

01 CHAPTER

第1章

HTML入门必学

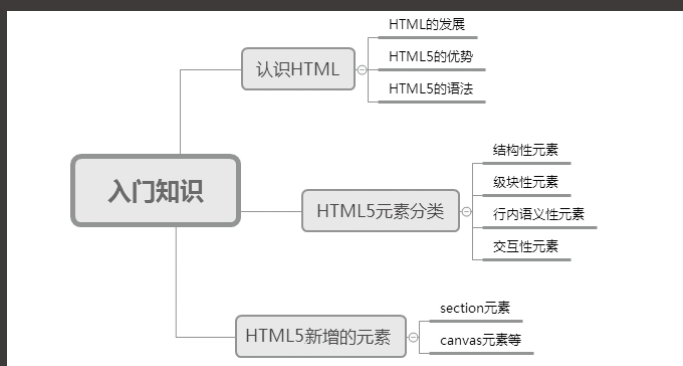
内容概要

HTML 是目前在网络上应用最为广泛的语言，是构成网页文档的主要语言。HTML 文档是由 HTML 标签组成的描述性文本，HTML 标签可以设置文字、图形、动画、声音、表格和链接等。HTML 是一种规范，一种标准，它通过标记符号来标记要显示的网页中的各个部分。

学习目标

- ◆ 了解 HTML 的基础概念和发展史
- ◆ 了解 HTML 的基本结构
- ◆ 掌握 HTML5 的新增功能
- ◆ 学会 HTML5 新增元素的使用

知识导图



课时安排

- ◆ 理论知识 1 课时
- ◆ 上机练习 1 课时





1.1 认识 HTML

HTML 的英文全称是 Hyper Text Markup Language，中文名称为超文本标记语言，是全球广域网上描述网页内容和外观的标准。HTML 被用来结构化信息，例如标题、段落和列表等，也可用来描述文档的外观和语义。由蒂姆·伯纳斯-李（Tim Berners-Lee）给出原始定义，由 IETF 用简化的 SGML（标准通用标记语言）语法进行进一步发展的 HTML，后来成为国际标准，并由万维网联盟（W3C）进行维护。早期的 HTML 语法被定义得较为松散，以便不熟悉网络出版的人使用。此后，官方标准渐渐趋于严格，但浏览器继续显示不符合标准的 HTML。使用 XML 的严格规则的 XHTML（可扩展超文本标记语言）是 W3C 计划中 HTML 的接替者。虽然很多人认为它已经成为当前的 HTML 标准，但实际上是一个独立的、和 HTML 平行发展的标准。W3C 目前的建议是使用 XHTML 1.1、XHTML 1.0 或者 HTML 4.01。

HTML5 是标准通用标记语言下的一个应用超文本标记语言（HTML），已进行了五次重大修改。较之以前的版本，HTML5 不仅可表示 Web 内容，其新功能还会使 Web 成为一个成熟的平台。在 HTML5 上，视频、音频、图像、动画，以及同计算机的交互都将被标准化。

1999 年 12 月发布 HTML 4.01 后，后继的 HTML5 和其他标准被束之高阁，为了推动 Web 标准化运动的发展，一些公司联合起来，成立了一个叫作 Web Hypertext Application Technology Working Group（Web 超文本应用技术工作组，WHATWG）的组织。WHATWG 致力于 Web 表单和应用程序，而 W3C（World Wide Web Consortium，万维网联盟）专注于 XHTML 2.0。在 2006 年，双方决定合作创建一个新版本的 HTML，这个新版本的 HTML 就是今天所熟知的 HTML5。

HTML5 添加了很多语法特征，其中的 <audio><video> 和 <canvas> 元素同时集成了 SVG 内容。这些元素是为了更容易地在网页中添加并处理多媒体和图片内容而添加的。其他的新元素包括 <section><article><header><nav> 和 <footer>，它们则是为了丰富文档的数据内容。新的属性的添加也是为了同样的目的，同时 API 和 DOM 已经成为 HTML5 中的基础部分。HTML5 还定义了处理非法文档的具体细节，使得所有浏览器和客户端能都一致地处理语法错误。



