

C++程式設計

條件判斷

迴圈

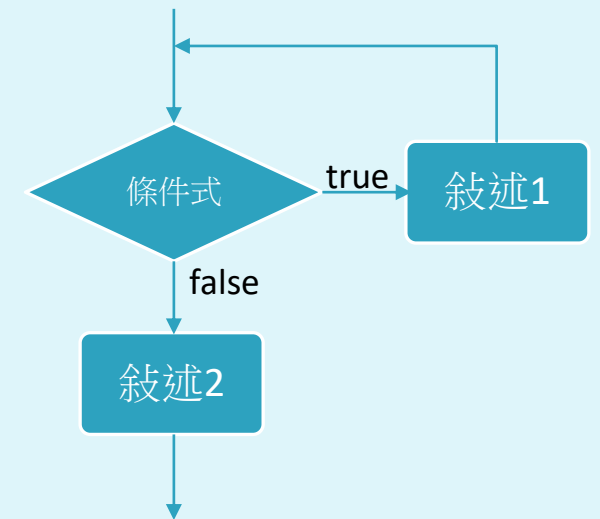
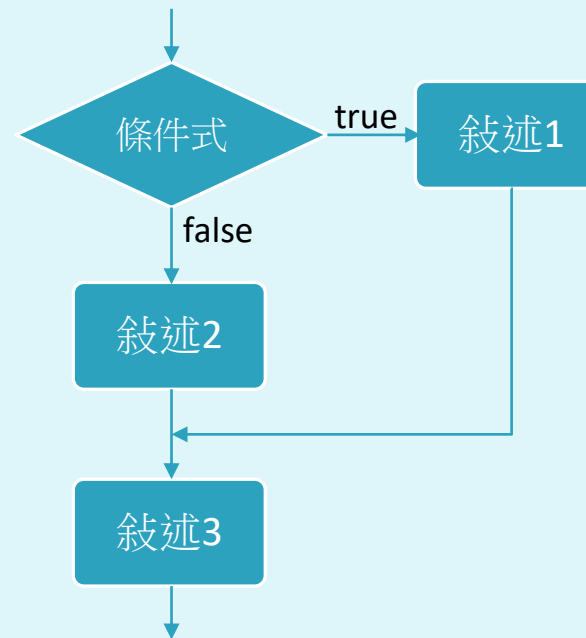
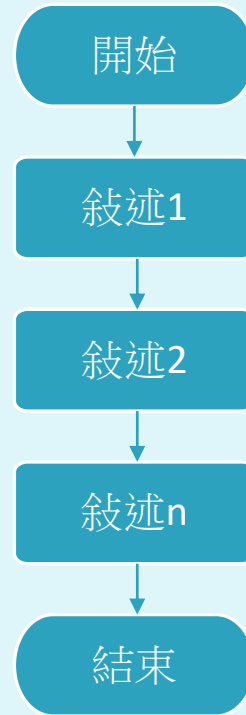
程式的結構

