第3章

HTML5中的重要元素

视频讲解

本章学习目标

- 熟练掌握文档元素的使用方法。
- 了解脚本和文本层次元素的使用方法。
- 理解元素公共属性的使用方法。

3.1 html 根元素

在 HTML 文档中，元素 html 代表了文档的根，其他所有元素都是在该元素的基础上进行延伸或拓展的，该元素也是 HTML 文档的最外层元素，因此也称该元素为根元素。

3.1.1 定义

html 元素可以在浏览器执行页面时，告知它执行的是一个 HTML 文档，在 HTML5 与 HTML4.0.1 中，该元素的差异不大，主要区别在于 xmlns 属性。在 HTML 4.0.1 中，该属性是必需的，因为它将在 HTML 转换成 xmlns 的过程中发挥作用；而在 HTML5 中，可以不必增加这个属性。另外，在 HTML5 中新增加了一个属性 manifest，用于指向文档缓存信息 URL。

3.1.2 属性

在 html 元素中有两个非常重要的属性，详细说明如表 3-1 所示。

表 3-1 html 元素的属性

属　性	值	描　述
manifest	URL	该 URL 指向描述文档缓存信息的地址
xmlns	http://www.w3.org/1999/xhtml	设置 XML namespace 的属性

实例 3-1　元素 html 的使用

1. 功能描述

在新建的页面中，显示“内容部分…”几个字符。

2. 实现代码

在 WebStorm 中新建一个 HTML 页面 3-1.html，加入代码如代码清单 3-1 所示。

代码清单 3-1　元素 html 的使用

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>元素 html 的使用</title>
    <style type="text/css">
        body {
            font-size: 10pt
        }
    </style>
</head>
    <body>
    内容部分
    …
    </body>
</html>
```

3. 页面效果

该页面在 Chrome 浏览器中执行的页面效果如图 3-1 所示。

图 3-1　元素 html 的使用效果

4. 源码分析

在实例 3-1 中，html 元素是最外层元素，它包括两个主要的部分：头部 head 与主体 body。其结构如图 3-2 所示。

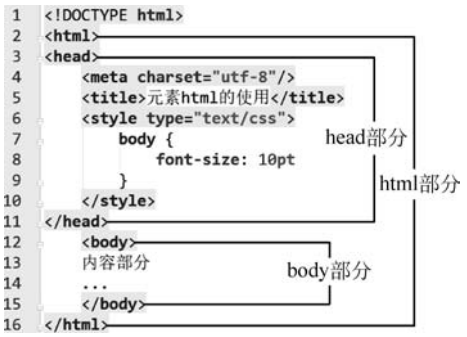

图 3-2　html 元素所包含的结构

3.2　文档元素

在一个 HTML 文档中包含两个部分：一个是头部分，由 head 元素包含；另一个部分是主体部分，由 body 元素包含。下面分别介绍它们在 HTML5 中的使用方法。

3.2.1　定义

由于 head、title、base、link、meta、style 元素常用于说明文档的相关信息，因此，这些元素也称为文档元素。

3.2.2　包含标签

head 元素是所有头部元素的载体，其中的元素可以包含 JavaScript 脚本或文件、CSS 样式或文件、或者其他说明文档、说明信息。head 元素包含的标签如表 3-2 所示。

表 3-2　head 元素中的标签及说明

标签名称	说　明	标签名称	说　明
base	定义页面中所有超链接的默认地址或目标	script	定义客户端脚本代码或文件
		style	定义 HTML 文档中的样式信息
link	导入页面中的样式文件	title	设置文档的标题内容
meta	定义页面中的相关信息		

base 元素可以设置页面中 URL 为空时的值，该元素有两个属性，一个是 href，表示当页面的 URL 为空时的超链接地址；另一个是 target，表示打开页面超链接的方式，如_blank 等。

```
...省略部分代码
<head>
  <base href="http://www.html5.com/" target="_blank"/>
</head>
<body>
  <a href="index.html">testPage</a>
</body>
...省略部分代码
```

上述代码中,单击 testPage 文字超链接时,将以新窗口的方式,打开 http://www.html5.com/ index.html 地址。

meta 元素可以设置页面的文档信息,如针对搜索引擎的关键字等,在 HTML5 中,该元素不再支持 scheme 属性,同时新增了一个 charset 属性,使页面字符集的定义更加方便。

