

由片語學習C程式設計

台灣大學資訊工程系劉邦鋒著

台灣大學劉邦鋒老師講授

August 19, 2016

判斷

```
if then  
if then else  
if then else if  
conditional expression  
switch
```

第四單元

判斷

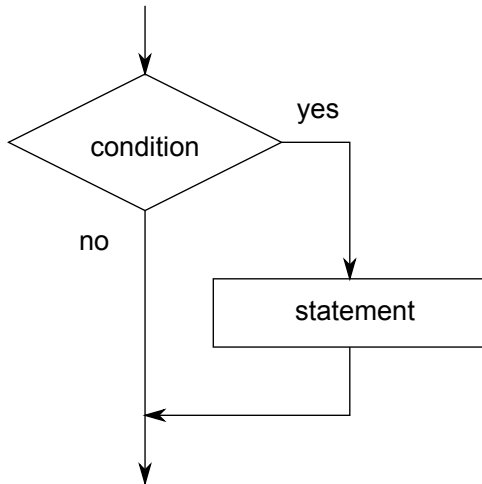
判斷

- 到目前為止我們寫的程式都是一行一行從頭執行到尾。
- 但有時我們需要依據某些條件，而執行不同部分的程式，這時我們就需要 **if** 判斷。

片語 1: if then 判斷

```
1  if (condition)  
2      statement;
```

- 根據 `condition` 決定是否執行 `statement`。
- 如果 `condition` 為 真，亦即非零，則會執行 `statement`，
- 如果 `condition` 為 偽，亦即零，則不會執行 `statement`。



範例程式 2: (if-then.c) 只顯示正數

```
1 #include <stdio.h>
2 main()
3 {
4     int i;
5     scanf("%d", &i);
6     if (i > 0)
7         printf("%d\n", i);
8 }
```

輸入

1

10

輸出

1

10

- 執行發現如果輸入為正數則 `printf` 會被執行。
- 如果輸入為零或負數則 `printf` 不會被執行。

片語 3: 複合敘述

```
1 {  
2     statement1;  
3     statement2;  
4     statement3;  
5 }
```

- **複合敘述** 將一組敘述合在一起當成一個敘述使用。
- 上例中用大括號 { } 將三個敘述包起來組成一個複合敘述。
- 三個敘述句尾都有分號。

片語 4: if 複合敘述

```
1  if (condition) {  
2      statement1;  
3      statement2;  
4      statement3;  
5  }
```

- 如果 condition 為 真，則依序執行 statement1，statement2，及 statement3。
- 如果 condition 為 偽，則不會執行。

範例程式 5: (product-positive.c) i 及 j 的乘積是否為正數

```
1  #include <stdio.h>
2  main()
3  {
4      int i;
5      int j;
6      scanf("%d", &i);
7      scanf("%d", &j);
8      if (i * j > 0) {
9          printf("%d\n", i);
10         printf("%d\n", j);
11     }
12 }
```

輸入

1 -4
2 5

輸出

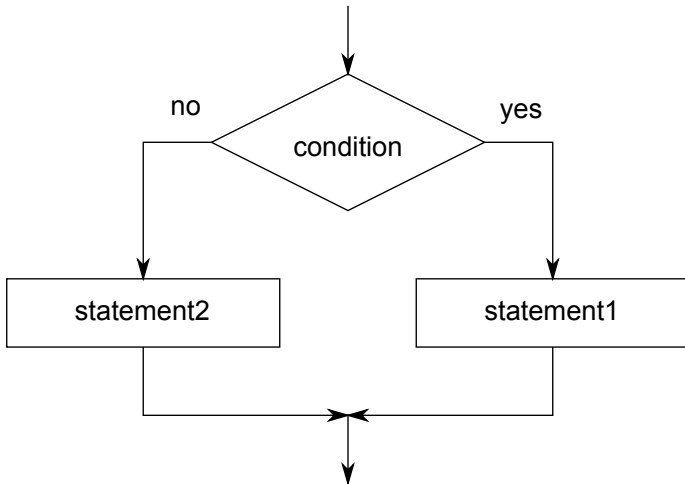
片語 6: if then else 判斷

```
1 if (condition)  
2     statement1;  
3 else  
4     statement2;
```