程式的結構包含：

1. 循序結構
2. 選擇結構
3. 重複結構

三種結構的共通點：

只有一個入口、一個出口



循序結構

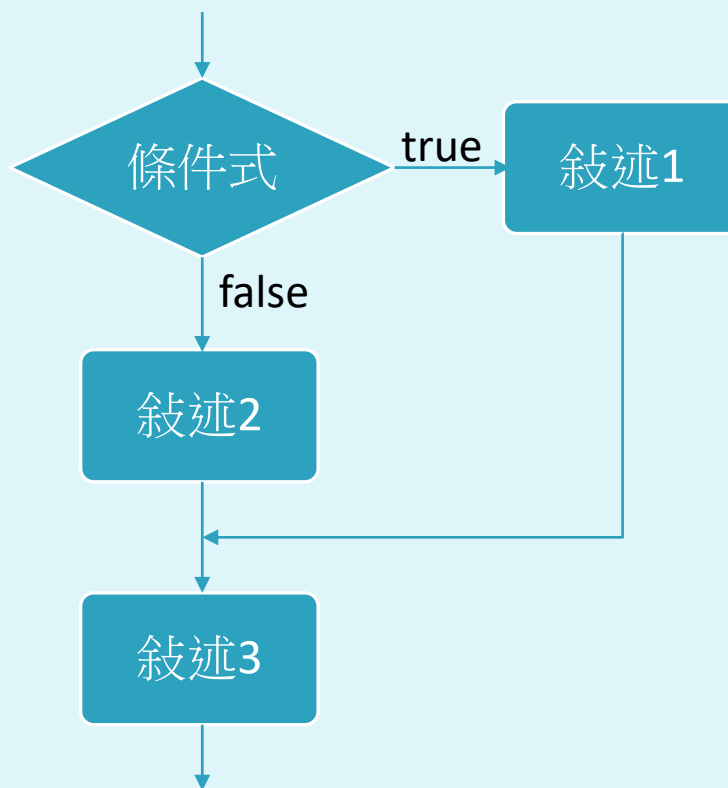
由上而下逐行執行



選擇結構

根據條件式成立與否，決定執行
哪些敘述

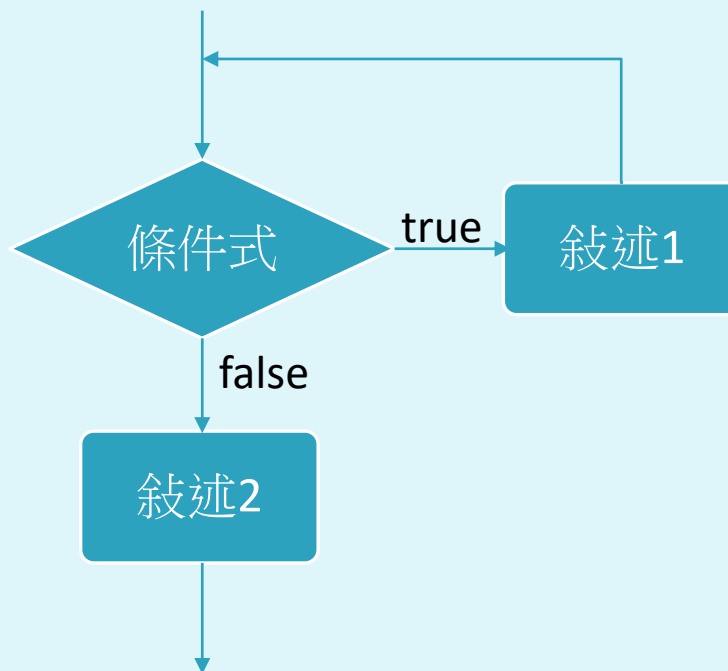
ex: if-else



重複結構

根據條件式成立與否，決定程式
區塊是否重複執行

ex: for 、 while 、 do while



選擇敘述

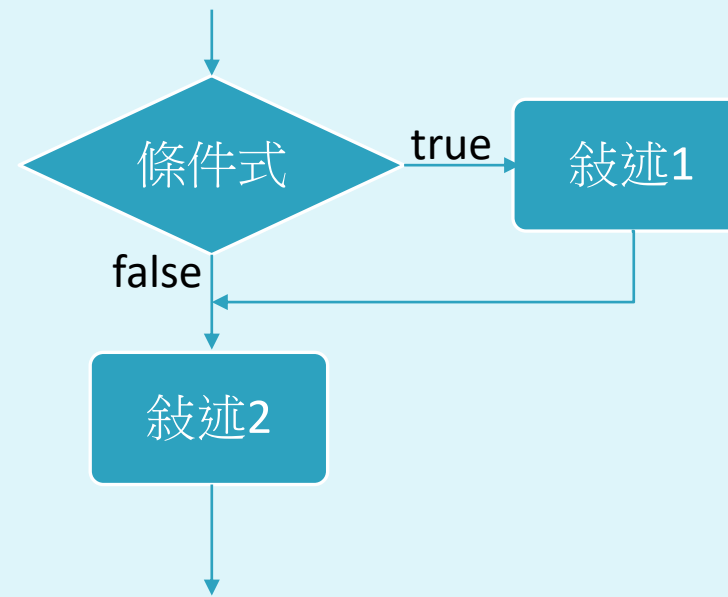
if敘述

- 格式:

```
if (條件式)
{
    敘述1;
}
敘述2;
```

- 說明:

1. 若條件成立(**true**), 則逐行執行大括號內敘述區塊. 否則跳至右大括號下一行敘述執行.
2. 若敘述區塊內只有**1**個敘述時, 可省略左、右大括號.



選擇敘述

```
1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      int w;
7      while(cin >> w)
8      {
9          if(w > 75)
10             cout << "體重過重!" << endl;
11             cout << "End." << endl;
12         }
13     return 0;
14 }
```

```
74
End.
75
End.
76
體重過重!
End.
```

