

第七讲：表单

学时计划：2 学时 理论，1 学时 实验

(实验四：在网页中使用表单和多媒体)

教学大纲：

- 1、表单基础
- 2、input
- 3、新增的表单属性
- 4、新增的表单元素
- 5、表单验证
- 6、讨论与思考

在开发 Web 应用程序的过程中，表单是页面上非常重要的内容，因为用户输入的内容都是在表单上完成的，用户和后台程序的交互也是通过表单实现的。

本讲详细介绍 HTML 5 中表单的元素、属性，以及 HTML 5 新增增加和扩展的元素和属性，并对表单验证、表单行为等内容进行了介绍。

一、表单基础

HTML 使用表单向服务器提交请求，表单、表单控件的主要作用就是收集用户输入，当用户提交表单时，用户输入的内容将被作为请求参数提交到远程服务器。因此，在 Web 编程中，表单主要用于收集用户输入的数据。

表单是进行用户交互的主要途径，因此表单在 Web 开发中有着重要的意义。

1.1 表单元素与表单属性

元素介绍：

form 元素用于生成输入表单，该元素不会生成可视化的部分。在 HTML 5 规范以前，其他的表单控件都必须放置在该元素内部。HTML 5 之后，表单元素增加了 form 属性，所以可以单独存在。

主要属性：

(1) action: 指定当点击表单内的“确认”或“提交”按钮时，该表单将提交给 action 指定的程序来接受和响应。该属性值可以是一个相对路径或者 URL。

(2) method: 指定提交该表单时，采用的数据传输方式，即提交表单时发送何种类型的请求。该属性值为“GET”或“POST”。

(3) name: 指定表单的唯一名称。

(4) target: 指定使用哪种方式来打开 action 指定的响应程序页面。

(5) enctype: 指定表单内的文字编码方式。该属性值为“application/x-www-form-urlencoded”、“multipart/form-data”、“text/plain”。

案例：

示例 7-1：表单提交方式

使用 GET 方式提交表单：

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>表单属性-GET</title>
</head>

<body>
<form method="get" name="UserLogin" action="7-1-S.php">
<p>账号： <input type="text" name="UserName"
maxlength="20" width="30">
<p>口令： <input type="password" name="UserPWD"
maxlength="20" width="30">
<p><input type="submit" name="Submit" value="提交表单">
</form>
</body>
</html>
```

利用 POST 方式提交表单：

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>表单属性-POST</title>
```

```
</head>

<body>
<form method="post" name="UserLogin" action="7-1-S.php">
<p>账号: <input type="text" name="UserName"
      maxlength="20" width="30">
<p>口令: <input type="password" name="UserPWD"
      maxlength="20" width="30">
<p><input type="submit" name="Submit" value="提交表单">
</form>
</body>
</html>
```

1.2 标签

元素介绍:

label 元素主要用在表单元素中定义标签, 以实现对其他表单元素的说明。

元素说明:

(1) <label>元素可以使用 id、style、class 等核心属性, 但通常不需要 value 属性, 因为该元素不传递值。

(2) <label>元素只是输出了文本内容。使用该标签而不直接使用文字内容的原因在于, 使用该标签可以使得和该元素相关联的元素形成焦点。部分浏览器是支持。

案例:

示例 7-2: 标签

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>label</title>
</head>

<body>
<form method="post" name="UserLogin" action="7-2-S.php">
<p>
  <label for="UserName">账号: </label>
  <input type="text" name="UserName" width="30">
</p>
```

```
<label for="UserPWD">口令: </label>
  <input type="password" name="UserPWD" width="30">
<p><input type="submit" name="Submit" value="提交表单">
</form>
</body>
</html>
```

1.3 按钮

元素介绍:

button 元素用于定义一个按钮，在该元素内部可以包含普通文本、文本格式化标签、图像等内容。

主要属性:

(1) disabled: 指定该元素是否可用。在进行权限管理时，可以指定该元素不可用等。

(2) name: 指定该元素的唯一名称。

(3) type: 指定该元素所定义的按钮类型，属性值是“button”、“reset”、“submit”之一。

(4) value: 指定该元素的按钮初始值。

元素说明:

(1) <button>和<input>元素都可以实现按钮功能，但前者的功能更加全面。

(2) 不要在<button>元素内放置图像映射，鼠标和键盘的一些操作会影响按钮的行为。

(3) 可以在<button>元素内放置图像，实现图像按钮。

案例:

示例 7-3: 按钮

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>button</title>
</head>

<body>
```

```
<form name="UserLogin" method="get" action="7-3-S.php">

<input type="text" width="30"><br>

<input type="password" width="30"><br>
<button type="submit" style="margin:0px; border:none;">
  
</button>
</form>
</body>
</html>
```

