

### 本章学习目标

- 了解 HTML5 新增标签的作用
- 掌握使用 audio 和 video 标签的方法

本章介绍几个 HTML5 的新增标签,体会 HTML5 新增标签给网页设计带来的便利。

## 5.1 输入类型 number

随着浏览器的发展,结合以往网页开发时的需求,HTML5 引入了新的标签,这会降低我们开发网页的难度。查看网页 [https://www.w3school.com.cn/html/html\\_form\\_input\\_types.asp](https://www.w3school.com.cn/html/html_form_input_types.asp),可以看到 HTML5 引入的新的输入类型,这里仅简单介绍几个。这些输入类型,有的相当于网页中 JavaScript 的作用,对输入数据按要求进行了验证,但不需要用户自己书写 JavaScript 了。

输入类型 number 用于应该包含数字值的输入字段,能够对数字做出限制。在提交表单时,浏览器会自动验证 number 域的值。输入类型 number 适用的浏览器如表 5.1 所示。只有高版本的浏览器才支持 HTML5 新引入的标签。

表 5.1 输入类型 number 适用的浏览器

IE	Firefox	Opera	Chrome
NO	4.0	9.0	10.0



**【例 5-1】** 输入类型 number 示例。

ch5\_01.html 内容如下:

```

01 <!DOCTYPE html >
02 <html >
03 <head >
04 <meta charset = "utf - 8"/>
05 <title>输入类型 number</title>
06 </head>
07 <body >
08 <form action = "https://www.w3school.com.cn/example/html5/demo_form.asp">
09 数量(1 到 5 之间):

```

```
10 <input type = "number" name = "quantity" min = "1" max = "5">  
11 <input type = "submit">  
12 </form >  
13 </body >  
14 </html >
```

运行上述代码后出现如图 5.1 所示界面。



图 5.1 number 输入类型验证

(1) 输入一个范围 1~5 的数字 3,提交可以得到如图 5.2 所示的界面。



图 5.2 正确输入数据时的响应

(2) 输入非数字时,例如 tiankong,提交可以得到如图 5.3 所示的界面。



图 5.3 输入字母时的验证

(3) 修改 form 的 action="#",输入一个不是 1~5 的数字 6,提交可以得到如图 5.4 所示的界面,说明验证是由浏览器完成的。



图 5.4 输入 6 时的验证



## 5.2 输入类型 email

email 类型用于应该包含 e-mail 地址的输入域。在提交表单时,浏览器会自动验证 email 域的值,验证的格式为带有@的字符串。用法参见例 5-2。

**【例 5-2】** 输入类型 email 示例。

ch5\_02.html 内容如下:

```
01 <!DOCTYPE html >
02 <html >
03   <head >
04     <meta charset = "utf - 8" />
05     <title>输入类型 email</title>
06   </head >
07   <body >
08     <form action = "# ">
09       E - mail: <input type = "email" name = "user_email" />
10       <input type = "submit">
11     </form >
12   </body >
13 </html >
```

输入一个 email 值 58700865. qq. com, 单击提交后响应如图 5.5 所示, 提示我们输入的字符串不是电子邮件地址的格式, 因为不含有@符号。



图 5.5 输入类型 email 格式不对时的验证

## 5.3 输入类型 url

url 类型用于应该包含 URL 地址的输入域。在提交表单时,浏览器会自动验证 url 域的值,输入类型要符合 url 格式。

用法参见例 5-3。

**【例 5-3】** 输入类型 url 示例。

ch5\_03.html 内容如下:

```
01 <!DOCTYPE html >
02 <html >
03   <head >
```



```
04     <meta charset = "utf - 8" />
05     <title>输入类型 url</title>
06 </head>
07 <body>
08     <form action = "#">
09     Homepage: <input type = "url" name = "user_url" /><br />
10     <input type = "submit">
11     </form>
12 </body>
13 </html>
```

运行上述代码后出现如图 5.6 所示界面,因未带有协议名,不符合 url 格式,故有提示。



图 5.6 输入类型 url 格式不对时的验证

## 5.4 输入类型 date

输入类型 date 用于应该包含日期的输入字段。若浏览器支持,日期选择器会出现在输入字段中。借助于图形输入可以规范日期输入格式,确保日期符合要求。

用法参见例 5-4。

**【例 5-4】** 输入类型 date 示例。

ch5\_04.html 内容如下:

```
01 <!DOCTYPE html>
02 <html>
03     <head>
04         <meta charset = "utf - 8" />
05         <title>输入类型 date</title>
06     </head>
07     <body>
08         <form action = "#">
09         生日<input type = "date" name = "bdaytime">
10         <input type = "submit">
11         </form>
12     </body>
13 </html>
```

