

# 附录 各章习题参考答案

## 第 1 章 习题参考答案

1. 答: 传统的网站一般是采用静态网页技术制作的静态网站。在静态网站中所有的内容以 HTML 语言编写, 存储在静态网页文件中, 文件扩展名为: .htm、.html、.shtml、.xml 等。网页上可以有 GIF 动画、FLASH 动画、滚动字幕等, 但这些效果只是视觉上的, 并未构成动态网页。这里所讨论的静态网站中的“静”是指网页内容在用户发出请求之前就已经生成了(这就是用户每次总能看到相同页面的原因), Web 服务器只负责保存和传递 HTML 文件, 而不进行额外处理, 用户只能浏览网站所提供的信息。

动态网站是目前主流的网站技术, 动态网站在发出请求之后才生成内容, 这有两个明显的优点。首先, 服务器端可以根据用户提交的请求以及所提供的参数值生成需要的页面内容。如: 在一个电子商务网站提交用户名和密码, 那么用户将看到的下一个页面就是动态生成的页面, 它包含了该用户的帐户信息, 简而言之, 就像登录淘宝网之后可以看到个人帐户一样。其次, 服务器端可以根据最新的可用信息设置它所生成的页面内容。如很多网站都有正在使用这个网站的用户数量的显示。用户数量的值是实时信息, 是在 Web 服务器接收用户请求时获取的。

由上述特征可以看出, 静态网站和动态网站的主要区别在于: 静态网站内容是在用户发出请求之前就预先生成的, 而动态网站内容则是在用户发出请求之后服务器根据用户提供的指令执行产生的。

2. 答: 见本章 1.1.5 节。

3. 答: 见本章 1.3.2 和 1.3.3 节。

4. 答: 见本章 1.5 节的实例。

## 第 2 章 习题参考答案

1. 答: HTML(HyperText Markup Language)超文本标记语言, 是制作页面文档的主要编辑语言。HTML 文件是一种纯文本文件, 通常以 .htm 或 .html 作为文件扩展名。可以用各种类型的工具来创建或者处理 HTML 页面, 如记事本、写字板、FrontPage、Dreamweaver 等。

XML(eXtensible Markup language)可扩展标记语言, 将网络上的文档规范化, 并赋予标

记一定的含义,同时,XML 又不仅仅只是标记语言,它还提供了一个标准,用户可以利用这个标准定义新的标记语言,并为这个新的标记语言规定它所特有的一套标记。

XML 虽然数据转换能力强大,完全可以替代 HTML,但面对成千上万已有的基于 HTML 语言设计的网站,直接采用 XML 还为时过早。因此,在 HTML 4.0 的基础上,用 XML 的规则对其进行扩展,得到了 XHTML(eXtensible Hypertext Markup Language)可扩展超文本标记语言。XHTML 是为了使 HTML 向 XML 顺利过渡而定义的标记语言,它以 HTML 为基础,采用 XML 严谨的语法结构。

2. 下面以一个页面 welcome.aspx 为例,来说明 ASP.NET 的文档结构。该文档代码如下:

```
<%@Page Language="C#" AutoEventWireup="true" CodeFile="welcome.aspx.cs"
Inherits="welcome" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<script runat="server">
</script>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>无标题页</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<p>
欢迎使用 Visual Web Developer
&nbsp;
</p>
</div>
</form>
</body>
</html>
```

从上面的代码可以看到,一个完整的 ASP.NET 页面文档是由指令、文档类型声明、代码声明、服务器代码、文本和 XHTML 标记等部分组成。

3. 见本章 2.3 节。

4. 主要代码如下:

```
<body>
<form id="form1" runat="server">
<div>
<h1 style="text-align: center">如梦令</h1>
<h3 style="text-align: center">李清照</h3>
<hr />
<br />
<h4 style="text-align: center">常记溪亭日暮,沉醉不知归路。</h4>
```

```

        <h4 style="text-align: center;">兴尽晚回舟，误入藕花深处。</h4>
        <h4 style="text-align: center;">争渡，争渡，惊起一滩鸥鹭。</h4>
    </form>
</body>

```

5. 主要代码如下：

```

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="lianxi2-5.aspx.cs"
Inherits="lianxi.lianxi2_5" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>An example of image</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<asp:Image ID="Image1" ImageURL="~/lianxi2-5.jpg" width="200" height="150"
border="1" runat="server" />
</div>
</form>
</body>
</html>

