默认的是double类型，可以不加d，也可以加，如果是float类型，必须加f，不然系统会默认为double类型

ex. float num = 3.1415926f;

--------------------------------------------------------

string s = Console.Readline();

--------------------------------------------------------

隱式轉換

int a = 1;

double b = 1.333;

a = b;

--------------------------------------------------------

Convert轉換(字符串要是數字)

string a = "10"

int b = Convert.ToInt32(a);

Console.WriteLine(b);

//10

--------------------------------------------------------

佔位符 {0} {1} {2}

Console.WriteLine("你好 {0}, {1}", a, b);

-----------------------------------------------------------

enum MyDate

{

Mon = 0; Tue = 1;

}

MyDate date = MyDate.Mon;

int a = (int)date;

------------------------------------------------------------------

數組類型[] 數組名

int[] my\_array;

數組名 = new 數組元素類型[元素個數]

my\_array = new int[3];

int[] a = new int[3] {1, 2, 3};

int[] a = new int[] {1, 2, 3};

int[] a = {1, 2, 3};

遍歷數組

foreach(數組數據類型 臨時變量 in 數組名) //var item in num

{

}

--------------------------------------------------------------------------------

函數申明

static 返回職類型 函數名 ([形式參數])

{

}

函數重載:相同函數名，不同傳入參數類型

----------------------------------------------------------------------------------------

函數高級參數

ref

函數外必須為變量賦值

形參和實參前都必須加ref關鍵字 A(ref num); static void A(ref int num)

out

可返回多個不同類型參數 Number(a,b, out max, out min);

-------------------------------------------------------------------------

[訪問修飾符] class 類名

{

}

實例化

類名 對象名 = new 類名();

----------------------------------------------------------------------

class Apple

{

private string color;

public string Color

{

get { return color;}

set { color = value;} //value違系統關鍵字

}

}

操作時

Apple apple = new Apple();

apple.Color = "red";

Console.WriteLine(apple.Color);

---------------------------------------------------------------------------------

多態

1.虛方法 2.抽象類 3.接口

1.虛方法: 父類中使用virtual關鍵字修飾 子類中用override進行重寫

2.抽象方法: 連方法體都沒有

抽象類不能被實例化

關鍵字: abstract;

抽象類中的抽象方法在繼承的子類中必須重寫!!!!!

3.接口

當抽象類中所有的方法都是抽象方法的時候，就可以把這個抽象類用另一種形式變現出，叫接口

接口名通常大寫I開頭

接口實現不需要abtract

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

1.list类型是引用类型

2.引用本身是类似于一个“保存地址的值变量”  
所以从方法外部传入引用到方法里，那么其实引用本身是复制了一份副本来给方法里使用的，只是说这个复制的引用副本和之前的引用的内容（也就是所指向的对象内存地址）是一样的，所以通过引用操作对象的数据时，可以看到2个引用都操作的同一个对象；但如果你是修改了引用副本本身的值内容（将该引用指向了一个新的对象的内存地址），那么是不会影响到之前方法外的那个引用的，所以修改后会发现2个引用所指向的对象不同了  
  
而如果对象引用参数前加上了ref，那么方法参数所传递的不再是引用的副本，而是引用的地址了（即通过引用的地址找到引用，再读出引用里保存的内存地址值，再根据则个地址值去找到真正要操作的对象），所以如果此时你再修改这个引用的值时，会根据引用的地址找到方法外的那个引用，然后修改其内容，所以会发现方法外的引用也会指向新的对象了

3这里有三段代码

你可以看看，体会一下：

<https://www.cnblogs.com/xcsn/p/7345165.html>

文黨註釋: /// 給類或方法註釋

C# decimal類型

Decimal d = 1.1M (需加後綴) +-7.9\*10^28/10^0~28

X??y

若x為null, 則計算結果為y , 若x不為null, 計算結果為x

[int.Parse()](https://msdn.microsoft.com/zh-tw/library/b3h1hf19%28v=vs.110%29.aspx)處理對象的是整數，所以遇到null時，會發生ArgumentNullException。如果數字字串為非整數，例如4.5,則會發生FormatException。所以這個method處理能力比較低，只能是**string**物件，且string資料要正確，不能為null，也不能為非整數。所以通常會使用[int.TryParse()](https://msdn.microsoft.com/zh-tw/library/f02979c7%28v=vs.110%29.aspx)以確認是否轉換成功，若回傳false，則需要有額外的邏輯去處理它。