1.2 HTML5 的优势

HTML5 与以往的 HTML 版本不同，HTML5 在字符集 / 元素和属性等方面做了大量的

改进。在讨论 HTML5 编程之前，首先带领大家了解 HTML5 的一些优势，以便为后面学习编程做好铺垫。

1.2.1 强大的交互性

HTML5 与之前的版本相比，在交互上做了很大文章。以前所能看见的页面中的文字都是只能看，不能修改。而在 HTML5 中只需添加一个 `contenteditable` 属性，即可看见页面内容变为可编辑。



小试身手 制作一个可以被编辑的页面

代码如下：

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title> 无标题文档 </title>
</head>
<body>
<p> 不能被用户编辑：关关雎鸠，在河之洲。窈窕淑女，君子好逑。 </p>
<p contenteditable="true"> 可以被用户编辑：关关雎鸠，在河之洲。窈窕淑女，君子好逑。 </p>
</body>
</html>
```

只需要在 `p` 标签内部加入 `contenteditable` 属性，并且让其值为真。在浏览器中显示的效果如图 1-1 所示。

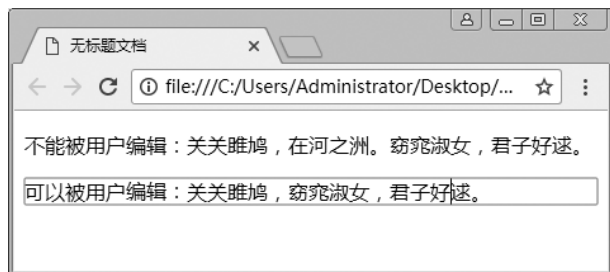


图 1-1

通过图 1-1 可以看出 HTML5 在交互方面给用户提供了很大便利与权限，但是 HTML5 的强大交互远不止这一点。除了对用户展现出了非常友好的态度之外，对开发者也是如此。例如，在一个文本输入框输入提示字提醒用户“请输入您的账号”等这样的操作来提醒用户页面中的某些输入框的功能，在 HTML5 以前需要写大量的 JavaScript 代码来完成这一操作，但现在只需要一个 `placeholder` 属性即可轻松搞定，节省了开发人员大量的时间与精力。



代码如下：

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title> 无标题文档 </title>
</head>
<body>
<form action="#" method="post">
<p><input type="text" value="" placeholder=" 输入您的用户名 "></p>
<p><input type="password" value="" placeholder=" 再输入您的密码 "></p>
</form>
</body>
</html>
```

代码的运行效果如图 1-2 所示。

HTML5 除了为用户和开发人员提供便利，还考虑到了各大浏览器厂商。例如，以前要在网页中看视频需要 Flash 插件，无形中增加了浏览器的负担，而现在只需要一个简单的 video 即可满足用户在网页中看视频的需求。

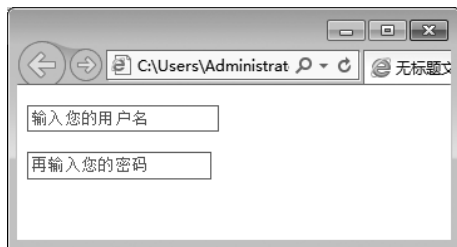


图 1-2

1.2.2 使用 HTML5 的优势

这里列出了使用 HTML5 的原因。

（1）简单

HTML5 使得创建网站更加简单。新的 HTML 标签，如 <header><footer><nav><section><aside> 等，使得阅读者更加容易访问内容。以前，即使定义了 class 或者 id 也无法了解给出的 div 是什么。使用新的语义学的定义标签，可以更好地了解 HTML 文档。

（2）视频和音频支持

以前想要在网页上实现视频和音频的播放都需要借助 Flash 等第三方插件完成，而在 HTML5 中可以直接使用标签 <video> 和 <audio> 访问资源。HTML5 视频和音频标签基本将其视为图片：<video src=""/>。至于参数，如宽度和高度或者自动播放，只需像其他 HTML 标签一样定义：<video src="url" width="640px" height="380px" autoplay/>。

HTML5 将以前烦琐的过程变得简单，然而一些过时的浏览器可能对 HTML5 的支持度并不是很友好，需要添加更多代码让其正常工作。但这个代码还是比 `<embed>` 和 `<object>` 简单得多。

