

Struct 2
&
動態記憶體配置

資訊之芽語法班 2015 c2251393

Struct review

- 宣告

```
struct my_type {  
    int a , b , c;  
    char d , e;  
    double f;  
};
```

- 使用

```
my_type x;  
x.a += 5;  
x.c = x.a + x.b;  
printf("%c" , x.d);  
scanf("%d" , &(x.b));  
my_type arr[ 100 ];
```

struct 可不是只有這樣

struct 可以包其他struct

```
struct point2d {
    int x , y;
};

struct point3d {
    point2d p;
    int z;
};
```

- 用法差不多

```
point3d a;
a.p.x = 5; // 讀成(a.p).x 注意喔<<
// (a.p)是個point2d
a.p.y = 10;
a.z = 4;
point3d arr[ 100 ];
```

more examples

```
struct scores_type {
    int math , pe , chinese;
}
struct student_type {
    char name[ 100 ];
    scores_type scores;
}
```

```
student_type gin;
scanf("%s" , gin.name);
gin.scores.math = 80;
gin.scores.math += 22;
gin.scores.pe = gin.scores.math / 2;
gin.scores.chinese = gin.scores.math + gin.scores.pe;
print("name %s,math %d",gin.name,gin.scores.math);
```

pointer to a struct variable

- int 有指標, char有指標, 陣列也有指標
- 沒道理我的student_type沒有

```
student_type gin;  
student_type *good_student = &gin;  
printf("good student %s" , (*gin).name);  
// (*gin)會得到一個student_type的東西  
// 然後就可以(*gin).blah了
```

- 所以...

```
void f( student_type *arr , int size )  
{  
    for( int i = 0 ; i < size ; i++ )  
        printf("%s" , arr[i].name);  
}  
  
student_type arr[ 100 ];  
f( arr , 5 ); // 真自然~~~
```

`(*ptr).blah` 好煩

- 於是有了 ->

```
student_type *ptr = &gin;  
printf("%s", ptr->name);  
ptr->score.math = 5;
```

- 原本要....

```
printf("%s", (*ptr).name);  
(*ptr).score.pe = 555;
```

- 是否簡潔多了呢？我覺得有啦XDDD

struct也可以包指標喔><

```
struct class_type {
    student_type *a , *b;
    int *t;
};
class_type t;
t->a = &gin;
printf("%s" , t.a->name); // 看成(t.a)->name
// (t.a)是一個student_type的指標
```


自己包自己？

- struct裡不能包一個自己

```
struct a_type {  
    a_type x; // error  
}
```

- 但是可以包一個指到另一個同樣struct的指標

```
struct student_type {  
    student_type *love;  
}  
student_type pus , heen;  
pus.love = &heen;
```

struct 掰掰 new/delete你好

我愛指標

- 我可以宣告一個指標然後直接存取它的值嗎？

```
int *a;  
*a = 514;
```

- 會壞掉 因為a指到的位置是一片未知黑暗

怎麼辦？

- 指標的本質是他指到了某個存在的變數
- 可是我不想多開一個變數給他命名很麻煩ㄟ
- 於是有了new

```
int *ptr = new int;  
*ptr = 514;
```

- Or...

```
int *ptr = new int(514);
```

更多new的用法

- new 一坨陣列

```
int *ptr2arr = new int [ 100 ];  
for( int i = 0 ; i < 100 ; i++ )  
    ptr2arr[ i ] = i;
```

- new 一個struct

```
student_type *ptr = new student_type;  
scanf( "%s" , ptr->name );
```

- new 一坨struct的陣列

```
student_type *ptr2arr = new student_type [ 100 ];  
for( int i = 0 ; i < 100 ; i++ )  
    scanf( "%s" , ptr2arr[ i ].name );
```

事情是有兩面的 動態記憶體配置也是

- 有new就有delete

```
student_type *ptr = new student_type;  
delete ptr;
```

- new一個陣列就可以delete一個陣列

```
student_type *ptr2arr = new student_type [ 100 ];  
delete []ptr2arr;
```