

系统实现

5.1 登录界面模块

在本节中,通过登录界面模块的实现,掌握如何在 MVC 中创建视图和控制器。

5.1.1 登录界面模块概要

登录界面作为进入 OA 系统的唯一入口,主要用来对进入 OA 系统的用户进行安全性检查,以防止非法用户进入系统。同时使用了验证码技术,防止使用其他程序进行非法登录。登录界面运行效果如图 5-1 所示。



图 5-1 登录界面

5.1.2 登录界面模块相关知识

登录界面模块的相关知识点如下:

- HTML 语言。
- CSS 样式。

- Layout(布局)。
- ViewStart。
- 视图。
- 路由机制。
- 控制器。

5.1.3 登录界面模块实现

1. 业务流程图

登录界面模块的业务流程图如图 5-2 所示。

2. 数据库设计

登录界面模块用到数据表 RL_Employee(员工), 具体表信息请参照第 4 章中表 4-36 所示。

3. HTML 概述

HTML 指的是超文本标记语言(Hyper Text Markup Language), 是一种简单标记语言, 不是一种编程语言。通常网页文档就是用 HTML 编写的超本文档, 通过浏览器(如 Microsoft Internet Explorer)解释 HTML 文档, 显示网页。

1) HTML 文档基本结构

HTML 文档可以描述包含文字、图形、动画、声音等在内的网页内容。完整的 HTML 文档由标题、段落、表格等多种对象组成, 这些对象在这里称为元素, HTML 文档用 HTML 标签(HTML tag)来分离并描述这些元素。

HTML 文档基本结构如下所示:

```
<html>                                //文档的开始标签
  <head>                                //文档头部的开始标签
    <title>                              //文档标题的开始标签
    ...                                  //文档头部信息
    </title>                             //文档标题的结束标签
  </head>                               //文档头部的结束标签
  <body>                                //文档体的开始标签
  ...                                  //网页显示的内容信息,如文字、图形、动画等
  </body>                               //文档体的结束标签
</html>                                //文档的结束标签
```

在 HTML 文档基本结构中, <html>...</html> 是所有 HTML 文档的开始和结束标签, <head>、<title>、<body> 等 HTML 标签都包含在 <html> 标签里面。<head>...</head> 标签是 HTML 文档的头部标签, 里面包含了 <title> 标签用于存

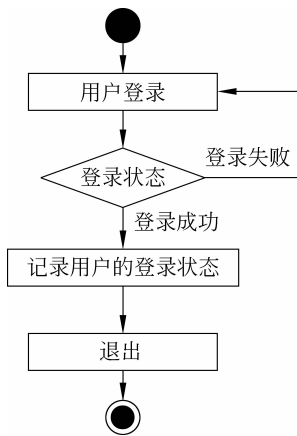


图 5-2 登录流程图

放网页页面的标题。<body>...</body>标签是 HTML 文档体标签,网页需要呈现的内容(文字、图形、动画、声音等)都存放在这个标签里。

HTML 标签用来对各种元素进行分离和描述。标签符用“<”和“>”表示。标签符中的内容即为元素。例如,<title>欢迎使用 ASP. NET MVC</title>,其中,title 是元素名,内容“欢迎使用 ASP. NET MVC”属于元素 title,<title>是用来说明元素 title 的标签。

【提示】 标签通常成对出现,有开始标签和结束标签,例如<html>...</html>、<body>...</body>等;但也有单独出现的时候,例如换行标签
、水平直线标签<hr>等。因此标签可以划分为单标签和双标签。在实际 HTML 代码编写中标签并不区分大小写,项目中为了规范性通常提倡采用小写。

标签的属性特指对标签这个对象特征的描述。不同的标签具有不同的属性值,所有标签的属性都放置在开始标签的尖括号里,标签与属性间用一个空格分隔,属性与属性的值之间用等号相连,不同属性之间同样也是用一个空格分隔。双标签属性的基本格式如下:

<标签 属性 0=属性值 0 属性 1=属性值 1 ... 属性 n=属性值 n>...</标签>

2) HTML 文档的基本标签

除了以上提到的标签外,HTML 文档的基本标签还有很多,包括文本标签、列表标签、表格标签、表单类标签、框架标签和图形图像标签等。在这里着重介绍后续章节中所运用的标签基础知识。

