<!DOCTYPE html>

模块：C# 系统内置委托
时间：2017-08-04
编写人：小帅
版本：V1.0

C# 系统内置委托包含有：Action<T>、Func<T>、Predicate<T>；

Action<T>

Encapsulates a method that has a single parameter and does not return a value.

命名空间：System
程序集：mscorlib(mscorlib.dll)

语法

```
public delegate void Action(T obj);
```

演变过程

No1、explicitly define a delegate

```
using System;
using System.Windows.Forms;

delegate void DisplayMessage(string message);

public class TestCustomDelegate
{
   public static void Main()
   {
      DisplayMessage messageTarget;

      if (Environment.GetCommandLineArgs().Length > 1)
         messageTarget = ShowWindowsMessage;
      else
         messageTarget = Console.WriteLine;

      messageTarget("Hello, World!");
   }

   private static void ShowWindowsMessage(string message)
   {
      MessageBox.Show(message);
   }
}
```

No2、instantiating the Action delegate instead of explicitly defining a new delegate

```
using System;
using System.Windows.Forms;

public class TestAction1
{
   public static void Main()
   {
      Action messageTarget;

      if (Environment.GetCommandLineArgs().Length > 1)
```

```
         messageTarget = ShowWindowsMessage;
      else
         messageTarget = Console.WriteLine;

      messageTarget("Hello, World!");

   }

   private static void ShowWindowsMessage(string message)
   {
      MessageBox.Show(message);
   }
}
```

No3、use the Action delegate with anonymous methods

```
using System;
using System.Windows.Forms;

public class TestAnonMethod
{
   public static void Main()
   {
      Action<string> messageTarget;

      if (Environment.GetCommandLineArgs().Length > 1)
         messageTarget = delegate(string s) { ShowWindowsMessage(s); };
      else
         messageTarget = delegate(string s) { Console.WriteLine(s); };

      messageTarget("Hello, World!");
   }

   private static void ShowWindowsMessage(string message)
   {
      MessageBox.Show(message);
   }
}
```

No4、assign a lambda expression to an Action delegate instance

```
using System;
using System.Windows.Forms;

public class TestLambdaExpression
{
   public static void Main()
   {
      Action messageTarget;
      if (Environment.GetCommandLineArgs().Length > 1)
         messageTarget = s => ShowWindowsMessage(s);
      else
         messageTarget = s => Console.WriteLine(s);

      messageTarget("Hello, World!");
   }

   private static void ShowWindowsMessage(string message)
   {
      MessageBox.Show(message);
   }
}
```