```

## 第3章 习题参考答案

1. 答：见 3.1.2、3.1.3 节。

2. 答：ASP.NET 的页面包含两个部分：一部分是可视化元素，包括标签、服务器控件以及一些静态文本等；另一部分是页面的程序逻辑，包括事件处理句柄和其他程序代码。ASP.NET 提供两种模式来组织页面元素和代码：一种是单一文件模式，另一种是后台代码模式。两种模式功能是一样的，可以在两种模式中使用同样的控件和代码，但使用的方式不同。

### (1) 单一文件模式

在单一文件模式下，页面的标签和代码在同一个.aspx 文件中，程序代码包含在<script runat="server"></script>的服务器程序脚本代码块中间，并且代码中间可以实现对一些方法和属性以及其他代码的定义，只要在类文件中可以使用的都可以在此处进行定义。运行时，单一页面被视为继承 Page 类。

### (2) 后台代码模式

后台代码页面模式将可视化元素和程序代码分别放置在不同的文件中，如果使用 C#，则可视化页面元素为.aspx 文件，程序代码为.cs 文件，根据使用语言的不同，代码后缀也

不同, 这种模式也被称为代码隐藏模型。

ASP.NET 的代码隐藏模型, 把一个程序文件分为一个.aspx 文件和一个对应的.aspx.cs 文件, 前者是界面代码(主要用 html 编写), 后者是一些控制代码(主要用 c#编写)。.aspx 文件顶部的页面设置把两个文件联系在一块。在进行程序设计过程中, 每一个控件都可以来触发事件。这些事件的代码单独在一个文件中, 而网页的页面设计在单独的一个文件中, 两个基本上分离的, 这样可以使代码文件更简洁。

3. 答: ASP.NET 网页是作为代码在服务器上运行的, 因此, 要使页面中的按钮或其他内容得到处理, 必须将该信息提交到服务器。每次页面提交时, 都会在服务器端运行其代码, 然后把运行的结果呈现给用户。ASP.NET 页面的处理循环如下。

(1) 用户通过客户端浏览器请求页面, 页面第一次运行, 执行初步处理。程序员可以通过编程对页面进行初始化等操作。

(2) 执行的结果以标记的形式呈现给浏览器, 浏览器对标记进行解释并显示。

(3) 用户输入信息或从可选项中进行选择, 或者单击按钮。如果用户单击的是一个超链接而不是按钮, 页面就会定位到另一个页面, 而该页面不会被进一步处理。

(4) 页面发送到 Web 服务器, 在 ASP.NET 中称此为“回发”, 也就是说页面发送回其自身。例如用户正在访问 Default.aspx 页面, 则单击该页面上的某个按钮可以将该页面发送回服务器, 发送的目的还是 Default.aspx。

(5) 在 Web 服务器上, 该页再次运行, 并且使用用户输入或选择的信息。

(6) 服务器将执行操作后的页面以 HTML 或 XHTML 标记的形式发送到客户端的浏览器。

4. 答: 参见 3.3 节。

5. 答: Web 配置文件 web.config 是 Web 应用程序的数据设定文件, 它是一份 XML 文件, 内含 Web 应用程序相关设定的 XML 标记, 可以用来简化 ASP.NET 应用程序的相关设定。该文件位于 Web 应用程序的任何目录中, 统一命名为 web.config, 它决定了所在目录及其子目录的配置信息, 并且子目录下的配置信息覆盖其父目录的配置, 即子目录如果没有 web.config 文件, 就是继承父目录 web.config 文件的相关设定; 如果子目录有 web.config 文件, 就会覆盖父目录 web.config 文件的相关设定。在运行状态下, ASP.NET 会根据远程 URL 请求, 把访问路径下的各个 web.config 配置文件叠加, 产生一个唯一的配置集合。

作为网络应用程序, 程序在执行之前有时需要初始化一些重要的变量, 而且这些工作必须发生在所有程序执行之前, ASP.NET 的 Global.asax 文件便是为此目的而设计的。每个 ASP.NET 应用程序都可以有一个 Global.asax 文件。一旦将其放在适当的虚拟目录中 ASP.NET 就会把它识别出来并且会自动使用该文件。

另外, 由于 Global.asax 在网络应用程序中的特殊地位, 它被存放的位置也是固定的。必须存放在当前应用所在的虚拟目录的根目录下。如果放在虚拟目录的子目录中, Global.asax 文件将不会起任何作用。