(1) 文本标签。用来进行文字大小、段落、显示方式等控制。这些控制文字显示效果的标签也称为格式化标签。

- <Hi>...</Hi>是与文字段落有关的标题标签,属于双标签。标题标签从<H1>...</H1>到<H6>...</H6>共分 6 级。标题标签之间的文字是文档中的标题,<H1>与</H1>之间的文字显示效果最大、最粗。以此类推,<H6>与</H6>之间的文字最终显示效果最小、最细。
- <i>...</i>是文字斜体标签。<i>与</i>之间的文字最终以斜体的形式显示。

例 5.1 一个简单的 HTML 文档,展示基本标签的使用。

代码如下:

```
1 <html>
2     <head>
3         <title>HTML 文档基本标签之文本标签</title>
4     </head>
5     <body>
6         <H1>H1 字体标题</H1>
7         <H2>H2 字体标题</H2>
8         <H3>H3 字体标题</H3>
9         <H4>H4 字体标题</H4>
10        <H5>H5 字体标题</H5>
```

```
11      <H6>H6 字体标题</H6>
12      <i>斜体字体->自动生成编号</i>
13  </body>
14 </html>
```

代码描述：例 5.1 中的代码通过常用的代码编辑工具 NotePad 或记事本编辑，保存类型以 .html 后缀结尾的文件“HTML 文档基本标签学习.html”，使用浏览器（默认采用 Microsoft Internet Explorer, IE）打开该文件，运行结果如图 5-3 所示。

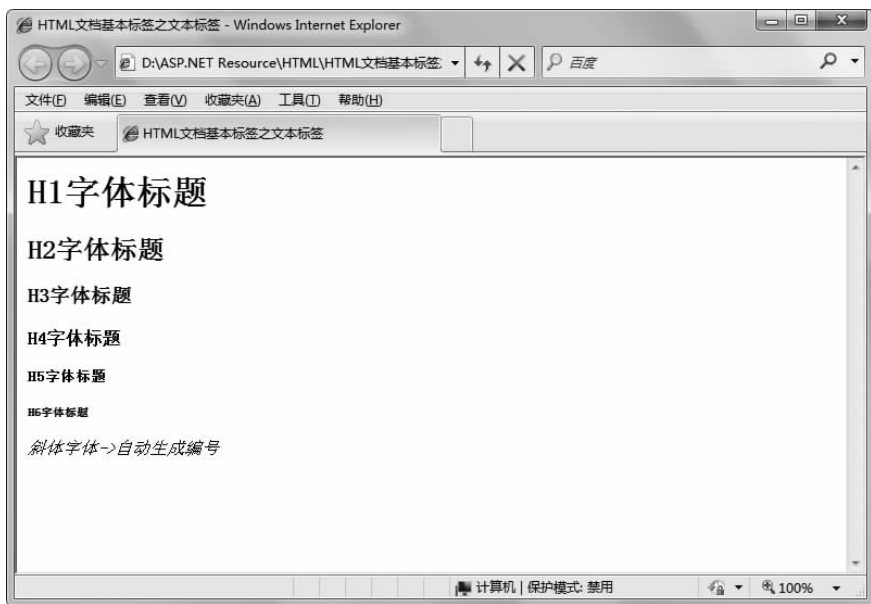


图 5-3 文本标签在 IE 中的运行结果

(2) 列表标签。分为有序列表标签和无序列表标签两种。另外，包含在有序列表标签和无序列表标签之间的是列表内容标签。

例 5.2 一个简单的 HTML 文档，展示无序列表标签的使用。

代码如下：

```
1 <html>
2   <head>
3     <title>HTML 文档基本标签之列表标签</title>
4   </head>
5   <body>
6     <ul>
7       <li>HTML 文档基本标签学习之列表标签 1</li>
8       <li>HTML 文档基本标签学习之列表标签 2</li>
9     </ul>
10  </body>
11 </html>
```

代码描述：例 5.2 中的代码通过常用的代码编辑工具 NotePad 或记事本编辑，保存为以 .html 后缀结尾的文件“HTML 文档基本标签之列表标签.html”，使用浏览器打开该文件，运行结果如图 5-4 所示。

