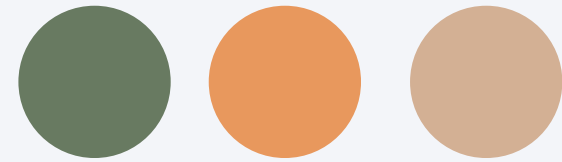


第二次社課

資訊研究社

社課時間:3/1、3/29、4/12、4/26、5/24、6/14



理論上的課表



快速複習進度



函式 (functions)



遞迴 (recursion)



陣列與結構 (arrays and structures)



基礎資料結構 (basic data structures)，包括：佇列 (queues) 和堆疊 (stacks)



基礎演算法 (basic algorithms)，包括：排序 (sorting) 和搜尋 (searching)

function
函式(函數)

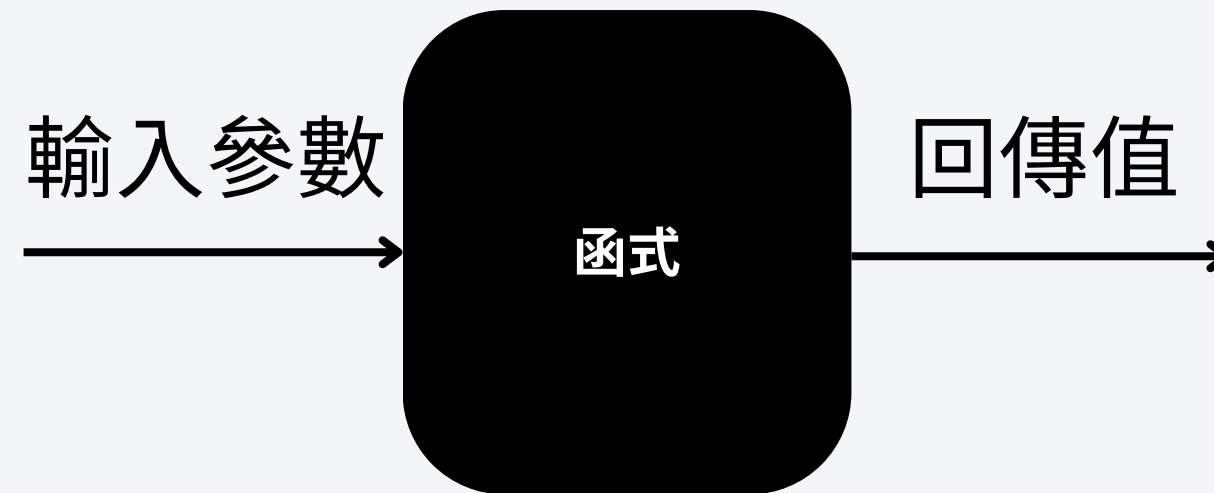
什麼是函式(function)

函式跟數學的函數差不多

可以把他想成

預先寫好的程式,只需要輸入數值就能用

可以用於實現某些特定功能



f=5x+3用函式來表示就像是

```
#include<bits/stdc++.h>
using namespace std;
int function(int x){
    int answer = 5*x+3;
    return answer;
}
int main(){
    int x;
    cin>>x;
    cout<<function(x);
}
```

$f=5x+3y+3$ 用函式來表示就像是

```
#include<bits/stdc++.h>
using namespace std;
int function(int x,int y){
    int answer = 5*x+3*y+3;
    return answer;
}
int main(){
    int x,y;
    cin>>x>>y;
    cout<<function(x,y);
}
```

看完範例大概也知道怎麼寫了吧

函式定義

資料型態 function的名字(資料型態 參數名字){

//你的code

return 回傳值;

}

其實也可以寫成這樣

```
#include<bits/stdc++.h>
using namespace std;
int function(int x,int y){
    return 5*x+3*y+3;
}
int main(){
    int x,y;
    cin>>x>>y;
    cout<<function(x,y);
}
```

C 常見的內建資料型態

資料型態	名稱	大小 (bytes)	範例
短整數 (Short Integer)	short int	2	32
整數 (Integer)	int	4	32
長整數 (Long Integer)	long int	4	32
字元 (Character)	char	1	'3'
單精度浮點數 (Single Precision Floating Point)	float	4	3.2
雙精度浮點數 (Double Precision Floating Point)	double	8	3.2
無	void	(無)	(無)

《實作相依》：意指語言標準內容並沒有強制的規定，
在使用不同編譯器或設定的情況下，可能會不一樣

大小是 《實作相依》

**函式的不同使用方法
會有
不同寫法**

有回傳值(範例)

```
#include<bits/stdc++.h>
using namespace std;
int f(int x){
    return x=5*x+3;
}
int main(){
    cout<<f(4);
}
```

沒有回傳值(範例)

```
#include<bits/stdc++.h>
using namespace std;
void f(int x){
    int ans = x*5+3;
    cout<<ans;
}
int main(){
    f(6);
}
```

int main
詳細看這

main也是一個函數

當程式開始執行時，系統會呼叫函數 main

```
return 0;
```


在main函式中的 return 0;
最主要的用處是當作一個程式的結束
回傳值可以是任意數(return 12345;也可以)

最後的一題小題目

bmi計算機

w/h/h

試著用剛才看到的function寫寫看

解答(有回傳)

```
#include<bits/stdc++.h>
using namespace std;
float bmicalc(float h,float w){
    return w/h/h;
}
int main(){
    float h,w;
    cin>>h>>w;
    cout<<bmicalc(h,w);
}
```

解答(無回傳)

```
#include<bits/stdc++.h>
using namespace std;
void bmicalc(float h,float w){
    float ans = w/h/h;
    cout<<fixed<<setprecision(2); //控制小數點後有幾位(非必要)
    cout<<ans;
}
int main(){
    bmicalc(1.7,50);
}
```

**這次內容比較短
但下次講遞迴希望有在聽的朋友能記得
不然要一直查還是挺麻煩的**