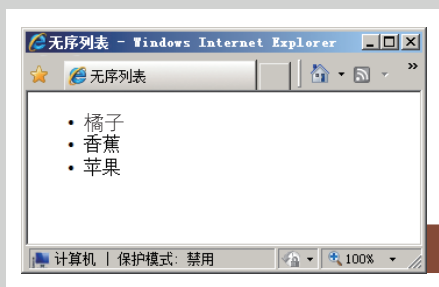




## 第2章



# HTML5的高级标签

### ● 本章目标

完成本章内容的学习以后，您将：

- ▶ 掌握列表标签。
- ▶ 掌握表格标签。
- ▶ 掌握表单标签。

### ● 本章素材下载

- ▶ 请访问课工场UI/UE学院：[kgc.cn/uiue](http://kgc.cn/uiue)  
(教材版块) 下载本章需要的案例素材。



## 本章简介

列表在网页制作中占据着重要的位置，许多精美、漂亮的网页中都使用了列表。本章将向大家介绍列表的概念及相关的使用方法，通过练习掌握列表应用的技巧，从而使读者可以制作出精美的网页。同时，在制作网页时，表格是一种不可或缺的数据展示工具，使用表格可以灵活地实现数据展示，表格在很多页面中还发挥着页面排版的作用。表单是实现用户与网页之间信息交互的基础，通过在网页中添加表单可以实现诸如会员注册、用户登录、提交资料等交互功能。本章将主要讲解如何在网页中制作表单，并使用表单元素创建表单。

# 理论讲解



参考视频  
H5 高级标签

## 2.1 列表

在网页制作中，列表有很多使用场合，如常见的树形可折叠菜单、购物网站的商品展示等。既然列表可以发挥如此巨大的作用，那么下面首先来了解一下什么是列表。

### 2.1.1 列表简介

什么是列表？简单来说，列表就是数据的一种展示形式。图 2.1 所示的数据信息就是采用列表完成的。



图 2.1 淘宝公益信息



除了图 2.1 所示的页面效果以外,在不同的场合使用列表有不同的效果。例如,在百度词典中,对于字词等的解释也可以使用列表来完成,如图 2.2 所示。



图 2.2 百度词典

通过以上的介绍,相信大家大致了解了什么是列表、列表可以做什么。那么接下来看看在 HTML5 中是如何对列表进行分类的。

### 2.1.2 列表的分类

HTML5 支持的列表形式总共有以下三种。

#### 1. 无序列表

无序列表是一个项目列表,使用项目符号标记无序的项目。在无序列表中,各个列表项之间没有顺序级别之分,它通常使用一个项目符号作为每个列表项的前缀。

#### 2. 有序列表

同样,有序列表也由一个个列表项组成,列表项既可使用数字标记,也可使用字母标记。

#### 3. 定义列表

当无序列表和有序列表都不适合时,可通过定义列表来完成数据展示,所以定义列表不仅是一个项目列表,而是项目及其注释的组合。在使用定义列表时,每一列项目前不会添加任何标记。

### 2.1.3 列表的应用

通过前面的列表介绍，大家已经了解了 HTML5 中列表的作用及列表的分类，那么，该如何使用列表呢？这就是下面将要讲解的内容——列表的使用方法。

#### 1. 无序列表的应用

无序列表使用 `<ul>` 标签作为无序列表的声明，使用 `<li>` 标签作为每个列表项的起始，在浏览器中查看到的页面效果如图 2.3 所示，可以看到 3 个列表项前面均有一个实体圆心。



图 2.3 无序列表

图 2.3 所示页面对应的代码如示例 1 所示。

#### 示例 1

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title> 无序列表 </title>
</head>
<body>
<ul>
<li> 橘子 </li>
<li> 香蕉 </li>
<li> 苹果 </li>
</ul>
</body>
</html>
```

如果希望使用无序列表，且列表项前的项目符号改用其他项目符号怎么办呢？`<ul>` 标签有一个 `type` 属性，这个属性的作用就是指定在显示列表时所采用的项目符号类型。`type` 属性的取值不同，显示的项目符号的形状也不同，其取值说明如表 2-1 所示。



表 2-1 &lt;ul&gt; 标签中 type 属性的取值

取 值	说 明
disc	项目符号显示为实体圆心，默认值
square	项目符号显示为实体方心
circle	项目符号显示为空心圆

示例 2 中分别使用了不同的 type 属性值来定义列表的项目符号。

### 示例 2

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title> 无序列表 </title>
</head>
<body>
<h4>type=circle 时的无序列表 :</h4>
<ul type="circle">
  <li> 橘子 </li>
  <li> 香蕉 </li>
  <li> 苹果 </li>
</ul>
<h4>type=disc 时的无序列表 :</h4>
<ul type="disc">
  <li> 橘子 </li>
  <li> 香蕉 </li>
  <li> 苹果 </li>
</ul>
<h4>type=square 时的无序列表 :</h4>
<ul type="square">
  <li> 橘子 </li>
  <li> 香蕉 </li>
  <li> 苹果 </li>
</ul>
</body>
</html>
```

在浏览器中查看页面效果，如图 2.4 所示。

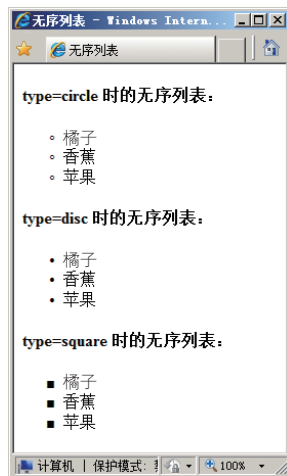


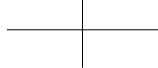
图 2.4 无序列表的 type 属性

## 2. 有序列表的应用

无序列表与有序列表的区别就在于，有序列表的各列表项有先后顺序，所以有序列表会使用数字进行标识。有序列表使用 `<ol>` 标签作为有序列表的声明，使用 `<li>` 标签作为每个列表项的起始。有序列表的代码应用如示例 3 所示。