ASP.NET 在应用程序中添加了“全局应用程序类”, 也就是 Global.asax。该文件是应

---

用程序用来保持应用程序级的事件、对象和变量的。一个 ASP.NET 应用程序只能有一个 Global.asax 文件，位于应用程序根目录下。

## 第 4 章 习题参考答案

1. 答：代码如下。

```
public class MainClass
{
    public static void Main()
    {
        Console.WriteLine(Foo(30));
    }
    public static int Foo(int i)
    {
        if (i <= 0)
            return 0;
        else if (i > 0 && i <= 2)
            return 1;
        else return Foo(i-1) + Foo(i-2);
    }
}
```

2. 答：代码如下。

```
int[] array = new int[*];
int temp = 0;
for (int i = 0; i < array.Length - 1; i++)
{
    for (int j = i + 1; j < array.Length; j++)
    {
        if (array[j] < array[i])
        {
            temp = array[i];
            array[i] = array[j];
            array[j] = temp;
        }
    }
}
```

3. 答：CTS，通用语言系统；CLS，通用语言规范；CLR，公共语言运行库。

4. 答：从值类型接口转换到引用类型装箱。从引用类型转换到值类型拆箱。

5. 答：数组和 string 都没有 Length()方法，只有 Length 属性。

6. 答：final - 修饰符（关键字）如果一个类被声明为 final，意味着它不能再派生出新的子类，不能作为父类被继承。因此 一个类不能既被声明为 abstract 的，又被声明为 final

的。将变量或方法声明为 `final`，可以保证它们在使用中 不被改变。被声明为 `final` 的变量必须在声明时给定初值，而在以后的引用中只能读取，不可修改。被声明为 `final` 的方法也同样只能使用，不能重载

`finally` – 再异常处理时提供 `finally` 块来执行任何清除操作。如果抛出一个异常，那么相匹配的 `catch` 子句就会 执行，然后控制就会进入 `finally` 块（如果有的话）。

`finalize` – 方法名。Java 技术允许使用 `finalize()` 方法在垃圾收集器将对像从内存中清除出去之前做必要的清理 工作。这个方法是由垃圾收集器在确定这个对象没有被引用时对这个对象调用的。它是在 `Object` 类中定义的，因此所有的类都继承了它。子类覆盖 `finalize()` 方法以整理系统资源或者执行其他清理工作。`finalize()` 方法是在垃圾收集器删除对像之前对这个对象调用的。

7. 答：它们被称为常成员变量和常成员函数，又称为类成员变量和类成员函数。分别用来反映类的状态。比如类成员变量可以用来统计类实例的数量，类成员函数负责这种统计的动作。

8. 答：using 引入名称空间或者使用非托管资源。new 新建实例或者隐藏父类方法。

## 第 5 章 习题参考答案

1. 答：System.Web.UI.UserControl。

2. 答：可以。

3. 答：HTML 控件和 Web 控件。

4. 答：

假设用户控件文件名为 WebUserControl.ascx，则在 Web 页面中注册的语句如下：

```
<%@ Register src="WebUserControl1.ascx" tagname="WebUserControl1"
tagprefix="uc1" %>
```

在创建控件实例时指定控件名称为 Mike，语句如下：

```
<uc1:WebUserControl1 ID="Mike" runat="server" />
```

5. 答：见本章表 5-2。

## 第 6 章 习题参考答案

1. 答：见本章 6.1.1、6.1.2、6.1.3 节内容。

2. 答：见本章 6.1.4 节内容。

3. 答：见本章 6.2.2、6.2.3、6.2.4 节内容。

## 第 7 章 习题参考答案

1.答: CSS(Cascading Style Sheet), 即层叠样式表, 是一种标记性语言, 用于控制网页样式并允许将样式信息与网页内容分离。使用 CSS 样式可以非常灵活地控制网页外观, 大大减轻实现精确布局定位、维护特定字体和样式的工作量。

CSS 中有两种定义样式的方法, 分别是内联式和级联式, 分别对应两类样式: 内联式样式和级联式样式。

### (1) 内联式样式

直接将样式控制放在单个 HTML 元素内, 称为内联式或行内样式。该样式通过 style 属性控制每个元素的外观。这种样式直观但是很繁琐, 除非具有相同样式的元素较少, 否则很少采用。

### (2) 级联式样式

在网页的 head 部分定义或导入的样式, 称为级联式样式。该样式可以实现将网页业务处理逻辑和呈现代码分离, 这样, 当修改某些元素的样式时, 只需要修改 head 部分定义或引入的样式, 该网页内所有具有相同样式的元素都会自动应用定义的或引入的样式。