1.4 列表框与下拉菜单

元素介绍:

select 元素用于创建列表框或下拉菜单，该元素需要和 option 元素结合使用，每个 option 元素代表一个列表项或菜单项。

select 元素只能够包含两种子元素:

<option>元素: 用于定义列表框或菜单项。

<optgroup>元素: 用于定义列表项或菜单组项。该元素内包含 <option>元素构成组。

主要属性:

(1) <select>元素可以指定 ID、style、class 等属性，还可以指定 onchange 事件属性，当该列表框或下拉列表项内的选中选项发生改变时，出发 onchange 事件。

<select>元素的其他属性有:

disabled: 指定该元素禁止使用。

multiple: 指定该元素内的选项是否可以是多项选择，默认是单向选择。

size: 指定该列表框内可以同时显示多少个列表项。

(2) <option>元素可以指定 ID、style、class 等属性，还可以指定 onclick 等事件属性。

<option>元素的其他属性有:

disabled: 指定该元素禁用。

selected: 指定该选项初始为选中状态。

value: 指定该选项被选中时对应的值。

(3) <optgroup>元素可以指定 ID、style、class 等属性，还可以指定 onclick 等事件属性。

<optgroup>元素的其他属性有：

lable: 指定该选项组的标签，此属性为必须填写的项目。

disabled: 指定该选项组禁止使用。

案例：

示例 7-4：列表框与下拉菜单

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>select</title>
</head>

<body>
<form>
<h1>用户注册</h1>
<p>账号名称： <input type="text" name="UserName" width="30">
<p>账号口令： <input type="password" name="UserPWD" width="30">
<p>重复口令： <input type="password" name="ReUserPWD" width="30">
<p>所在地区：
  <select name="Area" size="2">
    <optgroup label="-----中国-----">
      <option value="BeiJing">北京</option>
      <option value="ShangHai">上海</option>
      <option value="GuangZhou">广州</option>
      <option value="HeNan">河南</option>
      <option value="HeBei">河北</option>
    </optgroup>
    <optgroup label="-----世界-----">
      <option value="YanZhou">亚洲</option>
      <option value="FeiZhou">非洲</option>
      <option value="MeiZhou">美洲</option>
    </optgroup>
  </select>
<p><input type="submit" name="Submit" value="进行注册">
</form>
</body>
</html>
```

1.5 文本区域

元素介绍：

textarea 元素用于生成多行文本框。

主要属性：

(1) <textarea>元素可以指定 ID、style、class 等元素，还可以指定 onclick 等事件属性。

(2) cols: 指定文本域的宽度，此属性为必须填写的项目。

(3) rows: 指定文本域的高度，此属性为必须填写的项目。

(4) disabled: 指定该元素禁止使用。

(5) readonly: 指定该元素为只读状态。

案例：

示例 7-5：文本框

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>textarea</title>
</head>

<body>
<form>
<p>新闻标题：<input type="text" name="NewsTitle" style="width:370px;">
<p>新闻内容：<textarea name="NewsContent" cols="50" rows="10">请在此
输入新闻内容，不得少于 80 个字。</textarea>
<p>发布说明：<textarea name="NewsPubInfo" cols="50" rows="2"
readonly>请认真填写新闻内容，不要进行强制排版。感谢您的支持。此文本
框为只读属性。</textarea>
<p><input type="submit" value="发布新闻">
</form>
</body>
</html>
```

