

HTML5多媒体与嵌入

多媒体是我们可以看到和听到的一切,如文本、图片、音乐、声音、动画和视频等。大多数多媒体存储在 媒体文件中,以独立的文件形式存在,一般通过文件的扩展名区分不同类型的多媒体,HTML5可以通过多 种方式使用多媒体。本章首先重点介绍图像元素和响应式图像,然后介绍音视频文件的格式及如何在 HTML5 中使用,接下来介绍嵌入元素,最后介绍在"叮叮书店"项目首页添加超链接和多媒体元素的操作 过程。

本章要点

- 图像
- HTML5 音频/视频

- 响应式图像
- 嵌入元素

5.1 图像

HTML5 经常使用位图和矢量图两种类型的图像。

1. 位图

位图由排列在网格中的点(即像素)组成。图像是由网格中每个像素的位置和颜色值决定的,每个像素 被指定一种颜色。

编辑位图图像时,修改的是像素,而不是线条和曲线。位图图像与分辨率有关,这意味着描述图像的数 据被固定到一个特定大小的网格中。放大位图图像将使这些像素在网格中重新分布,这会使图像的边缘呈 锯齿状。图 5.1 中自行车位图放大后,可以看出明显的栅格化。

位图图像的主要格式有 GIF、JPEG 和 PNG。

2. 矢量图

矢量图使用包含颜色和位置信息的直线和曲线(矢量)呈现图像。如图 5.2 所示的一辆自行车的图像, 可以使用一系列描述车子轮廓的路径来定义。车子的颜色由其轮廓(即笔触)的颜色和该轮廓所包围区域 (即填充)的颜色决定。





矢量图与分辨率无关,这意味着当更改矢量图的颜色、移动矢量图、调整矢量图的大小、更改矢量图的形状时,其外观品质不会发生变化。图 5.2 中自行车的轮廓放大了 6 倍,但图像依然清晰。

矢量图的主要格式有 CDR、AI、WMF、EPS 和 SVG。

表 5.1 列出了 HTML5 可以使用的图像标签。

标签	描述
$	定义图像
< picture >	通过包含零个或多个< source >元素和一个< img >元素为不同的显示设备提供图像。浏览器会选择最匹配的< source >元素,否则选择< img >元素 src 属性中的 URL
< source >	定义媒体源
< map >	定义带有可单击区域的图像映射
<area/>	定义图像地图中的可单击区域
< figure >	定义媒体内容的分组,以及它们的标题
< figcaption >	定义 <figure>元素的标题</figure>

表 5.1 图像标签

5.1.1 < img >标签

图像由标签定义,是空元素。Web标准并没有给出必须支持的图像格式,一般浏览器都 支持 JPEG、GIF、PNG、BMP、ICO和 SVG等格式。表 5.2列出了标签常用属性。

属 性	值	描述
alt	text	必需,图像的替换文本
src	URL	必需,图像的 URL
height	pixels 或 %	可选,图像的高度
width	pixels 或 %	可选,图像的宽度
srcset	text	定义允许浏览器选择的图像集
sizes	text	定义一组媒体条件,当媒体条件为真时,根据图像预期布局宽度的值从 srcset 图像集选择一个合适的图像显示
usemap	URL	将图像定义为客户器端图像映射

表 5.2 < img>标签常用属性

1. 显示位图

要在页面上显示图像,必须使用 src 属性声明图像的 URL 地址,格式如下。

< img src = "url">

其中, URL 指图像文件的位置, 浏览器将图像显示在文档中图像标签出现的地方。

当浏览器不能显示图像时(如无法载入图像或浏览器禁止图像显示),可以在显示图像的位置上显示 alt 属性定义的文本。为页面上的每幅图像加上替换文本属性有利于更好地显示信息。例如:

< img >元素的内容和大小由外部资源的图像文件所决定,而不是元素自身,这样的元素称为替换元素。 由于搜索引擎需要读取图像的文件名并纳入 SEO,所以应该给图像文件定义一个描述性的文件名。

网络上大多数图片是有版权的,在未得到授权之前不要把 src 属性指向其他网站上的图像。

由于图像是独立文件存在的,如某个 HTML 文件包含 10 幅图像,要正确显示这个页面,需要加载 10 个 图像文件和一个 HTML 文件,HTTP 需要 11 次请求才能完成,加载图片是需要时间的,所以要合理地在文 档内容中加入图像,如果过度使用图像,用户在浏览该页面时,会增加很多不必要的等待时间。

【例 5.1】 img. html 说明了< img >标签的用法, 如图 5.3 所示。源码如下。

```
< head >
< title > img 标签</title >
</head >
< body >
```



视频讲解

HTML5+CSS3+ES6前端开发项目实战(微课视频版)

```
< img src = "images/w3c_home_nb.png" alt = "">< br >
```

< span > 鼠标指针指向图像, 会显示 title 属性值。</ span > < br >

< br >

```
<span>如果无法显示图像,将显示 alt 属性值。</span>< br>
```

```
< img src = "images/noabout - bookstore. jpg" alt = "叮叮书店">
```

</body>

提示: < img>标签的 alt 属性不能省略,否则在 https://validator. w3. org/检验时会提示错误。

2. 添加矢量图

HTML5 支持 SVG 矢量图,可缩放矢量图形(Scalable Vector Graphics, SVG)是一种用于描述二维矢量 图形的基于 XML 的标记语言。SVG 于 1999 年推出,2003 年成为 W3C 推荐标准,2011 年 SVG 1.1 成为推 荐标准的第2个版本。SVG 2.0 正在制定当中。