（3）文档声明

没错，就是 `doctype`，没有更多内容了，是不是非常简单？不需要拷贝粘贴，也没有多余的 `head` 标签。最大的好处在于，除了简单，它能在每一个浏览器中正常工作，即使是声名狼藉的 IE6 也没有问题。

（4）结构清晰、语义明确的代码

如果对简单、优雅、容易阅读的代码有所偏好的话，HTML5 绝对是一个为你量身定做的东西。HTML5 允许写出简单清晰且富于描述的代码。符合语义学的代码允许分开样式和内容。看看这个典型的简单拥有导航功能的 `header` 代码：

```
<div id="header">
<h1>Header Text</h1>
<div id="nav">
<ul>
<li><a href="#">Link</a></li>
<li><a href="#">Link</a></li>
<li><a href="#">Link</a></li>
</ul>
</div>
</div>
```

是不是很简单？但是使用 HTML5 后会使得代码更加简单并且富有含义：

```
<header>
<h1>Header Text</h1>
<nav>
<ul>
<li><a href="#">Link</a></li>
<li><a href="#">Link</a></li>
<li><a href="#">Link</a></li>
</ul>
</nav>
</header>
```

HTML5 可以通过使用语义学的 HTML `header` 标签描述内容来解决 `div` 及其 `class` 定义问题。以前需要大量使用 `div` 来定义每一个页面内容区域，但是使用新的 `<section>` `<article>` `<header>` `<footer>` `<aside>` 和 `<nav>` 标签，可以让代码更加清晰，便于阅读。

（5）强大的本地存储

HTML5 中引入了新特性本地存储，这是一个非常酷炫的新特性。有一点像比较老的技术 cookie 和客户端数据库的融合。但是它比 cookie 更好用，存储量也更大。因其支持多个 Windows 存储，拥有更好的安全性，且浏览器关闭后数据也可以被保存。

本地存储在 HTML5 工具中不需要第三方插件就能实现。能够将数据保存到浏览器中意味着可以简单地创建一些应用特性，例如：保存信息、缓存数据，以及加载上一次的 application 状态。

（6）交互升级

人们都喜欢对用户有反馈的动态网站，享受互动的过程。HTML5 中的 `<canvas>` 标签可以做更多的互动和动画，就像使用 Flash 达到的效果，经典游戏“水果忍者”就可以通过 canvas 画图功能来实现。

（7）HTML5 游戏

前几年，基于 HTML5 开发的游戏非常火爆。近两年虽然受到了一些冲击，但如果能找到合适的盈利模式，HTML5 依然是在手机端开发游戏的首选技术。

（8）移动互联网

移动设备如今已经占领世界。我们的生活只需要一部智能手机即可满足某些需求，如用手机支付、使用手机端订购外卖。HTML5 是最移动化的开发工具。随着 Adobe 宣布放弃移动 Flash 开发，用户将会考虑使用 HTML5 来开发 Web 应用。当手机浏览器完全支持 HTML5，那么开发移动项目将会和设计更小的触摸显示一样简单。这里有很多的 meta 标签允许用户优化移动：viewport，允许用户定义 viewport 宽度和缩放设置；全屏浏览器，IOS 指定的数值允许 Apple 设备全屏模式显示；Home screen icons，就像桌面收藏这些图标可以用来添加收藏到 IOS 和 Android 移动设备的首页中。



1.3 HTML5 语法

HTML5 中的语法和之前版本有些变化，因为 HTML5 设计为化繁为简的准则文档，类型和字符说明等都进行了简化，下面进行说明。

1.3.1 文档类型声明

DOCTYPE 声明是 HTML 文件中必不可少的，位于文件第一行，在 HTML4 中，它的声明方法如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

在 HTML5 中，刻意不使用版本声明，一份文档将会适用于所有版本的 HTML。HTML5 中的 DOCTYPE 声明方法（不区分大小写）如下：

```
<!DOCTYPE html>
```

另外，当使用工具时，也可以在 DOCTYPE 声明方式中加入 SYSTEM 识别符，声明方法如下：

```
<!DOCTYPE HTML SYSTEM"about:legacy-compat">
```

在 HTML5 中，像这样的 DOCTYPE 声明方式是允许的，字母不区分大小写，引号不区分单引号和双引号。

知识拓展

使用 HTML5 的 DOCTYPE 会触发浏览器以标准兼容模式显示页面。众所周知，网页有多种显示模式，浏览器会根据 DOCTYPE 识别该使用哪种模式，以及使用什么规则验证页面。