### 示例 3

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title> 有序列表 </title>
</head>
<body>
<p> 有序列表 </p>
<ol>
<li> 橘子 </li>
<li> 香蕉 </li>
<li> 苹果 </li>
</ol>
</body>
</html>
```



在浏览器中查看页面效果，如图 2.5 所示。

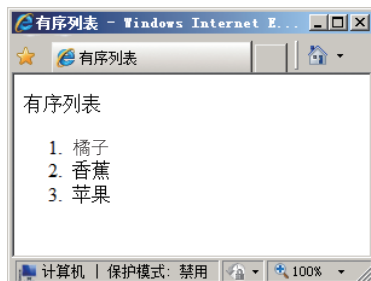


图 2.5 有序列表

与无序列表一样，有序列表的项目符号也可以进行设置。在 `<ol>` 标签中也存在一个 `type` 属性，作用同样是用于修改列表的项目符号。`type` 属性值的说明如表 2-2 所示。

表 2-2 `<ol>` 标签中 `type` 属性的取值

取 值	说 明
1 (数字)	使用数字作为项目符号
A/a	使用大写 / 小写字母作为项目符号
I/i	使用大写 / 小写罗马数字作为项目符号

不同的 `type` 属性取值，会导致列表显示的效果不同，代码如下例 4 所示。

#### 示例 4

```

<!-- 省略部分代码 -->
<h4>type=1 时的有序列表 </h4>
<ol type="1">
    <li>橘子 </li>
    <li>香蕉 </li>
    <li>苹果 </li>
</ol>
<h4>type=a 时的有序列表 </h4>
<ol type="a">
    <li>橘子 </li>
    <li>香蕉 </li>
    <li>苹果 </li>
</ol>
<!-- 省略部分代码 -->

```

在浏览器中查看页面效果，如图 2.6 所示。

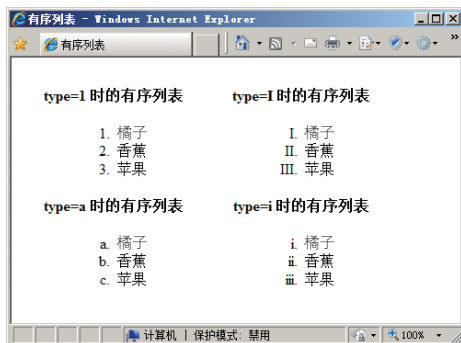


图 2.6 设置有序列表的 type 属性

### 3. 定义列表的应用

定义列表是一种很特殊的列表形式，它是标题及注释的结合。定义列表的语法相对于无序列表和有序列表不太一样，它使用 `<dl>` 标签作为列表的开始，使用 `<dt>` 标签作为每个列表项的起始，而对于每个列表项的定义则使用 `<dd>` 标签来完成。下面使用定义列表的方式来完成图 2.7 的效果。

从图 2.7 中可以看出，第一行文字“所属学院”类似于一个题目，而第二行文字“计算机应用”属于对第一行题目的解释，这种显示风格就是定义列表，其代码如下例 5 所示。

#### 示例 5

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title> 定义列表 </title>
</head>
<body>
<dl>
  <dt> 所属学院 </dt>
  <dd> 计算机应用 </dd>
  <dt> 所属专业 </dt>
  <dd> 计算机软件工程 </dd>
</dl>
</body>
</html>

```

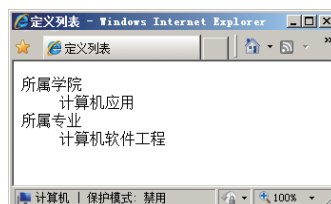


图 2.7 定义列表

到这里，我们已经学习了 HTML5 中三种列表的使用方法，归纳起来如表 2-3 所示。





表 2-3 三种列表的比较

类 型	说 明	项 目 符 号
无序列表	以 <ul> 标签来实现 以 <li> 标签定义列表项	通过 type 属性设置项目符号 包括 disc (默认)、square 和 circle
有序列表	以 <ol> 标签来实现 以 <li> 标签定义列表项	通过 type 属性设置项目顺序 包括 1 (数字, 默认)、A (大写字母)、 a (小写字母)、I (大写罗马数字) 和 i (小写罗马数字)
定义列表	以 <dl> 标签来实现 以 <dt> 标签定义列表项 以 <dd> 标签定义内容	无项目符号或显示顺序



注意

列表常用场合及列表使用中的注意事项如下:

- 无序列表中的每项都是平级的, 没有级别之分, 并且列表中的内容一般是相对简单的标题性质的网页内容。而有序列表则会依据列表项的顺序进行显示。
- 在实际的网页应用中, 无序列表 (ul-li) 比有序列表 (ol-li) 应用得更加广泛, 有序列表 (ol-li) 一般用于显示带有顺序编号的特定场合。
- 定义列表 (dl-dt-dd) 一般适用于带有标题和标题解释性内容或者图片和文本内容混合排列的场合。

## 2.2 表格

表格是块状元素, 发明表格标签的初衷是用于显示表格数据。例如, 学校中常见的考试成绩单、选修课课表, 企业中常见的工资账单等。

### 2.2.1 为什么使用表格

#### 1. 简单通用

由于表格采用行列式的简单结构, 以及它在生活中的广泛使用, 对它的理解和代码编写都很方便。

#### 2. 结构稳定

表格每行的列数通常一致, 同行单元格高度一致且水平对齐, 同列单元格宽度一致且垂直对齐。这种严格的约束形成了一个不易变形的长方形盒子结构, 堆叠排列起来结构很稳定。