二、input

表单中最重要的元素是 input。该元素通过使用 type 属性，可以改变元素的性质。

在 HTML 5 中，大幅度的增加和改良了 input 元素的种类，并且可以通过新增的这些种类，实现常用的需要 JavaScript 才能够实现的功能。

本部分主要介绍 input 元素的各种类型。

2.1 input 的属性

表 7-1 input 的属性

属性	属性值	描述
accept	list_of_mime_types	规定可通过文件上传控件提交的文件类型。 (仅适用于 type="file")
alt	text	规定图像输入控件的替代文本。 (仅适用于 type="image")
autocomplete	on off	规定是否使用输入字段的自动完成功能。
autofocus	autofocus	规定输入字段在页面加载时是否获得焦点。 (不适用于 type="hidden")
checked	checked	规定当页面加载时是否预先选择该 input 元素。 (适用于 type="checkbox" 或 type="radio")
disabled	disabled	规定当页面加载时是否禁用该 input 元素。 (不适用于 type="hidden")
form	formname	规定输入字段所属的一个或多个表单。
formaction	URL	覆盖表单的 action 属性。 (适用于 type="submit" 和 type="image")
formenctype	见注释	覆盖表单的 enctype 属性。 (适用于 type="submit" 和 type="image")
formmethod	get post	覆盖表单的 method 属性。 (适用于 type="submit" 和 type="image")
formnovalidate	formnovalidate	覆盖表单的 novalidate 属性。 如果使用该属性，则提交表单时不进行验证。
formtarget	_blank _self _parent _top framename	覆盖表单的 target 属性。 (适用于 type="submit" 和 type="image")
height	pixels %	定义 input 字段的高度。(适用于 type="image")
list	datalist-id	引用包含输入字段的预定义选项的 datalist 。
max	number date	规定输入字段的最大值。 请与 "min" 属性配合使用，来创建合法值的范围。
maxlength	number	规定文本字段中允许的最大字符数。

min	number date	规定输入字段的最小值。 请与 "max" 属性配合使用，来创建合法值的范围。
multiple	multiple	如果使用该属性，则允许一个以上的值。
name	field_name	规定 input 元素的名称。 name 属性用于在提交表单时搜集字段的值。
pattern	regexp_pattern	规定输入字段的值的模式或格式。 例如 pattern="[0-9]" 表示输入值必须是 0 与 9 之间的数字。
placeholder	text	规定帮助用户填写输入字段的提示。
readonly	readonly	指示字段的值无法修改。
required	required	指示输入字段的值是必需的。
size	number_of_char	规定输入字段中的可见字符数。
src	URL	规定图像的 URL。(适用于 type="image")
step	number	规定输入字的的合法数字间隔。
type	button checkbox date datetime datetime-local email file hidden image month number password radio range reset submit text time url week	规定 input 元素的类型。
value	value	对于按钮：规定按钮上的文本 对于图像按钮：传递到脚本的字段的符号结果 对于复选框和单选按钮：定义 input 元素被点击时的结果。 对于隐藏、密码和文本字段：规定元素的默认值。 注释：不能与 type="file" 一同使用。 注释：对于 type="checkbox" 以及 type="radio"，是必需的。
width	pixels %	定义 input 字段的宽度。(适用于 type="image")

2.2 电子邮件类型

元素介绍：

在 HTML 4 中，如果需要输入电子邮件地址，只能使用 text 类型。HTML 5 中新增加电子邮件类型。浏览器会自动检验输入的内容是否为合法的电子邮件地址，如果不能通过验证，则表单就不会发送。

案例：

示例 7-6：电子邮件类型

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>电子邮件类型</title>
</head>

<body>
<form action="form.php" method="post" name="FomeDemo">
<p>请输入电子邮件地址： <input type="email" name="AddressEmail">
<p><input type="submit" value="确定" name="FormSubmit">
</form>
</body>
</html>
```

2.3 URL 类型

元素介绍：

URL 类型和 email 类型相同，但是主要是针对于输入网址。该类型只接受绝对 URL，如果输入无效的 URL，会提示错误。

案例：

示例 7-7：URL 类型

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>URL 类型</title>
</head>

<body>
```

```
<form action="form.php" method="post" name="FomeDemo">
<p>请输入个人网站地址: <input type="url" name="AddressURL">
<p><input type="submit" value="确定" name="FormSubmit">
</form>
</body>
</html>
```

讨论:

使用 email 和 url 属性时, iPhone 的输入键盘发生了变化。



2.4 搜索类型

元素介绍:

search 类型不能够控制输入, 它的作用是指示浏览器输入框为搜索框。该类型和 text 类型看上去相同, 但是如果用户输入内容, 在输入框的后面会出现一个删除按钮。