SVG 用于标记图形,而不是内容,可以使用一些基本元素创建简单图形,如< circle >和< rect >,也可以使 用更高级的功能元素,如<feColorMatrix>(使用变换矩阵转换颜色)和< animate >(矢量图形动画)等。

HTML5 可以直接嵌入 SVG 标记,称为内联 SVG, Internet Explorer 9、Firefox、Opera、Chrome 和 Safari 都支持内联 SVG。

同样可以使用标签显示 SVG 图像。

【例 5.2】 SVG. html 使用了内联 SVG 方式和< img>标签显示矢量图, 如图 5.4 所示。源码如下。

```
扫一扫
             < head >
                 <title>SVG 矢量图</title>
             </head>
             < body >
                 < svq xmlns = "http://www.w3.org/2000/svq" version = "1.1" baseProfile = "full" width = "200" height =
             "100">
                     <rect width = "100 % " height = "100 % " fill = "red" />
                     <circle cx = "100" cy = "51" r = "40" fill = "green" />
                      <text x = "100" y = "65" font - size = "36" text - anchor = "middle" fill = "white"> SVG </text >
                 </sva>
                 < img src = "images/svg.svg" alt = "SVG">
             </body>
                                                     X
           S img标签
                                 ×
                                    +
           ← → C ① 127.0.0.1:5500/chapter5/img.... ☆ ★ A
                                                                                                              SVG 失量图
                                                                                              +
                                                                                          ×
          鼠标指针指向图像,会显示title属性值。
```



图 5.3 img. html 页面显示

如果无法显示图像,将显示alt属性值。

同叮叮书店



内联 SVG 要使用< svg>标签, xmlns 属性定义 SVG 命名空间, width 和 height 属性设置 SVG 画布的宽 度和高度。version 属性定义 SVG 版本, baseProfile 属性说明正确渲染内容所需最小的 SVG 语言配置, 值为 full 表示适用于个人计算机。version 和 baseProfile 属性是必不可少的。

如果可能的话,使用 SVG 替代 JPEG、GIF 或 PNG,这样能够轻松解决多屏幕分辨率的问题,而且也比位 图图像小得多。可以通过网站 https://www.vectorizer.io/在线把位图转换为矢量图。

卯 栃 讲 解

68

5.1.2 < map >标签和< area >标签

<map>和<area>标签用于创建图像地图,图像地图是指已被分为多个区域(图像的一部分)的图像,这 些区域称为热点,可以创建多个热点,热点支持超链接。

<map>元素必须使用 name 属性定义 image-map 名称, name 属性与标签的 usemap 属性相关联, 创建图像与映射之间的关系。

<map>元素包含<area>元素,定义图像映射中的可单击区域。表 5.3列出了<area>标签常用属性。

属 性	值	描述 送
coords	x1,y1,x2,y2 x,y,radius x1, y1, x2, y2,, xn,yn	定义可单击区域坐标。coords 属性与 shape 属性配合使用,规定区域的尺寸、形状和位置。图像左上角的坐标是"0,0" x1,y1,x2,y2:如果 shape 属性为"rect",该值规定矩形左上角和右下角的坐标 x,y,radius:如果 shape 属性为"circ",该值规定圆心的坐标和半径 x1,y1,x2,y2,,xn,yn:如果 shape 属性为"poly",该值规定多边形各边的坐标。 如果第1个坐标和最后一个坐标不一致,那么为了关闭多边形,浏览器必须添加最 后一对坐标
href	URL	定义此区域的目标 URL
nohref	nohref	规定该区域没有相关的链接
shape	default rect,circle,poly	定义区域的形状。default 规定全部区域, rect 定义矩形区域, circle 定义圆形, poly 定义多边形区域
target	_blank _self	规定在何处打开链接文档。其中,_blank为在新窗口中打开被链接文档;_self为 默认,在相同的窗口中打开被链接文档

表 5.3 < area >标签常用属性

例 5.2 中 map. html 说明了< map >和< area >标签的用法, 如图 5.5 所示。源码如下。



图 5.5 map. html 页面显示

当鼠标指针指向图像中的"权威指南"4个字和熊的头部时,鼠标指针变成手状,并且在浏览器窗口的状态栏中显示链接的地址。

5.1.3 < figure >标签和< figcaption >标签

如果需要为图像搭配说明文字,可以使用< figure >标签,< figure >标签规定独立的内容,如图像;< figcaption >标签配合使用,定义< figure >标签的标题,说明< figure >标签的内容。

【例 5.3】 figure. html 说明了< figure >标签的用法,如图 5.6 所示。源码如下。

```
扫一扫

</
```



图 5.6 figure. html 页面显示

5.2 响应式图像

在网页上显示图像,最理想的是当访问网站时根据不同的设备提供不同的分辨率图像或不同尺寸的图像,响应式图像技术可以通过让浏览器提供多个图像文件解决这个问题,如使用相同的图像但包含多个不同的分辨率,或者使用相同图像的裁剪和不同的图像以适应不同的空间分配。

5.2.1 像素与设备像素比

像素是图像分辨率的单位,分辨率又称为解析度,图像分辨率是图像清晰度或浓度的度量标准,通常以 横向和纵向点的数量——像素来表示。像素可以看作组成图像的小方格,这些小方格都有一个明确的位置 和被分配的色彩数值,小方格颜色和位置决定了图像呈现出来的样子。通常情况下,图像的分辨率越高,所 包含的像素就越多,图像就越清晰。

