

程式設計 (一)

CH4. 條件控制

Ming-Hung Wang 王銘宏

tonymhwang@cs.ccu.edu.tw

Department of Computer Science and Information Engineering
National Chung Cheng University

Last Semester, 2021

本章目錄

1. 布林值
2. if 選擇敘述式
3. if...else 選擇敘述式
4. 關係運算子
5. 邏輯運算子
6. 三元運算子
7. switch 多重選擇敘述式

布林值 (Boolean, 真假值)

只能儲存 1 (true) 與 0 (false) 的資料型態，並不是 C 語言的基本型態。

使用版本 C99 以後的 C 語言，只要引入 `stdbool.h` 標頭檔，就可以使用布林值資料型態 `bool`。

stdbool.h. 內容，
定義了 bool 資料型態、true=1、false=0

 stdbool.h

```
1  #ifndef STDBOOL_H
2  #define STDBOOL_h
3
4  #define bool _Bool
5  #ifndef true
6      #define true 1
7  #endif
8
9  #ifndef false
10     #define false 0
11 #endif
12
13 #endif
```

stdbool.h 所定義的 true 與 false

```
1 //boolean algebra & if else
2 #include <stdio.h>
3 #include <stdbool.h>
4
5 int main(){
6     bool flag;
7     printf("size of bool: %d\n", sizeof(bool));
8     flag = true;
9     printf("true = %d\n", flag);
10    flag = false;
11    printf("false = %d\n", flag);
12 }
13
```

"D:\codeblocks\boolean algebra & if else.exe"
size of bool: 1
true = 1
false = 0

布林值

雖然在 `stdbool.h` 中有定義 `bool` 與 `true`、`false`，我們在平時寫 C 語言時，還是會使用 0 代表 `false`，非 0 的數字 (通常是 1) 代表 `true`。當然，如果你有使用其他語言的經驗，並已經習慣使用布林值，在撰寫 C 語言時還是可以使用 `stdbool.h` 所定義的 `bool` 與 `true`、`false`。

if 選擇敘述式

if 選擇敘述式

中文裡的句型為：如果 (條件) 就 (執行...)
(轉換成 C 語言)

```
if (條件句) {  
    敘述句1;  
    敘述句2;  
    ...  
}
```

如果敘述句只有 1 個，可以省略大括號：

```
if (條件句)  
    敘述句1;
```


if 選擇敘述式

- 如果括號裡的值非 0 (true)，則會運行 if 的敘述
- 如果括號裡的值為 0 (false)，則不運行

```
1 //boolean algebra & if else
2 #include <stdio.h>
3
4 int main(){
5     int flag = 1;
6     if(flag){
7         printf("The flag is true.\n");
8     }
9 }
10
```

"D:\codeblocks\boolean algebra & if else.exe"

The flag is true.

if 選擇敘述式

```
1 //boolean algebra & if else
2 #include <stdio.h>
3
4 int main(){
5     int val;
6     int isEven = 1;
7     scanf("%d", &val);
8     if(val % 2){
9         printf("%d is odd number.\n", val);
10        isEven = 0;
11    }
12    if(isEven){
13        printf("%d is even number.\n", val);
14    }
15 }
```

"D:\codeblocks\boolean algebra & if else.exe" "D:\codeblocks\boolean algebra & if else.exe"

```
13      20
13 is odd number.  20 is even number.
```

if...else 選擇敘述式

if...else 選擇敘述式

中文裡的句型為:

如果 (條件) 就 (執行...), 否則 (執行...)
轉換成 C 語言:

```
if (條件句) {  
    敘述句1;  
    敘述句2;  
    ...  
}  
else {  
    敘述句1;  
    敘述句2;  
    ...  
}
```

如果敘述句只有 1 個, 也可以省略大括號

if...else 選擇敘述式

```
1 //boolean algebra & if else
2 #include <stdio.h>
3
4 int main(){
5     int val;
6     int isEven = 1;
7     scanf("%d", &val);
8     if(val % 2){
9         printf("%d is odd number.\n", val);
10        isEven = 0;
11    }
12    else{
13        printf("%d is even number.\n", val);
14    }
15 }
```

"D:\codeblocks\boolean algebra & if else.exe" "D:\codeblocks\boolean algebra & if else.exe"

```
27 64
27 is odd number. 64 is even number.
```

否則如果 else if

中文裡的句型為：

如果 (條件) 就 (執行...), 否則如果 (條件) 就 (執行...),
否則 (執行...)