```
...省略部分代码
<head>
  <meta name="keywords" content="HTML5,UI, Web 前端开发"/>
</head>
...省略部分代码
```

上面的代码中,通过 meta 元素定义了针对搜索引擎的关键字,值为“HTML5,UI,Web 前端开发”,便于搜索引擎对该页面的检索。

实例 3-2 文档元素的使用

1. 功能描述

在新建的页面<head>元素中,加入该元素所包含的各类标签,并定义超级链接的样式,单击“请单击我”时,将展示相应样式效果并进入 base 元素设置的默认地址。

2. 实现代码

在 WebStorm 中新建一个 HTML 页面 3-2.html,加入代码如代码清单 3-2 所示。

代码清单 3-2 文档元素的使用

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>文档元素的使用</title>
    <base href="http://www.html5.com/" target="_blank"/>
    <meta name="keywords" content="HTML5, CSS, JavaScript"/>
    <meta name="description" content="用于检测页面的文档元素"/>
    <style type="text/css">
        a {
            padding: 8px;
            font-size: 13px;
            text-decoration: none;
        }
        a:hover {
            border: solid 1px #ccc;
            background-color: #eee;
        }
    </style>
</head>
<body>
```

```
    <a href="index.html">请单击我</a>
</body>
</html>
```

3. 页面效果

该页面在 Chrome 浏览器中执行的页面效果如图 3-3 所示。

图 3-3　文档元素的使用效果

4. 源码分析

在 head 元素所包含的文档元素中，许多元素增加了用于 HTML5 中执行的新属性，同时，也有属性在 HTML5 中不再被支持。link 元素在 HTML5 中不再支持的属性有 charset、rev、target，其他属性同样可以在 HTML5 中执行。

需要说明：在一个页面文档中，base 与 title 元素只能使用一次，并且必须包含在 head 元素中。同时，base 元素应排在其他元素之前，以便于其他元素能调用 base 元素的属性；而其他元素可以重复使用多次。

3.3　脚本

为了增加页面的互动性，需要对文档编写客户端脚本，最为常用的语言是 JavaScript，通过编写客户端的脚本语言，可以实现对页面文档进行验证表单、变更内容等操作。

3.3.1　定义

在页面文档中，用于标识脚本的标签有两个：一个是 script，该元素既可以包含脚本语言，也可以通过 src 属性导入一个脚本文件，通过元素必选属性 type，选择脚本的 mime 类型；另外一个是 noscript 元素，它是一个检测工具，用于 script 中的脚本内容未被执行时显示的内容，即浏览器如果支持 script 中的脚本，则不会显示 noscript 中的内容。

3.3.2　属性

在脚本元素 script 中，使用属性设置脚本格式和类型，详细说明如表 3-3 所示。

表 3-3 script 元素的属性

属性名称	值	描述
async	true 或 false	定义是否异步执行脚本,此属性为 HTML5 新增
charset	charset	设置脚本中使用的字符编码,此属性 HTML5 不再支持
language	JavaScript 等	定义脚本的语言类型,此属性 HTML5 不再支持
xml:space	preserve	此属性 HTML5 不再支持

script 元素中的 async 属性为 HTML5 新增,该属性有两个取值 true 或 false,当取值为 true 时,脚本在页面中执行的方式是异步的,即在页面解析的过程中执行;当取值为 false 时,脚本将立即执行,页面也会等脚本执行完成后继承解析。

实例 3-3 脚本元素的使用

1. 功能描述

在新建的页面中,增加一个 id 值为 txtContent 的文本框,一个按钮,当单击按钮时,通过页面中加入的 JavaScript 脚本代码,获取文本框中的内容,显示在页面中。

2. 实现代码

在 WebStorm 中新建一个 HTML 页面 3-3.html,加入代码如代码清单 3-3-1 所示。

代码清单 3-3-1 脚本元素的使用

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>脚本元素的使用</title>
    <link href="Css/css3.css"
        rel="stylesheet" type="text/css">
    <script type="text/javascript" async="true">
        function Btn_Click() {
            var strTxt =
            document.getElementById("txtContent").value;
            var strDiv = document.getElementById("divShow");
            strDiv.style.display = "block";
            strDiv.innerHTML = "您输入的字符是:" + strTxt;
        }
    </script>
    <noscript>您的浏览器不支持 JavaScript!</noscript>
</head>
<body>
    <input type="text" id="txtContent"
           class="inputtxt">
    <input type="button" value="请单击我"
           class="inputbtn" onClick="Btn_Click();">
    <div id="divShow" class="divShow"></div>
</body>
</html>
```

为了增加页面的浏览效果，在实例 3-3 中，导入了一个 CSS 样式文件 css3.css，其代码如下列代码清单 3-3-2 所示。

代码清单 3-3-2 实例 3-3 的样式文件