## 2.2.2 表格的基本结构

先看一看表格的基本结构。表格是由指定数目的行和列组成的，如图 2.8 所示。

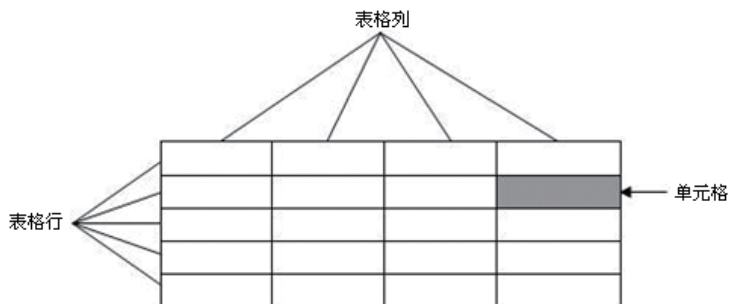


图 2.8 表格的基本结构

### 1. 单元格

单元格为表格的最小单位，一个或多个单元格纵横排列组成表格。

### 2. 表格行

一个或多个单元格横向堆叠形成行。

### 3. 表格列

由于表格单元格的宽度必须一致，因此单元格纵向排列形成列。

## 2.2.3 表格的基本语法

创建表格的基本语法如下：

```
<table>
  <tr>
    <td> 第 1 个单元格的内容 </td>
    <td> 第 2 个单元格的内容 </td>
    .....
  </tr>
  <tr>
    <td> 第 1 个单元格的内容 </td>
    <td> 第 2 个单元格的内容 </td>
    .....
  </tr>
</table>
```



创建表格一般分为三步。

第一步：创建表格标签 `<table>……</table>`。

第二步：在表格标签 `<table>……</table>` 里创建行标签 `<tr>……</tr>`，可以有多行。

第三步：在行标签 `<tr>……</tr>` 里创建单元格标签 `<td>……</td>`，可以有多个单元格。

为了显示表格的轮廓，一般还需要设置 `<table>` 标签的“border”边框属性，指定边框的宽度。例如，在页面中添加一个 2 行 3 列的表格，对应的 HTML5 代码如下例 6 所示。

### 示例 6

```
<!DOCTYPE html>
<html>
<head>
<title> 基本表格 </title>
</head>
<body>
<table border="2">
<tr>
<td>1 行 1 列的单元格 </td>
<td>1 行 2 列的单元格 </td>
<td>1 行 3 列的单元格 </td>
</tr>
<tr>
<td>2 行 1 列的单元格 </td>
<td>2 行 2 列的单元格 </td>
<td>2 行 3 列的单元格 </td>
</tr>
</table>
</body>
</html>
```

在浏览器中查看页面效果，如图 2.9 所示。

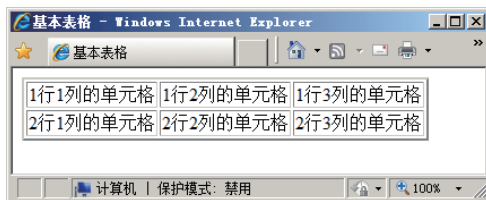


图 2.9 创建基本表格

## 2.2.4 表格的对齐方式

表格的对齐方式用来控制表格在网页中的显示位置，常见的对齐方式有默认对齐、左对齐、居中对齐和右对齐。而实现表格对齐的属性就是 **align** 属性，**align** 属性有 3 个值，分别对应左对齐、居中对齐、右对齐，当省略该属性时，系统自动采用默认对齐方式。

(1) 默认对齐。表格一经创建，便显示为默认对齐。默认对齐状态下表格以实际尺寸显示在左侧，如果旁边有内容，这些内容会显示在表格的下方，不会在表格的两侧进行排列。

(2) 居中对齐。有时候，希望表格显示在页面的中间位置，这样会使页面显得对称，浏览效果较好，这时候就需要对表格设置居中对齐。

(3) 左对齐、右对齐。如果对表格设置左对齐或者右对齐，表格会显示在页面的左侧或者右侧，其他内容会自动排列在表格旁边的空白位置。

表格的左对齐和右对齐在网页应用中相对比较少，一般用于显示广告，如网页中常见的画中画广告等。

(4) 单元格对齐。除了表格可以设置对齐方式外，单元格也同样可以设置对齐方式，单元格对齐则分为水平对齐和垂直对齐两个方向。水平对齐与垂直对齐的属性及取值，如表 2-4 所示。

表 2-4 单元格的对齐方式

属 性	值	说 明
align (水平对齐方式)	left	左对齐
	center	居中对齐
	right	右对齐
valign (垂直对齐方式)	top	顶端对齐
	middle	居中对齐
	bottom	底端对齐
	baseline	基线对齐

例如，通过下面的代码，将单元格的对齐方式改为水平右对齐、垂直底端对齐。

```
<table width="500" cellpadding="0" cellspacing="0" border="1">
  <tr>
    <td align="right" valign="bottom">……</td>
    <td>……</td>
  </tr>
</table>
```



在实际的开发过程中，表格的对齐方式通常会使用 CSS 样式表进行控制，使用属性进行对齐控制的场合比较少。

## 2.3 表单

表单在网页中应用比较广泛，如申请电子邮箱，用户需要首先填写注册信息，然后才能提交申请。又如希望登录邮箱收发电子邮件，也必须在登录页面中输入用户名及密码才能进入邮箱，这就是典型的表单应用。

