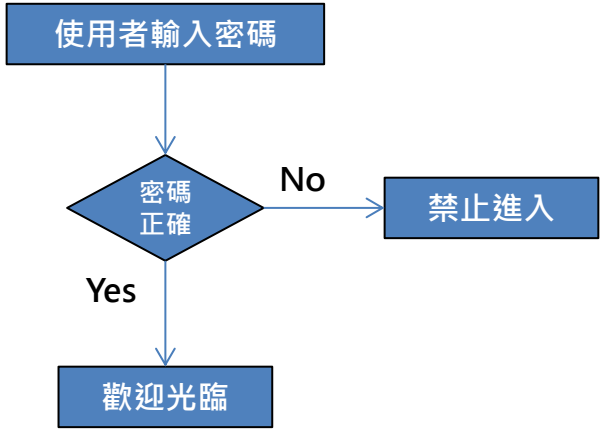
條件分支結構 - if

程式不是只能一直線的執行下去

- C/C++程式會從 main 函數裡的第一行開始，依序執行下去。
- BUT！這並不是一成不變的。
- 例如：使用者輸入密碼後，根據密碼正確與否，接下去要執行的動作顯然不會相同。

密碼判斷



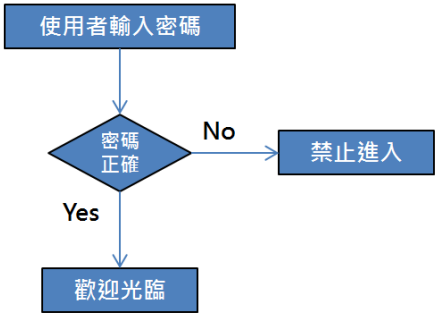

```
string password = "";

cout << "請輸入密碼:";
cin >> password;

if(password=="nlhs")
{
    cout << "歡迎光臨" << endl;
}
else
{
    cout << "禁止進入" << endl;
}
```

這裡有2個 = 符號

密碼判斷




```
string password = "";

cout << "請輸入密碼:";
cin >> password;

if(password=="nlhs")
{
    cout << "歡迎光臨" << endl;
}
else
{
    cout << "禁止進入" << endl;
}
```

if 後面的小括號裡是一個條件判斷式，結果不是 True 就是 False

結果是 True 執行這區塊

結果是 False 執行這區塊

※ 大括號內是一個程式區塊(Block)，可以包含多行程式碼。

這段程式的執行結果為何

```
string password = "";

cout << "請輸入密碼:";
cin >> password;

if(password=="nlhs")
{
    cout << "歡迎光臨" << endl;
}
else
{
    cout << "禁止進入" << endl;
}
cout << "程式結束" << endl;
```

(1)
```
請輸入密碼:nlhs
歡迎光臨
程式結束
```

(2)
```
請輸入密碼:nlhs
歡迎光臨
```

(3)
```
請輸入密碼:HaHaHaHa
禁止進入
程式結束
```

(4)
```
請輸入密碼:HaHaHaHa
禁止進入
```

這段程式的執行結果為何

```
string password = "";

cout << "請輸入密碼:";
cin >> password;

if(password=="nlhs")
{
    cout << "歡迎光臨" << endl;
}
else
{
    cout << "禁止進入" << endl;
}
cout << "程式結束" << endl;
```

(1) ✓

```
請輸入密碼:nlhs
歡迎光臨
程式結束
```

(2)

```
請輸入密碼:nlhs
歡迎光臨
```

(3) ✓

```
請輸入密碼:HaHaHaHa
禁止進入
程式結束
```

(4)

```
請輸入密碼:HaHaHaHa
禁止進入
```

else 可以是選擇性的(可以有，也可以沒有)

※ 假設 money 目前是 50。

```
cout << "請輸入帳號:";
cin >> id;

if(id=="Harry")
{
    money = money + 100;
}

cout << id << " 歡迎回到遊戲中" << endl;
cout << "你手邊的金幣數為 " << money << endl;
```

```
請輸入帳號:Peter
Peter 歡迎回到遊戲中
你手邊的金幣數為 50
```