```
@charset "utf-8";
/* CSS Document */
body{
  font-size:12px
}
.inputbtn {
    border:solid 1px #ccc;
    background-color:#eee;
    line-height:18px;
    font-size:12px
}
.inputtxt {
    border:solid 1px #ccc;
    line-height:18px;

font-size:12px
}
.divShow{
    border:solid 1px #666;
    background-color:#eee;
    margin-top:5px;
    padding:5px;
    width:196px;
    display:none
}
```

3. 页面效果

该页面在 Chrome 浏览器中执行的页面效果如图 3-4 所示。

图 3-4 脚本元素的使用效果

4. 源码分析

在本实例的 script 元素中，设置 async 属性的值为 true，即允许脚本在页面解析时异步执行，HTML5 中新增的这个属性，可以在很大程度上缓解页面解析的压力，加速页面加载

的速度,同时又不会阻碍 script 元素中脚本的执行,如果是执行大量的 JavaScript 代码,其效果将更加明显。

3.4 文本层次语义

页面中常常需要显示一段文章或文字,我们称之为文本内容。为了使内容更加形象、生动,需要增加一些特殊功能的元素,用于突出文本间的层次关系或标为重点,这样的元素被称为文本层次语义标记。在 HTML5 中,mark 和 cite 就是常用文本层次语义元素。下面分别进行详细说明。

3.4.1 mark 元素

mark 元素是 HTML5 中新增的元素,主要功能是在文本中,用于突出高亮显示某一个或几个字符,旨在引起用户的特别注意。其使用方法与 em 或 strong 元素有相似之处,但相比而言,HTML5 中新增的 mark 元素在突出显示时,更加随意与灵活。

实例 3-4 mark 元素的使用

1. 功能描述

在页面中,使用 h5 元素创建一个标题,然后通过 p 元素对标题进行阐述,阐述中,为了引起用户的注意,使用 mark 元素高亮处理了某些字符。

2. 实现代码

在 WebStorm 中新建一个 HTML 页面 3-4.html,加入代码如代码清单 3-4 所示。

代码清单 3-4 mark 元素的使用

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>mark 元素的使用</title>
    <link href="Css/css3.css"
          rel="stylesheet" type="text/css">
</head>
<body>
    <h5>优秀开发人员的
        <mark>素质</mark>
    </h5>
    <p class="p3_5">
        一个优秀的 Web 页面开发人员,必须具有
        <mark>过硬</mark>
        的技术与
        <mark>务实</mark>
        的专业精神
    </p>
```

```
</body>
</html>
```

3. 页面效果

该页面在 Chrome 浏览器中执行的页面效果如图 3-5 所示。

图 3-5 mark 元素的使用效果

4. 源码分析

mark 元素的这种高亮显示的特征，除用于文档中突出显示外，还常用于查看搜索结果页中关键字的高亮显示，其目的主要是用于引起用户的注意，突出显示关键字的所在位置。

虽然 mark 元素在使用效果上与 em 或 strong 元素有相似之处，但三者的出发点是不一样的。strong 元素是作者对文档中某段文字的重要性进行强调；em 元素是作者为了突出文章的重点而进行的设置；mark 元素是数据显示时，以高亮的形式显示某些字符，与原作者本意无关。

3.4.2 cite 元素

cite 元素创建一个引用标记，用于文档中参考文献的引用说明，如书名或文章名称。一旦在文档中使用了该标记，将以斜体的样式展示在页面中，区别于段落中的其他字符。

实例 3-5 cite 元素的使用

1. 功能描述

在页面中，通过 p 元素显示一段文档，然后，在文档的下面使用 cite 元素标识这段文档所引用的书名。

2. 实现代码

在 WebStorm 中新建一个 HTML 页面 3-5.html，加入代码如代码清单 3-5 所示。

代码清单 3-5 cite 元素的使用

```
<!DOCTYPE html>
<html>
```

```
<head>
    <meta charset="UTF-8">
    <title>cite元素的使用</title>
    <link href="Css/css3.css"
          rel="stylesheet" type="text/css">
</head>
<body>
    <h5>jQuery</h5>
    <p>
        jQuery是继Prototype之后的一个优秀的JavaScript框架,
        深受全球开发者的欢迎...</p>
    <p>
        ---引自 <<<cite>jQuery权威指南</cite>>> ---
    </p>
</body>
</html>
```