通俗地讲，表单就是一个将用户信息组织起来的容器。网页将需要用户填写的内容放置在表单容器中，当用户单击“提交”按钮的时候，表单会将数据统一发送给服务器。

表单的应用比较常见，典型的应用场景如下：

- 登录、注册：登录时填写用户名、密码，注册时填写姓名、电话等个人信息。
- 网上订单：在网上购买商品，一般要求填写姓名、联系方式、付款方式等信息。
- 调查问卷：回答对某些问题的看法，以便形成统计数据，方便分析。
- 网上搜索：输入关键字，搜索想要的可用信息。

为了方便用户操作，表单提供了多种表单元素，如图 2.10 所示的页面中，除了最常见的单行文本框之外，还有密码框、单选按钮、下拉列表框、提交按钮等。图 2.10 所示是人人网用户注册页面，该页面就是由一个典型的表单构成的。



图 2.10 人人网用户注册页面

### 2.3.1 表单的内容

创建表单后，就可以在表单中放置控件以接收用户的输入。这些控件通常放在

`<form>……</form>` 标签对之间一起使用，也可以在表单之外用来创建用户界面。在网上“冲浪”时，经常会见到一些常用的控件。例如，让用户输入姓名的单行文本框，让用户输入密码的密码框，让用户选择性别单选按钮，以及让用户提交信息的提交按钮等。

不同的表单控件有不同的用途。如果要求用户输入的仅仅是一些文字信息，如“姓名”“备注”“留言”等，一般使用单行文本框或多行文本框；如果要求用户在指定的范围内做出选择，一般使用单选按钮、复选框和下拉列表框，如图 2.10 中“性别”“生日”选择等常采用这些控件；如果要把填写好的表单信息提交给服务器，一般使用提交按钮，如图 2.10 中的“立即注册找好友”按钮。除此之外，还有一些不太常用的表单控件，在这里就不一一列举了。

### 2.3.2 表单标签及其属性

HTML5 中使用 `<form>` 标签来实现表单的创建。该标签用于在网页中创建表单区域，属于一个容器标签，其他表单标签需要在它的范围内才有效，`<input>` 标签便是其中的一个，用以设定各种输入资料的方法。表单标签有两个常用的属性，如表 2-5 所示。

表 2-5 `<form>` 标签的属性

属 性	说 明
action	此属性指示服务器上处理表单输出的程序。一般来说，当用户单击表单上的提交按钮后，信息发送到 Web 服务器上，由 action 属性所指定的程序处理。语法为 <code>action = "URL"</code> 。如果 action 属性的值为空，则默认将单提交到本页
method	此属性告诉浏览器如何将数据发送给服务器，它指定向服务器发送数据的方法（用 post 方法还是用 get 方法）。如果值为 get，浏览器将创建一个请求，该请求包含页面 URL、一个问号和表单的值。浏览器会将该请求返回给 URL 中指定的脚本进行处理。如果将值指定为 post，表单上的数据会作为一个数据块发送到脚本，而不使用请求字符串。语法为 <code>method = (get   post)</code>

下面制作一个最基本的表单，然后使用 `post` 方法将表单提交给“`result.html`”页面，代码如下例 7 所示。

#### 示例 7

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title> 文本框 </title>
</head>
<body>
<form method="post" action="result.html">
<p> 名字 :<input name="name" type="text" > </p>

```