1.3.2 字符编码

在 HTML4 中，使用 <meta> 元素的形式指定文件中的字符编码，如下：

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" >
```

在 HTML5 中，可以使用对 <meta> 元素直接追加 charset 属性的方式指定字符编码，如下：

```
<meta charset="utf-8">
```

两种方法都有效，即通过 content 元素的属性来指定，但是不能同时混合使用两种方式。在以前的网站代码中存在的标记方式，在 HTML5 中将被认为是错误的，如下：

```
<meta charset="utf-8" http-equiv="Content-Type" content="text/html; charset=utf-8" >
```

从 HTML5 开始，文件的字符编码推荐使用 UTF-8。

1.3.3 省略引号

属性两边既可以用双引号，也可以用单引号。HTML5 在此基础上做了一些改进，当属性值不包括空字符串、<、>、=、单引号、双引号等字符时，属性值两边的引号可以省略。

以下写法都是合法的：

```
<input type="text">  
<input type='text'>  
<input type=text>
```



1.4 HTML5 元素分类

HTML5 新增了很多元素，也废除了不少元素，根据现有的标准规范，把 HTML5 的元素按等级定义为结构性元素、级块性元素、行内语义性元素和交互性元素四大类。

1.4.1 结构性元素

结构性元素主要负责 Web 上下文结构的定义，确保 HTML 文档的完整性，这类元素包括以下几个。

- ◎ **Section**：在 Web 页面应用中，该元素可以用于区域的章节表述。
- ◎ **Header**：页面头部，注意区别于 head 元素。head 元素中的内容往往是不可见的，header 元素往往在一对 body 元素之中。
- ◎ **Footer**：页面底部，通常会在这里标出网站的一些相关信息。例如，关于我们、法律声明、邮件信息和管理入口等。
- ◎ **Nav**：是专门用于菜单导航、链接导航的元素，是 navigator 的缩写。
- ◎ **Article**：用于表示一篇文章的主体内容，即文字集中显示的区域。

1.4.2 级块性元素

级块性元素主要完成 Web 页面区域的划分，确保内容的有效分隔，这类元素包括以下几个。

- ◎ **Aside**：用以表示注记、贴士、侧栏、摘要、插入的引用等，作为补充主体的内容。
从一个简单页面显示上看，就是侧边栏，可以在左边，也可以在右边。从一个页面的局部看，就是摘要。
- ◎ **Figure**：是对多个元素组合并展示的元素，通常与 figcaption 联合使用。
- ◎ **Code**：表示一段代码块。
- ◎ **Dialog**：用于表达人与人之间的对话，该元素还包括 dt 和 dd 这两个组合元素，它们常常同时使用，dt 用于表示说话者，而 dd 则用来表示说话者说的内容。

1.4.3 行内语义性元素

行内语义性元素主要完成 Web 页面具体内容的引用和表示，是丰富内容展示的基础，这类元素包括以下几个。

- ◎ Meter: 表示特定范围内的数值，可用于工资、数量、百分比等。
- ◎ Time: 表示时间值。
- ◎ Progress: 用来表示进度条，可通过对其 max、min、step 等属性进行控制，完成进度的表示和监视。
- ◎ Video: 视频元素，用于支持和实现视频文件的直接播放，支持缓冲预载和多种视频媒体格式，如 MPEG-4、OGGV 和 WEBM 等。
- ◎ Audio: 音频元素，用于支持和实现音频文件的直接播放，支持缓冲预载和多种音频媒体格式。

1.4.4 交互性元素

交互性元素主要用于功能性的内容表达，会有一定的内容和数据的关联，是各种事件的基础，这类元素包括以下几个。

- ◎ Details: 用来表示一段具体的内容，但是内容默认可能不显示，通过某种手段（如单击），legend 交互才会显示。
- ◎ Datagrid: 用来控制客户端数据与显示，可以由动态脚本及时更新。
- ◎ Menu: 主要用于交互表单。
- ◎ Command: 用来处理命令按钮。



1.5 HTML5 中新增的元素

在 HTML5 中，新增了很多元素。使用这些新的元素，前端设计人员可以更加省力和高效地制作出好看的网页。下面将对所有新增元素的使用方法进行简单介绍。

1.5.1 section 元素

<section> 元素定义文档中的节（section、区段）。比如章节、页眉、页脚或文档中的其他部分。

在 HTML4 中，div 元素与 section 元素具有相同功能，其语法格式如下：