3. 页面效果

该页面在 Chrome 浏览器中执行的页面效果如图 3-6 所示。

图 3-6　cite 元素的使用效果

4. 源码分析

在 HTML5 中,cite 元素基本兼容了在 HTML4 中的全部功能,但定义时更加严格。在使用该元素时,除包含标题或书名外,不允许包含更多的其他引用信息,如作者姓名、出版日期等。

3.5　公共属性

在 HTML5 完成替换 HTML4 的过程中,无论是新增或是改良的元素,都有一些共同的属性,我们称之为公共属性。为了进一步了解这些公共属性,下面选择几个常用的公共属性逐一进行介绍。

3.5.1　draggable 属性

draggable 属性的功能是设置用户是否允许拖动元素,该属性有三个值,分别为 true、

false、auto。如果设置为 true,则可以用鼠标选中元素后,进行拖动的操作。

实例 3-6 draggable 属性的使用

1. 功能描述

在页面中,使用 article 元素显示一段文字,并设置 article 元素的属性 draggable 值为 true,当用户选中这段文字移动鼠标指针时,可以实现拖动的效果。

2. 实现代码

在 WebStorm 中新建一个 HTML 页面 3-6.html,加入代码如代码清单 3-6 所示。

代码清单 3-6 draggable 属性的使用

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>draggable 属性的使用</title>
    <link href="Css/css3.css"
          rel="stylesheet" type="text/css">
</head>
<body>
    <h5>元素的拖动属性</h5>
    <article draggable="true" class="p3_7">
          这是一段可以拖动的文字,选中后进行拖动.
    </article>
</body>
</html>
```

3. 页面效果

该页面在 Chrome 浏览器中执行的页面效果如图 3-7 所示。

图 3-7 draggable 属性的使用效果

4. 源码分析

在 HTML5 中,当某个元素的 draggable 属性值为 true 时,则表示该元素可以被拖动,在拖动的过程中,元素的内容和样式与拖动前是相同的,也不允许修改。

3.5.2 hidden 属性

在 HTML5 中,绝大部分的元素都支持 hidden 属性,该属性只有两个取值:true 和 false。当 hidden 的取值为 true 时,元素不在页面中显示,但还存在于页面中;反之,则显示于页面中,该属性的默认值为 false,即元素创建时便显示出来。

实例 3-7 hidden 属性的使用

1. 功能描述

在页面的 nav 元素中设置两个相互排斥的单选按钮,一个用于显示 article 元素,另一个用于隐藏 article 元素,通过编写相应的 JavaScript 代码实现上述功能。

2. 实现代码

在 WebStorm 中新建一个 HTML 页面 3-7.html,加入代码如代码清单 3-7 所示。

代码清单 3-7 hidden 属性的使用

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>hidden 属性的使用</title>
    <link href="Css/css3.css" rel="stylesheet"
          type="text/css">
    <script type="text/javascript" async="true">
        function Rdo_Click(v) {
            var strArt = document.getElementById("art");
            if (v)
                strArt.removeAttribute("hidden");
            else
                strArt.setAttribute("hidden", "true");
        }
    </script>
</head>
<body>
    <h5>元素的隐藏属性</h5>
    <nav style="padding-top:5px;padding-bottom:5px">
        <input type="radio" id="rdoHidden_1"
               onClick="Rdo_Click(1)"
               name="rdoHidden" value="1"
               checked="true"/>显示
        <input type="radio" id="rdoHidden_2"
               onClick="Rdo_Click(0)"
               name="rdoHidden" value="0"/>隐藏
    </nav>
    <article id="art" class="p3_8">
        今天是一个好天气啊,蓝蓝的天空,飘着朵朵白云。
    </article>
</body>
</html>
```

3. 页面效果

该页面在 Chrome 浏览器中执行的页面效果如图 3-8 所示。

图 3-8　hidden 属性的使用效果

4. 源码分析

在页面的 JavaScript 代码中，根据单击单选按钮时传来的不同值，向 article 元素添加或删除 hidden 属性，从而实现该元素显示或隐藏的页面效果。