2.答: 有两种方式, 一种是代码方式, 操作方法为: 创建一个新的网页, 系统自动打开页面【源】视图, 利用系统提供的智能提示功能, 可以方便地设置各种元素的样式。另一种方式为可视化方式, 即在【源】视图或者【设计】视图下选中某个标记元素, 然后设置属性面板中 style 属性值。

3.答: 自从 1996 年 CSS1 的推出, W3C 组织就建议把所有网页上的对象都放在一个盒(box)中, 设计师可以通过创建定义来控制这个盒的属性, 这些对象包括段落、列表、标题、图片以及层。盒子模型主要定义 4 个区域: 内容(content)、边框距(padding)、边界(border)和边距(margin)。前面【例 7-5】中定义的层就是一个典型的盒。对于初学者, 经常会搞不清楚 margin, background-color, background-image, padding, content, border 之间的层次、关系和相互影响。这里给出一个盒子模型图, 如下图所示。

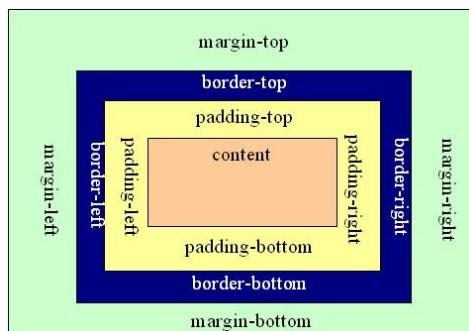


图 盒子模型

理解盒子模型就可以理解层与层之间定位的关系以及层内部的表达样式。其中, margin



属性负责层与层之间的距离, padding 属性负责内容和边框之间的距离。

4. 答: 见本章 7.3.1、7.3.4 节的内容及示例。

5. 答: 见本章 7.4.1、7.4.2 节内容。

## 第 8 章 习题参考答案

1. 答: ADO.NET 是 .NET Framework 提供的数据库访问的类库, ADO.NET 对 Microsoft SQL Server、Oracle 和 XML 等数据源提供一致的访问。应用程序可以使用 ADO.NET 连接到这些数据源, 并检索和更新所包含的数据。

ADO.NET 用于数据库访问的类库包含 .NET Framework 数据库提供程序和 DataSet 两个组件。 .NET Framework 数据库提供程序与 DataSet 之间的关系如下图所示。

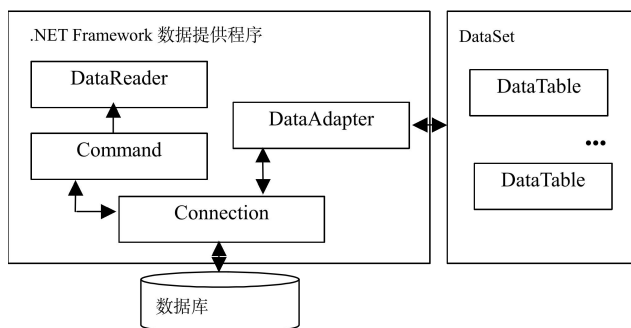


图 ADO.NET 的组成

2. 答: .NET Framework 数据库提供程序包含以下 4 个核心类。

- Connection: 建立与数据源的连接。
- Command: 对数据源执行操作命令, 用于修改数据、查询数据和运行存储过程等。
- DataReader: 从数据源获取返回的数据。
- DataAdapter: 用数据源数据填充 DataSet, 并可以处理数据更新。

3. 答: DataSet 是 ADO.NET 的断开式结构的核心组件。设计 DataSet 的目的是为了实现独立于任何数据源的数据访问, 可以把它看成是内存中的数据库, 是专门用来处理数据源中读出的数据。

DataSet 的优点就是离线式, 一旦读取到数据库中的数据后, 就在内存中建立数据库的副本, 在此之后的操作, 直到执行更新命令为止, 所有的操作都是在内存中完成的。不管底层的数据库是哪种形式, DataSet 的行为都是一致的。

DataSet 是数据表(DataTable)的集合, 它可以包含任意多个数据表, 而且每个 DataSet 中的数据表对应一个数据源中的数据表(Table)或者是数据视图(View)。DataSet 的结构如下图所示。