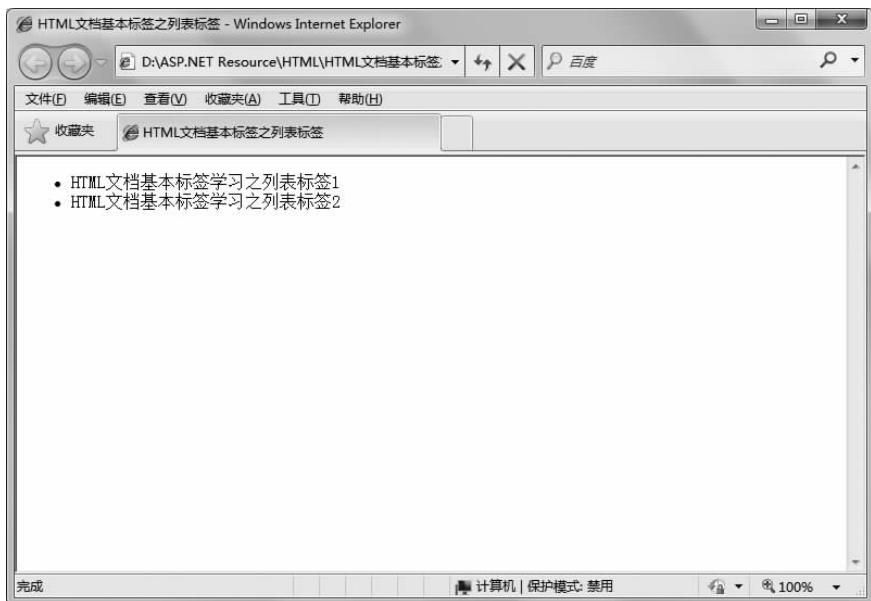


图 5-4 无序列列表标签在 IE 中的运行结果

(3) 表格标签。HTML 的表格实际上和 Excel 的表格一样，同样由行和列组成。在这里创建表格通常会用到以下标签：

- `<table>...</table>`，定义表格的标签。
- `<tr>...</tr>`，表格行标签。
- `<td>...</td>`，表格列标签。
- `<th>...</th>`，表格头部标题标签。

例 5.3 一个简单的表格标签 HTML 文件。

代码如下：

```
1 <html>
2   <head>
3     <title>HTML 文档基本标签之表格标签</title>
4   </head>
5   <body>
6     <table border=1>
7       <tr>
8         <th>编号</th>
9         <th>采购用户</th>
10        <th>创建日期</th>
11      </tr>
12      <tr>
```

```
13         <td>pc20140505</td>
14         <td>采购用户 1</td>
15         <td>20140505</td>
16     </tr>
17 </table>
18 </body>
19 </html>
```

代码描述：在例 5.3 代码中，第 6 行 `<table>` 中使用了表格标签中的属性 `border`，该属性用来设置表格边框线宽度，默认值为 0，即不显示边框，这里为了方便展示，将 `border` 属性值设置为 1。用记事本编辑这段代码，保存为以 `.html` 后缀结尾的文件“HTML 文档基本标签之表格标签.html”，使用浏览器打开该文件，运行结果如图 5-5 所示。