案例:

示例 7-8: 搜索类型

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>搜索类型</title>
</head>

<body>
<form action="form.php" method="post" name="FomeDemo">
<p>请输入搜索内容: <input type="search" name="SearchText">
<p><input type="submit" value="进行搜索" name="FormSubmit">
</form>
</body>
</html>
```

2.5 电话号码类型

元素介绍:

tel 类型和 URL、email 类型不同, 它没有规定特殊的语法。该类型也是用于指示浏览器。

案例:

示例 7-9: 电话号码类型

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>电话号码类型</title>
</head>

<body>
<form action="form.php" method="post" name="FomeDemo">
<p>请输入电话号码: <input type="tel" name="telephoneNum">
<p><input type="submit" value="确定" name="FormSubmit">
</form>
</body>
</html>
```

2.6 数字类型

元素介绍:

number 类型只能接受数字值, 该类型通常和 min、max、step 属

性结合使用。

min 属性：决定了该输入框能够接受的最小值。

max 属性：决定了该输入框能够接受的最大值。

step 属性：决定了该输入框接受值递增或者递减的步长。

案例：

示例 7-10：数字类型

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>数字类型</title>
</head>

<body>
<form action="form.php" method="post" name="FomeDemo">
<p>请输入数字：<input type="number" name="InputNum">
<p>请输入 10-15 之间的数字：<input type="number" min="10" max="15"
name="InputNum2">
<p>请输入 10-20 之间的偶数：<input type="number" min="10" max="20"
step="2" name="InputNum3">
<p><input type="submit" value="确定" name="FormSubmit">
</form>
</body>
</html>
```

2.7 范围类型

元素介绍：

range 类型允许用户选择一个范围的数值。通过一个滑动快让用户进行选择。range 必须要制定 min、max 来进行输入值得上限制和下限值的定义。可以设置 step 来进行步长的定义。

案例：

示例 7-11：范围类型

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>范围类型</title>
</head>
```

```
<body>
<form action="form.php" method="post" name="FomeDemo">
<p> 请输入数字： <input type="range" min="1" max="100" step="5"
name="RangeNum">
<p><input type="submit" value="确定" name="FormSubmit">
</form>
</body>
</html>
```

2.8 日期类型

元素介绍：

date 类型用于创建一个更好的日期输入框，该类型会自动创建一个日历供用户选择。

案例：

示例 7-12：日期类型

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>日期类型</title>
</head>

<body>
<form action="form.php" method="post" name="FomeDemo">
<p>请选择航班日期： <input type="date" name="DateValue">
<p><input type="submit" value="确定" name="FormSubmit">
</form>
</body>
</html>
```

2.9 周类型

元素介绍：

week 类型使用上和 date 类型基本相同，但是只能够选择周。

案例：

示例 7-13：周类型

```
<!doctype html>
```

```
<html>
<head>
<meta charset="utf-8">
<title>周类型</title>
</head>

<body>
<form action="form.php" method="post" name="FomeDemo">
<p>请选择工作周: <input type="week" name="WeekValue">
<p><input type="submit" value="确定" name="FormSubmit">
</form>
</body>
</html>
```

2.10 月份类型

元素介绍:

month 类型和 date 类型相似,但是只允许选择月份。

案例:

示例 7-14: 月类型

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>月类型</title>
</head>

<body>
<form action="form.php" method="post" name="FomeDemo">
<p>请选择月份: <input type="month" name="WeekValue">
<p><input type="submit" value="确定" name="FormSubmit">
</form>
</body>
</html>
```

2.11 时间类型

元素介绍:

time 类型与 date 类似,但是只能选择时间。它的格式包括小时和分钟。

案例：

示例 7-15：时间类型

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>时间类型</title>
</head>

<body>
<form action="form.php" method="post" name="FomeDemo">
<p>请选择出发的时间： <input type="date" name="GoDate"><input
type="time" name="GoTime">
<p><input type="submit" value="确定" name="FormSubmit">
</form>
</body>
</html>
```

2.12 日期与时间类型

元素介绍：

datetime 类型支持输入完整的日期和时间。

案例：

示例 7-16：日期与时间类型

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>日期与时间类型</title>
</head>

<body>
<form action="form.php" method="post" name="FomeDemo">
<p>请选择出发的时间： <input type="datetime" name="GoDateTime">
<p><input type="submit" value="确定" name="FormSubmit">
</form>
</body>
</html>
```