轉換成 C 語言：

```
if (條件句) {  
    ...  
}  
else if (條件句) {  
    ...  
}  
else {  
    ...  
}
```

if...else 選擇敘述式

```
1 //boolean algebra & if else
2 #include <stdio.h>
3
4 int main(){
5     int val;
6     scanf("%d", &val);
7     if(val % 2){
8         printf("%d is odd number.\n", val);
9     }
10    else if(val % 3){
11        printf("%d is even number but not multiple of 3.\n", val);
12    }
13    else{
14        printf("%d is even number and is multiple of 3.\n", val);
15    }
16 }
```

"D:\codeblocks\boolean algebra & if else.exe"

```
20
20 is even number but not multiple of 3.
```

"D:\codeblocks\boolean algebra & if else.exe"

```
60
60 is even number and is multiple of 3.
```

if...else 選擇敘述式

else if 可以無限疊加，最後的 else 也可以選擇不寫

```
1 //boolean algebra & if else
2 #include <stdio.h>
3
4 int main(){
5     int val;
6     scanf("%d", &val);
7     if(val % 2){
8         printf("%d isn't multiple of 2.\n", val);
9     }
10    else if(val % 3){
11        printf("%d isn't multiple of 3.\n", val);
12        printf("%d is multiple of 2.\n", val);
13    }
14    else if(val % 5){
15        printf("%d isn't multiple of 5.\n", val);
16        printf("%d is multiple of 2 and 3.\n", val);
17    }
18    else if (val % 7){
19        printf("%d isn't multiple of 7.\n", val);
20        printf("%d is multiple of 2, 3 and 5.\n", val);
21    }
22 }
```

"D:\codeblocks\boolean algebra & if else.exe"

```
70
70 isn't multiple of 3.
70 is multiple of 2.
```

"D:\codeblocks\boolean algebra & if else.exe"

```
66
66 isn't multiple of 5.
66 is multiple of 2 and 3.
```

"D:\codeblocks\boolean algebra & if else.exe"

```
60
60 isn't multiple of 7.
60 is multiple of 2, 3 and 5.
```

"D:\codeblocks\boolean algebra & if else.exe"

```
99
99 isn't multiple of 2.
```


關係運算子

判斷大於、小於、等於、不等於等關係

關係運算子

| 名稱 | 運算子 | 語法 |
|------|-----|------------|
| 大於 | > | $a > b$ |
| 大於等於 | >= | $a \geq b$ |
| 小於 | < | $a < b$ |
| 小於等於 | <= | $a \leq b$ |
| 等於 | == | $a == b$ |
| 不等於 | != | $a != b$ |

== (等於) 不要跟 = (指派) 搞混了喔!

關係運算子

關係運算子使用範例 1:

```
1 //boolean algebra & if else
2 #include <stdio.h>
3
4 int main(){
5     int val;
6     scanf("%d", &val);
7     printf("%d < 10 : %d\n%d > 10 : %d\n%d == 10 : %d\n",
8           val, val < 10, val, val > 10, val, val == 10);
9     if(val < 10)
10        printf("%d is smaller than 10\n", val);
11    else if (val > 10)
12        printf("%d is smaller than 10\n", val);
13    else
14        printf("%d is 10\n", val);
15 }
```

| "D:\codeblocks\boolean algebra & if else.exe" | "D:\codeblocks\boolean algebra & if else.exe" | "D:\codeblocks\boolean algebra & if else.exe" |
|---|---|---|
| -30 | 50 | 10 |
| -30 < 10 : 1 | 50 < 10 : 0 | 10 < 10 : 0 |
| -30 > 10 : 0 | 50 > 10 : 1 | 10 > 10 : 0 |
| -30 == 10 : 0 | 50 == 10 : 0 | 10 == 10 : 1 |
| -30 is smaller than 10 | 50 is smaller than 10 | 10 is 10 |

關係運算子使用範例 2:

```
1 //boolean algebra & if else
2 #include <stdio.h>
3
4 int main(){
5     int val, isntSixMuti = 0;
6     scanf("%d", &val);
7     if(val % 2 != 0){
8         printf("%d isn't multiple of 2.\n", val);
9         isntSixMuti = isntSixMuti + 1;
10    }
11    if(val % 3 != 0){
12        printf("%d isn't multiple of 3.\n", val);
13        isntSixMuti = isntSixMuti + 1;
14    }
15    if(isntSixMuti){
16        printf("%d isn;t multiple of 6.\n", val);
17    }
18 }
```