- 如果 condition 為 真，亦即非零，則會執行 statement1。
- 如果 condition 為 偽，亦即零，則會執行 statement2。

判斷

```
if then  
if then else  
if then else if  
conditional expression  
switch
```



範例程式 7: (abs.c) 顯示絕對值

```
1  #include <stdio.h>
2  main()
3  {
4      int i;
5      int k;
6      scanf("%d", &i);
7      if (i > 0)
8          k = i;
9      else
10         k = -i;
11     printf("%d\n", k);
12 }
```

顯示絕對值

- 讀入一整數 i ，並顯示其絕對值。
- 如果 $i > 0$ 為 **真**，則我們直接設 k 為 i ，
- 否則 $i \leq 0$ ，則我們會設 k 為 $-i$ 。
- 不管是那一種狀況， k 都會被設為 i 的絕對值。

輸入

1

-100

輸出

1

100

片語 8: 將i 及 j 的較大值設為 max

```
1  if (i > j)  
2      max = i;  
3  else  
4      max = j;
```

片語 9: 將 i 及 max 的較大值設為 max

```
1  if (i > max)
2      max = i;
```

範例程式 10: (max-3.c) 決定三個數中的最大值

```
1  #include <stdio.h>
2  main()
3  {
4      int i, j, k, max;
5      scanf("%d", &i);
6      scanf("%d", &j);
7      scanf("%d", &k);
8      if (i > j)
9          max = i;
10     else
11         max = j;
12     if (k > max)
13         max = k;
14     printf("%d\n", max);
15 }
```

輸入

```
1 4  
2 9  
3 5
```

輸出

```
1 9
```

片語 11: if then else 判斷

```
1  if (condition) {  
2      statement1;  
3      statement2;  
4      statement3;  
5  } else {  
6      statement4;  
7      statement5;  
8      statement6;  
9  }
```

- 我們也可以在 **then** 和 **else** 的部分都使用複合敘述，就像片語 11 一樣。
- 這樣我們就可以在 **then** 和 **else** 的部分使用多個敘述來完成工作。

範例程式 12: (sort-print.c) 讀入兩個整數並由小到大印出

```
1  #include <stdio.h>
2  main()
3  {
4      int i, j;
5      scanf("%d", &i);
6      scanf("%d", &j);
7      if (i > j) {
8          printf("%d\n", j);
9          printf("%d\n", i);
10     } else {
11         printf("%d\n", i);
12         printf("%d\n", j);
13     }
14 }
```


輸入

```
1 56  
2 21
```

輸出

```
1 21  
2 56
```

巢狀 if-then-else

片語 13: else if

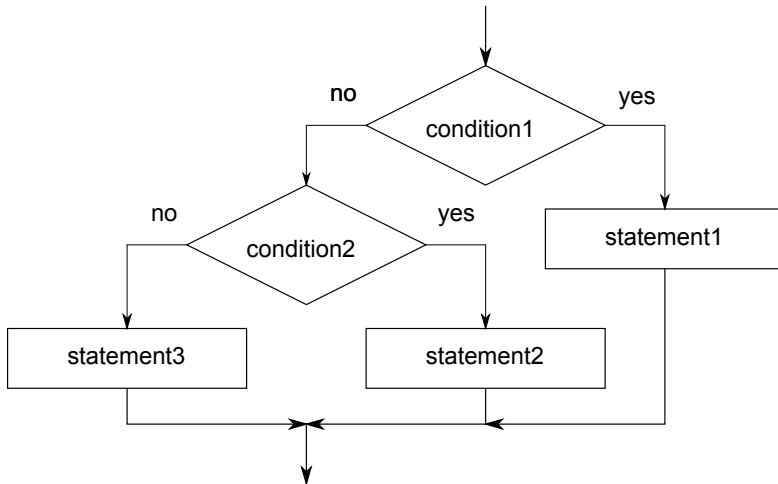
```
1  if (condition1)
2      statement1;
3  else if (condition2)
4      statement2;
5  else
6      statement3;
```

