

變數與指定運算子

寫一個程式，
在**記憶體**放置一個**整數**，然後印出來。
例如：在記憶體放置3。

問題



為什麼放置記憶體？

記憶體是電腦在運作過程中，
暫時存放資料的地方。



位置	存放的資料
第一個櫃子	3
num1	3

- 記憶體猶如有編號的櫃子。
- 使用記憶體前，須先告知。

變數宣告

第5行：

使用記憶體前，須先告知。

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      int num;
6      num=3;
7      printf("%d\n",num);
8      return 0;
9  }
```

變數名稱→變數類別→資料型態

編號→變數名稱

第6行：
指定值給變數
指定運算子=

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      int num;
6      num=3;
7      printf("%d\n",num);
8      return 0;
9  }
```

變數與指定運算子

變數定義的屬性(以int num說明)

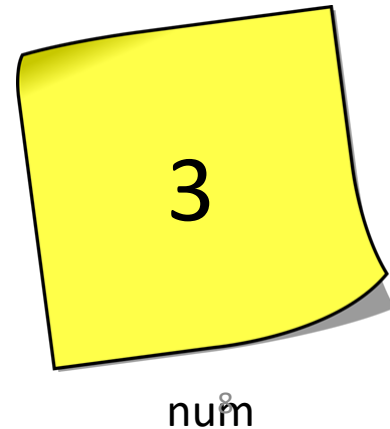
- 型態：
指的是數或者字。
數先簡單區分整數與小數(浮點數)。
int代表的是整數(integer)；
float代表的是浮點數(float)。
- 名字：
一個變數必須有名字，例如: num。
 1. 由英文字母(大小寫被視為不同)、數字以及底線組成。
 2. 不能用數字開頭。
 3. 組成: 限一個字詞。
- 值:
被賦予的值，可以使用指定運算子或者是輸入函式改變其值。
- 地址:
一個變數相對應於電腦記憶體中的位址。

指定運算子 = ：指定一個值或字元給變數

```
int num;
```

```
num = 3;
```

- 宣告一個常數，同時設置賦值為3。
- 變數num的值一開始設為3。
- 如果未設定初始值，則變數的初始值可能是任意數。



輸出—使用printf

- 必須包含標頭檔#include <stdio.h> 。
- printf("%d", num);
 - " " : 代表一個字串的開始與結束。
 - % : 跳脫字元，代表printf要顯示的內容。
 - d : 以十進位(decimal)的形式顯示變數值，
d要顯示的值就是num(假設num=3)。
 - , : 逗號用來分隔一個函式的眾多的數，
例如: 輸出兩個變數。 printf("%d %d" , num1, num2);

延伸的概念

1：設定兩個變數並印出

```
#include <stdio.h>
```

```
int main( ){
```

```
    int a;
```

```
    int b;
```

或 `int a,b;`

```
    a=11;
```

```
    b=12;
```

```
    printf( "%d \n" ,a);
```

```
    printf( "%d\n" ,b);
```

```
    return 0;
```

```
}
```

2：除了印常數外，也可以加入其他字詞

```
#include <stdio.h>
```

```
int main( ){
```

```
    int cat;
```

```
    cat=3;
```

```
    printf( "There are %d cats. \n" ,cat);
```

```
    return 0;
```

```
}
```

3：浮點數如何處理？

(數據型態不只這些，本節先說明這些)

數據型態		
數	表示方法	格式符號
整數	int	%d
浮點數	float	%f

浮點數定義程式碼-printf

```
#include <stdio.h>
```

```
int main( ){
```

```
    float a;
```

```
    a=11.11;
```

```
    printf( "%f" ,a);
```

```
    return 0;
```

```
}
```

輸出:

11.110000

4：如何顯示小數點後指定的位數？ (例如: 小數點後第2位)-printf

```
#include <stdio.h>
```

```
int main( ){
```

```
    float a;
```

```
    a=11.11;
```

```
    printf("%.2f",a);
```

```
    return 0;
```

```
}
```

輸出：

11.11