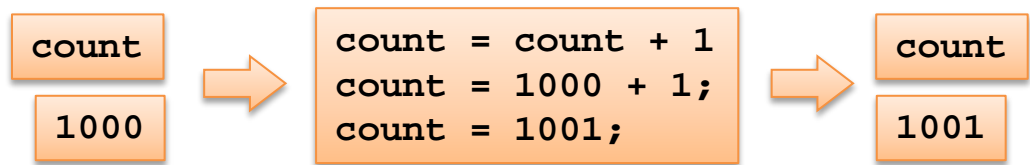


# 運算符號與運算式

# 運算符號

- + - \* /  
一般運算



- =  
把值設定給變數

- `count = count + 1;`

表示要把變數 `count` 的值加 1 之後在存回相同的變數 `count`。

# 求餘數要用 % 符號

```
#include <stdio.h>
#define SEC_PER_MIN 60
int main(void){
    scanf("%d", &sec);
    min = sec / SEC_PER_MIN;
    left = sec % SEC_PER_MIN;
    printf("%d seconds is %d minutes, %d seconds.\n",
           sec, min, left);
    printf("Enter next value (<=0 to quit):\n");
    scanf("%d", &sec);
}
```

# 累加符號 ++ 與 --

```
#include <stdio.h>
int main(void)
{
    int lower = 0, upper = 10;
    while (lower < upper) {
        lower++;
        upper--;
        printf("lower = %d, upper = %d \n", lower,
upper);
    }
    return 0;
}
```

```
lower = 1, upper = 9
lower = 2, upper = 8
lower = 3, upper = 7
lower = 4, upper = 6
lower = 5, upper = 5
```

# a++ vs. ++a

```
#include <stdio.h>
int main(void)
{
    int a = 1, b = 1;
    int aplus, plusb;
    aplus = a++;
    plusb = ++b;
    printf("a    aplus    b    plusb \n");
    printf("%1d%6d%6d%6d\n", a, aplus, b, plusb);
    return 0;
}
```

輸出：

a	aplus	b	plusb
2	1	2	2

# Expressions 與 Statements

- Expressions:

`a + 5`

`x = 3 * 4`

`k > 3`

- Statements:

`a + 5;`

`x = 3 * 4;`

`k > 3;`

# Assignment 符號

**+=** , **--** , **\*=** , **/=** , **%=**

**x += 5;**

**x = x + 5;**

**y %= 7;**

**y = y % 7;**

# 型別轉換

- 當 `expression` 或 `statement` 中所出現的變數或常數之間型別不同時，C 會將變數或常數轉成相同型別。
  - 在 `expression` 中的 `char` 會轉換成 `int`，譬如
    - `i <= 'Z'`
  - 當兩種型別混用時，位階較低的型別會轉成位階較高的；位階高低順序如下：
    - `double, float, unsigned long, long, unsigned int, int`
  - 在等式的 `statement` 中，型別會被轉換成等號左邊的變數的型別。
    - `X_double = Y_int + Z_int;`