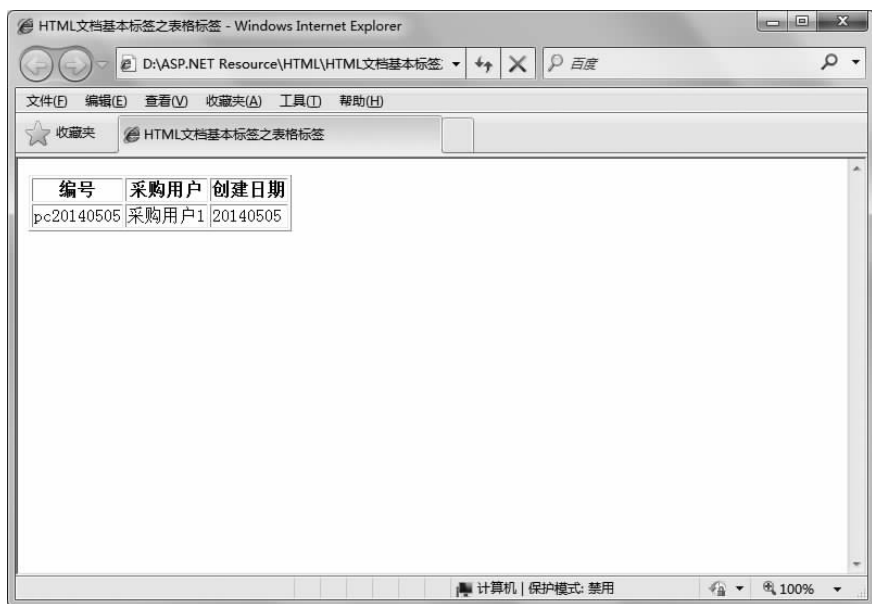


图 5-5 表格标签在 IE 中的运行结果

(4) 表单类标签。这类标签在 HTML 文档中使用 `<form>...</form>` 来定义，主要用于接收用户输入的信息，再将信息自动上传到 Web 服务器，由 Web 服务器对表单进行处理，处理后的结果要么存放在服务器数据库中，要么直接反馈到客户端浏览器中。

创建表单会用到以下标签：

- `<form>...</form>`，定义表单的标签。
- `<select>...</select>`，定义下拉列表的标签。
- `<textarea>...</textarea>`，定义文本框的标签（用于定义多行文字输入区域）。
- `<input/>`，定义一个表单的输入域。

例 5.4 表单类标签的实现。

代码如下：

```
1 <html>
2   <head>
3     <title>HTML 文档基本标签之表单类标签</title>
4   </head>
5   <body>
6     <form id="form1" runat="server">
7       <!-- 下拉列表: -->
8       <select name=select1>
9         <option>下拉选项 1
10        <option>下拉选项 2
11        <option>下拉选项 3
12      </select>
13      <!-- 多行文字输入域: -->
14      <textarea cols=20 name=textarea1 rows=5></textarea>
15      <!-- 一个表单输入域: -->
16      <input type="text" name=text1 size=10>
17    </form>
18
19    <form>新的段落</form>
20  </body>
21 </html>
```

代码描述: 在例 5.4 的代码中,从第 6 行<form>开始到第 19 行</form>结束是一个双标签。第 19 行的<form>…</form>标签在运行后会产生新的段落。用记事本编辑这段代码,保存为以.html 后缀结尾的文件“HTML 文档基本标签之表单类标签.html”,使用浏览器打开该文件,运行结果如图 5-6 所示。

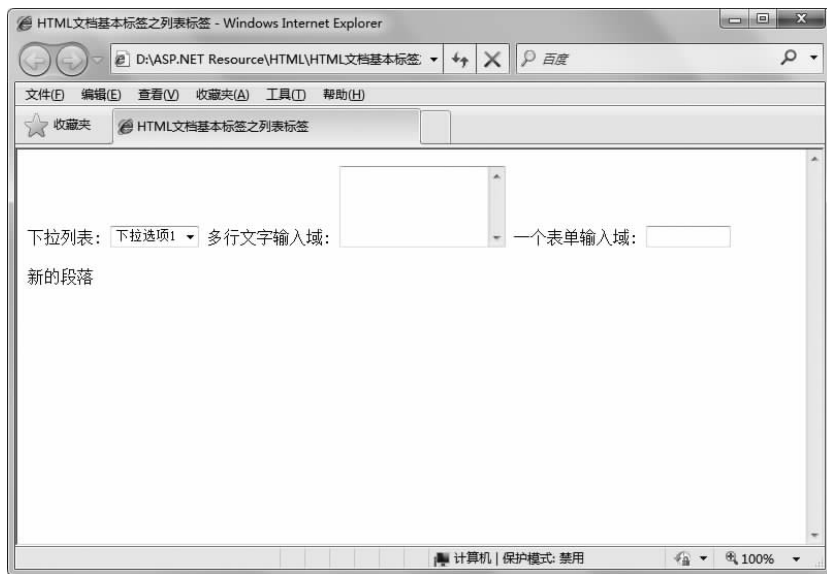


图 5-6 表单类标签在 IE 中的运行结果

(5) 框架标签。框架特指用来分割浏览器窗口的工具,在一个浏览器窗口中可以布局多个框架,各框架之间相互独立,多个框架的集合组成了框架集,通过标签<frameset>...</frameset>定义。