- 一個 if-then-else 的 else 部分是另外一個 if 敘述。
- 這種多層次的 if-then-else 結構稱為 **巢狀 if-then-else**。

- ① 如果 `condition1` 為 真，則只執行 `statement1` 並結束。
- ② 如果 `condition1` 為 偽，
 - ① 如果 `condition2` 為 真 只執行 `statement2` 並結束。
 - ② 如果 `condition2` 為 偽 只執行 `statement3` 並結束。

判斷

```
if then  
if then else  
if then else if  
conditional expression  
switch
```



利用變數存判斷結果

- 使用 `else if` 判定閏年。
- 我們把判定的結果存在 `k` 中。
 - 如果 `k` 為 1，則 `year` 為閏年。
 - 如果 `k` 為 0，則 `year` 為平年。

範例程式 14: (leap-year-else-if.c) 使用 else if 判定閏年

```
1  #include <stdio.h>
2  main()
3  {
4      int year, k;
5      scanf("%d", &year);
6      if (year % 400 == 0)
7          k = 1;
8      else if ((year % 4 == 0) && (year % 100 != 0))
9          k = 1;
10     else
11         k = 0;
12     printf("%d\n", k);
13 }
```

輸入

1

1900

輸出

1

0

範例程式 15: (bad-leap-year-else-if.c) 使用多層 if 判定閏年

```
1 #include <stdio.h>
2 main()
3 {
4     int year;
5     int k;
6     scanf("%d", &year);
7     if (year % 400 != 0)
8         if ((year % 4 == 0) && (year % 100 != 0))
9             k = 1;
10        else
11            k = 0;
12    else
13        k = 1;
14    printf("%d\n", k);
15 }
```


輸入

1 1377

輸出

1 0

風格要點

使用 **縮排** 將 `if` 及其對應的 `else` 對齊能增加可讀性。

學習要點

一個 `else` 會對應到往上最近的一個尚未對應的 `if`。

判斷式值

片語 16: 判斷式值

1 `(cond)? expression1 : expression2`

- 如果 `cond` 為真，則算式的值是 `expression1`，否則為 `expression2`。

範例程式 17: (cond-abs.c) 使用判斷式值計算絕對值

```
1  #include <stdio.h>
2  main()
3  {
4      int i;
5      int k;
6      scanf("%d", &i);
7      k = (i > 0)? i: -i;
8      printf("%d\n", k);
9  }
```

範例程式 18: (cond-max-3.c) 決定三個數中的最大值

```
1  #include <stdio.h>
2  main()
3  {
4      int i, j, k, max;
5      scanf("%d", &i);
6      scanf("%d", &j);
7      scanf("%d", &k);
8      max = (i > j)? i : j;
9      if (k > max)
10         max = k;
11     printf("%d\n", max);
12 }
```

範例程式 19: (cond-leap-year.c) 使用判斷式值決定潤年.

```
1  #include <stdio.h>
2  main()
3  {
4      int year, k;
5      scanf("%d", &year);
6      k = (year % 400 == 0)? 1 :
7          ((year % 4 == 0) && (year % 100 != 0)) ? 1 : 0;
8      printf("%d\n", k);
9      scanf("%d", &year);
10     k = ((year % 400 == 0) ||
11         ((year % 4 == 0) && (year % 100 != 0)))? 1 : 0;
12     printf("%d\n", k);
13 }
```

輸入

1	1980
2	1900

輸出

1	1
2	0

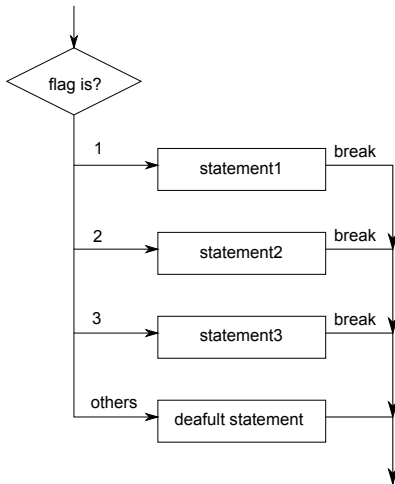
風格要點

如非必要應該盡量避免複雜的判斷式值。

片語 20: switch 判斷

```
1  switch (flag) {  
2  case 1:  
3      statement1;  
4      break;  
5  case 2:  
6      statement2;  
7      break;  
8      ...  
9  case n:  
10     statementn;  
11     break;  
12 default:  
13     default_statement;  
14 }
```

- 根據旗標變數 `flag` 的值決定要執行的 `statement`.



