



# 單一選擇

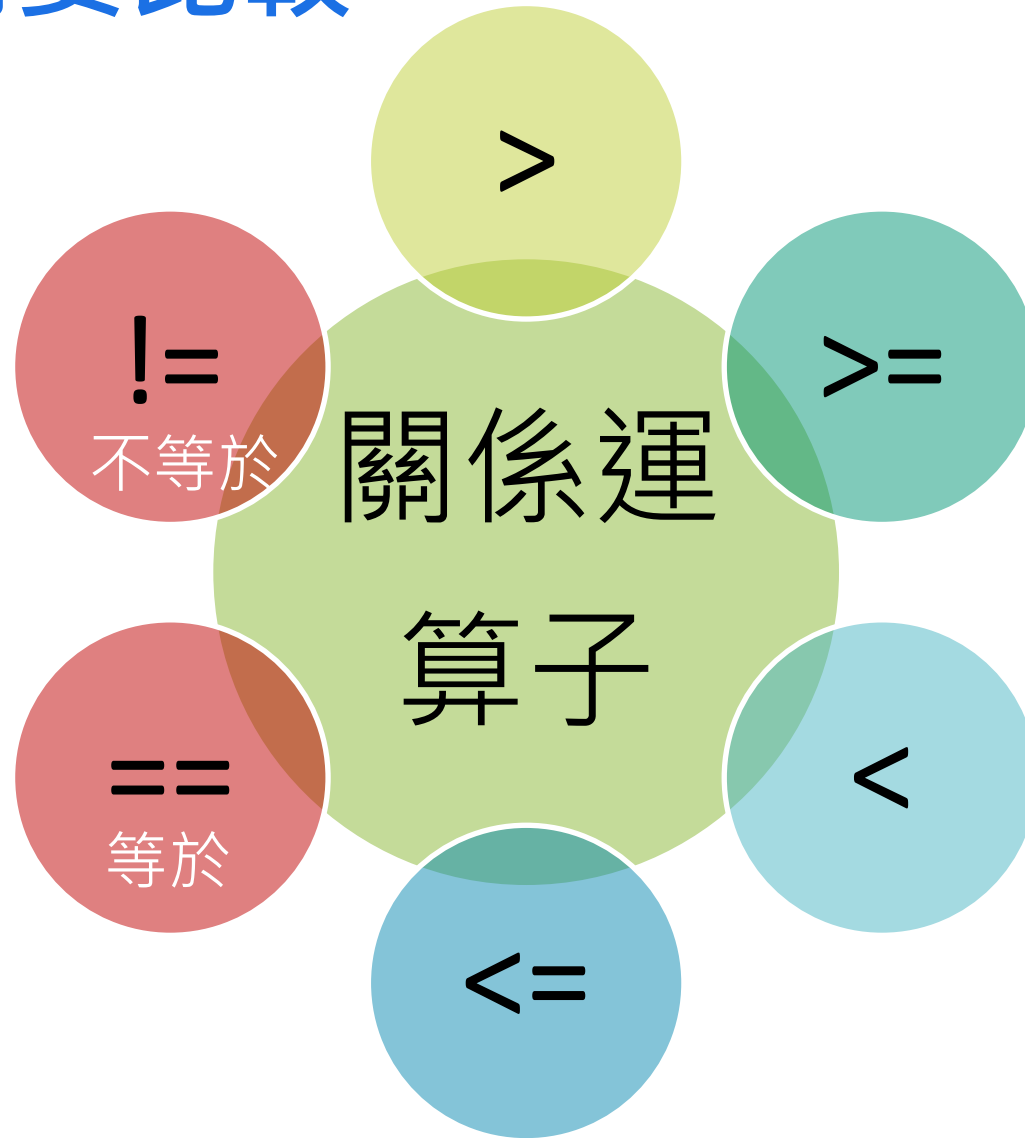


# C/C++ 語言提供三種選擇結構

選擇結構	描述
if選擇敘述式	單一選擇敘述式，只選擇或跳過一項動作
if...else選擇敘述式	雙重選擇敘述式，會取兩種不同動作之一
switch	依運算式的不同，選擇執行許多動作之一

本單元主要學習**if**選擇敘述式

# 選擇之前，需要比較



# 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。

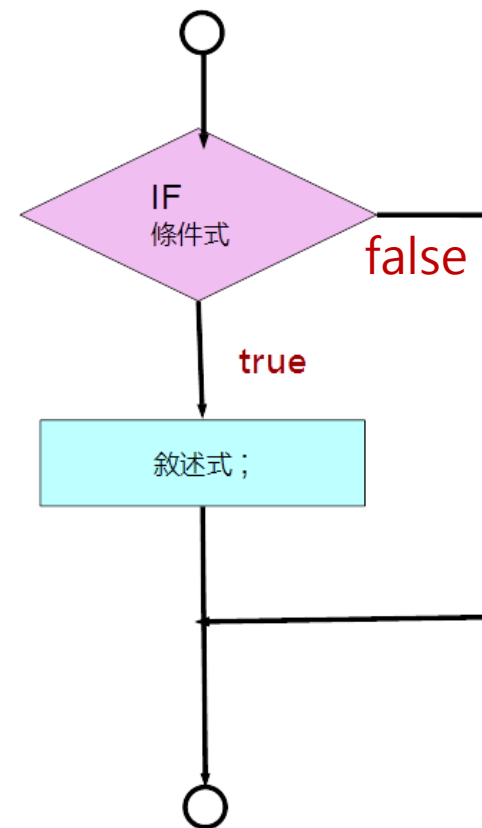
寫一個程式，  
判斷一個整數是正數嗎？

問題



# if選擇敘述式流程圖

選擇結構	描述
if選擇敘述式	單一選擇敘述式， 只選擇或跳過一項動作



# 參考程式碼

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int a;
```

寫法1

```
    scanf("%d",&a);
```

```
    if(a>0){
```

```
        printf("%d是正數\n",a);
```

```
    }
```

```
    return 0;
```

```
}
```

- 如果 $a > 0$ ，印出正數。
- `if(a>0) printf("%d是正數\n",a);`  
可以寫成一行，如寫法2
- if條件式後面只有一個句子的情形，`{}`可省略，如寫法3。

```
if(a>0) printf("%d是正數\n",a);
```

寫法2

```
if(a>0)
```

```
    printf("%d是正數\n",a);
```

寫法3



# 延伸的概念



# if條件式後面，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6行與第7行是一起的，理論上可以寫成一行：  
if(n>0) printf( "%d是正數" );
- 從程式可讀性的觀點來看，還是分成兩行的寫法較佳。

```
1 #include <stdio.h>
2 int main()
3 {
4     int n;
5     scanf("%d",&n);
6     if(n>0)
7         printf("%d是正數\n",n);
8     return 0;
9 }
```

# 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 <stdio.h>
2 int main()
3 {
4     int n;
5     scanf("%d",&n);
6     if(n>0)
7         printf("%d是正數\n",n);
8     return 0;
9 }
```

# 縮排有利於程式碼外觀與程式碼可讀性

- C語言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 <stdio.h>
2 int main()
3 {
4     int n;
5     scanf("%d",&n);
6     if(n>0)
7         printf("%d是正數\n",n);
8     return 0;
9 }
```

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

上圖寫法較好