選擇敘述

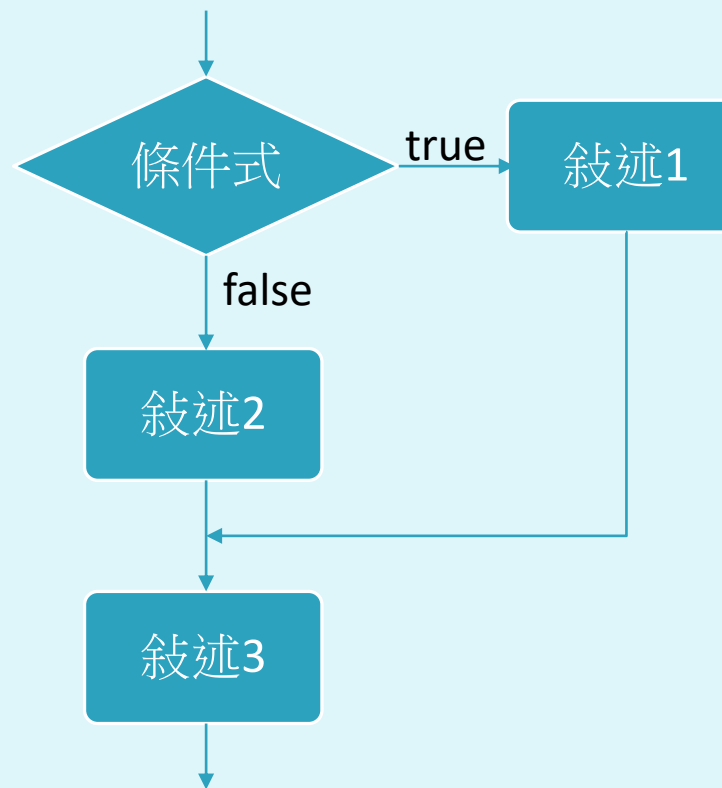
if-else敘述

- 格式:

```
if (條件式)
{
    敘述1;
}
else
{
    敘述2;
}
敘述3;
```

- 說明:

1. 若條件成立(true), 則執行if敘述區塊. 否則執行else敘述區塊.
2. 若敘述區塊只有1個敘述時, 可省略左、右大括號.



選擇敘述

```
1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      int w;
7      while(cin >> w)
8      {
9          if(w > 60 && w < 75)
10             cout << "體重標準!" << endl;
11         else
12             cout << "體重體重過輕或過重." << endl;
13         cout << "End." << endl;
14     }
15     return 0;
16 }
```

```
61
體重標準!
End.
55
體重體重過輕或過重.
End.
74
體重標準!
End.
75
體重體重過輕或過重.
End.
```

選擇敘述

條件運算子

- 說明: 意義同if-else敘述,但較為簡潔

條件運算子	說明
? :	依據條件式決定執行哪一個運算式

- 格式: `傳回值 = 條件式 ? 運算式1 : 運算式2`

- 若條件成立則傳回運算式1的結果,否則傳回運算式2的結果

選擇敘述

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a,b,m;
7      while(cin>>a>>b)
8      {
9          if(a<b)
10             m=a;
11          else
12             m=b;
13             cout << m << "比較小" << endl;
14     }
15     return 0;
16 }
```