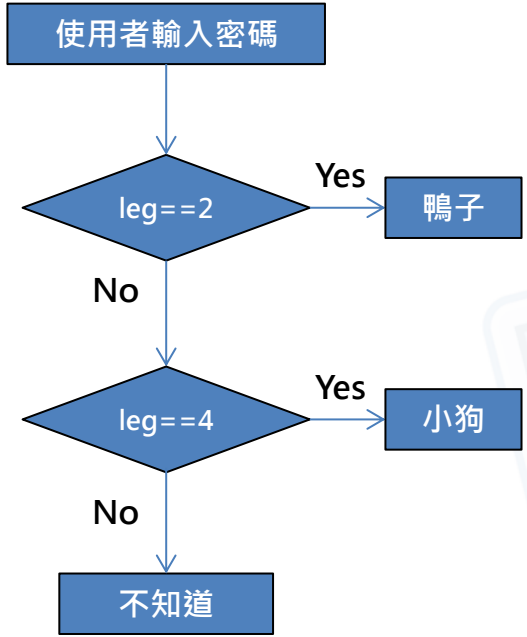
```
請輸入帳號:Harry
Harry 歡迎回到遊戲中
你手邊的金幣數為 150
```

else 後面可以再串接 if

```
// 猜動物
int leg = 0;

cout << "他有幾隻腳?";
cin >> leg;

if(leg==2)
{
    cout << "鴨子" << endl;
}
else if(leg==4)
{
   cout << "小狗" << endl;
}
else
{
   cout << "我不知道這是什麼。" << endl;
}
```

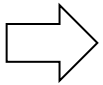


程式區塊內若只有一行，可省略大括號

```
// 猜動物
int leg = 0;

cout << "他有幾隻腳?";
cin >> leg;

if(leg==2)
{
    cout << "鴨子" << endl;
}
else if(leg==4)
{
   cout << "小狗" << endl;
}
else
{
   cout << "我不知道這是什麼。" << endl;
}
```



```
// 猜動物
int leg = 0;

cout << "他有幾隻腳?";
cin >> leg;

if(leg==2)
    cout << "鴨子" << endl;
else if(leg==4)
    cout << "小狗" << endl;
else
    cout << "我不知道這是什麼。" << endl;
```

注意：雖然它看起來比左邊的版本短，
但是執行速度並不會因此而較快

不可以隨便省略大括號

- 比較一下這兩段程式。

```
if( exp > 20000 )
{
    MaxHP = MaxHP + 10;
    MaxMP = MaxMP + 5;
}
```

如果 exp 大於 20000

則 MaxHP 增加 10 且 MaxMP 增加 5

如果 exp 沒有大於 20000

則 MaxHP 和 MaxMP 的值都不變

```
if( exp > 20000 )
    MaxHP = MaxHP + 10;
    MaxMP = MaxMP + 5;
```

如果 exp 大於 20000

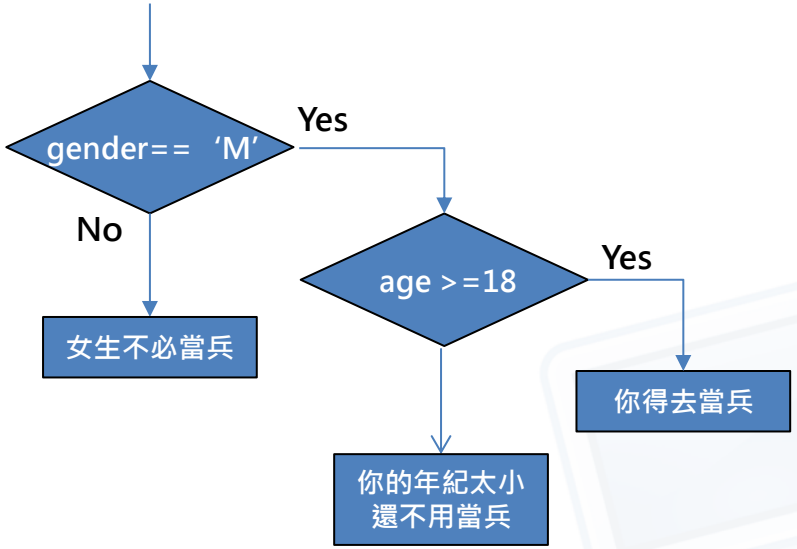
則 MaxHP 增加 10 且 MaxMP 增加 5

不管怎樣 MaxMP 都增加 5

程式區塊裡還可以有程式區塊

```
int age = 20;
char gender = 'M';

if (gender == 'M')
{
    if (age >= 18)
    {
        cout << "你得去當兵!" << endl;
    }
    else
    {
        cout << "你的年紀太小，還不用當兵!" << endl;
    }
}
else
{
    cout << "女生不必當兵!" << endl;
}
```



※ 這種結構叫做「巢狀(nested)結構」。

比較運算子

- a==b 中的 == 是用來比較a和b是否相等，這種運算子叫做「比較運算子」。
- 比較的結果是一個布林值(Boolean)，布林值只有兩種可能
  - True(真)
  - False(假)

