

## C#中ref的用法

其实这个问题很容易理解，首先在C#中传递方法参数缺省是“值拷贝”模式，也就是说对于值类型(ValueType)变量直接拷贝一份，而对于引用类型则拷贝一个指向同一对象的引用副本传递给方法，因此即使不使用ref关键字，我们也可以在方法内部改变该引用所指向对象的内部状态，但是某些时候我们需要在方法内部创建一个新的对象实例，并使得原有引用指向这个新的对象。那么问题就来了，由于现在存在两个引用，我们改变的只是传递到方法的引用副本，而该副本在超出方法作用域后既失去作用，而原有的引用依然指向原有对象。因此我们需要使用ref关键字，那么传递给方法的不再是引用副本，而是引用本身。我们就可以改变原有引用对象实例了。

```
public class Data
{
    public int i = 10;
}

public class Class1
{
    public static void Test1(Data d)
    {
        // 参数d只是一个引用副本，和原引用变量d同时指向同一个对象因此都可以修改该对象的状态
        d.i = 100;
    }

    public static void Test2(Data d)
    {
        //创建新的Data对象，并将参数d指向它。此时参数d和原有引用d分别指向2个不同的Data对象
        // 当超出Test方法作用范围时，参数d和其引用的对象将失去引用，等待GC回收。
        d = new Data();
        d.i = 200;
    }

    public static void Test3(ref Data d)
    {
        //由于使用ref关键字，因此此处的参数d和原变量d为同一引用，并没有创建副本，所以创建
        //新的Data对象是可行的。
        d = new Data();
        d.i = 300;
    }

    public static void Main(string[] args)
    {
        Data d = new Data();
        Console.WriteLine(d.i); // 10

        Test1(d);
        Console.WriteLine(d.i); // 100

        Test2(d);
        Console.WriteLine(d.i); // 100

        Test3(ref d);
        Console.WriteLine(d.i); // 300
    }
}
```