```
1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      int a,b,m;
7      while(cin >> a >> b)
8      {
9          m = (a<b) ? a : b;
10         cout << m << "比較小" << endl;
11     }
12     return 0;
13 }
```

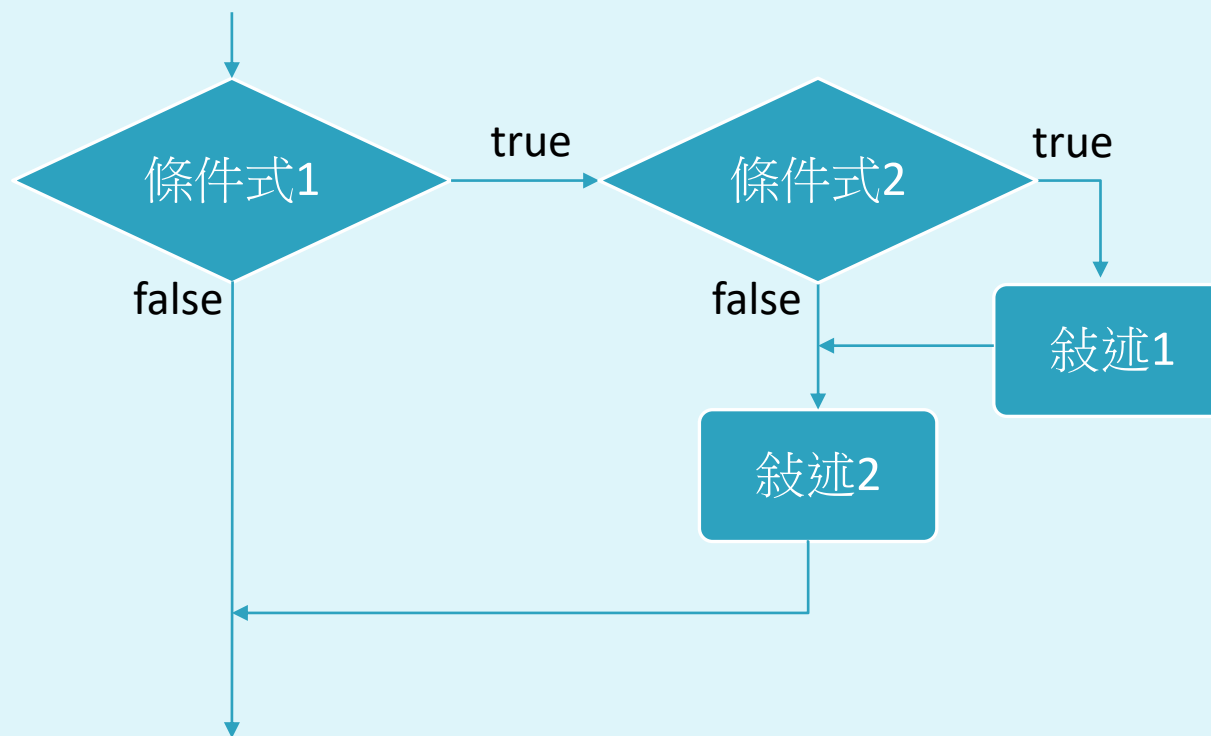
```
3 5
3比較小
9 1
1比較小
```

選擇敘述

巢狀if敘述

◦ 格式:

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

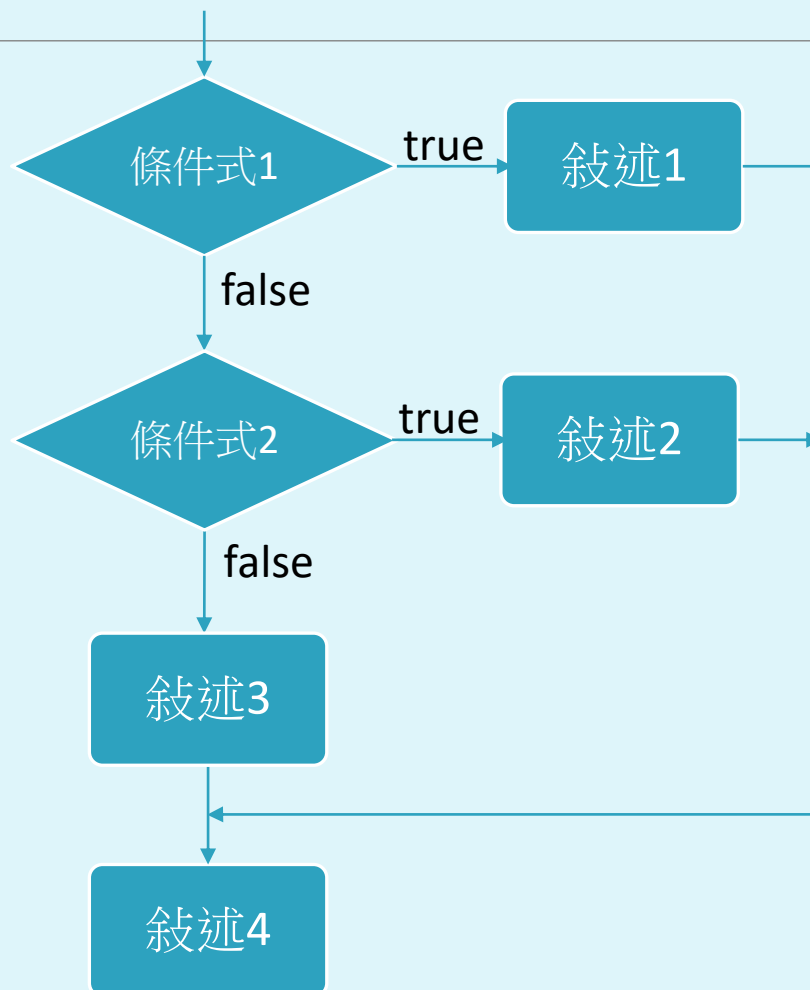


選擇敘述

巢狀if敘述

◦ 格式:

```
if (條件式1)
{
    敘述1;
}
else
{
    if(條件式2)
    {
        敘述2;
    }
    else
    {
        敘述3;
    }
}
```



選擇敘述

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int w;
6      while (cin >> w)
7      {
8          if (w > 75)
9              cout << "過重!" << endl;
10         else
11             if (w > 60)
12                 cout << "正常!" << endl;
13             else
14                 cout << "過輕!" << endl;
15     }
16 }
```

```
80
過重!
70
正常!
60
過輕!
```

switch敘述

格式:

```
switch(運算式)
```

```
{
```

```
    case 選擇值1:
```

```
        敘述主體1;
```

```
        break;
```

```
    case 選擇值2:
```

```
        敘述主體2;
```

```
        break;
```

```
    ...
```

```
    case 選擇值n:
```

```
        敘述主體n;
```