"D:\codeblocks\boolean algebra & if else.exe"

```
15
15 isn't multiple of 2.
15 isn;t multiple of 6.
```

"D:\codeblocks\boolean algebra & if else.exe"

```
35
35 isn't multiple of 2.
35 isn't multiple of 3.
35 isn;t multiple of 6.
```

關係運算子

注意，C 語言的關係運算子
不能 像數學一樣連續比較：

```
1 //boolean algebra & if else
2 #include <stdio.h>
3
4 int main(){
5     int val;
6     scanf("%d", &val);
7     if(0 < val < 5){ //Logic Error
8         printf("%d is between 0 and 5\n", val);
9     }
10 }
11
12 "D:\codeblocks\boolean algebra & if else.exe"
```

10
10 is between 0 and 5

關係運算子

$0 < \text{val} < 5$, $\text{val} = 10$ 時的運算:

$\Rightarrow 0 < 10 < 5$

$\Rightarrow \underline{0 < 10} < 5$

$\Rightarrow 1 \text{ (true)} < 5$

$\Rightarrow \underline{1 < 5}$

$\Rightarrow 1 \text{ (true)}$

要解決這個問題，
需要用到下一節介紹的邏輯運算子。

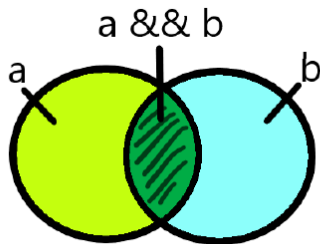
邏輯運算子

計算且、或、非等邏輯運算

邏輯運算子

| 名稱 | 運算子 | 語法 |
|--------|-----|--------|
| 邏輯 AND | && | a && b |
| 邏輯 OR | | a b |
| 邏輯 NOT | ! | !a |

邏輯 AND (且)



| $\begin{smallmatrix} a \\ b \end{smallmatrix}$ | 0 | 1 |
|--|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

邏輯 AND 範例：

```
1 //boolean algebra & if else
2 #include <stdio.h>
3
4 int main(){
5     int val;
6     scanf("%d", &val);
7     if(0 < val && val < 5){
8         printf("%d is between 0 and 5\n", val);
9     }
10    else{
11        printf("%d is not between 0 and 5\n", val);
12    }
13 }
```

"D:\codeblocks\boolean algebra & if else.exe"

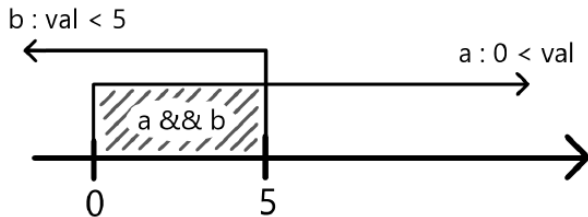
```
3
3 is between 0 and 5
```

"D:\codeblocks\boolean algebra & if else.exe"

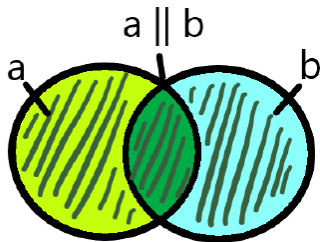
```
10
10 is not between 0 and 5
```

邏輯運算子

$0 < \text{val} \ \&\& \ \text{val} < 5$ 邏輯解釋：



邏輯 OR (或)



| $\begin{array}{c} a \\ \backslash \\ b \end{array}$ | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 1 |

邏輯 OR 範例：

```
1 //boolean algebra & if else
2 #include <stdio.h>
3
4 int main(){
5     int val;
6     scanf("%d", &val);
7     if(val < 2 || 7 < val ){
8         printf("%d is not between 2 and 7\n", val);
9     }
10    else{
11        printf("%d is between 2 and 7\n", val);
12    }
13 }
```

"D:\codeblocks\boolean algebra & if else.exe"

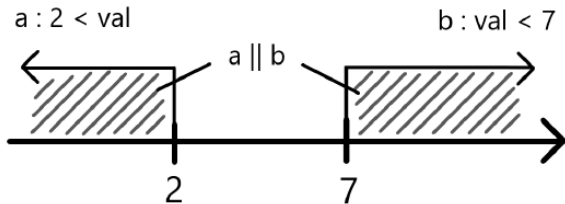
1
1 is not between 2 and 7

"D:\codeblocks\boolean algebra & if else.exe"