[Convert.ToInt32()](https://msdn.microsoft.com/en-us/library/system.convert.toint32%28v=vs.110%29.aspx)有多個Overload method，所以可以處理各種型別的物件。處理到null會回傳0，並不會產生Exception。同時，如果有小數點後一位的話，會以4捨6入來處理整數的轉換(如果是5則回傳偶數的整數)。

Out ref 區別

一張含有 文字 的圖片

自動產生的描述

Parmas??????

static

一張含有 文字 的圖片

自動產生的描述

靜態方法只能使用靜態屬性

一張含有 文字 的圖片

自動產生的描述

ArrayList

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

List

一張含有 文字 的圖片

自動產生的描述

List 泛式規定儲存數據 減少裝箱開箱運算 減少強制轉換類型出錯

Dictionary



一張含有 文字 的圖片

自動產生的描述

劉老師

WPF 是 windows form 新版

ASP.NET Web forms(OLD) ASP.NET MVC(NEW)

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

一張含有 文字, 收據 的圖片

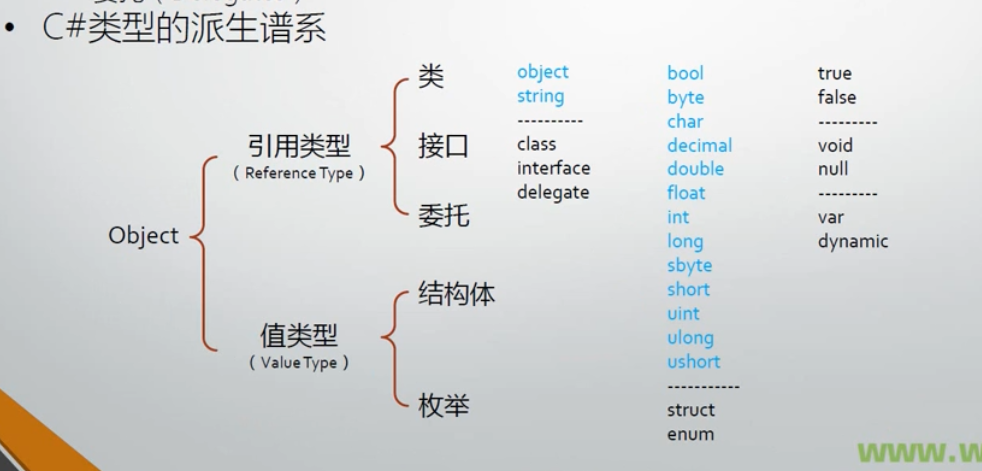
自動產生的描述

一張含有 文字 的圖片

自動產生的描述

一張含有 文字, 收據 的圖片

自動產生的描述



一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

Typeof操作符作用

一張含有 文字 的圖片

自動產生的描述

Var 使用

一張含有 文字 的圖片

自動產生的描述

New

一張含有 文字 的圖片

自動產生的描述

Checked unchecked 操作符

隱式轉換 顯示轉換

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

Operator explicit implicit



Public static explicit operator A(B b){}

A a= (A)b;

Is as 操作符

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述若 o as teacher 為true, 返回o地址，否則回傳null

Int 不能接收 null 值

但是可以 nullable<int> x = null;

等同於 int? x = null;

?? null合併操作符

Int? x = null;

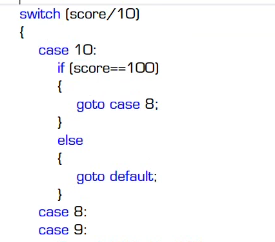
Int y = x??1; // 如果x是null，把1賦值給y

標籤語句

一張含有 文字 的圖片

自動產生的描述

Case也是標籤語劇



Try catch finally

Throw

Finally內的代碼不論有沒有抓到錯誤都會執行

Catch(錯誤名 名字)

{

Console.writeline(名字.message);

}

也可在catch內寫throw把異常 throw給掉用此方法的那行

Ienumerator

Ienumerator enumer = intArrasy.getenumerator();

Enumer.movenext();

Enumer.reset();

Enumer.current;