像素可分为设备像素和设备独立像素。

1. 设备像素(物理像素)

显示器分辨率是指计算机显示器本身的物理分辨率,对液晶显示器(Liquid Crystal Display,LCD)来说, 是指显示屏上的像素点数量,这些像素称为物理像素,也叫设备像素(Device Pixels),设备像素已经在生产制 造时固定,不能改变。

2. 设备独立像素(逻辑像素)

设备独立像素(Device Independent Pixels)是操作系统定义的一种像素单位,也叫逻辑像素或设备无关

像素,允许应用程序以设备独立像素为单位进行测量,然后系统将应用程序的设备独立像素测量值转换为适 合特定设备的物理像素。

在 Chrome 浏览器开发者工具的控制台中输入 screen. width 和 screen. height 得到的数值就是整个屏幕 设备独立像素的宽度和高度,这个值不会随页面内容的缩放或浏览器窗口大小而改变。

提示:可以通过操作系统的分辨率设置改变设备独立像素的大小。

在 Web 开发中一般使用 CSS 像素(CSS Pixel),即 CSS 样式代码中使用的逻辑像素,可以看作设备独立像素,CSS 像素是一个相对单位,相对的是设备像素。设备像素和 CSS 像素一般使用 px 作为单位,在浏览器 缩放比例为 100%的情况下,一个 CSS 像素大小等于一个设备像素。

在不同的设备或不同的环境中,CSS中的1px所代表的设备像素长度是不同的,也就是说CSS中的1px 并不总是代表设备像素的1px,特别是在移动设备上。

3. 设备像素比

设备像素比(Device Pixel Ratio, DPR)是设备像素和设备独立像素的比例,即设备像素比=物理像素/逻辑像素,表示设备独立像素和设备像素的转换关系。

在 Chrome 浏览器开发者工具的控制台中输入 window. devicePixelRatio 可以得到这个值。

CSS 像素也是设备独立像素,所以通过 devicePixelRatio 的值可以知道该设备上一个 CSS 像素代表多少 个物理像素。当设备像素比为1时,使用1(1×1,横向和纵向)个设备像素显示1个 CSS 像素;当设备像素 比为2时,使用4(2×2,横向和纵向)个设备像素显示1个 CSS 像素;当设备像素比为3时,使用9(3×3,横 向和纵向)个设备像素显示1个 CSS 像素。

5.2.2 < img>标签的 srcset 和 sizes 属性

标签提供了 srcset 和 sizes 两个新属性,可以根据设备的宽度显示不同的分辨率图像或让高分辨率设备显示高质量图像。例如:

< img

srcset = "images/320.jpg 320w,images/640.jpg 640w,images/800.jpg 800w"
sizes = "(max - width:320px) 320px,(max - width:640px) 640px,800px"
src = "images/800.jpg" alt = "新措">

srcset 属性定义了允许浏览器选择的图像集,属性值由逗号分隔的列表组成,列表项由图像 URL 与图像 宽度或像素密度描述符组成。其中,图像宽度是一个正整数,单位使用 w(像素),表示图像分辨率的真实大 小;像素密度是一个正浮点数,单位使用 x,表示设备像素比,1x 是默认值。图像 URL 与图像宽度或像素密 度用空格分隔,图像宽度和像素密度不能同时使用。

例如,images/320.jpg 320w 列表项中,images/320.jpg 表示图像的 URL,320w 表示这个图像实际的宽 度为 320 像素。

sizes 属性定义了一组媒体条件和资源大小,属性值由逗号分隔的列表组成,列表项由媒体条件和资源大小组成,用空格分隔。媒体条件的设定可参照 10.3 节中的 CSS 媒体查询。资源大小是指图像预期布局的宽度,也就是显示指定图像的最大宽度,可能影响指定图像显示的大小,使用 CSS 长度单位 px。当媒体条件为 真时,浏览器根据资源大小的值从 srcset 图像集选择一个合适的图像显示。如果没有设置 srcset 属性,那么 sizes 属性不起作用。

例如,(max-width:320px) 320px 列表项中,(max-width:320px)表示媒体条件,意思是当可视窗口的宽度是 320 像素或更少时条件为真; 320px 是资源大小,表示当媒体条件为真时显示指定图像填充的最大宽度。

浏览器从 srcset 图像集选择图像的原则是在实际尺寸大于或等于资源大小值的图像中,选择最接近的 那一幅图像,当浏览器成功匹配第1个媒体条件时,剩下的都会被忽略,所以媒体条件的列表顺序最好是根 据具体条件从小到大排列(如 max-width)或从大到小排列(如 min-width)。

sizes 属性列表中最后一个资源大小默认是没有媒体条件的,当没有任何一个媒体条件为真时,它会起作用。

71

扫一扫

初频讲解

【例 5.4】 img_srcset. html 使用< img >标签 srcset 属性根据设备像素比在相同的 CSS 像素宽度下,显示不同的分辨率图像,实现让高分辨率设备显示高质量图像。源码如下。

```
< head>
    <title> img标签的 srcset 属性</title>
    <title> img标签的 srcset 属性</title>
    <ti>< style>
        * {padding: 0;margin: 0;}/* 去除页边距 */
        img {width: 320px;}
    </style>
</head>
</body>
    <ti>< div>
        <img srcset = "images/320.jpg 1x, images/640.jpg 2x, images/800.jpg 3x" alt = "新措">
        <img srcset = "images/320.jpg 1x, images/640.jpg 1.5x, images/800.jpg 2x" alt = "新措">
        </div>
</body>
</body>
```