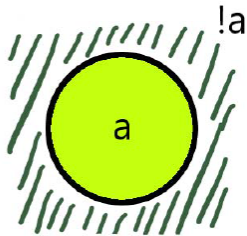
5
5 is between 2 and 7

邏輯運算子

$val < 2 \parallel 7 < val$ 邏輯解釋：



邏輯 NOT (非)



| | | |
|----|---|---|
| a | 0 | 1 |
| !a | 1 | 0 |

邏輯 NOT 範例：

```
1 //boolean algebra & if else
2 #include <stdio.h>
3
4 int main(){
5     int val;
6     scanf("%d", &val);
7     if(!(2 < val && val < 7)){
8         printf("%d is not between 2 and 7\n", val);
9     }
10    else{
11        printf("%d is between 2 and 7\n", val);
12    }
13 }
```

"D:\codeblocks\boolean algebra & if else.exe"

```
10
10 is not between 2 and 7
```


三元運算子 (:?)

一個有時候可以代替 if else 用的運算子

三元運算子

語法：

條件式 ? 成立傳回值 : 失敗傳回值

例如原本 if ... else 的敘述：

```
if (a > b)
    max = a;
else
    max = b;
```

等價於

```
max = (a > b) ? a : b;
```

三元運算子

原本 if ... else if ... else 的敘述：

```
if (a > b)
    cmp = 1;
else if (a < b)
    cmp = -1;
else
    cmp = 0;
```

等價於

```
cmp = (a > b) ? 1 : (a < b) ? -1 : 0;
```

重複使用太多三元運算子會導致程式難以閱讀，
盡量避免使用太多三元運算子。

switch 多重選擇敘述式

switch 多重選擇敘述式

有時候我們會需要判斷一個變數的不同數值情況：

```
1 //switch case
2 #include <stdio.h>
3
4 int main(){
5     int day;
6     scanf("%d", &day);
7     if (day == 1)
8         printf("MON\n");
9     else if (day == 2)
10        printf("TUE\n");
11    else if (day == 3)
12        printf("WED\n");
13    else if (day == 4)
14        printf("THU\n");
15    else if (day == 5)
16        printf("FRI\n");
17    else
18        printf("WEEKEND\n");
19 }
```

switch 多重選擇敘述式

當一個判斷式都在判斷同一個變數時，
可以選擇格式比較工整的 switch case

```
7   if (day == 1){
8       printf("MON\n");
9   }
10  else if (day == 2){
11      printf("TUE\n");
12  }
13  else if (day == 3){
14      printf("WED\n");
15  }
16  else if (day == 4){
17      printf("THU\n");
18  }
19  else if (day == 5){
20      printf("FRI\n");
21  }
22  else
23      printf("WEEKEND\n");
24  }
```

```
7   switch(day){
8       case 1:
9           printf("MON\n");
10          break;
11       case 2:
12           printf("TUE\n");
13          break;
14       case 3:
15           printf("WED\n");
16          break;
17       case 4:
18           printf("THU\n");
19          break;
20       case 5:
21           printf("FRI\n");
22          break;
23       default:
24           printf("WEEKEND\n");
25  }
```

switch 多重選擇敘述式

switch case 格式：

```
switch (變數) {  
    case 符合的結果1:  
        ...  
        break;  
    case 符合的結果2:  
        ...  
        break;  
    case 符合的結果3:  
        ...  
        break;  
    default: //若不符合上面的case  
        ...  
}
```

break 的意思是離開 switch 的大括號，如果全部沒有加上 break，會一直執行到大括號結束為止。

switch 多重選擇敘述式

執行序會從符合的 case 進入，並碰到 break 離開：

```
1 //switch case
2 #include <stdio.h>
3
4 int main(){
5     int val;
6     scanf("%d", &val);
7     switch (val){
8         case 1:
9             printf("> A\n");
10        case 2:
11            printf("> B\n");
12        case 3:
13            printf("> C\n");
14            break;
15        case 4:
16        case 5:
17            printf("> OOX\n");
18    }
19 }
```

"D:\codeblocks\switch case.exe"

```
2
> B
> C
```

"D:\codeblocks\switch case.exe"

```
4
> OOX
```


參考資料： Deitel, H. M., & Deitel, P. J. (2015). C: How to program.
Upper Saddle River, N.J: Prentice Hall.