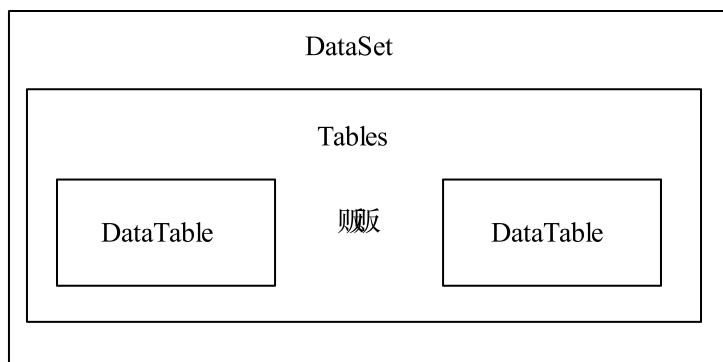


图 DataSet 的结构

4. 答: ASP.NET 数据访问程序的开发流程有以下几个步骤:
- (1) 利用 Connection 对象创建数据连接。
  - (2) 利用 Command 对象数据源执行 SQL 命令。
  - (3) 利用 DataReader 对象读取数据源的数据。
  - (4) DataSet 对象与 DataAdapter 对象配合, 完成数据的查询和更新操作。
5. 答: 见 8.3 节内容。
6. 答: 见 8.4 节内容。

## 第 9 章 习题参考答案

1. 答: 绑定格式为<%# 变量或表达式 %>。
2. 答: ASP.NET 中用于绑定数据的常用列表控件主要有以下几种: DropDownList、ListBox、CheckBoxList、RadioButtonList、BulletedList 等。
3. 答: 请参考本章的 9.3.4 节【例 9-6】。
4. 答: 请参考本章的 9.4.1 节【例 9-7】。
5. 答: 请参考本章的 9.4.2 节【例 9-8】。

## 第 10 章 习题参考答案

1. 答: 当用户与浏览器中的页面进行交互时, 会触发页面元素对象的响应事件, 客户端捕获相应的事件后, 如果需要提交给服务器进行处理, 客户端就将要处理的数据 (包括状态描述) 转换为 XML 或 JSON 格式的字符串, 再利用异步传送方式将这些数据提交给服务器; 服务器进行处理后, 同样利用 XML 或 JSON 格式和异步传送方式将处理结果返回给客户端; 客户端再从返回结果中提取需要的部分, 并将提取的部分利用 JavaScript 对

网页进行“悄悄地”局部更新，而不是刷新整个页面。

2. 答：（1）页面局部刷新。在读取数据的过程中，用户所面对的不是白屏，而是原来的页面状态，或者正在更新的信息提示状态，只有当接收到全部数据后才更新相应部分的内容，而这种更新也是瞬间的，用户几乎感觉不到。

（2）把一些由服务器承担的工作转移到客户端处理，这样可以充分利用客户端闲置的处理能力，减轻服务器和网络传输的负担。

（3）使用基于标准化的并被广泛支持的技术，不需要安装插件即可运行。

（4）使 Web 中的界面与应用分离，也可以说是数据与呈现分离。

3. 答：ScriptManager 控件是 ASP.NET AJAX 的核心，它提供处理页面上的所有 ASP.NET AJAX 控件(UpdatePanel、UpdateProgress 等)的支持，没有该控件的存在其他 ASP.NET AJAX 控件是不能工作的，并且所有需要支持 ASP.NET AJAX 的 ASP.NET 页面上只能有一个 ScriptManager 控件。另外，ScriptManager 控件还可以生成相关的客户端代理脚本，以便能够在客户端脚本中访问 Web Service。

4. 答：源代码如下。

```

【解答】源代码如下：
<!DOCTYPE html>
<html>
<head runat="server">
<title></title>
<style type="text/css">
.highlightTip{ background-color: #FFFFCC; }
</style>
</head>
<body>
<form id="form1" runat="server">
<asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>
<div>
<table style="width: 50%; height: 130px">
<tr><td style="text-align:right; width: 280px;">用户名</td>
<td style="text-align:left">
<asp:TextBox ID="TextBoxUserName" runat="server" AutoPostBack="True">
</asp:TextBox>
</td>
</tr>
<tr>
<td style="text-align:right;">密码</td>
<td style="text-align:left">
<asp:TextBox ID="TextBoxPassword" runat="server"
TextMode="Password" Width="149px"></asp:TextBox>
<asp:BalloonPopupExtender ID="TextBox1_BalloonPopupExtender" runat="server"
TargetControlID="TextBoxPassword"
BalloonPopupControlID="LabelTip1"></asp:BalloonPopupExtender>

