
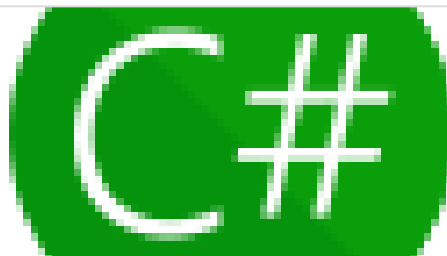


查詢 內存空間

指標相關運算子 - 存取記憶體和取值記憶體位置

了解使用指標時您可以使用的 C# 運算子。您可以使用這些運算子來存取記憶體、編制記憶體位置的索引，以及在記憶體位置取值儲存體

 <https://learn.microsoft.com/zh-tw/dotnet/csharp/language-reference/operators/pointer-related-operators>

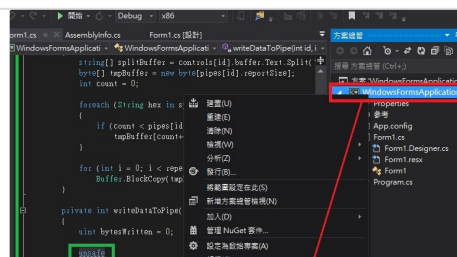


編譯器 允許執行unsafe (C# 查詢 內存空間的指令 一定要放在unsafe 內)

C# 編譯器選項 /unsafe

(1). 程式中遇到使用unsafe關鍵字，會發現編譯錯誤，請選擇"專案名稱"後按"滑鼠右鍵"，點選"屬性"。(2). 在"建置"標籤頁勾選"容許Unsafe程式碼"後就可以編譯成功。

 <http://science-boy-not-difficult.blogspot.com/2017/02/c-unsafe.html>



fixed語法說明：

重拾C#日常积累：Fixed初始化固定大小的指针_boonya的博客-CSDN博客

Fixed的说明Fixed 语句可防止垃圾回收器重新定位可移动的变量。Fixed 语句仅允许存在于不安全的上下文中。Fixed 可用于创建固定大小的缓冲区。fixed 语句将为托管变量设置一个指针，并在该语句的执行过程中“单边锁定”该变量。如果没有 fixed，指向可移动的托管变量的指针将几乎没有什么用处，因为垃圾回收可

 <https://blog.csdn.net/boonya/article/details/79930128>

建立一個陣列num_list，然後再整個賦值給another_list

C# 內要查詢內存位置，只能查詢單一元素，"無法直接查詢整個陣列"

所以查詢兩個陣列內第0個元素的內存位置

```
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

            unsafe
            {
                int[] num_list = new int[] { 50, 100, 150 };

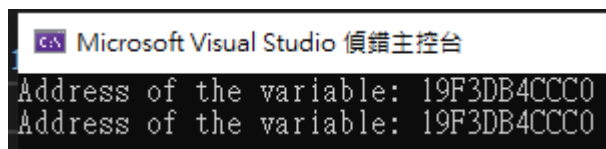
                int[] another_list = num_list;

                fixed (int* p = &num_list[0])

                    Console.WriteLine($"Address of the variable: {(long)p:X}");

                fixed (int* q = &another_list[0])
                    Console.WriteLine($"Address of the variable: {(long)q:X}");
            }
        }
    }
}
```

執行結果，位置相同



```
C:\> Microsoft Visual Studio 偵錯主控台
Address of the variable: 19F3DB4CCC0
Address of the variable: 19F3DB4CCC0
```


=====

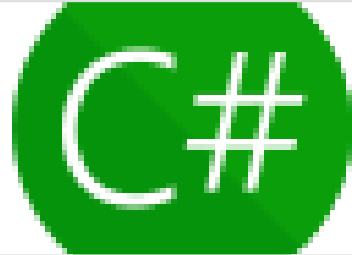
(long)p:X}的意思：

以十六進位制 輸出

如何在十六進位字串和數數值型別之間轉換 - C# 程式設計手冊


瞭解如何在十六進位字串和數數值型別之間轉換。請參閱程式碼範例，並檢視其他可用的資源。

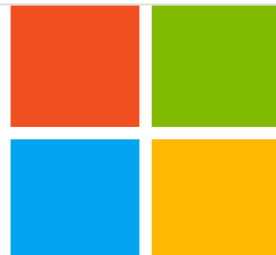
 <https://learn.microsoft.com/zh-tw/dotnet/csharp/programming-guide/types/how-to-convert-between-hexadecimal-strings-and-numeric-types>



Console.WriteLine Method (System)

Writes the specified data, followed by the current line terminator, to the standard output stream.

 <https://learn.microsoft.com/en-us/dotnet/api/system.console.writeline?view=net-8.0>



```
using System;
class Sample
{
    enum Color {Yellow = 1, Blue, Green};
    static DateTime thisDate = DateTime.Now;

    public static void Main()
    {
        Console.Clear();

        // Format a negative integer or floating-point number in various ways.
        Console.WriteLine("Standard Numeric Format Specifiers");
        Console.WriteLine(
            "(C) Currency: . . . . . {0:C}\n" +
            "(D) Decimal: . . . . . {0:D}\n" +
            "(E) Scientific: . . . . . {1:E}\n" +
            "(F) Fixed point: . . . . . {1:F}\n" +
            "(G) General: . . . . . {0:G}\n" +
            "    (default): . . . . . {0} (default = 'G')\n" +
            "(N) Number: . . . . . {0:N}\n" +
            "(P) Percent: . . . . . {1:P}\n" +
            "(R) Round-trip: . . . . . {1:R}\n" +
            "(X) Hexadecimal: . . . . . {0:X}\n",
            -123, -123.45f);
    }
}
```