在本例中,img{width:320px;}这个 CSS 样式会应用在图片上,图像的宽度在屏幕上是 320 像素(CSS 像素)。在 Firefox 浏览器 84 以上版本中直接打开浏览,按 Ctrl+Shift+M 组合键,进入 Web 开发者响应式设 计模式,改变设备的像素比,可以看到,当访问页面的设备具有标准和低分辨率时,一个设备像素表示一个 CSS 像素,images/320.jpg 会被加载(1x)。如果设备有高分辨率,两个或更多的设备像素表示一个 CSS 像 素,images/640.jpg(2x)和 images/800.jpg(3x)会被加载,如图 5.7 所示。



图 5.7 img_srcset. html 页面显示

【例 5.5】 img_srcset_sizes. html 使用< img >标签 srcset 和 sizes 属性根据不同的分辨率切换显示不同 尺寸的同一图像,实现根据设备的宽度显示不同的分辨率图像。源码如下。

在 Firefox 浏览器 84 以上版本中直接打开浏览,按 Ctrl+Shift+M 组合键,进入 Web 开发者响应式设计 模式,可以看到在设备的像素比始终为1的情况下,可视窗口的宽度为 320 像素时,显示 images/320.jpg 图 像,可视窗口的宽度设为 640 像素时,显示 images/640.jpg 图像,如图 5.8 所示。图中下边的图像看起来比 上边的图像小是由于图像预期布局的宽度设置得比较小((max-width:320px) 280px)。



图 5.8 img_srcset_sizes. html 页面显示

提示:目前 Chrome 浏览器对标签 srcset 和 sizes 属性支持不完整。

5.2.3 < picture >标签

< picture >标签通过包含零个或多个< source >标签和一个< img >标签为不同的显示设备或场景提供不同的图像。浏览器会选择最匹配的子< source >标签,如果没有,就选择< img >标签 src 属性中的 URL。

< source >标签为< picture >、< audio >和< video >标签指定多个媒体源,是一个空元素,用于为不同浏览器支持的多种格式提供相同的媒体内容,以便与多种浏览器兼容。表 5.4 列出了< source >标签常用属性。

属性	值	描述述
media	text	媒体条件。如果媒体条件匹配结果为 false,那么< source >元素会被跳过,仅在 < picture >元素中使用
srcset	text	定义允许浏览器选择的图像集,当< source >元素是< picture >或< video >元素的直接子 元素时,该属性才有效
type	MIME	为 srcset 属性指向的资源指定 MIME 类型。如果浏览器不支持指定类型,那么 < source >元素会被跳过

表 5.4 < source >标签常用属性

media 属性媒体条件的设定可参照 10.3 节中的 CSS 媒体查询。

```
【例 5.6】 picture.html使用<picture>和< source>标签在不同的分辨率时加载裁剪的不同图像。源码
如下。
< head>
< title>picture标签</title>
</head>
< body>
< picture>
< source media = "(max - width:320px)" srcset = "images/320 - 1. jpg">
< source media = "(max - width:320px)" srcset = "images/640 - 1. jpg">
< img src = "images/800. jpg" alt = "新措">
</picture>
```

</body>

在 Chrome 浏览器中按 F12 键进入开发者工具界面,按 Ctrl+Shift+M 组合键打开设备工具栏,将 Responsive 下拉菜单中的宽度分别设置为 320、640、800。可以看到,在不同的分辨率下,加载不同的图像,如 图 5.9 所示。



图 5.9 picture. html 页面显示

响应式图像的优势如下。

(1) 可以加载适当大小的图像文件,让带宽得到充分利用。

- (2) 可以加载不同裁剪并具有不同宽高比的图像,以适应不同宽度布局的变化。
- (3) 可以加载更高像素密度的图像,显示更清晰。

5.3 音视频

传统的 Web 技术不能在 Web 中嵌入音频和视频,HTML5 标准的提出,可以直接在页面中插入音视频。 HTML5 音频/视频标签如表 5.5 所示。

表 5.5 音频/视频标签

标签	描 述
< video >	定义视频
< track >	定义用在媒体播放器中的文本轨道
< audio >	定义音频

5.3.1 < video >标签

< video >标签定义视频,如电影片段或其他视频流。使用< video >标签播放视频时不需要任何插件,只要浏览器支持 HTML5 就可以。表 5.6 列出了< video >标签常用属性。

74

属性	值	描述	
autoplay	autoplay	视频就绪后自动播放	
controls	controls	显示视频播放器控件,如"播放"按钮	
height	pixels	视频播放器的高度	
loop	loop	循环播放	
width	pixels	视频播放器的宽度	
poster	URL	定义视频下载时显示的图像,或用户单击"播放"按钮前显示的图像	
preload	auto metadata none	定义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 auto:(默认值)表示预加载全部的音频/视频 metadata:仅加载音频/视频的元数据 none:不加载音频/视频	
src	URL	播放视频的 URL	
muted	muted	静音	

表 5.6 < video >标签常用属性

< video >标签只要有 src 属性就可以使用,如

<video src = "multimedia/Wildlife.mp4"></video>

对于不支持<video>标签的浏览器,可以在元素内容中添加替换文字,如

< video src = "multimedia/Wildlife.webm" controls = "controls" autoplay = "autoplay">您的浏览器不支持 video 元素</video >