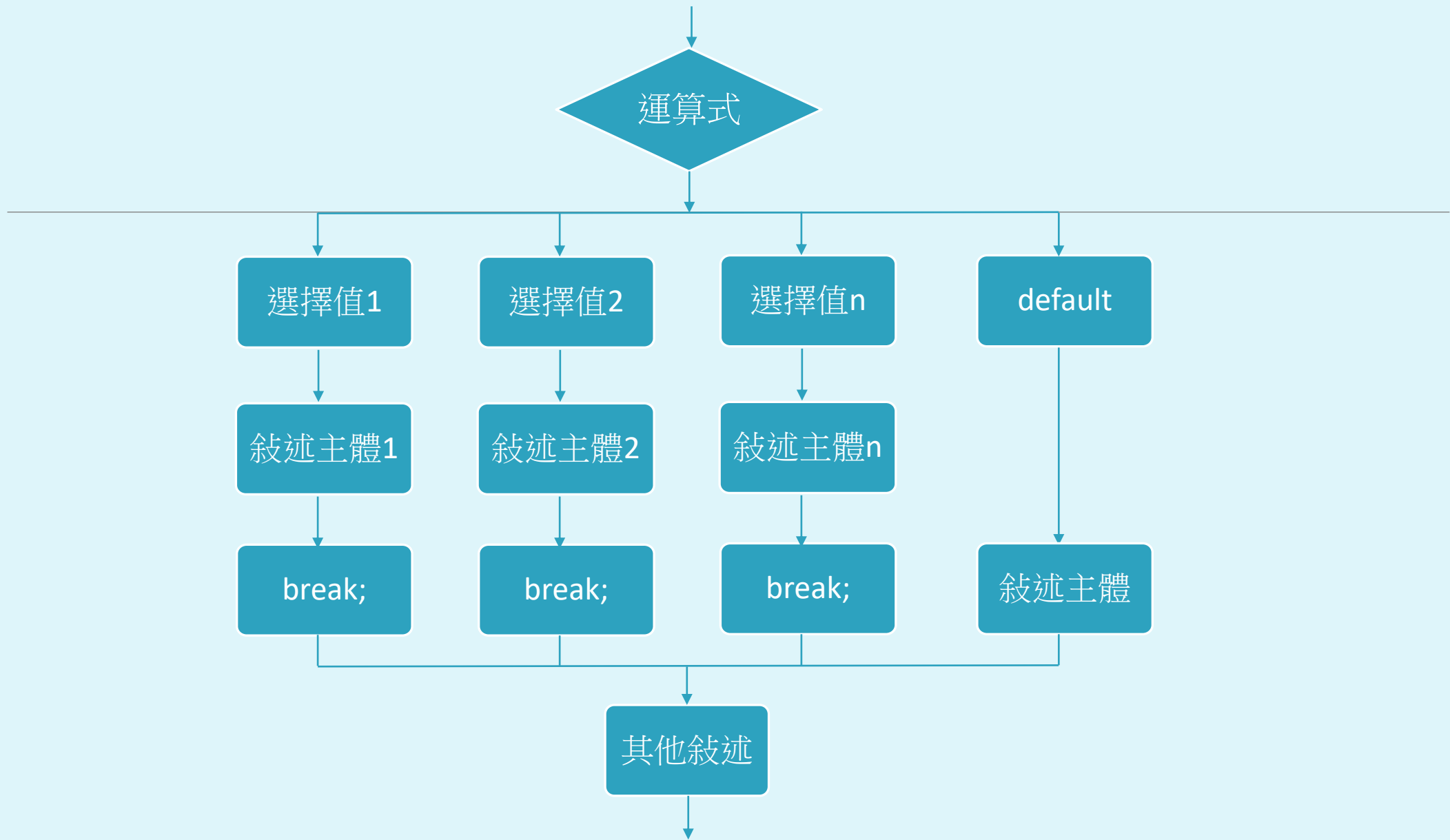
```
        break;
```

```
    default:
```

```
        敘述主體;
```

```
}
```

選擇值為字元或整數常數




```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      char ch;
7      while (cin >> ch)
8      {
9          switch (ch)
10         {
11             case 'w':
12                 cout << "up" << endl;
13                 break;
14             case 's':
15                 cout << "down" << endl;
16                 break;
17             case 'a':
18                 cout << "left" << endl;
19                 break;
20             case 'd':
21                 cout << "right" << endl;
22                 break;
23             default:
24                 cout << "..." << endl;
25         }
26     }
27 }

```

```

w
up
s
down
a
left
d
right
h
...
j
...