```

```

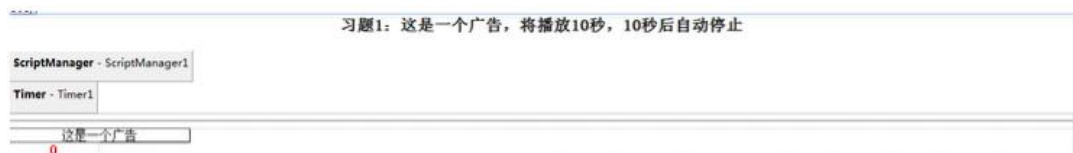
<asp.Label ID="LabelTip1" runat="server" Text="请输入 6 个以上字符" />
</td>
</tr>
<tr>
<td style="text-align:right;" >出生日期: </td>
<td style="text-align:left;" >
<asp.TextBox ID="TextBoxBirthday" runat="server"></asp.TextBox> <asp.CalendarExtender
ID="CalendarExtender1" DaysModeTitleFormat="yyyy 年 M 月" TodaysDateFormat="yyyy-MM-dd"
Format="yyyy-MM-dd" TargetControlID="TextBoxBirthday" StartDate="1900-1-1" runat="server" />
</td>
</tr>
<tr>
<td colspan="2" style="text-align:center;">
<asp.Button ID="Button1" runat="server" Text="确定" />
<asp.Button ID="Button2" runat="server" Text="取消" />
</td>
</tr>
</table>
<asp.RequiredFieldValidator ID="R1" runat="server" ControlToValidate="TextBoxUserName"
Display="None" ErrorMessage="<b>不能为空</b><br /><br />必须输入姓名。" />
<asp.ValidatorCalloutExtender ID="V1" TargetControlID="R1" HighlightCssClass="highLightTip"
runat="server" />
<asp.RequiredFieldValidator ID="R2" runat="server" ControlToValidate="TextBoxPassword"
Display="None" ErrorMessage="<b>不能为空</b><br /><br />必须输入密码。" />
<asp.ValidatorCalloutExtender ID="V2" TargetControlID="R2" HighlightCssClass="highLightTip"
runat="server" />
<asp.RequiredFieldValidator ID="R3" runat="server" ControlToValidate="TextBoxBirthday"
Display="None" ErrorMessage="<b>不能为空</b><br /><br />必须输入姓名。" />
<asp.ValidatorCalloutExtender ID="V3" TargetControlID="R3" HighlightCssClass="highLightTip"
runat="server" />
</div>
</form>
</body>
</html>

```

## 第 11 章 习题参考答案

1. 答: 本题的解决办法很多, 本解答给出其中一种解决办法。本题中使用数字来模拟广告播放的次数。

在 AdExample.aspx 页面实现此程序需要的效果。页面设计如下图所示。



AdExample.aspx 的设计界面源码如下:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="TimerExample.aspx.cs"
Inherits="Exise.Ch08.TimerExample"%>
<!DOCTYPE html>
<head id="Head1" runat="server">
<title></title>
<link href="jQuery/ui-1.8.18/redmond/jquery-ui.css" rel="stylesheet" type="text/css" />
<style>
body { text-align: center; font-size: 14px; }
#UpdatePanel1 { width: 600px; margin: 0 auto; border: 1px solid; height: 100px; }
</style>
</head>
<body>
<form id="form1" runat="server">
<h3>
习题: 这是一个广告, 将播放 10 秒, 10 秒后自动停止
</h3>
<asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>
<asp:Timer ID="Timer1" runat="server" ontick="Timer1_Tick" Interval="1000"></asp:Timer>
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
<ContentTemplate>
<hr />
<div id="div4" style="width: 200px; text-align: center; border: 1px solid;" runat="server"> 这是一个
广告</div>
<div id="div1" runat="server" style="width: 100px; font-size: larger; color: Red; text-align: center;
font-weight: bolder;">0</div>
</ContentTemplate>
<Triggers>
<asp:AsyncPostBackTrigger ControlID="Timer1" />
</Triggers>
53
</asp:UpdatePanel>
</form>
</body>
</html>
```

AdExample.aspx.cs 的代码:

```
namespace Exise.Ch08
{
    public partial class AdExample : System.Web.UI.Page
```

```

{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (IsPostBack==false)
        {
            Timer1.Interval = 1000;

            Timer1.Enabled = true;
        }
    }
    protected void Timer1_Tick(object sender, EventArgs e)
    {
        string str = div1.InnerText;
        int v1 = int.Parse(str) + 1;
        if (v1 <= 10)
        {
            div1.InnerText=v1.ToString();
            div4.InnerText = "广告播放第" + v1.ToString() + "次";
        } else {
            Timer1.Enabled = false;
            div4.InnerText = "广告播放完毕";
        }
    }
}