1. 视频格式

由于版权的原因,目前<video>标签支持3种视频编码。

(1) Ogg: 带有 Theora 视频编码和 Vorbis 音频编码的 Ogg(.ogv)文件。

(2) MPEG4: 带有 H. 264 视频编码和 AAC 音频编码的 MPEG4(. m4v、. mp4)文件。

(3) WebM: 带有 VP8 视频编码和 Vorbis 音频编码的 WebM(. webm)文件。

提示:不同浏览器和移动设备系统对视频编码格式支持的情况不完全一样。

2. < source >标签

为了解决浏览器对视频格式的兼容情况,可以使用< source >标签为同一个媒体数据指定多个播放格式 与编码方式,确保浏览器可以从中选择一种自己支持的视频格式进行播放。也就是将 src 属性从< video >标 签中移除,将它放在几个单独的< source >标签中。例如:

```
< video controls = "controls" autoplay = "autoplay">

< source src = "multimedia/Wildlife.ogv" type = "video/ogg">

< source src = "multimedia/Wildlife.webm" type = "video/webm">

< source src = "multimedia/Wildlife.mp4" type = "video/mp4">

您的浏览器不支持 video 元素。
```

每个< source >标签都有一个 type 属性,包含了视频文件的 MIME 类型,这个属性是可选的,但是建议添加这个属性,浏览器会通过检查这个属性迅速地跳过那些不支持的格式。如果没有 type 属性,浏览器选择自上而下,尝试加载每个文件,直到找到所支持的格式为止,这样会消耗掉大量的时间和资源。

不同格式视频文件的转换,可以在网络搜索一些免费工具软件进行,如 Free Video Converter,可从 http://www.freemake.com/free_video_converter/下载,该软件支持 AVI、MP4、WMV、MKV、MPEG、3GP、 DVD、MP3、iPod、iPhone、PSP、Android 等众多格式的转换。

3. < track >标签

< track >标签为< video >标签之类的媒介规定外部文本轨道,如用于规定字幕文件或其他包含文本的文件,当媒介播放时,这些文件是可见的。表 5.7列出了< track >标签常用属性。

属性	值	描述	
default	default	规定该轨道是默认的	
kind	captions chapters subtitles	表示轨道属于何种文本类型 captions:在播放器中显示的简短说明 chapters:定义章节,用于导航媒介资源 subtitles:定义字幕,用于在视频中显示字幕	
label	label	轨道的标签或标题	
src	URL	轨道的 URL	
srclang	language_code	轨道的语言,若 kind 属性值是"subtitles",则该属性必需	

表 5.7 < track >标签常用属性

HTML5 视频外挂字幕英文简称 webVTT(Video Text Track),是以.vtt 为扩展名的纯文本文件。 webVTT 是 UTF-8 编码格式的文本文件,内容示例如下。

WEBVTT

< head >

```
00:00:01.000 --> 00:00:04.000
在海边,奔腾着一群骏马
```

00:00:05.000 --> 00:00:07.000 惊散了鸟儿

webVTT 文件中的每项为一个 cue(提示信息),以箭头分隔开始时间和结束时间,时间格式为 hours: minutes:seconds:milliseconds,必须严格遵守,时、分、秒必须为两位数字,不足的以 0 填补,毫秒必须是 3 位 数字。对应的文本在下一行,文本可以是一行或多行,文本中不能有空行。

提示: 文本轨道会使网站更容易被搜索引擎抓取到。

【例 5.7】 video. html 是一个< video >元素播放带有字幕的视频,其中的两个字幕之一是默认的,如图 5.10 所示。源码如下。

扫一扫

```
<title>HTML视频</title></head><body>< video controls = "controls" autoplay = "autoplay">< source src = "multimedia/Wildlife.mp4" type = "video/mp4">< source src = "multimedia/Wildlife.webm" type = "video/webm">< source src = "multimedia/Wildlife.ogv" type = "video/ogg">< track kind = "subtitles" src = "multimedia/Wildlife - zh.vtt" srclang = "zh" label = "中文" default =</td>"default">< track kind = "subtitles" src = "multimedia/Wildlife - en.vtt" srclang = "en" label = "english">您的浏览器不支持 HTML5 视频。</video></video></body>
```

提示:必须发布到 Web 服务器上进行浏览才能显示字幕。视频加载后,低版本浏览器可能需单击视频播放器控件 CC 按钮,才能显示字幕。



图 5.10 video. html 页面显示

5.3.2 < audio >标签

HTML5使用< audio >标签播放音频,其常用属性和< video >标签一样。

目前< audio >元素支持3种音频编码。

(1) Ogg: 全称应该是 OGGVobis, 是一种新的音频压缩格式。Ogg 是完全免费、开放和没有专利限制的, 文件扩展名是. ogg。

(2) MP3: 是一种音频压缩技术,其全称是动态影像专家压缩标准音频层面 3(Moving Picture Experts Group Audio Layer 3),简称为 MP3,可以大幅度地降低音频数据量。将音乐以 1:10 甚至 1:12 的压缩率 压缩成容量较小的文件,对于大多数用户来说重放的音质与最初的无压缩音频相比没有明显的下降。

(3) WAV: 微软公司(Microsoft)开发的一种声音文件格式,符合 RIFF(Resource Interchange File Format)文件规范,被 Windows 平台及其应用程序所广泛支持,标准格式的 WAV 文件和 CD 格式一样,采用 44.1k 的采样率,16 位量化数字,声音文件质量和 CD 相差无几。

3种格式中,WAV格式音质最好,但是文件较大;MP3压缩率较高,音质比WAV要差;Ogg与MP3在相同位速率(Bit Rate)编码情况下,Ogg体积更小,并且Ogg是免费的。