```

```

1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int month;
7      while (cin >> month)
8      {
9          switch (month)
10         {
11             case 3:
12             case 4:
13             case 5: cout << "春天" << endl;
14                     break;
15             case 6:
16             case 7:
17             case 8: cout << "夏天" << endl;
18                     break;
19             case 9:
20             case 10:
21             case 11: cout << "秋天" << endl;
22                     break;
23             case 12:
24             case 1:
25             case 2: cout << "冬天" << endl;
26                     break;
27             default:
28                 cout << "不存在!" << endl;
29         }
30     }
31     return 0;
32 }

```

```

3
春天
6
夏天
9
秋天
13
不存在!

```

```

1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      int month;
7      while (cin>>month)
8      {
9          switch(month)
10         {
11             case 3 ... 5:
12                 cout << "春天" << endl;
13                 break;
14             case 6 ... 8:
15                 cout << "夏天" << endl;
16                 break;
17             case 9 ... 11:
18                 cout << "秋天" << endl;
19                 break;
20             case 12:
21             case 1 ... 2:
22                 cout << "冬天" << endl;
23                 break;
24             default:
25                 cout << "不存在!" << endl;
26         }
27     }
28 }

```

```

3
春天
6
夏天
9
秋天
13
不存在!

```

廻卷

```
1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      int sum;
7      sum = 1+2+3+4+5;
8      cout << "1+2+...+5=" << sum << endl;
9
10     return 0;
11 }
```

1+2+...+5=15

迴圈 - for

for迴圈

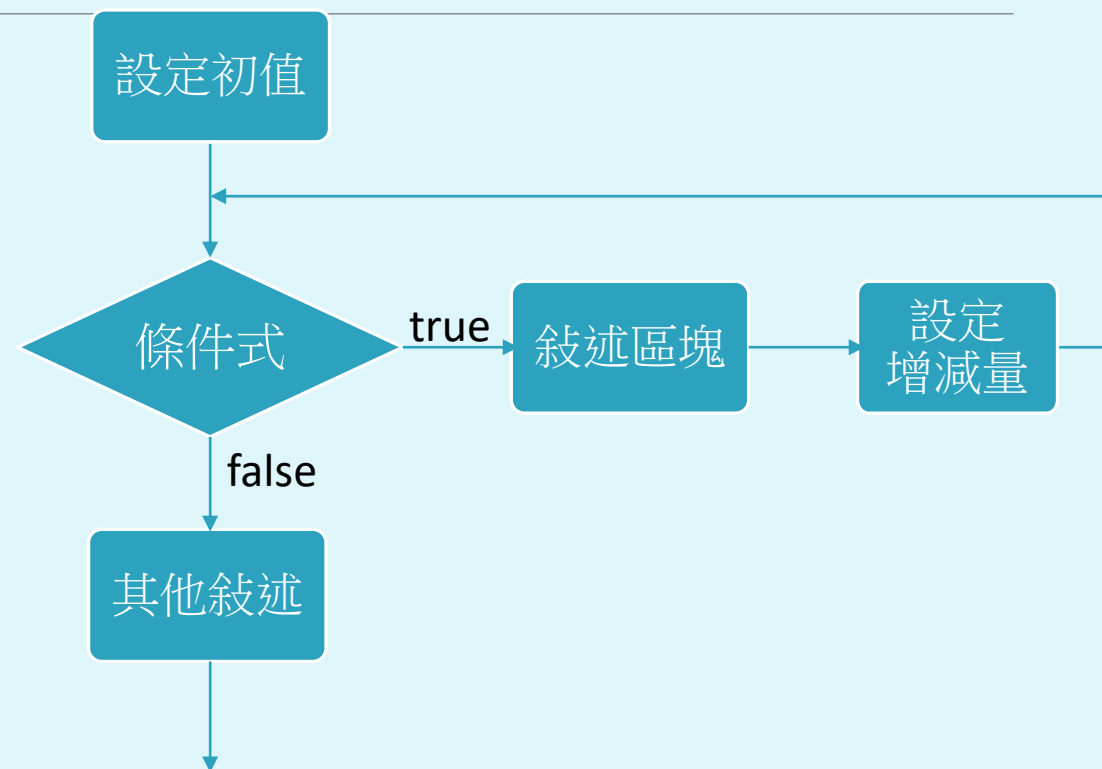
◦ 格式:

```
for (設定初值; 條件式; 設定增減量)
{
    敘述區塊;
}
```

◦ 說明:

1. 設定計次變數的初始值
2. 條件成立則執行敘述區塊,否則跳出迴圈
3. 執行完敘述區塊後,計次變數遞增(減),再回步驟2重新判斷

*若敘述區塊只有1個敘述時,可省略左、右大括號.



