

# 結構

系程-11/27  
荊輔翔

# 開始之前

- 複習:

副函式-

用副函式印出"hello world!!"

# 結構

- 變數宣告

# 結構

Int  
Float  
char  
等等....



結構

struct

# 結構

一種自定義的資料型態，可包函多種資料

比較：

陣列：每個元素均具有相同的資料型態

結構：不同的資料型態成員可同時存在一個結構內。

# 結構

定義一：

```
struct 結構名稱{  
    資料型態 變數名稱-1;  
    資料型態 變數名稱-2;  
    .....  
} 結構變數名稱，.....;
```

# 結構

定義二：

```
struct 結構名稱{
```

```
    資料型態 變數名稱-1;
```

```
    資料型態 變數名稱-2;
```

```
    .....
```

```
};
```

```
struct 結構名稱 結構變數名稱-1 , 結構變數名  
                稱-2 , .... ;
```



# 結構

結構變數設定初值的方式類似陣列。

在結構變數宣告後，利用等號(=)及大括號{}來給予初值。

# 結構

```
struct sale{  
    char name[20];  
    char item[20];  
    float prise;  
} mysale = {"Acme Industries",  
             "Left-handed widget",  
             1000.00 };
```

# 結構

- 資料存取

一般我們會用 “.” 或 “->” 來存取資料

“.” 存取一般結構變數之成員資料

“->” 存取指標型態結構變數之成員資料

# 結構

- 範例

```
struct example{
```

```
    int x;
```

```
    int y;
```

```
} first, second;
```

- 

```
first.x=5;
```

```
first.y=11;
```

```
second.x=7;
```

```
second.y=first.x+first.y;
```

```
>>> second.y=16
```

# 結構

- `#include <stdio.h>`
- `#include <stdlib.h>`
- `int main ()`
- `struct data`
- `char name[10];`
- `int math;`
- `};`
- `struct data student = {"Mary Wang", 74};`
- `printf("學生姓名: %s\n", student.name);`
- `printf("數學成績: %d\n", student.math);`
- `}`

# 結構

練習:

用結構撰寫一隻程式，包含 “姓名” “班級” “計概成績”  
“英文成績” “數學成績”

並至少可供三人輸入!!



# 結構

## 巢狀結構

# 結構

就是結構中的結構



# 結構

- struct{
- int month;
- int day;
- int year;
- } **DATE**;

- struct{
- int hour;
- int min;
- int sec;
- } **TIME**;

- struct{
- **DATE** date;
- **TIME** time;
- } log;

# 結構

巢狀結構成員存取範例

**log**

**log.date**

**log.date.month**

**log.date.year**

**log.time**

**log.time.min**

**log.time.sec**

# 結構

## 練習:

利用巢狀結構算出四座標長方形的面積

EX: left.up.x

left.up.y

Right.up.x

Right.up.y

# 結構

## 練習:

- 在結構中使用陣列

EX:

```
struct entry{
```

- **char name[12];**
- **char phone[8];**
- **};**
- **struct entry list[50];**



# 結構