提示:不同浏览器对于< audio >标签的音频格式支持情况不完全一样。

一般提供 Ogg 和 MP3 格式,就可以支持所有主流浏览器了。

不同格式音频文件的转换,可以在网络上搜索一些免费工具软件进行,如 Free Audio Converter,可从 http://www.freemake.com/free_audio_converter/下载,支持 MP3、WMA、WAV、FLAC、AAC、M4A、OGG 等 30 多种音频格式的转换。

【例 5.8】 audio. html 说明了< audio >标签的用法, 如图 5.11 所示。源码如下。





图 5.11 audio. html 页面显示

5.4 其他嵌入元素

除了把图像、视频和音频嵌入页面上的这些元素外,还有在网页中嵌入各种内容类型的标签:<iframe>、<embed>和<object>。<iframe>标签用于嵌入其他网页;另外两个标签<embed>和<object>用来嵌入多种类型外部内容,如嵌入 PDF、SVG 和 Flash。

5.4.1 < iframe >标签

<iframe>标签允许将其他Web文档嵌入当前页面中。表 5.8列出了<iframe>标签常用属性。

属性	值	描述
src	URL	规定在 iframe 中显示的文档 URL
height	pixels, %	对象的高度,默认值为150px
width	pixels, %	对象的宽度,默认值为 300px
		显示在 <iframe>框架的 HTML5 内容,必须是有</iframe>
srcdoc	HTML_code	效的语法。如果浏览器不支持 srcdoc 属性,则相
		应地会显示 src 属性(若已设置)规定的文件
	"":应用以下所有的限制	
	allow-same-origin: 允许 iframe 内容被视为与包含	
	文档有相同的来源	对框架中的内容施加额外限制。该属性的值可以
sandbox	allow-top-navigation: 允许 iframe 内容从包含文档导航	为空以应用所有限制,也可以以空格分隔的标记
	allow-forms: 允许表单提交	解除特定的限制
	allow-scripts: 允许脚本执行	
	allow-popups: 允许弹出窗口	

表 5.8 < iframe >标签常用属性

黑客或破解者时常将<iframe>标签作为网站的攻击目标,所以要始终使用 sandbox 属性。Sandbox(沙箱)是一种用于安全地运行程序的机制,沙箱技术按照安全策略限制程序对系统资源的使用,进而防止其对系统进行破坏。

sandbox 属性不应该同时添加 allow-scripts 和 allow-same-origin,在这种情况下,嵌入式内容可以绕过阻止站点执行脚本的同源安全策略,并使用 JavaScript 完全关闭沙箱。

有时嵌入第三方内容(如百度地图)非常有意义,为了防止可能带来的安全隐患,只在必要时嵌入,在嵌入时最好使用超文本传输安全协议(Hypertext Transfer Protocol Secure,HTTPS)。

【例 5.9】 iframe. html 说明了< iframe >标签的用法,如图 5.12 所示。源码如下。

```
< head >
< title > iframe 标签</title >
</head >
< body >
```

```
<iframe src = "img.html" sandbox = ""></iframe >
<iframe srcdoc = "<h2 > iframe 标签</h2 > iframe 标签" src = "img.html"></iframe >
<iframe src = "images/svg.svg" sandbox = ""></iframe >
</body >
```



图 5.12 iframe. html 页面显示

<iframe>是一个内联元素,所以这3个<iframe>标签中的内容显示在一行上。

5.4.2 < embed >标签

< embed >标签定义嵌入的内容,如插件,< embed >是一个空标签。一旦对象嵌入页面中,对象将成为页面的一部分。表 5.9 列出了< embed >标签常用属性。

表 5.9 < embed >	▶标签常用属性
-----------------	---------

	1	
属性	值	描述
src	URL	嵌入内容的 URL
height	pixels	对象的高度
type	MIME_type	定义嵌入内容的类型
width	pixels	对象的宽度

【例 5.10】 embed. html 在页面中嵌入了一个 Flash 视频和一个 PDF 文档,如图 5.13 所示。源码如下。

```
< head >
< title > embed 标签</title >
```



```
< body >
```

< embed src = "multimedia/buick.swf" type = "application/x - shockwave - flash" width = "480" height = "360">
< embed src = "multimedia/Pixel.pdf" type = "application/pdf" width = "480" height = "360">

</body>



图 5.13 embed. html 页面显示



提示:现在主流浏览器设置已经默认屏蔽 Adobe Flash Player。

5.4.3 < object >标签

< object >标签定义一个嵌入对象,表示引入一个外部资源,如图像、音频、视频、Java Applets、ActiveX、 PDF和 Flash。表 5.10列出了< object >标签常用属性。

属性	值	描述
data	URL	嵌入内容的 URL
height	pixels	对象的高度
type	MIME_type	定义嵌入内容的类型
width	pixels	对象的宽度

表 5.10 < object >标签常用属性



80

【例 5.11】 object. html 在页面中嵌入了一个 PDF 文档,如图 5.14 所示。源码如下。

```
< head>
    <title > object 标签</title>
</head>
<body>
    <br/>
    <object data = "multimedia/Pix</pre>
```

<object data = "multimedia/Pixel.pdf" type = "application/pdf" width = "480" height = "360"> 如果没有安装 PDF 插件,请下载此文件。 </object>