```
<div>...</div>
```

示例代码如下：

```
<div>HTML5+CSS3</div>
```

在 HTML5 中 section 语法格式如下：

```
<section>...</section>
```

示例代码如下：

```
<section>HTML5+CSS3</section>
```

1.5.2 article 元素

`<article>` 元素定义外部的内容。

外部内容可以是来自外部新闻提供者的一篇新文章，或者来自 blog 的文本，或者是来自论坛的文本，抑或是其他外部源内容。

在 HTML4 中，div 元素与 article 元素具有相同功能，其语法格式如下：

```
<div>...</div>
```

示例代码如下：

```
<div>HTML5+CSS 3 </div>
```

在 HTML5 中 article 语法格式如下：

```
< article >...</ article >
```

示例代码如下：

```
< article >HTML5+CSS3</ article >
```

1.5.3 aside 元素

`<aside>` 元素用于表示 article 元素内容之外的，并且与 aside 元素内容相关的一些辅助信息。

在 HTML4 中，div 元素与 aside 元素具有相同功能，其语法格式如下：

```
<div>...</div>
```

示例代码如下：

```
<div>HTML5+CSS3</div>
```

在 HTML5 中 aside 语法格式如下：

```
< aside >...</ aside >
```

示例代码如下：

```
< aside >HTML5+CSS3</ aside >
```

1.5.4 header 元素

<header> 元素表示页面中一个内容区域或整个页面的标题。

在 HTML4 中，div 元素与 header 元素具有相同功能，其语法格式如下：

```
<div>...</div>
```

示例代码如下：

```
<div>HTML5+CSS3</div>
```

在 HTML5 中 header 语法格式如下：

```
<header>...</header>
```

示例代码如下：

```
<header>HTML5+CSS3</header>
```

1.5.5 fhgroup 元素

<fhgroup> 元素用于组合整个页面或页面中一个内容区块的标题。

在 HTML4 中，div 元素与 fhgroup 元素具有相同功能，其语法格式如下：

```
<div>...</div>
```

示例代码如下：

```
<div>HTML5+CSS 3</div>
```

在 HTML5 中 fhgroup 语法格式如下：

```
<fhgroup>...</fhgroup>
```

示例代码如下：

```
<fhgroup>HTML5+CSS3</fhgroup>
```

1.5.6 footer 元素

<footer> 元素用于组合整个页面或页面中一个内容区块的脚注。

在 HTML4 中，div 元素与 footer 元素具有相同功能，其语法格式如下：

```
<div>...</div>
```

示例代码如下：

```
<div>
XXX 大学计算机系 2016 届学员 <br/>
李磊 <br/>
139xxxx2505<br/>
2017-03-12
</div>
```

在 HTML5 中 footer 语法格式如下：

```
<footer>...</footer>
```

示例代码如下：

```
<footer>
XXX 大学计算机系 2016 届学员 <br/>
李磊 <br/>
139xxxx2505<br/>
2017-03-12
</footer>
```

1.5.7 nav 元素

<nav> 元素定义导航链接的部分。

在 HTML4 中，使用 ul 元素替代 nav 元素，其语法格式如下：

```
<ul>...</ul>
```

示例代码如下：

```
<ul>
<li>items01</li>
<li>items02</li>
<li>items03</li>
<li>items04</li>
</ul>
```

在 HTML5 中 nav 语法格式如下：

```
<nav>...</nav>
```

示例代码如下：

```
<nav>
<a href="">items01</a>
<a href="">items02</a>
<a href="">items03</a>
<a href="">items04</a>
</nav>
```

1.5.8 figure 元素

<figure> 元素用于对元素进行组合。

在 HTML4 中，示例代码如下：

```
<dl>
<h1>HTML5</h1>
<p>HTML5 是当今最流行的网络应用技术之一 </p>
</dl>
```

在 HTML5 中 figure 语法格式如下：