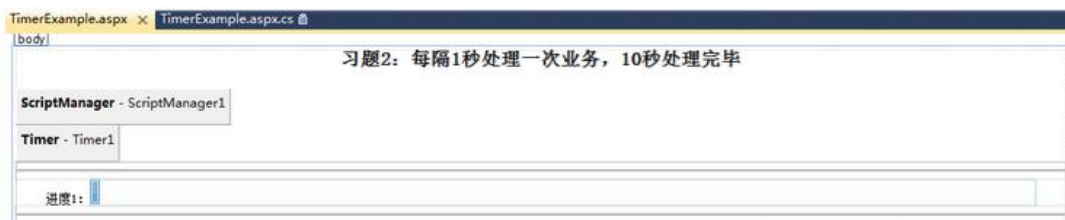
```

程序运行效果如下图所示。

### 习题1：这是一个广告，将播放10秒，10秒后自动停止



2. 答：【解答】本题的解决办法很多，本解答给出其中一种解决办法。  
在 TimerExample.aspx 页面实现此程序需要的效果。页面设计如下：



TimerExample.aspx 页面的设计器代码如下：

```

<%@Page Language="C#" AutoEventWireup="true" CodeBehind="TimerExample.aspx.cs"
Inherits="Exise.Ch08.TimerExample"%>

```

```

<!DOCTYPE html>
<html>
<head id="Head1" runat="server">
<title></title>
<link href="~/jQuery/ui-1.8.18/redmond/jquery-ui.css" rel="stylesheet" type="text/css" />
<style>
body { text-align: center; font-size: 12px; }
#UpdatePanel1 { width: 600px; margin: 0 auto; border: 1px solid; }
#div4 { width: 200px; margin: 0 auto; border: 1px solid; }
.ui-progressbar { display: inline-block; width: 90%; }
.ui-progressbar-value { text-align: right; }
</style>
</head>
<body>
<form id="form1" runat="server">
<h3>
习题：每隔 1 秒处理一次业务，10 秒处理完毕
</h3>
<asp:ScriptManager ID="ScriptManager1" runat="server"></asp:ScriptManager>
<asp:Timer ID="Timer1" runat="server" ontick="Timer1_Tick"></asp:Timer>
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
<ContentTemplate>
<hr />
进度 1:
<div class="ui-progressbar ui-widget ui-widget-content ui-corner-all">
<div id="div1" runat="server" style="width: 0%;".class="ui-progressbar-value
ui-widget-header ui-corner-left">
</div>
</div>
<hr />
</ContentTemplate>
<Triggers>
<asp:AsyncPostBackTrigger ControlID="Timer1" />
</Triggers>
</asp:UpdatePanel>
</form>
</body>
</html>

```

TimerExample.aspx.cs 的后台代码如下：

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;

```

```

using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
namespace Exise.Ch08
{
    public partial class TimerExample : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (IsPostBack == false)
            {
                div1.InnerText = "0";
                Timer1.Interval = 1000;
                Timer1.Enabled = true;
            }
        }
        protected void Timer1_Tick(object sender, EventArgs e)
        {
            int v1 = int.Parse(div1.InnerText.TrimEnd('%')) + 10;
            if (v1 <= 100)
            {
                div1.InnerText = div1.Style["width"] = v1 + "%";
            }
            else
            {
                div1.InnerText = div1.Style["width"] = "100%";
                Timer1.Enabled = false;
            }
        }
    }
}

```

程序的运行效果:

3. 编写一个了解学生业余活动的调查页面，页面中包含对输入信息的验证功能。

**【解答】** 本题的解决方法比较多，本例子给出其中一种解决方法。

程序运行效果如下图所示:

validateExample.aspx

页面的设计代码如下:

```

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="validateExample.aspx.cs"
Inherits="Exise.Ch08.validateExample" %>
<!DOCTYPE html>
<head runat="server">
<title></title>