</body>



图 5.14 object. html 页面显示

嵌入插件是一种传统技术,一般用于内部网或企业项目中。出于安全的考虑,最好不要在外网中使用。

5.5 "叮叮书店"项目首页超链接和图像的使用

"叮叮书店"项目除了首页(index. html)外,还需要建立"图书分类"页面(category. html)、"电子书"页面 (ebook. html)、"客户服务"页面(contact. html)、"关于书店"页面(about. html)、购物车(cart. html)、显示图书 详细内容页面(details. html)和试读页面(read. html),这些页面可以通过超链接访问。

首页内容除了文本和超链接外,还需要使用图像,如书的封面和广告等。

启动 Visual Studio Code,打开"叮叮书店"项目首页 index. html(3.8节创建),进入代码编辑区,添加超链接和图像。

1. 顶部广告

将光标定位到< div id="top-advert" class="center">后面,按Enter 键,输入以下代码。

2. 网站 Logo

将光标定位到< div id="logo">后面,按 Enter 键,输入以下代码。

< img src = "images/logo.png" alt = "叮叮书店">

3. 购物车

将光标定位到li id = "cart-position">后面,把< span class = "icon-cart"> < sup > 2 </sup >改为超链接,代码如下。

< a href = "cart.html" class = "cart - head">< span class = "icon - cart">< sup > 2 </sup >

将光标定位到<div class="cart-thumb">后面,按Enter键,输入以下代码。

< a href = " # .">< img src = "images/recommend3.jpg" alt = "">

将光标定位到< div class="cart-tittle">后面,把"Spring Boot 开发实战"改为超链接,代码如下。

< a href = "♯.">Spring Boot 开发实战

将光标定位到下一个<div class="cart-thumb">后面,按Enter键,输入以下代码。

< a href = " # .">< img src = "images/new2.jpg" alt = "">

将光标定位到下一个< div class="cart-tittle">后面,把"Kubernetes 权威指南"改为超链接,代码如下。

< a href = " # . "> Kubernetes 权威指南

将光标定位到后面,把"去购物车"改为超链接,代码如下。

去购物车

4. 导航菜单

将光标定位到<nav class="center">后面,把无序列表项中的内容改为超链接,代码如下。

提示: 表示空链接,对于不确定的链接暂时可以使用空链接替代,等确定后再修改。 表示链接到页面开始位置,表示链接到当前位置。

5. 图书分类

将光标定位到<h3>图书分类</h3>后面,把无序列表项中的内容改为超链接,代码如下。

编程语言

```
< a href = " # . "> Python </a > 
     < a href = " # .">Java </a>
     < a href = " # . "> Android </a > 
     < a href = " # . ">C 语言</a>
     < a href = " # .">C # </a>
  大数据
  < a href = " #.">数据挖掘</a>
     < a href = " #."> SQL 语言</a>
     < a href = " # .">Mysgl </a>
     < a href = " # . "> Oracle </a>
  < a href = " #.">人工智能</a>
>网页制作
  < a href = " # . "> HTML5 </a>
     < a href = " # . ">CSS3 </a>
     < a href = " # .">JavaScript </a>
     < a href = "#, ">网页设计</a>
  < a href = " # . ">图形图像</a>
```

6. 横幅广告

82

将光标定位到< div id="banner">后面,按 Enter 键,输入以下代码。

```
<dl>
        <dl>
            <dt>< a href = "#." id = "a1">1 </a>< a href = "#." id = "a2">2 </a>< a href = "#." id = "a3">3 </a>
< a href = "#." id = "a4">4 </a>< a href = "#." id = "a5">5 </a></dt>
        <d>></dt>
        <d>>< a href = "#." id = "a4">4 </a>< a href = "#." id = "a5">5 </a></dt>
        <d>></dt>
        <d>></dt>
        <d>></dd>
        <dd>></dd>
        <d>></dd>
        <dd>></dd>
        <
```

7. 用户新闻

将光标定位到<h3>Hi~欢迎逛叮叮!</h3>后面,把下面两个<div>标签中的内容改为超链接,代码如下。

```
    < div >< a href = " # ." class = "login">登录</a> | < a href = " # ." class = "login">注册</a></div >< div >< a href = " # ." id = "btn - new">新人福利</a> < a href = " # ." id = "btn - vip"> VIP 会员</a></div >
```

将光标定位到<div><h4>叮叮快报</h4></div>后面,把无序列表项中的内容改为超链接,代码如下。

8. 本周推荐

将本周推荐、最近新书和最近促销中</h3>后面的"查看更多< div class="title-cover"></div>"改为超链接,代码如下。

```
<a href = "#.">查看更多<div class = "title - cover"></div ></a>
```

将光标定位到< div class="recommend-description">后面,按 Enter 键,输入以下代码。

< a href = "details. html">< img src = "images/recommend1. jpg" alt = "">

同样,将光标定位到下面3个< div class="recommend-description">后面,按 Enter 键,分别输入以下代码。

```
< a href = "details.html">< img src = "images/recommend2.jpg" alt = ""></a>
< a href = "details.html">< img src = "images/recommend3.jpg" alt = ""></a>
```

< a href = "details.html">< img src = "images/recommend4.jpg" alt = "">

将本周推荐、最近新书和最近促销中的< span class="icon-cart">改为超链接,代码如下。

< a href = " # ." class = "main - content - cart">< span class = "icon - cart">

9. 最近新书

将光标定位到<div class="new-description">后面,按Enter键,输入以下代码。

< a href = " # .">< img src = "images/new1.jpg" alt = "">

同样,将光标定位到下面3个<div class="new-description">后面,按Enter键,分别输入以下代码。

< a href = " # .">< img src = "images/new2.jpg" alt = "">
< a href = " # .">< img src = "images/new3.jpg" alt = "">
< a href = " # .">< img src = "images/new4.jpg" alt = "">

