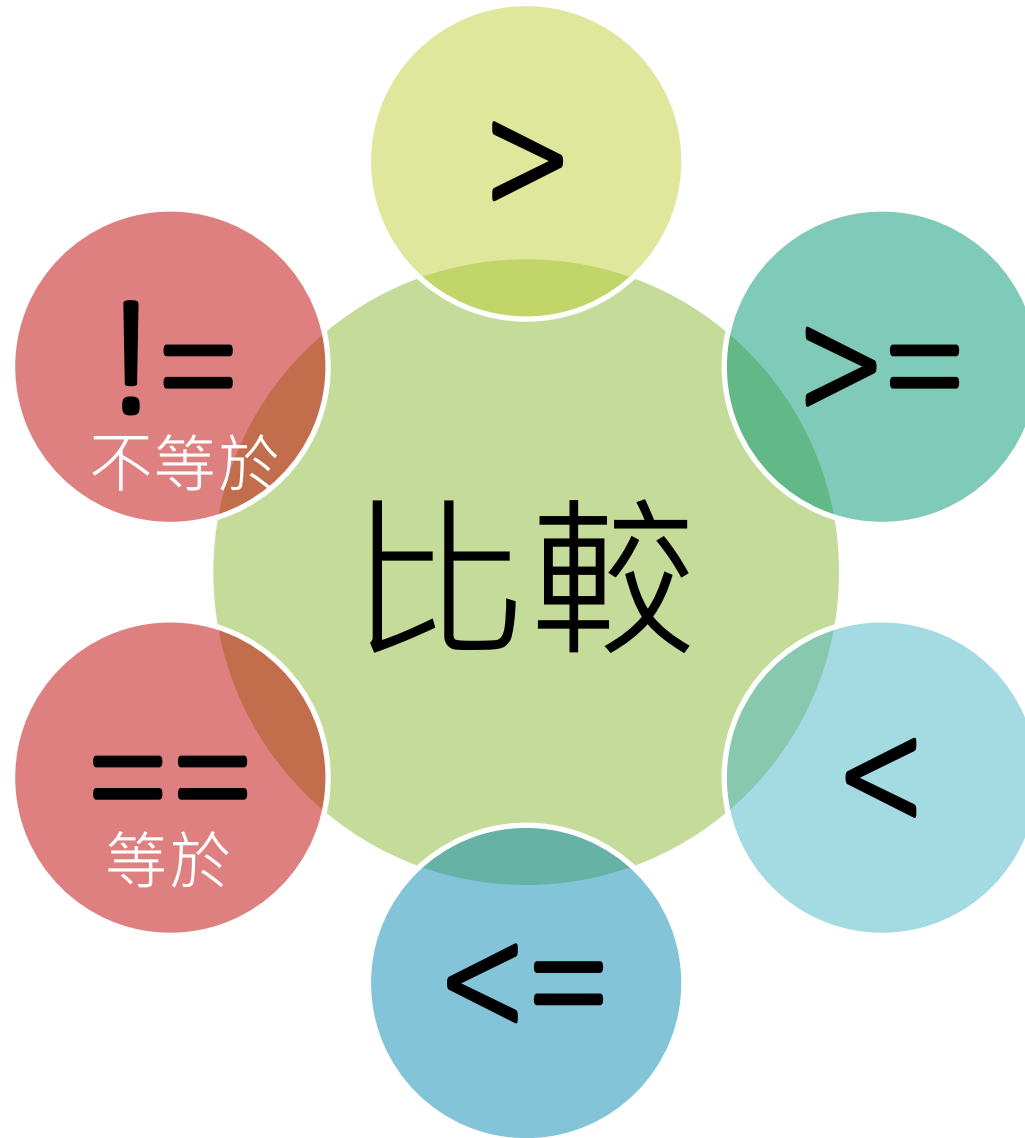


單一選擇



較量
才剛開始呢!



關係運算子(relational operator) 或是 比較運算子(comparison operator) (兩者相同)

屬於二元運算子(binary operator)

關係/比較運算子	運算式	運算子的意義
==	$x==y$	等於
!=	$x!=y$	不等於
>	$x>y$	大於
<	$x<y$	小於
>=	$x>=y$	大於等於
<=	$x<=y$	小於等於