```



```

<style type="text/css">
* { font-family: Verdana; font-size: 96%; }
.tl { display: inline-block; width: 100px; text-align: right; }
label { width: 10em; float: left; }
label.error { float: none; color: red; padding-left: .5em; vertical-align: top; }
p { clear: both; }
.submit { margin-left: 12em; }
</style>
</head>
<body>
<form id="form1" runat="server">
<div>
<div style="margin: 0 auto; width: 500px; min-width: 500px; border: 1px solid;">
<h3>
学生业余活动调查页面
</h3>
<span class="tl">
学生学号: </span> <asp:TextBox ID="TextBox1" runat="server" />
<br />
<span class="tl">学生姓名: </span> <asp:TextBox ID="TextBox2" runat="server" /> <br />
<span class="tl">
选择你最喜欢的运动项目: </span>
<asp:DropDownList ID="DropDownList1" runat="server" >
<asp:ListItem Text="无" Value="无"></asp:ListItem>
<asp:ListItem Text="足球" Value="足球"></asp:ListItem>
<asp:ListItem Text="篮球" Value="篮球"></asp:ListItem>
<asp:ListItem Text="乒乓球" Value="乒乓球"></asp:ListItem>
</asp:DropDownList>
<br />
<asp:Button ID="Button1" runat="server" Text="提交" Style="margin: 10px auto 0 180px;" />
<hr />
<asp:Label ID="Label1" runat="server" Text="" />
</div>
</div>
</form>
<script src=" ./jQuery/js/jquery-1.7.1.min.js" type="text/javascript"></script>
<script src=" ./jQuery/validate-1.9/jquery.validate.min.js" type="text/javascript"></script>
57 <script src=" ./jQuery/validate-1.9/jquery.validate.messages_cn.js" type="text/javascript"></script>
<script>
$(function () {
$("#Button1").click(function () {
$("#form1").validate({ rules: { TextBox1: { required: true, rangelength: [8, 10] }, TextBox2: { required:
true, maxlength: 10 } }, messages: { TextBox1: { required: "请输入 8-10 个数字字符" }, TextBox2:
{ required: "最多输入 10 个中文字符" }

```

```

    }
    });
    });
    });
</script>
</body>
</html>

```

validateExample.aspx.cs 的页面代码如下:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
namespace Exise.Ch08
{
    public partial class ValidateExample : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            Button1.Click += new EventHandler(Button1_Click);
        }
        void Button1_Click(object sender, EventArgs e)
        {
            Label1.Text = string.Format("你的学号是{0}，姓名是{1}，最喜欢的运动项目是:{2}<br/>",
this.TextBox1.Text,this.TextBox2.Text,this.DropDownList1.Selected.Value);
        }
    }
}

```

## 第 12 章 习题参考答案

### 一、单选题

1. B 2. A 3. A 4. B 5. D

### 二、多选题

1. BC 2. ABCDF 3. AB 4. AE 5. ABCDE

### 三、简答题

1. 答: 白页(White Page) 包括了地址, 联系方法, 和已知的企业标识;

黄页(Yellow page)包括了基于标准分类法的行业类别;

绿页(Green Page)则包括了关于该企业所提供的 Web 服务的技术信息, 其形式可能是一些指向文件或是 URL 的指针, 而这些文件或 URL 是为服务发现机制服务的。

2. 答:

优点: 简便性、独立于语言及平台、功能抽象、可发现性、提高开发效率;

局限: 闭合系统、对性能要求高、可靠性要求高、技术局限性。

## 第 13 章 习题参考答案

1. 答: 参见 13.1.1 节。

2. 答: 为了使一个 XML 文档结构完整, XML 必须遵守一定规则。常见的 XML 文档规则如下:

(1) 文档必须以 XML 版本声明开始。

(2) 含有数据的元素必须有起始标记和结束标记。每个起始标记必须以相应的结束标记结束。如果一个文档未能结束一个标记, 浏览器将报告一个错误信息, 并且不会以任何形式显示任何文档的内容。

(3) 不含数据并且仅使用一个标记的元素必须以 `</>` 结束。

(4) 文档只能包含一个能够包含全部其他元素的根元素。如: `<students>` 元素。

(5) 元素只能嵌套不能重叠。

(6) 属性值必须加引号。如: `<address telephone="021-30008655">` 中 `telephone` 属性的值不能写成 `< address telephone="021-30008655">`。

3. 答: 参见 13.1.3 节。

4. 答: 参见 13.2 节。

5. 答: 参见 13.4 节。

## 第 14 章 习题参考答案

1. 答: 互联网中国内比较知名的网上书店有当当网、卓越亚马逊、京东商城、红孩子、淘宝网、淘宝商城、苏宁易购 (图书作为附属产品)、中国互动出版网、新华书店网上书城等。

2. 答: 略 (打开指定的网上书店, 找到图书板块, 进行分类的查看即可)。