- <frameset>...</frameset>,定义框架集的标签。
- <frame>...</frame>,定义框架的标签。

例如,将图 5-4、图 5-5 和图 5-6 在一个浏览器窗口进行布局显示,可以通过例 5.5 实现。

例 5.5 框架类标签实现。

代码如下:

```

1 <html>
2     <head>
3         <title>HTML 文档基本标签之框架标签</title>
4     </head>
5     <frameset rows="300,*" cols="*" frameborder="yes" border="1"
        framespacing="0">
6         <frame src="HTML 文档基本标签之文本标签.html">
7         <frameset cols="300,*" frameborder="yes" border="1"
            framespacing="0">
8             <frame src="HTML 文档基本标签之表格标签.html">
9             <frame src="HTML 文档基本标签之表单标签.html">
10        </frameset>
11    </frameset>
12    <noframes></noframes>
13 </html>

```

代码描述: 例 5.5 中的代码通过常用的代码编辑工具 NotePad 或记事本编辑,保存为以 .html 后缀结尾的文件“HTML 文档基本标签之框架标签.html”,使用浏览器打开该文件,运行结果如图 5-7 所示。

4. CSS 样式

CSS 指的是层叠样式表(Cascading Styles Sheet),与 HTML 一样,是一种标记语言,需要通过浏览器解释执行。CSS 主要用于控制网页的显示效果,例如字体、颜色、背景等;而 HTML 则主要用于控制网页的整体结构,重点是网页内容。在采用 CSS 技术进行网页制作时,只需要对相应的代码做一些简单修改,就可以改变同一页面的不同部分或者不同页面的显示效果。

1) CSS 样式的组成

CSS 样式表由多个样式规则组成,每个样式规则有标准的格式,一个标准的 CSS 样式的定义通常由选择器(selector)、属性(property)、属性值(value)3 部分组成,其基本结构如下:

选择器 {属性 0: 属性值 0; 属性 1: 属性值 1; ...}

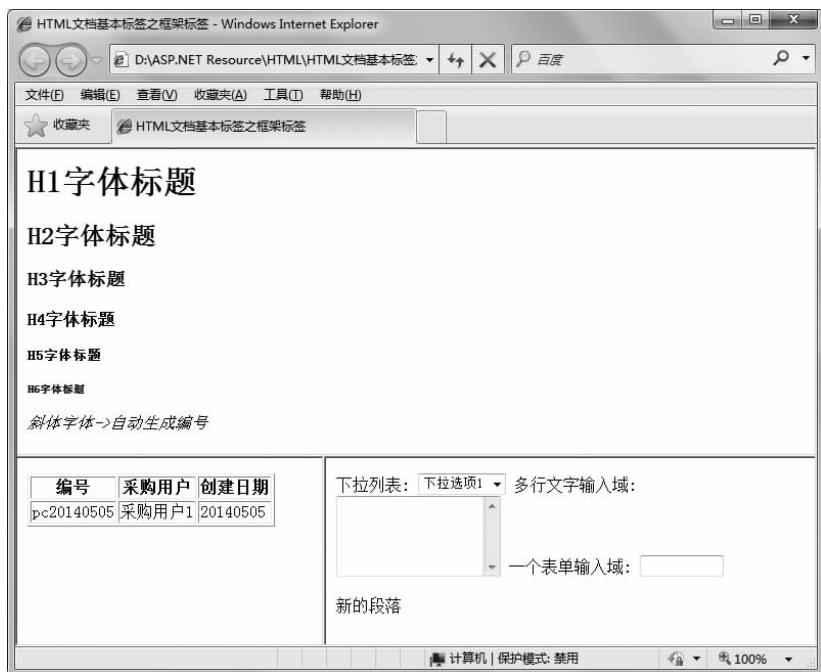


图 5-7 框架类标签在 IE 中的运行结果

在 5.1.3 节中首次提到了元素这个概念,在这里选择器其实就是设置规则的作用对象,作用对象通常是指 HTML 的元素,例如<p>、等。作用对象可以拥有多个属性,属性用来呈现网页的效果,每个属性只有一个属性值,多个属性之间用分号(;)分隔。