```
<p> 密码 :<input name="pass" type="password" > </p>  
<p>  
    <input type="submit" name="Button" value=" 提交 " >  
    <input type="reset" name="Reset" value=" 重填 " >  
</p>  
</form>  
</body>  
</html>
```

在浏览器中查看示例 7 的页面效果，如图 2.11 所示。

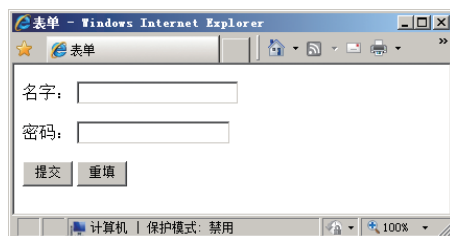


图 2.11 简单的表单

在示例 7 中，若把 `method="post"` 改为 `method="get"`，就变成了使用 `get` 方法将表单提交给“result.html”页面处理。这两种方法都是将表单数据提交给服务器上指定的程序进行处理，它们有什么区别呢？

先让大家看看采用 `post` 和 `get` 方法提交表单信息后浏览器地址栏的变化。

- 以 `post` 方式提交表单，在“名字”和“密码”后分别输入用户名“lucker”和密码“123456”，单击“提交”按钮，页面效果如图 2.12 所示。

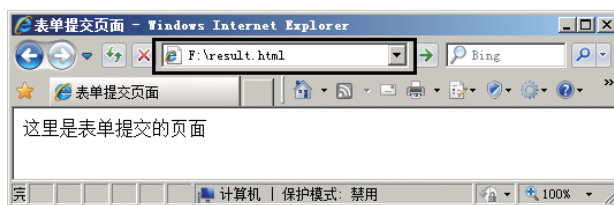


图 2.12 以 post 方式提交表单

注意：地址栏中的 URL 信息没有发生变化，这就是以 `post` 方式提交表单的特点。

- 以 `get` 方式提交表单，在页面单击“提交”按钮，页面效果如图 2.13 所示。

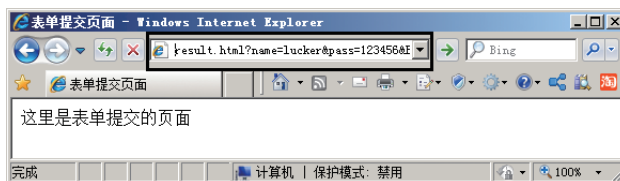


图 2.13 以 get 方式提交表单

采用 **get** 方法提交表单信息之后，在浏览器的地址栏中，**URL** 信息会发生变化。仔细观察不难发现，**URL** 信息中清晰地显示出表单提交的数据内容，即刚刚输入的用户名和密码都完全显示在地址栏中。



注意

通过对比图2.12和图2.13的效果，可以发现post和get两种提交方式之间的区别。

- (1) post提交方式不会改变地址栏状态，表单数据不会被显示。
- (2) 使用get提交方式，地址栏状态会发生变化，表单数据会在URL信息中显示。

所以，基于以上两点区别，post方式提交的数据安全性要明显高于get方式提交的数据。在日常开发中，建议大家尽可能地采用post方式来提交表单数据。

### 2.3.3 表单元素及格式

在图 2.10 中，可以看到用户注册时需要输入很多注册信息，而装载这些数据的控件，就称为表单元素。有了这些表单元素，表单才会有意义。那么如何在表单中添加表单元素呢？其实添加方法很简单，就是使用 **<input>** 标签，如示例 7 就使用 **<input>** 标签实现了向表单添加文本输入框、提交按钮、重置按钮的功能。**<input>** 标签有很多属性，下面对一些比较常用的属性进行整理，如表 2-6 所示。

表 2-6 <input> 标签的属性

属 性	说 明
type	此属性指定表单元素的类型。可用的选项有 text、password、checkbox、radio、submit、reset、file、hidden、image 和 button。默认选择为 text
name	此属性指定表单元素的名称。例如，如果表单上有几个文本框，可以按名称来标识它们，如 text1、text2 等
value	此属性是可选属性，它指定表单元素的初始值。如果 type 为 radio，则必须指定一个值
size	此属性指定表单元素的初始宽度。如果 type 为 text 或 password，则表单元素的大小以字符为单位。对于其他输入类型，宽度以像素为单位
maxlength	此属性用于指定可在 text 或 password 元素中输入的最大字符数。默认值为无限大
checked	此属性指定按钮是否是被选中的。当输入类型为 radio 或 checkbox 时，使用此属性

到目前为止，大家已经知道了如何在页面中添加表单，也掌握了如何向表单添加表单元素，那么这么多表单元素都该如何使用呢？下面选取几个常用的表单元素，来逐一学习其类型及常用的属性。

#### 1. 文本框

在表单中最常用、最常见的表单输入元素就是文本框 (**text**)，它用于输入单行文本信息，如用户名的输入框。若要在文档的表单里创建一个文本框，将表单元素 **type** 属性设为 **text** 就可以了。









并不能保证输入的数据安全。为了使数据安全，应该加强人为管理，采用数据加密技术等。

### 3. 单选按钮

单选按钮控件用于选择一组相互排斥的值，组中的每个单选按钮控件应具有相同的名称，用户一次只能单击一个单选按钮。只有从组中单击选按钮才会在提交的数据中提交对应的数值。在使用单选按钮时，需要一个显式的 `value` 属性。

#### 示例 10

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title> 单选按钮 </title>
</head>
<body>
<form method="post" action="">
性别：
  <input name="gen" type="radio" class="input" value="男" > 男 &nbsp;&nbsp;&nbsp;
  <input name="gen" type="radio" value="女" class="input"> 女
</form>
</body>
</html>
```

运行示例 10 的代码，在浏览器中预览效果，如图 2.16 所示。

如果希望在页面加载单选按钮时有一个默认选项，那么可以使用 `checked` 属性。例如，性别选项默认选中为“男”，则修改代码如下：

```
<input name="gen" type="radio" class="input" value="男" checked="checked"> 男
```

此时，再次运行示例 10，则页面效果如图 2.17 所示。



图 2.16 单选按钮效果

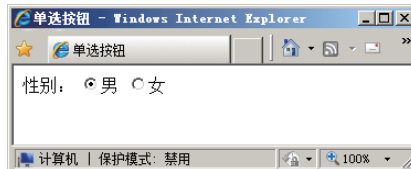


图 2.17 使用 checked 属性设置默认选项

### 4. 复选框

复选框与单选按钮有些类似，只不过复选框允许用户勾选多个选项。复选框的类型是 `checkbox`，即将表单元素的 `type` 属性设为 `checkbox` 就可以创建一个复选框。复选框的

命名与单选按钮有些区别，既可以多个复选框选用相同的名称，也可以各自具有不同的名称，关键是看如何使用复选框。用户可以勾选某个复选框，也可以取消勾选。一旦用户勾选了某个复选框，在表单提交时，会将该复选框的 **name** 值和对应的 **value** 值一起提交。

### 🔥 示例 11

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>checkbox</title>
</head>
<body>
<form method="post" action="">
  爱好：
  <input type="checkbox" name="interest" value="sports"> 运动
  <input type="checkbox" name="interest" value="talk"> 聊天
  <input type="checkbox" name="interest" value="play"> 玩游戏
</form>
</body>
</html>
```

运行示例 11 的代码，在浏览器中预览效果，如图 2.18 所示。

与单选按钮一样，复选框也可以设置默认选项，同样使用 **checked** 属性进行设置。例如，将爱好中的“运动”选项默认选中，则代码修改如下：