2.13 颜色类型

元素介绍：

color 类型用于定义一个拾色器。

案例：

示例 7-17：颜色类型

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>日期与时间类型</title>
</head>

<body>
<form action="form.php" method="post" name="FomeDemo">
<p>请选择色彩值： <input type="color" name="ColorValue">
<p><input type="submit" value="确定" name="FormSubmit">
</form>
</body>
</html>
```

三、新增的表单属性

HTML 5 为表单空间新增了大量属性，这些属性都是为 HTML 5 提供了一些非常实用和常用功能。以前，通常使用 JavaScript 实现这些功能。

3.1 占位字符：placeholder 属性

元素介绍：

placeholder 属性通常用于 search 类型，但也可以用于 text 等类型。它表示一个简短提示、单词或者词汇，可用于帮助用户输入正确的内容。这个属性的价值是可以在输入框内设置占位字符，占位字符会在输入框为空或者失去焦点时显示出来。

案例：

示例 7-18：占位字符

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
```

```
<title>占位字符</title>
</head>

<body>
<form action="form.php" method="post" name="FomeDemo">
<p><input type="search" name="SearchText" placeholder="请输入搜索关键词">
<p><input type="submit" value="进行搜索" name="FormSubmit">
</form>
</body>
</html>
<html>
<head>
```

3.2required 属性

元素介绍:

required 通过设置布尔值，可以防止输入框为空时提交表单。

案例:

示例 7-19: required 属性

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>required 属性</title>
</head>

<body>
<form action="form.php" method="post" name="FomeDemo">
<p><input type="search" name="SearchText" placeholder="请输入搜索关键词" required>
<p><input type="submit" value="进行搜索" name="FormSubmit">
</form>
</body>
</html>
```

3.3multiple 属性

元素介绍:

multiple 属性为布尔值，可以用在一些输入类型中（例如，email 或 file），从而支持在同一个输入框中输入多个值。在同一个输入框

中输入多个值时必须用半角状态的逗号分隔。

案例：

示例 7-20: multiple 属性

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>multiple 属性</title>
</head>

<body>
<form action="form.php" method="post" name="FomeDemo">
<p><input type="email" name="emailText" placeholder="请输入电子邮件地址" required multiple>
<p><input type="submit" value="确定" name="FormSubmit">
</form>
</body>
</html>
```

3.4 自动聚焦: autofocus 属性

元素介绍：

很多网站使用 JavaScript 脚本让 Web 表单的首个输入框自动获得焦点，例如 Google.com 首页会让搜索框自动获取焦点以便于直接输入搜索关键字。

autofocus 属性的作用就是自动的将焦点自动聚焦到特定的输入框中。如果浏览器不支持该属性，则会自动忽略该属性。对于此种情况，可以使用 JavaScript 脚本来实现。

案例：

示例 7-21: 电子邮件类型

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>autofocus 属性</title>
</head>

<body>
<form action="form.php" method="post" name="FomeDemo">
```

```
<p>
<input type="search" id="SearchText" placeholder="请输入搜索关键词"
size="50" required autofocus>
<script>
  if(!("autofocus" in document.createElement("input"))) {
    document.getElementById("SearchText").focus();
  }
</script>
<input type="submit" value="进行搜索" name="FormSubmit">
</form>
</body>
</html>
```

3.5 pattern 属性

元素介绍:

pattern 属性的作用是实现元素验证。它支持使用正则表达式定制验证规则。

案例:

示例 7-22: 邮政编码的验证

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>邮政编码的验证</title>
</head>

<body>
<form action="form.php" method="post" name="FormDemo">
<p>请输入邮编:
<input type="text" id="PostalCode" size="20" pattern="[0-9]{6}" title="请输入
六位数字的邮政编码。" required autofocus>
<script>
  if(!("autofocus" in document.createElement("input"))) {
    document.getElementById("PostalCode").focus();
  }
</script>
<input type="submit" value="确定" name="FormSubmit">
</form>
</body>
</html>
```

3.6 form 属性

元素介绍:

form 属性可用于在<form>元素之外声明表单元素。

案例:

示例 7-23: form 元素

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>邮政编码的验证</title>
</head>

<body>
<nav>
  <p><input type="search" name="SearchText" placeholder="请输入搜索关键词" form="FormDemo">
</nav>
<form action="Form.php" method="get" name="FormDemo">
<p>请输入邮编:
<input type="text" id="PostalCode" size="20" pattern="[0-9]{6}" title="请输入六位数字的邮政编码。" autofocus>
<script>
  if(!("autofocus" in document.createElement("input"))) {
    document.getElementById("PostalCode").focus();
  }
</script>
<input type="submit" value="确定">
</form>
</body>
</html>
```

四、新增的表单元素

HTML 5 除了将现有的元素通过增加属性的方式,极大的扩展了表单元素的丰富性外,还增加了一些新的元素。下面介绍几个较为实用的新元素。

4.1datalist

元素介绍:

datalist 元素是表单的特有元素。可以通过 list 属性预先创建一组列表项，后面可以作为输入框的输入提示。

案例：

示例 7-24: datalist 元素

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>datalist 元素</title>
</head>

<body>
<form>
<datalist id="TelephoneInfo">
  <option value="13526688704" label="Mobile Phone"></option>
  <option value="0371-65962530" label="Office Phone"></option>
  <option value="0371-63068118" label="House Phone"></option>
</datalist>
<p>请选择联系方式: <input type="tel" name="Telephone" id="Telephone"
list="TelephoneInfo">
<p><input type="submit" value="确定">
</form>
</body>
</html>
```

4.2 progress

元素介绍：

progress 元素是表单特有的元素，用来表示一个任务的完成进度。这个进度可以是不确定的，只是表示进度正在进行，但不清楚还有多少工作量没有完成。也可以用 0 到最大数字之间的数字来表示进度完成情况。

主要属性：

- (1) value: 表示完成了多少工作量。
- (2) max: 表示总共有多少工作量。

案例：

示例 7-25: progress 元素

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>progress 元素</title>
</head>

<body>
<h1>互联网应用技术研究所在研项目: </h1>
<p>天一影视:
<p><progress id="P-VOD" max="100" value="80"></progress>
<script>
    document.write(document.getElementById("P-VOD").value);
    document.write("%");
</script>
<p>智库网:
<p><progress id="P-Zhikoo" max="150" value="34"></progress>
<script>
    document.write(document.getElementById("P-Zhikoo").value);
    document.write("%");
</script>
</body>
</html>

```

4.3 meter

元素介绍:

meter 元素表示规定范围内的数量值。

主要属性:

(1) value: 在元素中表示出来的实际值。该属性值默认为 0, 可以给该属性制定一个浮点小数值。

(2) min: 指定规定范围时允许使用的最小值, 默认值为 0。

(3) max: 制定规定范围时允许使用的最大值, 默认值为 1。

(4) low: 规定范围的下限值。

(5) high: 规定范围的上限值。

(6) optimum: 最佳值, 该属性值必须位于 min 和 max 之间。

案例:

示例 7-26: meter 元素

```

<!doctype html>

```

```

<html>
<head>
<meta charset="utf-8">
<title>meter 元素</title>
</head>

<body>
<h1>计算机系统状态: </h1>
<p>CPU 使用率:
<p><meter id="CPU" value="5" min="0" max="100" low="2" high="85"
style="width:200px;"></meter>
<script>
    document.write(document.getElementById("CPU").value);
    document.write("%");
</script>
<p>内存使用率:
<p><meter id="Memory" value="868" min="0" max="2048" low="200"
high="1800" style="width:200px;"></meter>
<script>
    document.write(parseInt(document.getElementById("Memory").value/doc
ument.getElementById("Memory").getAttribute("max")*100));
    document.write("%");
</script>
</body>
</html>

```

4.4 output

元素介绍:

output 元素定义不同类型的输出，比如计算结果或者脚本的输出。output 元素必须从属于某个表单，也就是说，必须将它书写在表单内部，或者对它添加 form 属性。

此元素目前获得的浏览器支持比较少，可以确定的是 Opera 10 浏览器对此进行了支持。

案例:

示例 7-27: output 元素

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>output 元素</title>

```



```
</head>

<body>
<form>
请选择您的年龄：
<input type="range" name="YourAge" min="20" max="60" step="1">
<output onFormInput="value=YourAge.value">40</output>岁
</form>
</body>
</html>
```