迴圈 - for

```
1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      int sum=0;
7      for(int i=1;i<=5;i++)
8          sum = sum + i;
9      cout << "1+2+...+5=" << sum << endl;
10     return 0;
11 }
```

1+2+...+5=15

迴圈次數	i	Sum
1	i=1	sum=0+1
2	i=2	sum=0+1+2
3	i=3	sum=0+1+2+3
4	i=4	sum=0+1+2+3+4
5	i=5	sum=0+1+2+3+4+5

變數i為區域變數,有效範圍僅在for迴圈內

迴圈 - while

while迴圈

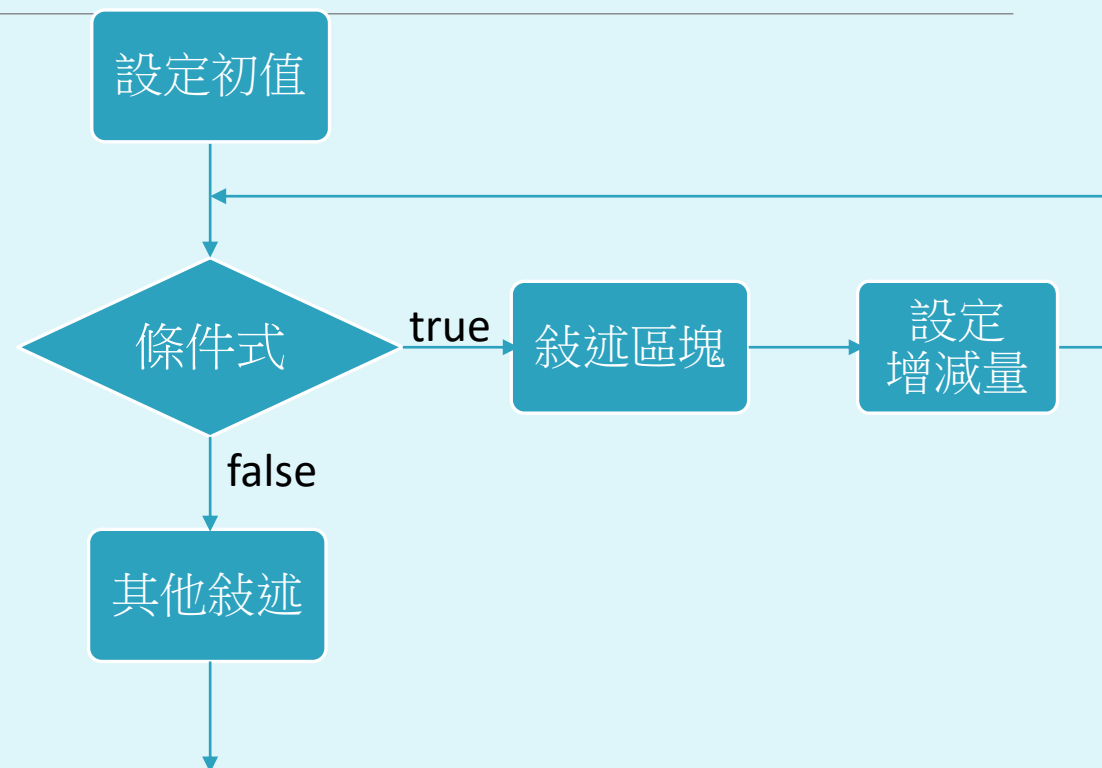
- 格式:

```
設定初值  
while (條件式)  
{  
    敘述區塊;  
    設定增減量;  
}
```

- 說明:

1. 設定計次變數的初始值
2. 條件成立則執行敘述區塊,否則跳出迴圈
3. 執行完敘述區塊後,計次變數遞增(減),再回步驟2重新判斷

*若無法事先得知迴圈執行次數,必須使用while迴圈



迴圈 - while

```
1  #include<iostream>
2  using namespace std;
3  int main()
4  {
5      int n;
6      cin>>n;
7      while (n!=8)
8      {
9          cout<<"Oops..."<<endl;
10         cin>>n;
11     }
12     cout<<"Ya!"<<endl;
13 }
```

```
10
Oops...
9
Oops...
8
Ya!
```


迴圈 - while

```
1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      int sum=0;
7      int i=1;
8      while(i<=5)
9      {
10         sum = sum + i;
11         i++;
12     }
13     cout << "1+2+...+5=" << sum << endl;
14     return 0;
15 }
```

