

委托的N种写法，你喜欢哪种？

一、委托调用方式

1. 最原始版本：

```
delegate string PlusStringHandle(string x, string y);
class Program
{
    static void Main(string[] args)
    {
        PlusStringHandle pHandle = new PlusStringHandle(plusString);
        Console.WriteLine(pHandle("abc", "edf"));

        Console.Read();
    }

    static string plusString(string x, string y)
    {
        return x + y;
    }
}
```

2. 原始匿名函数版：去掉“plusString”方法，改为

```
PlusStringHandle pHandle = new PlusStringHandle(delegate(string x, string y)
{
    return x + y;
});
Console.WriteLine(pHandle("abc", "edf"));
```

3. 使用Lambda（C#3.0+），继续去掉“plusString”方法（以下代码均不再需要该方法）

```
PlusStringHandle pHandle = (string x, string y) =>
{
    return x + y;
};
Console.WriteLine(pHandle("abc", "edf"));
```

还有更甚的写法（省去参数类型）

```
PlusStringHandle pHandle = (x, y) =>
{
    return x + y;
};
Console.WriteLine(pHandle("abc", "edf"));
```

如果只有一个参数

```
delegate void WriteStringHandle(string str);
static void Main(string[] args)
{
    //如果只有一个参数
    WriteStringHandle handle = p => Console.WriteLine(p);
    handle("lisi");
}
```

```

        Console.Read();
    }

```

二、委托声明方式

1. 原始声明方式见上述Demo

2. 直接使用.NET Framework定义好的泛型委托 Func 与 Action，从而省却每次都进行的委托声明。

```

static void Main(string[] args)
{
    WritePrint<int>(p => Console.WriteLine("{0}是一个整数", p), 10);

    Console.Read();
}

static void WritePrint<T>(Action<T> action, T t)
{
    Console.WriteLine("类型为：{0}，值为：{1}", t.GetType(), t);
    action(t);
}

```

3. 再加上个扩展方法，就能搞成所谓的“链式编程”啦。

```

class Program
{
    static void Main(string[] args)
    {
        string str = "所有童鞋：".plusString(p => p = p + " girl:  lisi、lili\r\n").plusString(p => p +
"boy: wangwu") ;
        Console.WriteLine(str);

        Console.Read();
    }
}

static class Extentions
{
    public static string plusString<TParam>(this TParam source, Func<TParam, string> func)
    {
        Console.WriteLine("字符串相加前原值为：{0}。。。。。。", source);
        return func(source);
    }
}

```

看这个代码是不是和我们平时写的"list.Where(p => p.Age > 18)"很像呢？没错Where等方法就是使用类似的方式来实现的。