3.5.3　spellcheck 属性

spellcheck 属性用于检测文本框或输入框中的拼音或语法是否正确，该属性的值为布尔值，即 true 或 false。如果为 true，则要检测对应输入框中的语法；反之，则不检测。

实例 3-8　spellcheck 属性的使用

1. 功能描述

在页面中分别创建两个 textarea 输入框元素，第一个元素将 spellcheck 属性设置为 true，即需要语法检测；另外一个元素的 spellcheck 属性设置为 false，即不要语法检测，并分别在两个输入框中输入文字，对比不同的检测效果。

2. 实现代码

在 WebStorm 中新建一个 HTML 页面 3-8.html，加入代码如代码清单 3-8 所示。

代码清单 3-8　spellcheck 属性的使用

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>spellcheck 属性的使用</title>
    <link href="Css/css3.css"
          rel="stylesheet" type="text/css">
</head>
```

```
<body>
    <h5>输入框中语法检测属性</h5>
    <p>需要检测<br/>
       <textarea spellcheck="true"
                 class="inputtxt"></textarea>
    </p>
    <p>不需要检测<br/>
       <textarea spellcheck="false"
                 class="inputtxt"></textarea>
    </p>
</body>
</html>
```

3. 页面效果

该页面在 Chrome 浏览器中执行的页面效果如图 3-9 所示。

图 3-9　spellcheck 属性的使用效果

4. 页面效果

在 HTML5 中，虽然各浏览器对 spellcheck 属性进行了很好的支持，但在 Chrome 浏览器中支持的元素是有差异的，即在该浏览器中，支持 textarea 输入框元素，而不支持 input 元素中的文本框，Firefox 和 Opera 浏览器需在“选项”菜单中手动设置，才能显示效果。

3.5.4　contenteditable 属性

在 HTML5 中，有一个非常便捷的属性，它可以直接对显示在页面中的文字进行编辑，该属性就是公共属性 contenteditable，该属性的取值为布尔型，即 true 或 false。如果在元素中将该属性的值设置为 true，那么，就可以对该元素显示的文本内容直接进行编辑了。

实例 3-9　contenteditable 属性的使用

1. 功能描述

在页面中分别创建两个 article 内容元素，第一个元素将 contenteditable 属性设置为 true，用于直接内容的编辑；第二个 article 元素保存编辑后的内容，当用户编辑完成后，单击“保存”按钮，则将编辑后的内容显示在第二个 article 内容元素中。

2. 实现代码

在 WebStorm 中新建一个 HTML 页面 3-9.html,加入代码如代码清单 3-9 所示。

代码清单 3-9 contenteditable 属性的使用

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>contenteditable 属性的使用</title>
    <link href="Css/css3.css"
          rel="stylesheet" type="text/css">
    <script type="text/javascript" async="true">
        function Btn_Click() {
            var strArt = document.getElementById("art_0")
                        .innerHTML;
            var objArt = document.getElementById("art_1");
            objArt.innerHTML = '<b>编辑后:</b>' + strArt;
        }
    </script>
</head>
<body>
    <h5>元素的内容编辑属性</h5>
    <article contenteditable="true"
             class="p3_10" id="art_0">
        一段可编辑的文字
    </article>
    <article class="p3_10" id="art_1">
    </article>
    <input type="button" value="保存"
           class="inputbtn" onClick="Btn_Click();">
</body>
</html>
```

3. 页面效果

该页面在 Chrome 浏览器中执行的页面效果如图 3-10 所示。

图 3-10 contenteditable 属性的使用效果

4. 页面效果

在 HTML5 中,大部分显示文本内容的元素都支持 contenteditable 属性,因此,该属性的使用给页面中用户的交互体验带来极大的方便。目前,暂无相关的 API 对编辑后的内容进行直接保存,如果需要保存,只能借助于 AJAX 或 jQuery 中的异步操作,更新对应的后台数据。

小结

在本章中,先从最基本的根元素 html 讲起,分门别类地对 HTML5 中的各重要元素采用理论辅助实例的方式进行介绍,最后介绍 HTML5 中各元素的公共属性,通过对这些 HTML5 中新增或改良元素的介绍,可以进一步加深对各使用元素的理解,为第 4 章 HTML5 中表单元素的学习打下扎实的实践基础。