运行上述代码后出现如图 5.7 所示界面。在我们输入日期时,会出现日期提示选择窗口,能保证我们的输入符合日期格式的要求,例如月份不能大于 12。注意在不同的浏览器里显示可能会略有不同。





图 5.7 输入类型 date 验证

## 5.5 表单属性 autocomplete

表单属性 autocomplete 适用于 < form > 标签和以下类型的 < input > 标签: text、search、url、telephone、email、password、datepickers、range 以及 color。当用户在自动完成域中开始输入时,浏览器应该在该域中显示填写的选项。

编码示例如下。

**【例 5-5】** 表单属性 autocomplete 示例。

ch5\_05.html 内容如下:

```
01 <!DOCTYPE html >
02 <html >
03   <head >
04     <meta charset = "utf - 8" />
05     <title>表单属性 autocomplete</title>
06   </head >
07   <body >
08     <form action = "# " method = "get" autocomplete = "on">
09       First name:< input type = "text" name = "fname" /><br />
10       Last name: < input type = "text" name = "lname" /><br />
11       E - mail: < input type = "email" name = "email" autocomplete = "off" /><br />
12     </form >
13   </body >
14 </html >
```

运行上述代码,首次输入测试信息后,出现如图 5.8 所示的界面。

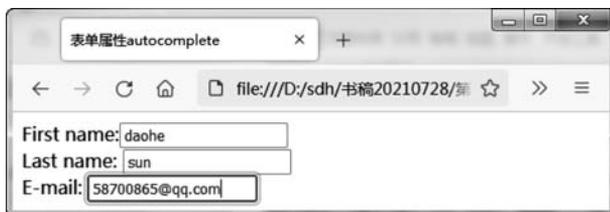


图 5.8 表单属性 autocomplete 的验证



按 F5 键刷新页面后,出现如图 5.9 所示界面。因 form 表单设置了 `autocomplete="on"`,浏览器应该在该域中显示填写的选项,所以第一个、第二个文本框会留下上次输入的结果。但 email 输入设置了 `autocomplete="off"`,所以不会留下上次输入。

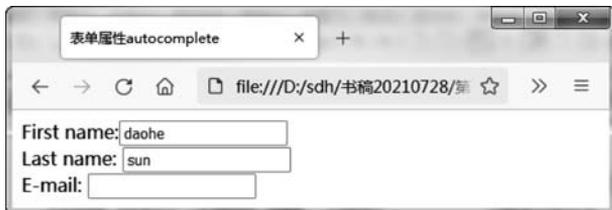


图 5.9 自动显示填写的选项

## 5.6 音频 audio

直到现在,仍然不存在一项关于在网页上播放音频的标准。今天,大多数音频是通过插件(例如 Flash)来播放的。然而,并非所有浏览器都拥有同样的插件。HTML5 规定了一种通过 `audio` 元素来包含音频的标准方法。`audio` 元素能够播放声音文件或者音频流。当前,`audio` 元素支持 Ogg Vorbis、MP3、WAV 三种音频格式。大多数高版本的浏览器都支持这三种音频格式。`<audio>` 标签用于定义声音,例如音乐或其他音频流。`<audio>` 标签是 HTML 5 的新标签,其常用属性如表 5.2 所示。

表 5.2 `<audio>` 标签的常用属性

属性	值	描述
<code>autoplay</code>	<code>autoplay</code>	音频在就绪后马上播放
<code>controls</code>	<code>controls</code>	向用户显示控件,例如播放按钮
<code>loop</code>	<code>loop</code>	每当音频结束时重新开始播放
<code>preload</code>	<code>preload</code>	音频在页面加载时进行加载,并预备播放。如果使用 <code>autoplay</code> ,则忽略该属性
<code>src</code>	<code>url</code>	要播放的音频的 URL