```
<input type="checkbox" name="cb1" value="sports" checked="checked"> 运动
```

运行效果如图 2.19 所示。



图 2.18 复选框效果



图 2.19 设置默认勾选的复选框

### 经验总结

单选按钮应具有相同的名称，便于互斥选择；而复选框的名称则要根据应用环境来确定是否相同。通常情况下，如果选项之间是并列关系，就需要设置为相同的名称，以便能够同时获取，例如兴趣爱好。一个人可以有多个兴趣爱好，这样复选框设置相同名称，在提交数据时能够一次性得到所有选择的兴趣爱好选项，否则，每个选项都需要单独进行读取，从而降低了效率。



## 5. 下拉列表框

下拉列表框主要是为了用户快速、方便、正确地选择一些选项，并且节省页面空间，它是通过 `<select>` 标签和 `<option>` 标签来实现的。`<select>` 标签用于显示可供用户选择的下拉列表，每个选项由一个 `<option>` 标签表示，`<select>` 标签必须包含至少一个 `<option>` 标签。相关代码如下：

```
<select name=" 指定列表名称 " size=" 行数 ">
  <option value=" 可选项的值 " selected="selected">……</option>
  <option value=" 可选项的值 ">……</option>
</select>
```

其中，在有多条选项可供用户滚动查看时，`size` 确定列表中可同时看到的行数；`selected` 表示该选项在默认情况下是被选中的，而且一个下拉列表框中只能有一个列表项默认被选中，如同单选按钮组那样。

### 示例 12

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title> 下拉列表框 </title>
</head>
<body>
出生日期 :
<input name="byear" value="yyyy" size="4" maxlength="4"> &nbsp;&nbsp;&nbsp;&nbsp; 年
<select name="bmon">
  <option value="">[ 选择月份 ]</option>
  <option value="1"> 一月 </option>
  <option value="2"> 二月 </option>
  <option value="3"> 三月 </option>
  <option value="4"> 四月 </option>
  <option value="5"> 五月 </option>
  <option value="6"> 六月 </option>
  <option value="7"> 七月 </option>
  <option value="8"> 八月 </option>
  <option value="9"> 九月 </option>
  <option value="10"> 十月 </option>
  <option value="11"> 十一月 </option>
```

```

    <option value="12"> 十二月 </option>
  </select> 月 &nbsp;
  <input name="bday" value="dd" size="2" maxlength="2" > 日
</form>
</body>
</html>

```

运行示例 12 的代码，在浏览器中预览效果，如图 2.20 所示。

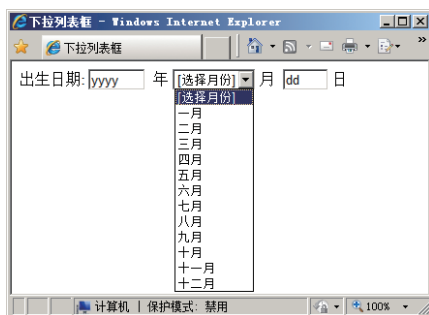


图 2.20 列表框效果

下拉列表框中添加的 `option` 选项会按照顺序进行排列，但是如果希望其中某个选项默认显示，就需要使用 `selected` 属性来进行设置。例如，让月份默认显示“十月”，则相应代码修改如下：

```
<option value="10" selected="selected"> 十月 </option>
```

设置了 `selected` 属性后，则下拉列表框会默认显示“十月”，如图 2.21 所示。



图 2.21 设置下拉列表框的默认显示

## 6. 按钮

按钮在表单中经常用到，在 HTML5 中按钮分为三种，分别是普通按钮（`button`）、提交按钮（`submit`）和重置按钮（`reset`）。普通按钮主要用来响应 `onclick` 事件，提交按钮用来提交表单信息，重置按钮用来清除表单中已填信息。相关代码如下：

```
<input type="reset" name="Reset" value=" 重填 ">
```

其中，`type="button"` 表示普通按钮；`type="submit"` 表示提交按钮；`type="reset"` 表示重置按钮。`name` 用来给按钮命名，`value` 用来设置显示在按钮上的文字。



并传递表单数据。

(3) **button** 按钮：属于普通的按钮，需要与事件关联使用。示例 13 的代码中为普通按钮添加了一个 **onclick** 事件，当用户单击该按钮时，将会显示该按钮的 **value** 值，页面效果如图 2.23 所示。

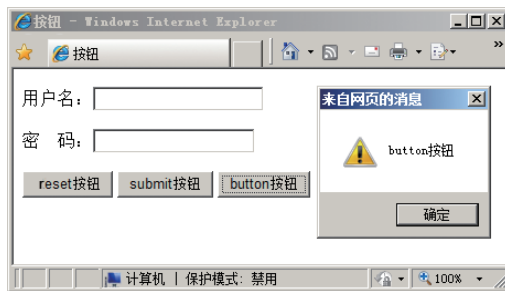


图 2.23 普通按钮的 onclick 事件

**onclick** 事件是表单元素被单击时所激发的事件，并只限于按钮。在事件中可以调用相应的脚本代码，执行一些特定的客户端程序。

有时候，在页面使用按钮，显示的样式不美观，所以在实际开发过程中，往往会使用图片按钮来代替，如图 2.24 所示。



图 2.24 图片按钮的效果

实现图片按钮的效果有多种方法，比较简单的方法就是配合使用 **type** 和 **src** 属性，代码如下：

```
<input type="image" src="images/login.gif" />
```



注意

这种方式实现的图片按钮比较特殊，虽然 **type** 属性没有设置为“**submit**”，但仍然具备提交功能。

## 7. 多行文本域

当需要在网页中输入两行或两行以上的文本时，怎么办？显然，前面学过的文本框及其他表单元素都不能满足要求，这就应该使用多行文本框，它使用的标签是 **<textarea>**。相关代码如下：





```
<textarea name="textarea" cols=" 显示的列的宽度 " rows=" 显示的行数 ">
    文本内容
</textarea>
```

