

# 等號比較運算子 ==

實值型別相等：

```
using System;           //引用系統類別
using System.Linq.Expressions;
using System.Text;

namespace consoleapp01    //命名空間
{
    class Program          //Program類別
    {
        static void Main(string[] args)    //程式進入點
        {
            //      主程式內容

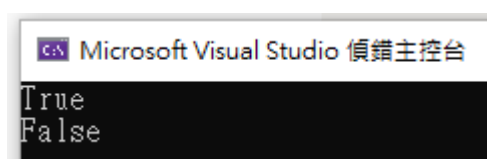
            int a = 1 + 2 + 3;
            int b = 6;

            //// output: True
            Console.WriteLine(a == b);

            char c1 = 'a';
            char c2 = 'A';

            // output: False 英文字母有分大小寫
            Console.WriteLine(c1 == c2);

        }
    }
}
```



參考型別相等：

這是官方文件說明

## 參考型別相等

根據預設，如果兩個非記錄參考類型運算元參考相同物件，則它們相等：

```
C# 複製

public class ReferenceTypesEquality
{
    public class MyClass
    {
        private int id;

        public MyClass(int id) => this.id = id;
    }

    public static void Main()
    {
        var a = new MyClass(1);
        var b = new MyClass(1);
        var c = a;
        Console.WriteLine(a == b); // output: False
        Console.WriteLine(a == c); // output: True
    }
}
```

如範例所示，使用者定義參考型別預設支援 `==` 運算子。不過，參考型別可以多載 `==` 運算子。若參考型別多載 `==` 運算子，請使用 [Object.ReferenceEquals](#) 方法檢查兩個該類型的參考是否參考相同的物件。

### 不過我改寫成這樣

```
using System;          //引用系統類別
using System.Linq.Expressions;
using System.Text;

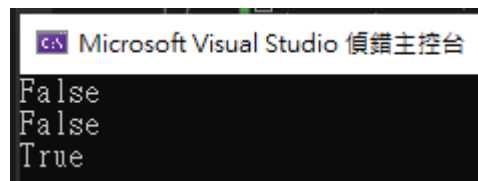
using System;          //引用系統類別
using System.Linq.Expressions;
using System.Text;

namespace consoleapp01    //命名空間
{
    public class Program
    {
        public class set_num
        {
            private int id;
            public set_num(int id)
            {
                this.id = id;
            }
        }
    }
}
```

```
public static void Main()
{
    var a = new set_num(1);
    var b = new set_num(1);

    var c = a;

    Console.WriteLine(a == b);
    Console.WriteLine(b == c);
    Console.WriteLine(c == a);
}
}
```



Microsoft Visual Studio 偵錯主控台

```
False
False
True
```