```
<figure>
<figcaption>HTML5</figcaption>
<p>HTML5 是当今最流行的网络应用技术之一 </p>
</figure>
```

1.5.9 video 元素

<video> 元素用于定义视频，例如电影片段等。

在 HTML4 中，示例代码如下：

```
<object data="movie.mp4" type="video/mp4">
<param name="" value="movie.mp4">
</object>
```

在 HTML5 中 video 语法格式如下：

```
<video width="320" height="240" controls>
<source src="movie.mp4" type="video/mp4">
<source src="movie.ogg" type="video/ogg">
```

您的浏览器不支持 Video 标签。

</video>

1.5.10 audio 元素

<audio> 元素用于定义音频，例如歌曲片段等。

在 HTML4 中，示例代码如下：

```
<object data="music.mp3" type="application/mp3">  
<param name="" value="music.mp3">  
</object>
```

在 HTML5 中 audio 语法格式如下：

```
<audio controls>  
<source src="music.mp3" type="audio/mp3">  
<source src="music.ogg" type="audio/ogg">  
您的浏览器不支持 audio 标签。  
</audio>
```

1.5.11 embed 元素

<embed> 元素定义嵌入的内容，比如插件。

在 HTML4 中，示例代码如下：

```
<object data="flash.swf" type="application/x-shockwave-flash"></object>
```

在 HTML5 中 embed 语法格式如下：

```
<embed src="helloworld.swf" />
```

1.5.12 mark 元素

<mark> 元素主要突出显示部分文本。

在 HTML4 中，span 元素与 mark 元素具有相同功能，其语法格式如下：

```
<span>...</span>
```

示例代码如下：

```
<span>HTML4 技术的运用 </span>
```

在 HTML5 中 mark 元素的语法格式如下：

```
<mark>...</mark>
```

示例代码如下：

```
<mark>HTML5 技术的运用 </mark>
```

1.5.13 progress 元素

<progress> 元素表示运行中的进程，可以使用 progress 元素显示 JavaScript 中耗费时间函数的进程。

在 HTML5 中 progress 元素的语法格式如下：

```
<progress></progress>
```

progress 元素是 HTML5 中新增的元素，HTML4 中没有相应的元素来表示。

1.5.14 meter 元素

<meter> 元素表示度量衡，仅用于已知最大值和最小值的度量。

在 HTML5 中 meter 元素的语法格式如下：

```
<meter></meter>
```

meter 元素是 HTML5 中新增的元素，HTML4 中没有相应的元素来表示。

1.5.15 time 元素

<time> 元素表示日期和时间。

在 HTML5 中 time 元素的语法格式如下：

```
<time></time>
```

time 元素是 HTML5 中新增的元素，HTML4 中没有相应的元素来表示。

1.5.16 wbr 元素

<wbr> (Word Break Opportunity) 元素规定在文本中的何处适合添加换行符。

在 HTML5 中 wbr 元素的语法格式如下：

```
<p> 尝试缩小浏览器窗口，以下段落的 "XMLHttpRequest" 单词会被分行： </p>  
<p> 学习 AJAX，您必须熟悉 <wbr>Http<wbr>Request 对象。 </p>
```

`<p><b>` 注意: `</b>` IE 浏览器不支持 `wbr` 标签。 `</p>`

`wbr` 元素是 HTML5 中新增的元素, HTML4 中没有相应的元素来表示。

1.5.17 canvas 元素

`<canvas>` 元素用于定义图形, 如图表和其他图像, 必须使用脚本来绘制图形。

在 HTML5 中 `canvas` 元素的语法格式如下:

```
<canvas id="myCanvas" width="500" height="500"></canvas>
```

`canvas` 元素是 HTML5 中新增的元素, HTML4 中没有相应的元素来表示。

1.5.18 command 元素

`<command>` 元素可以定义用户可能调用的命令 (比如单选按钮、复选框)。

在 HTML5 中 `command` 元素的语法格式如下:

```
<command onclick="cut()" label="cut"/>
```

`command` 元素是 HTML5 中新增的元素, HTML4 中没有相应的元素来表示。

1.5.19 datalist 元素

`<datalist>` 元素规定了 `<input>` 元素可能的选项列表。

`datalist` 元素通常与 `input` 元素配合使用。

在 HTML5 中 `datalist` 元素的语法格式如下:

