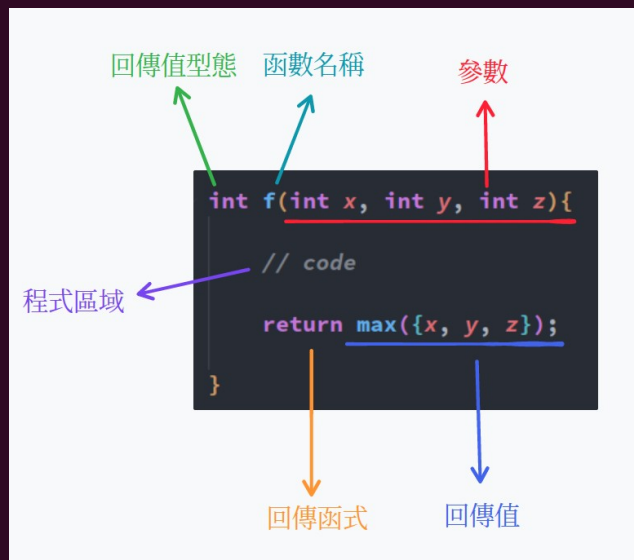
回傳值型別：也就是回傳值的型別

以上面的求最大值來說，就是 `int`

如果有一個函式想要求最大長度的字串

則是 `string`

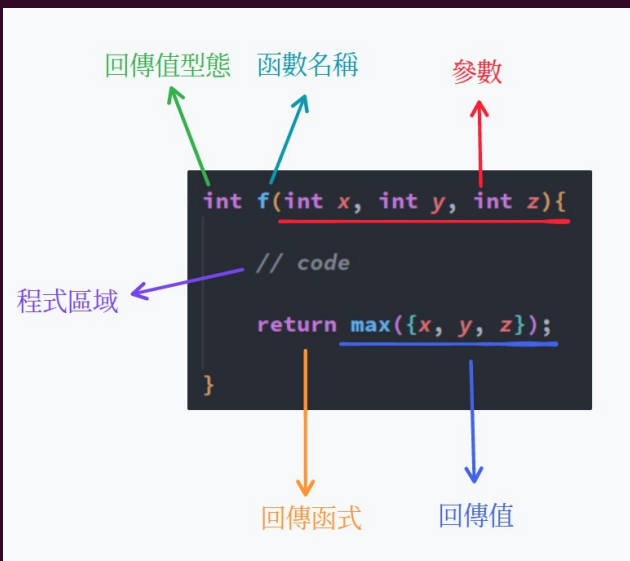
此外，如果沒有要回傳值的話就是 `void`



# 函式：~ 函式簡介 / 語法 / 參數 \$

參數：函式中可以使用的變數

可以在程式區域或回傳值被使用

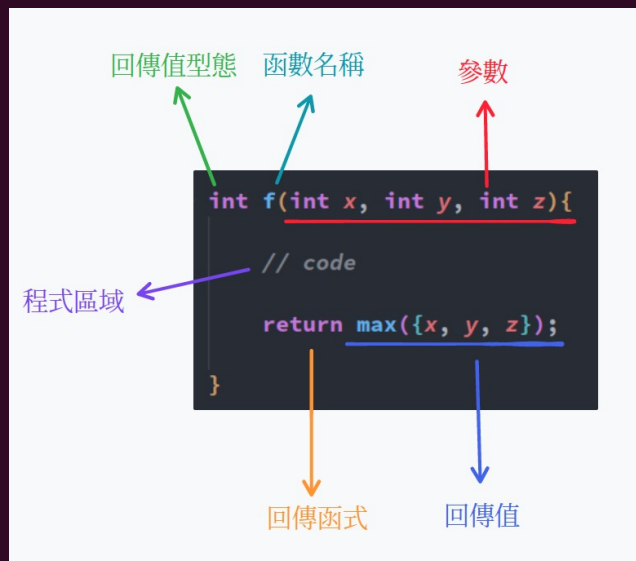


## 函式：~ 函式簡介 / 語法 / 回傳 \$

回傳函數：回報將要回傳值

放了 `return` 後就會立刻結束函式  
並且回傳回傳值

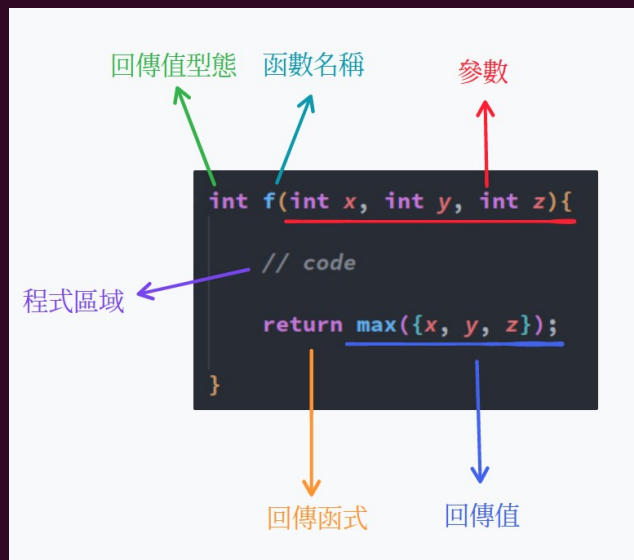
如果回傳值型態為 `void` 的話  
則直接寫成 `return`;



# 函式：~ 函式簡介 / 實做 \$

試著實做以下功能

1. 判斷兩數何者較大
2. 判斷兩字串何者長度較大
3. 轉換溫標



## 函式：~ 遞迴 / 用途 \$

> 遞迴只應天上有 凡人應當用迴圈

遞迴在初學階段可能不會有太多功用，因為大部分可以用迴圈代替

不過在未來學到 DFS、分治... 或是一些特殊的題目就必須使用到遞迴  
因此就有必要了解遞迴的概念：把問題變成小問題



## 函式：~ 遞迴 / 數學遞迴 \$

數學的遞迴需要有兩個東西

1. 中止條件
2. 遞迴關係式

費氏數列

$$\left\{ \begin{array}{l} f(1) = 1 \\ f(2) = 1 \\ f(x) = f(x-1) + f(x-2) \\ (x \geq 3) \end{array} \right.$$

函式：~ 遞迴 / 範例程式 \$

```
// function
int f(int x){
    // 中止條件
    if (x==1){
        return 1;
    }else if (x==2){
        return 1;
    }else{
        // 遞迴關係式
        return f(x-1)+f(x-2);
    }
}
```