



# 變數與 指定運算子

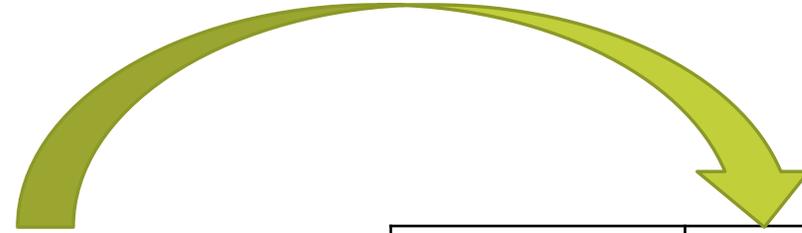
寫一個程式，  
在**記憶體**放置一個**整數**，然後  
印出來。  
例如：在記憶體放置3。

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。

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。

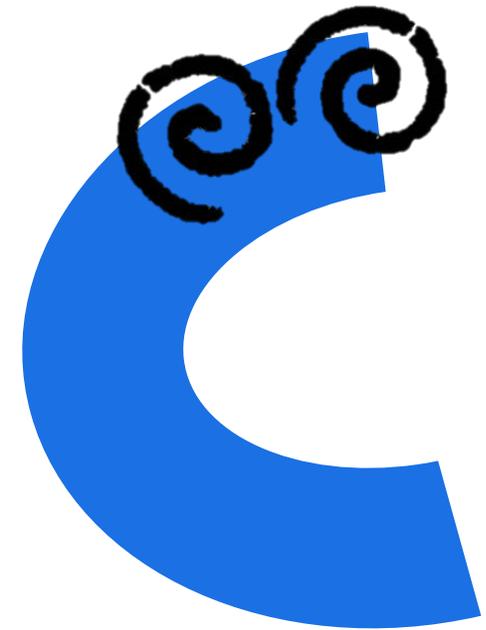
```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int main()
4 {
5     int num;
6     num=3;
7     printf("%d\n",num);
8     return 0;
9 }
```

資料型態→變數名稱

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)

## 資料型態：

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)。

## 變數名稱：

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。

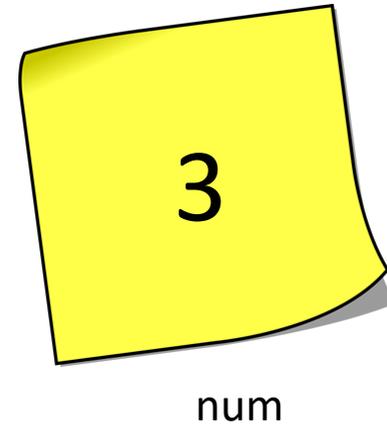
# 指定運算子 = : 指定一個值或字元給變數

```
int num;
```

```
num = 3;
```