比較運算子

- 常用的比較運算子如下

名稱	運算子	範例	範例運算結果
等於	==	2 == 3	False
不等於	!=	2 != 3	True
大於	>	2 > 3	False
小於	<	2 < 3	True
大於等於	>=	2 >= 3	True
小於等於	<=	2 <= 3	True

bool 型別

- int 型別
  - 值的範圍 ....., -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, ...
- bool 型別
  - 值的範圍 false, true
  - 電腦內部還是用數字來表示 false, true
    - 0 表示 false
    - 1 表示 true
    - 事實上在 C/C++ 中，非0即為 true

```
#include <iostream>

using namespace std;

int main()
{
    bool result;

    result  = 2>3;
    cout << result << endl;

    return 0;
}
```

很容易犯的錯

- 這個程式的輸出是？

```
int main()
{
    int theAnswer = 2;
    int whatYouGuess = 3;

    if(theAnswer = whatYouGuess)
        cout << "答對了!" << endl;
    else
        cout << "答錯了!..." << endl;

    return 0;
}
```

練習：帳密檢查

- 讀取帳號密碼後判斷，若帳號為NLHS 且密碼為1234，輸出“歡迎光臨”，否則輸出”身分驗證失敗”

```
#include <iostream>

using namespace std;

int main()
{
    string id = "";
    string password = "";

    cout << "請輸入帳號:";
    cin >> id;
    cout << "請輸入密碼:";
    cin >> password;

    // 把你的程式碼放在這裡

    return 0;
}
```

邏輯運算子

- 使用邏輯運算子，將用到比較運算子的運算式結合起來。
- 常用的邏輯運算子如下：

名稱	運算子	範例	範例運算結果
AND	&&	(2>2) && (4<5)	False
OR	||	(2>2) || (4<5)	True
NOT	!	!(2 > 3)	True

範例：帳號密碼判斷

```
string id = "";
string password = "";

cout << "請輸入帳號:";
cin >> id;
cout << "請輸入密碼:";
cin >> password;

if(id=="NLHS")
{
    if(password=="1234")
    {
        cout << "歡迎光臨" << endl;
    }
    else
    {
        cout << "身分驗證失敗" << endl;
    }
}
else
{
    cout << "身分驗證失敗" << endl;
}
```

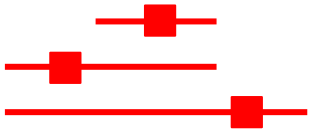
```
string id = "";
string password = "";

cout << "請輸入帳號:";
cin >> id;
cout << "請輸入密碼:";
cin >> password;

if(id=="NLHS" && password=="1234")
{
    cout << "歡迎光臨" << endl;
}
else
{
    cout << "身分驗證失敗" << endl;
}
```

運算子的優先順序

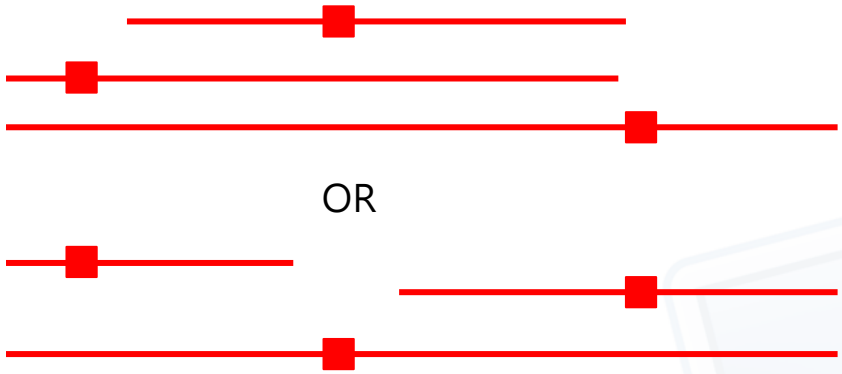
- 1 + 2 * 3 - 4



* 的優先權大於 + 和 -

+的優先權等於 -

- `if(id=="NLHS" && password=="1234")`



== 和 && 誰的優先權較高？

C++ 運算子的優先順序
http://en.cppreference.com/w/cpp/language/operator_precedence

範例：帳號密碼判斷

```
string id = "";
string password = "";

cout << "請輸入帳號:";
cin >> id;
cout << "請輸入密碼:";
cin >> password;

if(id=="NLHS" && password=="1234")
{
    cout << "歡迎光臨" << endl;
}
else
{
    cout << "禁止進入" << endl;
}
```

```
string id = "";
string password = "";

cout << "請輸入帳號:";
cin >> id;
cout << "請輸入密碼:";
cin >> password;

if( (id=="NLHS") && (password=="1234") )
{
    cout << "歡迎光臨" << endl;
}
else
{
    cout << "禁止進入" << endl;
}
```