```
<input list="browsers">
<datalist id="browsers">
  <option value="Internet Explorer">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

`datalist` 元素是 HTML5 中新增的元素, HTML4 中没有相应的元素来表示。

1.5.20 details 元素

`<details>` 元素规定了用户可见的或者隐藏的需求的补充细节。

`<details>` 元素是用于供用户开启关闭的交互式控件。任何形式的内容都能被放在

<details> 元素里边。

<details> 元素的内容对用户是不可见的，除非设置了 open 属性。

在 HTML5 中 details 元素的语法格式如下：

```
<details>
<summary>Copyright 1999-2011.</summary>
<p> - by Refsnes Data. All Rights Reserved.</p>
<p>All content and graphics on this web site are the property of the      company Refsnes </p>
</details>
```

details 元素是 HTML5 中新增的元素，HTML4 中没有相应的元素来表示。

1.5.21 datagrid 元素

<datagrid> 元素表示可选数据的列表，它以树形列表的形式显示。

在 HTML5 中 datagrid 元素的语法格式如下：

```
<datagrid>...</datagrid>
```

datagrid 元素是 HTML5 中新增的元素，HTML4 中没有相应的元素来表示。

1.5.22 keygen 元素

<keygen> 元素用于生成密钥。

在 HTML5 中 keygen 元素的语法格式如下：

```
< keygen name="security">
```

keygen 元素是 HTML5 中新增的元素，HTML4 中没有相应的元素来表示。

1.5.23 output 元素

<output> 元素表示不同类型的输出，例如脚本的输出。

在 HTML5 中 output 元素的示例代码如下：

```
<output></output>
```

在 HTML4 中的示例代码如下：

```
<span></span>
```

1.5.24 source 元素

<source> 元素用于为媒介元素定义媒介资源。

在 HTML5 中 source 元素的示例代码如下：

```
<source type="" src="" />
```

在 HTML4 中的示例代码如下：

```
<param>
```

1.5.25 menu 元素

<menu> 元素表示菜单列表，当希望列出表单控件时使用该标签。

在 HTML5 中 menu 元素的示例代码如下：

```
<menu>
<li>items01</li>
<li>items02</li>
</menu>
```



1.6 课堂练习

通过对本章内容的学习，大体掌握了 HTML5 的基础知识，接下来完成一个简单的练习，其效果如图 1-3 所示。

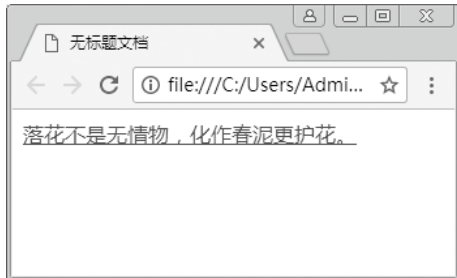


图 1-3

操作提示：

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title> 无标题文档 </title>
</head>
<body>
</body>
</html>
```

此段代码是 HTML5 的基础代码，图 1-3 的效果就是在此代码的基础上完成的。



强化训练

以上小节讲解了 HTML5 中新增的知识点，其中有新增的结构性元素，本小节为大家准备了一个强化练习，运行效果如图 1-4 所示。



图 1-4

完整的代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title> 课堂练习 </title>
</head>
<body>
<!-- 通常不推荐没有标题内容使用 section 元素 -->
<!-- 不要与 article 元素混淆 -->
<section>
  <h1> 香蕉 </h1>
  <p> 这是一种水果 </p>
</section>
<article>
  <h1> 苹果 </h1>
  <p> 这是一种水果 </p>
<section>
```

```
<h2> 红富士 </h2>
<p> 这个苹果很好吃！！ </p>
</section>
<section>
  <h2> 洛川苹果 </h2>
  <p> 这个是陕西盛产的苹果！！ </p>
</section>
</article>
<!-- 注意 section 和 article 的区别 -->
<section>
  <h1> 有很多水果 </h1>
  <article>
    <h2> 香蕉 </h2>
    <p> 食用香蕉的好处 </p>
  </article>
  <article>
    <h2> 苹果 </h2>
    <p> 苹果含有维生素 </p>
  </article>
  <article>
    <h2> 西瓜 </h2>
    <p> 西瓜很好吃，其中含有 ...</p>
  </article>
</section>
</body>
</html>
```