五、表单验证

在 HTML 5 中，在增加了大量的表单元素与属性的同时，也增加了大量在提交时对表单与表单内新增元素进行内容有效性验证的功能。

5.1 自动验证

在 HTML 5 中，例如 URL、email 等 input 类型自身已经实现了表单验证。可以通过对元素使用属性的方法，实现在表单提交时执行自动验证的功能。

自动验证的属性主要有：required、pattern、min、max、step 属性。这些属性在前面的章节已经做了详细的介绍。

5.2 显式验证

除了对 input 元素通过属性进行自动验证外，在 HTML 5 中，form、input、select、textarea 元素都具有 checkValidity 方法，以显式的对表单内所有元素内容或单个元素内容进行有效性验证。

checkValidity 方法以布尔值的形式返回验证结果。

案例：

示例 7-28：显式验证

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>checkValidity 验证</title>
</head>
```

```

<body>
<script language="javascript">
function check()
{
  var userEmail = document.getElementById("UserEmail");
  if(email.value=="")
  {
    alert("电子邮件地址必须填写。");
    return false;
  }
  else if(!email.checkValidity())
    alert("请输入正确的电子邮件地址格式。");
  else
    alert("您输入的电子邮件地址格式是正确的。");
}
</script>
<form id="UserAdd" onsubmit="return check();">
  <label for="UserEmail">请输入电子邮件地址: </label>
  <input type="email" id="UserEmail" name="UserEmail" />
  <input type="submit" />
</form>
</body>
</html>

```

5.3 取消验证

有些时候，可能需要对表单进行提交，但是需要临时取消对表单的验证，例如开发测试的时候，或者表单很大，用户填写完第一部分后可以先保存下来，等待第二部分填写完毕后一并提交时。

在 HTML 5 中，有两种方式来临时取消对表单的验证。

方法一：利用 form 元素的 novalidate 属性，取消整个表单内的元素的验证。

方法二：利用 input 元素或 submit 元素的 formnovalidate 属性，取消单个表单元素的验证。

通过取消验证的方式，可以实现“假提交”和“真提交”的效果。一个按钮不对表单进行任何验证，将数据保存到一个临时数据文件中，另外一个按钮对表单进行验证，将数据保存到数据库。

案例：

示例 7-29-1：取消整个表单的验证

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>取消验证</title>
</head>

<body>
<script language="javascript">
function check()
{
    var UserEmail = document.getElementById("UserEmail");
    if(email.value=="")
    {
        alert("电子邮件地址必须填写。");
        return false;
    }
    else if(!email.checkValidity())
        alert("请输入正确的电子邮件地址格式。");
    else
        alert("您输入的电子邮件地址格式是正确的。");
}
</script>
<form id="UserAdd" onsubmit="return check();" novalidate>
    <label for="UserEmail">请输入电子邮件地址： </label>
    <input type="email" id="UserEmail" name="UserEmail" />
    <input type="submit" />
</form>
</body>
</html>
```

示例 7-29-2：取消表单内单个元素的验证

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>取消验证</title>
</head>

<body>
<script language="javascript">
function check()
{
```

```
var userEmail = document.getElementById("UserEmail");
if(email.value=="")
{
    alert("电子邮件地址必须填写。");
    return false;
}
else if(!email.checkValidity())
    alert("请输入正确的电子邮件地址格式。");
else
    alert("您输入的电子邮件地址格式是正确的。");
}
</script>
<form id="UserAdd" onsubmit="return check();">
    <label for="UserEmail">请输入电子邮件地址: </label>
    <input type="email" id="UserEmail" name="UserEmail" formnovalidate
/>
    <input type="submit" />
</form>
</body>
</html>
```

六、讨论与思考

6.1 表单效果应该选择 HTML 5 还是 JavaScript?

(1) 在 HTML 5 中增强的元素和属性可以实现占位字符等效果，而这些效果目前都是通过 JavaScript 来实现的，那么那种方式的性能更好呢？

(2) 考虑到不支持 HTML 5 的浏览器，在通过 HTML 5 来实现表单效果时，应该如何做到让所有浏览器都实现效果？

6.2 表单验证可靠么？

(1) HTML 5 中提供的表单验证方式，是不是可靠的？

(2) 如何做到可靠的表单验证？