變數num的值一開始設為3。

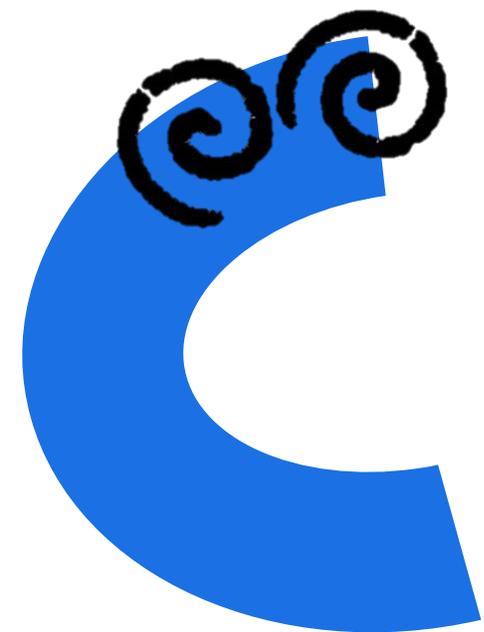
如果未設定初始值，則變數的初始值可能是任意數。



# 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？

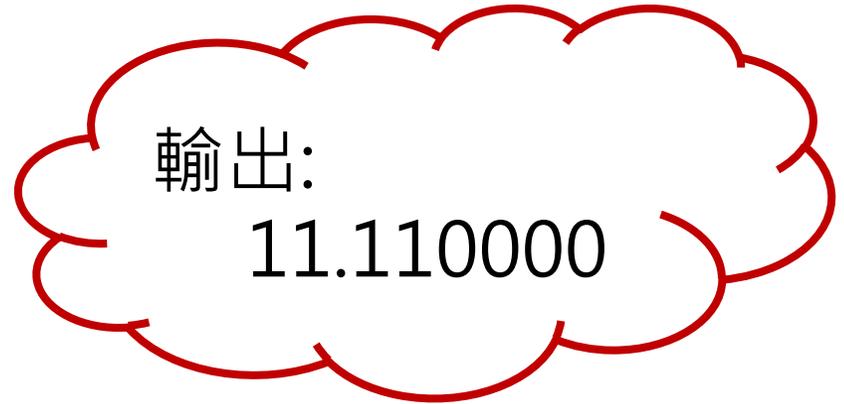
(資料型態不只這些，後面補充說明)

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

# 浮點數的輸出

---

```
#include <stdio.h>
int main(){
    float a;
    a=11.11;
    printf( "%f" ,a);
    return 0;
}
```

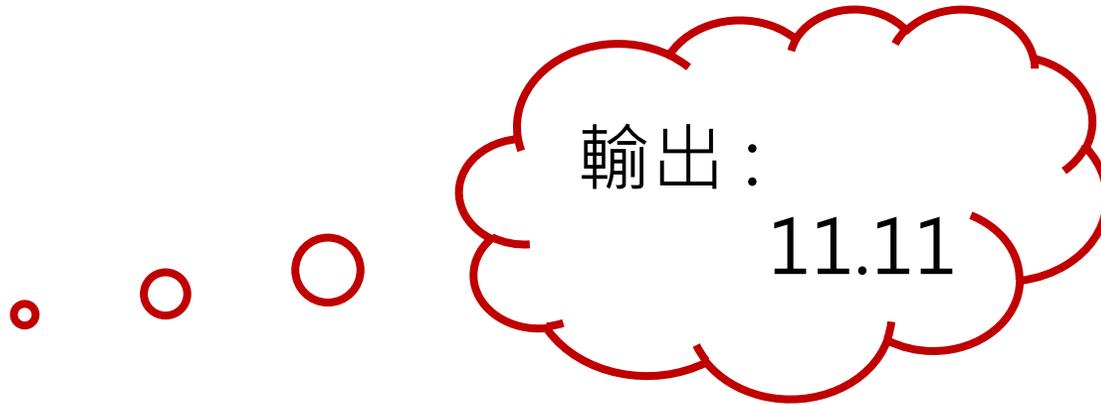


# 4：如何顯示小數點後指定的位數？ (例如: 小數點後第2位)-printf

---

```
#include <stdio.h>

int main(){
    float a;
    a=11.11;
    printf("%.2f",a);
    return 0;
}
```



# 資料型態

---

- 在程式執行的過程中，為了能夠方便有效的執行程式的各種運算，需要的資料型態有：整數、浮點數與文字等
- 電腦內部使用0與1(位元)來表示資料，並且儲存於記憶單元
- 任何資料都可以轉換成一連串位元的組合

# 常見的資料型態

資料型態	名稱	大小(bytes)	範圍	格式符號
整數	int	2bytes(16位元系統) 或 4bytes	-32768至32767或 -2147483648至2147483647	%d
長整數	long int	4bytes 或 8bytes	-2147483648至2147483647 或 -9223372036854775808至 9223372036854775807	%ld
長整數	long long int	8bytes	-9223372036854775808至 9223372036854775807	%lld
單精度浮點數	float	4bytes	$2.939 \times 10^{-38}$ 至 $3.403 \times 10^{+38}$ (7 sf)	%f
雙精度浮點數	double	8bytes	$5.563 \times 10^{-309}$ 至 $1.798 \times 10^{+308}$ (15 sf)	%lf
字元	char	1bytes	通常為-128至127或0至255	%c

大小並沒有強制規定，在使用不同編譯器或設定的狀況下，會有不同的結果