10. 最近促销

将光标定位到<div class="description">后面,按Enter键,输入以下代码。

< a href = " # .">< img src = "images/sale1.jpg" alt = "">

同样,将光标定位到下面3个<div class="description">后面,按Enter键,分别输入以下代码。

```
< a href = " # .">< img src = "images/sale2.jpg" alt = ""></a>
< a href = " # .">< img src = "images/sale3.jpg" alt = ""></a>
< a href = " # .">< img src = "images/sale4.jpg" alt = ""></a>
```

11. 畅销图书

将光标定位到<div class="curr">后面,按Enter键,输入以下代码。

<div class = "p - img">< img src = "images/selling1.jpg" alt = ""></div>

同样,将光标定位到下面4个<div class="curr">后面,按Enter键,分别输入以下代码。

```
< div class = "p - img">< img src = "images/selling2.jpg" alt = ""></div >
< div class = "p - img">< img src = "images/selling3.jpg" alt = ""></div >
< div class = "p - img">< img src = "images/selling4.jpg" alt = ""></div >
< div class = "p - img">< img src = "images/selling4.jpg" alt = ""></div >
< div class = "p - img">< img src = "images/selling5.jpg" alt = ""></div >
</div >
</div >
</div >
```

再将光标定位到第1个<div class="curr">后面,把<div class="curr">中的所有内容作为一个整体改为超链接,代码如下。

```
< a href = "#.">
        <div class = "p - img">< img src = "images/selling1.jpg" alt = ""></div>
        <div class = "p - name">深度学习 [deep learning]< strong > ¥ 43.50 </strong >
            <del > ¥ 52.00 </del >
            </div>
        <//a>
```

同样,下面4个<div class="curr">中的内容也参照上面进行修改。

12. 合作伙伴

将光标定位到< span class="icon-partner"></ span >合作伙伴</ h3 > </ div > 后面,把无序列表项 中的内容改为超链接,代码如下。

```
< a href = "#.">中国电子商务研究中心</a>< a href = "#.">清华大学出版社</a>
```

```
< a href = "#.">中国人民大学出版社</a>< a href = "#.">中国社会科学出版社</a>< a href = "#.">机械工业出版社</a>
```

13. 关于书店

84

将光标定位到<h3>< span class="icon-about">关于书店</h3></div>< div class="content"> 后面,按Enter键,输入以下代码。

```
< img src = "images/about - bookstore.jpg" alt = "">
```

14. 页脚

将光标定位到<h4>购物指南</h4>后面,把无序列表项中的内容改为超链接,代码如下。

```
< a href = "#.">购物流程</a>< a href = "#.">会员介绍</a>< a href = "#.">联系客服</a>
```

将光标定位到<h4>配送方式</h4>后面,把无序列表项中的内容改为超链接,代码如下。

< a href = " #.">上门自提< a href = " #.">限时达

将光标定位到<h4>支付方式</h4>后面,把无序列表项中的内容改为超链接,代码如下。

< a href = " #.">货到付款< a href = " #.">在线支付

将光标定位到<h4>售后服务</h4>后面,把无序列表项中的内容改为超链接,代码如下。

< a href = " # .">售后政策< a href = " # .">伤格保护

15. 版权信息

将光标定位到"京 ICP 证 000001 号音像制品经营许可证"后面,按 Enter 键,输入以下代码。

将光标定位到< div>Copyright©2020-2028 后面,把"叮叮书店"改为超链接,代码如下。

< a href = "index. html">叮叮书店

在浏览器预览,如图 5.15 所示。



图 5.15 "叮叮书店"项目首页预览示意图

5.6 小结

本章重点介绍了图像元素和响应式图像,介绍了 HTML5 如何使用音频和视频,简单介绍了嵌入元素, 详细讲解了"叮叮书店"项目首页超链接、图像的添加过程和基本操作。

5.7 习题

1. 选择题

(1) 关于以下两行 HTML5 代码,下列描述中正确的是()。

```
<ing src = "image.gif" alt = "picture">
```

- picture 。
- A. 两者都是将图像链接到网页
- B. 前者是链接后在网页显示图像,后者是在网页中直接显示图像
- C. 两者都是在网页中直接显示图像
- D. 前者是在网页中直接显示图像,后者是链接后在网页显示图像
- (2)下列关于网页中图像的说法不正确的是()。
 - A. 网页中的图像并不与网页保存在同一个文件中,每幅图像单独保存
 - B. 标签可以描述图像的位置、大小等属性
 - C. 标签可以直接描述图像上的像素
 - D. 图像可以作为超链接标签的内容

(3) 若要在页面中创建一个图像超链接,要显示的图像为 logo.gif,链接地址为 http://www.sohu.com/,下列用法中正确的是()。

A. < a href="http://www.sohu.com/">logo.gif

- B.
- C.

D.

(4) 下列标签中主要用来创建视频和 Flash 的是()。

A. < object > B. < embed > C. < form > D. < marquee >

(5)为了解决浏览器对视频格式的兼容情况,可以使用()标签为同一个媒体数据指定多个播放格 式与编码方式。

A. < source > B. < audio > C. < video > D. < track >

2. 简答题

(1) 嵌入图像的元素是什么? 它有哪些必需和常用属性?

- (2) 目前< video >标签支持哪些视频编码?
- (3) 如何解决不同浏览器对视频格式的兼容问题?
- (4) < map >和< area >标签的作用是什么?
- (5) 实现响应式图像有几种方法?