運算式(expression): 由運算子(operator)與運算元(operand)組成。

寫一個程式，
判斷一個數是否為正數？

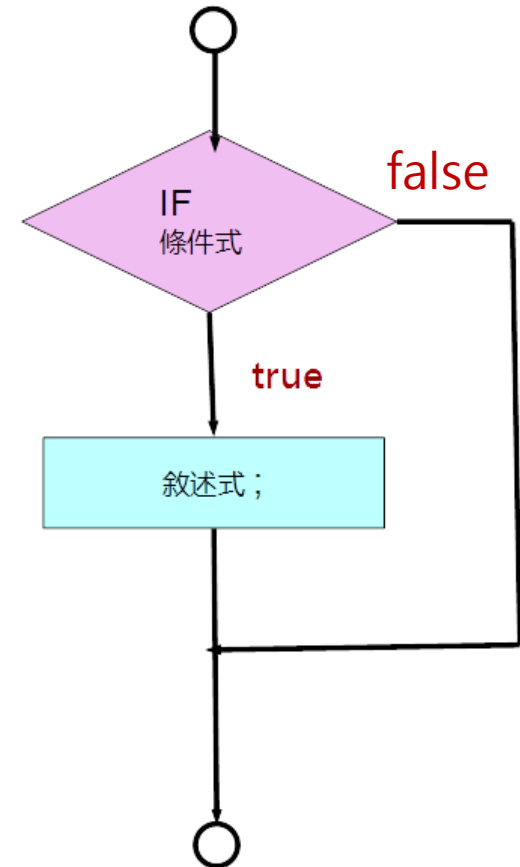
問題

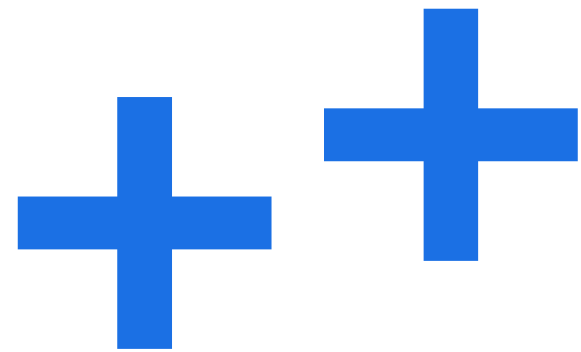


參考程式碼

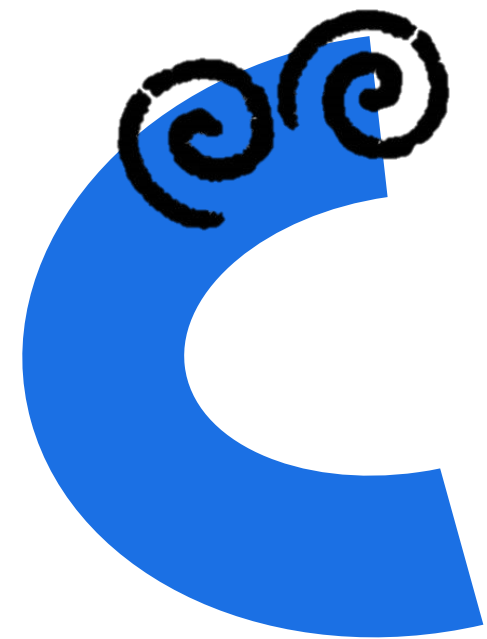
```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cin>>n;
    if(n>0){
        cout<<n<<"是正數"<<endl;
    }
    return 0;
}
```

這是一個句子。
如果 $a > 0$ ，印出正數。
如果只有一個句子，
{ }可省略。





延伸學習



if條件表達式後面，怎麼沒有分號作為結束？

- 句末要以分號 (;) 作結束的語法並沒有改變。
- 第7行與第8行是一起的，理論上可以寫成一行：
if(n>0) cout<<n<<"是正數"<<endl;
- 從程式可讀性的觀點來看，
還是分成兩行的寫法較佳。

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int n; /*宣告整數變數*/
6     cin>>n; /*輸入整數變數*/
7     if(n>0)
8         cout<<n<<"是正數"<<endl;
9     return 0;
10 }
```


if包含的條件式需要加大括號{ }嗎？

- 如果if後面的條件式只有一行，{ }可以被省略。

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int n; /*宣告整數變數*/
6     cin>>n; /*輸入整數變數*/
7     if(n>0)
8         cout<<n<<"是正數"<<endl;
9     return 0;
10 }
```

縮排有利於程式碼外觀與程式碼可讀性

- C是一種格式自由的語言, 編譯器只要求文法正確, 並不要求你用什麼樣子的格式來撰寫程式碼外觀。
- 注意: 雜亂無章的程式碼, 會影響除錯(debug)的效率。

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int n; /*宣告整數變數*/
6     cin>>n; /*輸入整數變數*/
7     if(n>0)
8     |     cout<<n<<"是正數"<<endl;
9     return 0;
10 }
```

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int n; /*宣告整數變數*/
6     cin>>n; /*輸入整數變數*/
7     if(n>0) {
8     |     cout<<n<<"是正數"<<endl;
9     |     }
10     return 0;
11 }
```

下圖寫法較好