其中，`cols` 属性用来指定多行文本框的列的宽度；`rows` 属性用来指定多行文本框的行数。在 `<textarea>……</textarea>` 标签对中不能使用 `value` 属性来赋初始值。

#### 示例 14

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title> 文本域 </title>
</head>
<body>
<form method="post" action="">
<H4> 填写个人评价 </H4>
<p>
    <textarea name="textarea" cols="40" rows="6">
        自信、活泼、善于思考……
    </textarea>
</p>
</form>
</body>
</html>
```

运行示例 14 的代码，在浏览器中预览效果，如图 2.25 所示。



图 2.25 多行文本框效果

## 8. 文件域

文件域用于实现文件的选择，在应用时只需把 `type` 属性设为“file”即可。在实际应用中，文件域通常应用于文件上传的操作，如选择需要上传的文本、图片等。

## 示例 15

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title> 文件域 </title>
</head>
<body>
<form action="" method="post" enctype="multipart/form-data">
  <p><input type="file" name="files" /><br/>
  <input type="submit" name="upload" value=" 上传 " /></p>
</form>
</body>
</html>

```

运行示例 15 的代码，在浏览器中预览效果，如图 2.26 所示。



图 2.26 文件域

如图 2.26 所示，文件域会创建一个不能输入内容的地址文本框和一个“浏览”按钮。单击“浏览”按钮，将会弹出“选择要加载的文件”对话框，选择文件后，路径将显示在地址文本框中，执行的效果如图 2.27 所示。

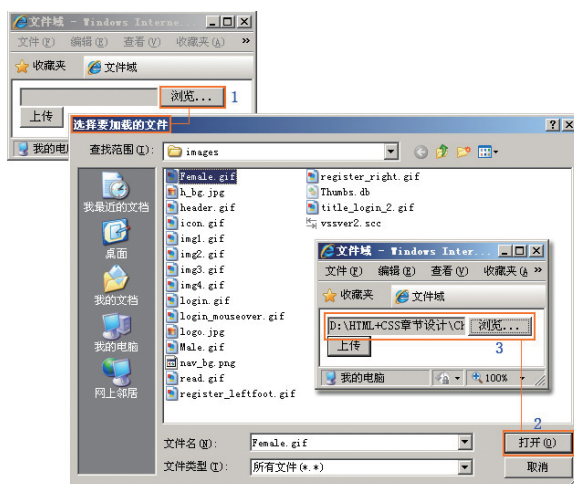


图 2.27 文件域与上传操作



在使用文件域时，需要特别注意的是包含文件域的表单，由于提交的表单数据包括普通的表单数据、文件数据等多部分内容，因此必须设置表单的“enctype”编码属性为“multipart/form-data”，这表示将表单数据分为多部分提交。