例 5.6 设置标题的文字效果。

代码如下:

```
1 h1 {font-family:"微软雅黑"; font-size:26pt; font-style:italic; color:#0000FF}
2 h2 {font-family:"宋体"; font-size:20pt; font-style:normal; color:#00FF00}
```

代码描述: 在例 5.6 的代码中,第 1 行的 h1 和第 2 行的 h2 都称为选择器,也就是作用对象;font-family(字体)、font-size(字体大小)、font-style(文字风格)和 color(颜色)称为属性;“微软雅黑”、26pt、italic、#0000FF 等都称为属性值。通过常用的代码编辑工具 NotePad 或记事本编辑以上代码,保存为以“.css”后缀结尾的样式文件,这里将该样式文件命名为 Style。

2) CSS 样式的引用

下面通过实例介绍在 HTML 文件中引用已经编辑好的 CSS 样式文件的方法。

例 5.7 CSS 样式结合 HTML 文件。

代码如下:

```
1 <html>
2 <head>
```

```

3      <link href="Style.css" rel="stylesheet" type="text/css"/>
4  </head>
5  <body>
6      <h1>欢迎使用 ASP.NET!</h1>
7      <h2>Welcome to use ASP.NET!</h2>
8  </body>
9 </html>

```

代码描述：例 5.7 定义了一个名为 css 的 HTML 文件，其中第 3 行 href="Style.css" 是引用的样式文件存放位置，rel="stylesheet" 指出该 link 和 href 之间的关联样式为样式表文件，type 指出该样式文件类型为文本文件。将例 5.6 的样式文件 Style.css 和例 5.7 的 HTML 文件 css.html 放在同一文件夹下，使用浏览器打开 css.html 文件，运行结果如图 5-8 所示。

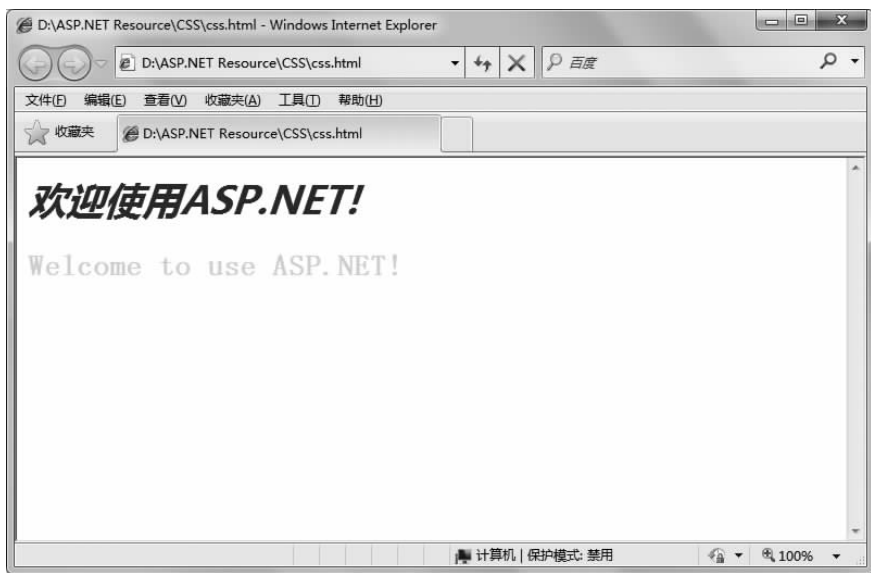


图 5-8 CSS 样式结合 HTML 文件在 IE 中的运行结果

为了实现登录界面的动画效果及验证码功能，在 Index.cshtml 视图文件中还引入了 teach-login.css 样式文件和简化版本的 JQuery 文件 jquery-1.8.2.min.js，这里限于篇幅，不做详细介绍。

例 5.8 登录页面的字体、背景色、图片等设置程序。

代码如下：

```

1 @{
2     Layout=null;
3 }
4 <!DOCTYPE html>
5 <html>
6 <head>

```