迴圈次數	i	Sum
1	i=1	sum=0+ 1
2	i=2	sum=0+1+ 2
3	i=3	sum=0+1+2+ 3
4	i=4	sum=0+1+2+3+ 4
5	i=5	sum=0+1+2+3+4+ 5

1+2+...+5=15

迴圈 - do while

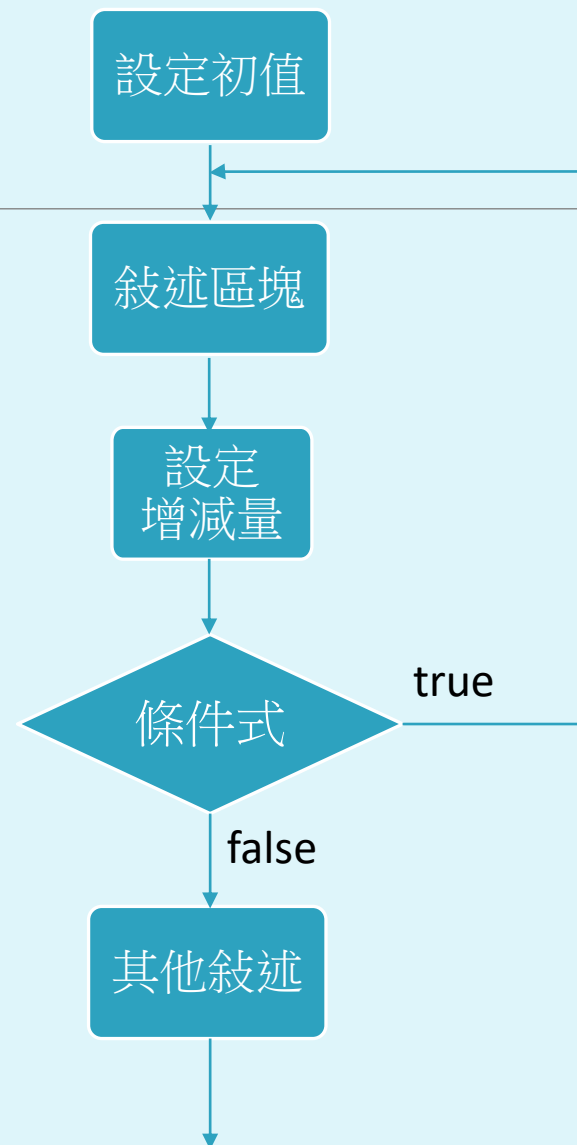
do while 迴圈

◦ 格式:

```
設定初值  
do  
{  
    敘述區塊;  
    設定增減量;  
} while (條件式);
```

◦ 說明:

1. 設定計次變數的初始值
2. 執行敘述區塊,計次變數遞增(減)
3. 條件成立則再回步驟2執行敘述區塊,否則跳出迴圈



迴圈 - do while

```
1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      string str;
7      do
8      {
9          cout << "輸入文字: ";
10         cin >> str;
11         cout << "你的輸入:" << str << endl;
12     } while (str != "goodbye");
13     return 0;
14 }
```

```
輸入文字: hello
你的輸入: hello
輸入文字: goodbye
你的輸入: goodbye
```

巢狀迴圈

```
1  #include<iostream>
2  using namespace std;
3
4  int main()
5  {
6      int i,j;
7      for(i=1;i<=3;i++)
8      {
9          for(j=1;j<=3;j++)
10             cout << i << "*" << j << "=" << i*j << "\t";
11             cout << endl;
12         }
13     return 0;
14 }
```

1*1=1	1*2=2	1*3=3
2*1=2	2*2=4	2*3=6
3*1=3	3*2=6	3*3=9

迴圈的跳離

break敘述: 直接跳離迴圈

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int n;
7      while (cin >> n)
8      {
9          if (n == 0)
10             break;
11          if (n % 4 == 0 && n % 100 != 0 || n % 400 == 0)
12             cout << "a leap year" << endl;
13          else
14             cout << "a normal year" << endl;
15      }
16      return 0;
17 }
```

```
1992
a leap year
1993
a normal year
1900
a normal year
2000
a leap year
0
```

```
Process returned 0 (0x0)   execution time : 10.258 s
Press any key to continue.
```

迴圈的跳離

continue敘述: 結束一輪迴圈

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for(int i=1;i<=10;i++)
7      {
8          if(i==5)
9              continue;
10         cout << i << ' ';
11     }
12 }
13
```

1 2 3 4 6 7 8 9 10

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      for(int i=1;i<=10;i++)
7      {
8          if(i==5)
9              break;
10         cout << i << ' ';
11     }
12 }
13
```

1 2 3 4