# 实战案例

## 实战案例 1——制作树形菜单页面

### 需求描述

模拟“我的电脑”中的磁盘文件管理，显示磁盘与文件夹之间的关系，完成效果如图 2.28 所示。

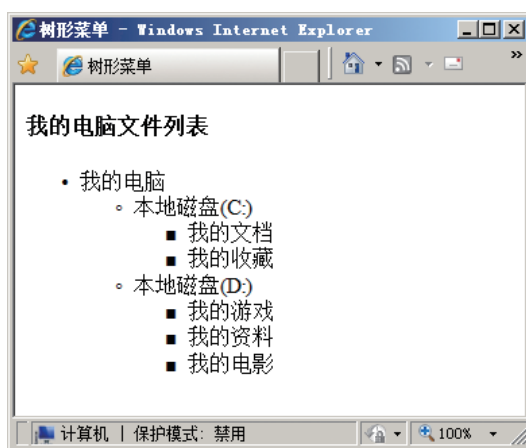


图 2.28 模拟树形菜单页面

### 技能要点

- 以 `<ul>` 标签来实现。
- 以 `<li>` 标签表示列表项。

### 实现思路

- 以 `<ul>` 标签和 `<li>` 标签来实现上述无序列表样式。

## 实战案例 2——制作模拟考试试卷页面

### 需求描述

模拟考试试卷选择题的题型格式，使用有序列表完成模拟试卷，完成效果如图 2.29 所示。

### 技能要点

- 以 `<ol>` 标签来实现。
- 以 `<li>` 标签表示列表项。



### 实现思路

以 `<ol>` 标签和 `<li>` 标签来实现上述有序列表样式。



图 2.29 模拟试卷选择题页面

## 实战案例 3——制作易趣网商品列表页面

### 需求描述

使用定义列表制作易趣网商品列表页面，页面效果如图 2.30 所示。

### 技能要点

- 以 `<dl>` 标签来实现。
- 以 `<dt>` 标签定义列表项。
- 以 `<dd>` 标签定义内容。

### 实现思路

把图片作为商品的标题性内容放在 `<dt>` 标签中，把价格和商品的简单介绍放在 `<dd>` 标签中。



图 2.30 易趣网商品列表页面

## 实战案例 4——制作淘宝店铺列表页面

### 需求描述

制作如图 2.31 所示的淘宝店铺列表页面。



图 2.31 淘宝店铺列表页面



### 技能要点

- 学会使用表格。
- 掌握表格、单元格常用属性的用法。
- 学会使用表格嵌套制作页面。

### 实现思路

- 构建网页结构，分析表格的嵌套关系。
- 合理地对表格进行嵌套。
- 对需要合并的单元格进行合并。
- 设置单元格内文本的对齐方式。
- 在单元格内插入内容。

## 实战案例 5——制作新浪微博页面

### 需求描述

制作如图 2.32 所示的新浪微博首页。



图 2.32 新浪微博首页

### 技能要点

- 学会使用表格。
- 掌握表格、单元格常用属性的用法。
- 学会使用表格嵌套制作页面。

### 实现思路

- 构建网页结构，分析表格的嵌套关系。
- 合理地对表格进行嵌套。
- 对需要合并的单元格进行合并。
- 设置单元格内文本的对齐方式。
- 在单元格内插入内容。

## 实战案例 6——制作网易邮箱登录页面

### 需求描述

制作如图 2.33 所示的网易邮箱登录页面。

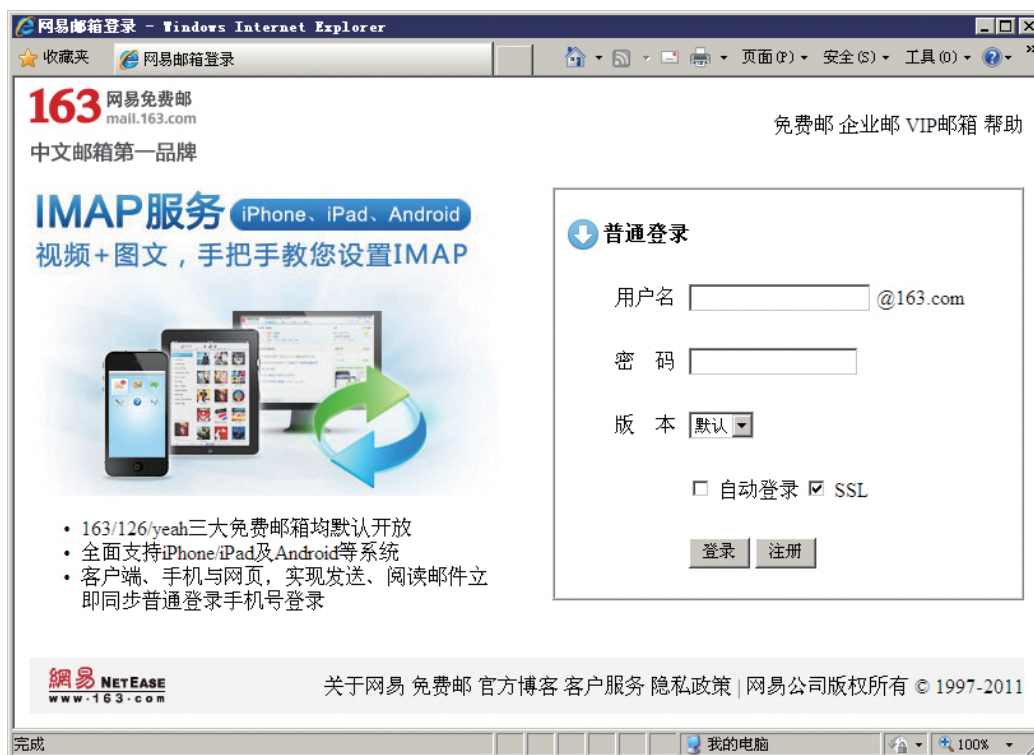


图 2.33 网易邮箱登录页面

### 技能要点

- 学会使用表格布局表单。
- 学会常见表单元素的创建方法。

### 实现思路





- 首先应当使用表格来制作页面的整体布局，在表格内插入图片、文本等内容。
- 在登录区域，插入一个表格，设置边框、填充，然后在表格内插入一个表单。
- 在表单中，再插入表格来控制表单元素的显示位置。
- 页面中用到的表单元素有单行文本框、密码框、下拉列表框、复选框、提交按钮和普通按钮。

## 实战案例 7——制作阿里巴巴会员注册页面

### 需求描述

- 制作如图 2.34 所示的阿里巴巴会员注册页面。
- 电子邮箱、登录名、密码最多能容纳的字符数是 32 个，验证码最多能容纳 5 个字符。
- 默认情况下，会员身份中的“买家”处于选中状态。
- 提交按钮使用课工场 UI/UE 学院 kgc.cn/uiue 提供的本章素材中相应的图片代替。



图 2.34 阿里巴巴会员注册页面

### 技能要点

- 学会使用表格布局表单。
- 学会常见表单元素的创建方法。

### 实现思路

- 构建网页结构，分析表格的嵌套关系，制作使用表格布局的页面。
- 在注册表单所在的位置插入表单。
- 在表单内插入表格，用来容纳文本及表单元素。
- 在对应的单元格内插入表单元素。

### 难点提示

- 会员身份中的单选按钮名称必须相同，单选按钮的默认选中状态可以在“属性”面板中进行设置，或者在“代码”视图下添加 `checked` 属性。
- 图片按钮使用 `type` 和 `src` 属性来实现。

## 实战案例 8——制作人人网注册页面

### 需求描述

- 制作如图 2.35 所示的人人网注册页面。
- 注册邮箱、密码、姓名和验证码最多能容纳的字符数分别是 50 个、16 个、8 个和 5 个。



图 2.35 人人网注册页面



- 默认情况下，性别中的“男”处于选中状态。
- 生日下拉列表框中的“1991年10月30日”处于选中状态。
- 提交按钮使用课工场 UI/UE 学院 [kgc.cn/uiue](http://kgc.cn/uiue) 提供的本章素材中相应的图片代替。

#### 技能要点

- 学会使用表格布局表单。
- 学会常见表单元素的创建方法。

#### 实现思路

- 构建网页结构，分析表格的嵌套关系，制作使用表格布局的页面。
- 在注册表单所在的位置插入表单。
- 在表单内插入表格，用来容纳文本及表单元素。
- 在对应的单元格内插入表单元素。