注意事項

- `flag` 必須是變數，而非算式。
- `case` 之後的必須是常數，不能是算式。
- `statement` 之後的 `break` 不可省。
- 可以把一開始的 `switch (flag)` 想像成一個 根據 `flag` 的值的“跳躍”。
- 當執行完對應的 `statement` 之後必須跳出 `switch`。

if as switch

片語 21: 使用 if 實作 switch。

```
1  if (flag == 1)
2      statement1;
3  else if (flag == 2)
4      statement2;
5      ...
6  else if (flag == n)
7      statementn;
8  else
9      default_statement;
```

範例程式 22: (switch-power.c) 計算 power

```
4  int i, j, power;  
5  scanf("%d", &power);  
6  scanf("%d", &i);  
7  switch (power) {  
8  case 1:  
9      j = i;  
10     break;  
11  case 2:  
12     j = i * i;  
13     break;  
14  case 3:  
15     j = i * i * i;  
16     break;  
17  default:  
18     j = 0;  
19  }  
20  printf("%d\n", j);
```

輸入

1 2
2 4

輸出

1 16

- 使用 `switch` 判斷 `power` 是 1、2、還是 3。
- 忘記在 `j = i * i` 之後加上 `break`。
- 當 `power` 為 1 及 3 時，程式都能正常執行。
- 但當 `power` 為 2 時，程式會正確的跳到 `case 2:`，但是因為下面沒有 `break`，所以會繼續執行，造成錯誤的結果。

範例程式 23: (switch-power-no-break.c) 計算 power

```
9  switch (power) {  
10 case 1:  
11     j = i;  
12     break;  
13 case 2:  
14     j = i * i;  
15 case 3:  
16     j = i * i * i;  
17     break;  
18 default:  
19     j = 0;  
20 }
```


判斷

```
if then  
if then else  
if then else if  
conditional expression  
switch
```

輸入

1 2
2 4

輸出

1 64

決定一年天數

- 利用 `switch` 根據 `flag` 的值跳躍到對應的 `case` 的特點判斷一個月的天數。
- 如果 `month` 是 1、3、5、7、8、10 或是 12，都會將 `days` 設為 31。
- `month` 是 4、6、9、或是 11，但都會將 `days` 設為 30。
- `month` 是 2，則會依據 `year` 將 `days` 設為 29 或 28。

範例程式 24: (month.c) 決定一個月的天數

```
4  int year, month, days;  
5  scanf("%d", &year);  
6  scanf("%d", &month);  
7  switch (month) {  
8  case 1: case 3: case 5: case 7: case 8:  
9  case 10: case 12:  
10     days = 31;  
11     break;  
12 case 4: case 6: case 9: case 11:  
13     days = 30;  
14     break;
```

```
16 case 2:
17     if ((year % 400 == 0) ||
18         (year % 4 == 0 && (year % 100 != 0)))
19         days = 29;
20     else
21         days = 28;
22     break;
23 default:
24     days = 0;
25 }
26 printf("%d\n", days);
```

輸入

1 1900
2 2

輸出

1 28

- 雖然 `switch` 可以不寫 `default`，但為了程式的正確性我們必須加上 `default` 用以處理例外的狀況。
- 使用者有可能輸入 1 到 12 之外的月份。程式必須能夠判別這種狀況，並將 `days` 設成一個特殊值 0，用以提醒使用者。

風格要點

為了程式的正確性我們必須在每個 `switch` 加上 `default` 用以處理例外的狀況。