### 【例 5-6】 `audio` 标签示例。

`ch5_06.html` 内容如下:

```
01 <!DOCTYPE html >
02 <html >
03   <head >
04     <meta charset = "utf - 8">
05     <title> audio 标签</title>
06   </head >
07   <body >
08     <audio src = ". /i/test.mp3" controls = "controls">
09     <audio src = ". /i/test.m4a" controls = "controls">
10       Your browser does not support the audio element.
11   </audio >
```



```
12 </body>  
13 </html>
```

运行上述代码后出现如图 5.10 所示的界面。



图 5.10 audio 标签示例

audio 元素允许多个 source 元素。source 元素可以链接不同的音频文件,浏览器将使用第一个可识别的格式。例 5-6 使用第一个 mp3 文件,适用于 Firefox、Opera 以及 Chrome 浏览器。当素材音频文件不符合要求时,可以使用格式转换器进行转换,如图 5.11 所示,可以把手机录音的 m4a 文件转换为 mp3 文件。



图 5.11 mp3 音频转换

## 5.7 视频 video

今天,大多数视频是通过插件(例如 Flash)来显示的。然而,并非所有浏览器都拥有同样的插件。HTML5 规定了一种通过 video 元素来包含视频的标准方法。当前,video 元素支持 Ogg、MPEG 4、WebM 三种视频格式。

Ogg: 带有 Theora 视频编码和 Vorbis 音频编码的视频文件。

MPEG4: 带有 H.264 视频编码和 AAC 音频编码的视频文件。

WebM: 带有 VP8 视频编码和 Vorbis 音频编码的视频文件。

<video>标签用于定义视频,例如电影片段或其他视频流。<video>标签是 HTML5 的新标签,其常用属性如表 5.3 所示。

表 5.3 &lt;video&gt; 标签的常用属性

属性	值	描述
autoplay	autoplay	如果出现该属性,则视频在就绪后马上播放
controls	controls	如果出现该属性,则向用户显示控件,例如播放按钮
height	pixels	设置视频播放器的高度
loop	loop	如果出现该属性,则当媒介文件完成播放后再次开始播放
preload	preload	如果出现该属性,则视频在页面加载时进行加载,并预备播放。如果使用 autoplay,则忽略该属性
src	url	要播放的视频的 URL
width	pixels	设置视频播放器的宽度

HTML5 <video>元素同样拥有方法、属性和事件,其中方法用于播放、暂停以及加载等,属性(例如时长、音量等)可以被读取或设置。

**【例 5-7】** video 标签示例。

ch5\_07.html 内容如下:

```
01 <!DOCTYPE html >
02 <html >
03   <head >
04     <meta http-equiv = "Content-Type" content = "text/html; charset = utf-8" />
05     <title> video 标签使用</title>
06   </head >
07   <body >
08     <video width = "320" height = "240" controls = "controls">
09       <source src = "IPAddress.wmv" type = "video/wav">
10       <source src = "IPAddress_H264.mp4" type = "video/mp4">
11       Your browser does not support the video tag.
12     </video >
13   </body >
14 </html >
```

controls 属性供添加播放、暂停和音量控件,包含宽度和高度属性。video 元素允许多个 source 元素,source 元素可以链接不同的视频文件,浏览器将使用第一个可识别的格式。例 5-7 中 IPAddress.wmv 不被识别,所以会使用 IPAddress\_H264.mp4。ch5\_07.html 的运行效果如图 5.12 所示。



图 5.12 video 运行示例



注意这里需要是 H. 264 视频编码和 AAC 音频编码的 MPEG4 文件,如果不是则无法播放。示例文件夹给出了一个不是 H. 264 视频编码的 IPAddress. mp4,可以尝试去播放它观察效果。当素材视频文件不符合要求时,可以使用格式转换器进行转换,如图 5.13 所示。单击“高级”,会出现 MP4 编码格式选择界面,如图 5.14 所示。可以把手机录制的 MP4 文件转换为符合要求的格式。



图 5.13 转换 mp4



图 5.14 mp4 格式选择

这里仅介绍上述几个 HTML5 标签,目前 HTML5 在实际系统使用中还需要考虑兼容性的问题。随着时间推移,浏览器会支持越来越多的 HTML5 标签,开发网页也会容易些。

## 习题 5

### 一、问答题

1. 列举几个 HTML5 的新增标签。
2. 简述 HTML5 的新特性。

### 二、操作题

1. 使用< audio >标签在网页上实现音频播放。
2. 使用< video >标签在网页上实现视频播放。