

第 1 章 页面描述语言

1.1 HTML 文档

超文本标记语言（Hyper Text Markup Language, HTML）是一种可以通过一系列标签将文字、图片等多种资源组合在一起的标记语言。浏览器打开的 Web 网页就是 HTML 文档。

自 HTML 诞生后已经更新了多个版本，如表 1.1 所示。

表 1.1 HTML 的版本

版本	发布时间	版本	发布时间
HTML	1991 年	HTML 4.01	1999 年
HTML+	1993 年	XHTML 1.0	2000 年
HTML 2.0	1995 年	HTML5	2012 年
HTML 3.2	1997 年	XHTML5	2013 年

学习完本章后可以使用 HTML 来创建简单的 Web 站点。

1.1.1 HTML 文档的创建与查看

1. HTML 文档的创建

HTML 编辑器是一个编写 HTML 的工具。使用 HTML 编辑器时，可以编辑主题、索引、自定义窗口和选择添加搜索页面。平常可以用记事本（Notepad）等文本编辑器来学习。选择一个简单的编辑器来学习是一个非常好的方法。创建 HTML 文档的参考步骤如下：

步骤 1：启动 Notepad（Windows 系统中），方法是单击“开始”→“所有程序”→“附件”→Notepad。

步骤 2：使用 Notepad 编辑 HTML，在 Notepad 中输入如下代码：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>HTML 基础教程</title>
</head>
<body>
  <h1>我的第一个标题</h1>
  <p>我的第一个段落</p>
</body>
</html>
```

步骤 3：保存 HTML 文档。在 Notepad 中选择“文件”→“另存为”命令，以.htm 或.html

扩展名保存，将该文档保存在常用的文件夹中。

步骤 4: 在浏览器中运行 HTML 文档。

启动浏览器，选择“文件”→“打开文件”命令，打开需要运行的 HTML 文档，显示结果如图 1.1 所示。

2. 查看 HTML 文档代码

如何查看 HTML 文档的源代码？如果对一个网页的实现方式感兴趣，可以用任何浏览器打开此网页，按快捷键 Ctrl+U 或 F12 查看源代码或者右击后在弹出的快捷菜单中选择“查看源”命令，均会打开一个窗口，其中包含页面的 HTML 代码。

通过右击查看代码如图 1.2 所示（Microsoft Edge 浏览器）。

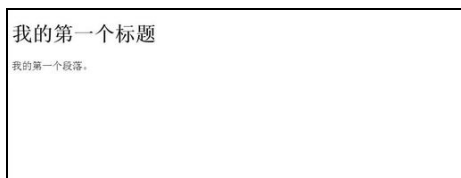


图 1.1 显示结果



图 1.2 通过右击查看代码

通过快捷键 F12 查看代码如图 1.3 所示（Chrome 浏览器）。

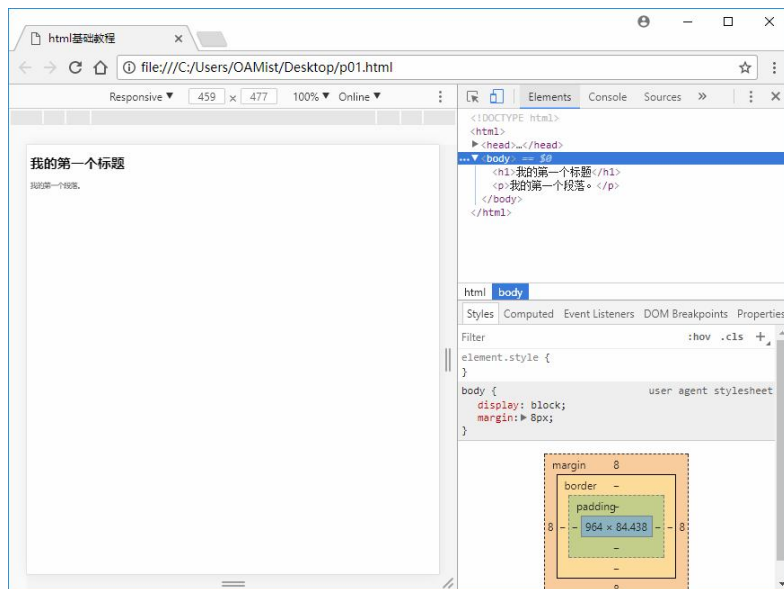


图 1.3 通过快捷键查看代码

1.1.2 HTML 文档结构

可以通过下面这个 HTML 示例来建立一个简单的 HTML 页面，以此来了解一下 HTML

文档的结构。

```
<!DOCTYPE html>
<html>
<body>
  <head>
    <meta charset="utf-8">
    <title>HTML 基础教程</title>
  </head>
  <h1>我的第一个标题</h1>
  <p>我的第一个段落</p>
</body>
</html>
```

其中，DOCTYPE 声明了文档类型。

<html>标签描述了文档类型，该标签的结束标志为</html>。

<body>标签定义了文档的主体，即网页的可视化内容，该标签的结束标志为</body>。

<h1>标签作为一个标题使用，该标签的结束标志为</h1>。

<p>标签作为一个段落显示，该标签的结束标志为</p>。

Web 浏览器（Chrome、Internet Explorer、Firefox 和 Safari）用于读取 HTML 文档，并将其作为网页显示。

浏览器并不是直接显示所有的标签，而是根据所输入的标签来显示内容。

下面对文档声明、头部元素、文档主体 3 个部分分别进行论述。

1. 文档声明

<!DOCTYPE>是标准通用标记语言的文档类型声明，有助于在浏览器中正确地显示网页。

因声明不区分大小写，所以下面每种方式都是正确的。

```
<!DOCTYPE html>
<!DOCTYPE HTML>
<!doctype html>
<!Doctype Html>
```

2. 头部元素

HTML 的头部元素包含文档的概要信息，也被称为元信息，如表 1.2 所示。

表 1.2 HTML 的头部元素

标签	描述
<head>	定义了文档的信息
<title>	定义了文档的标题
<base>	定义了页面链接标签的默认链接地址
<link>	定义了一个文档和外部资源之间的关系
<meta>	定义了 HTML 文档中的元数据
<script>	定义了客户端的脚本文件
<style>	定义了 HTML 文档的样式文件

(1) <head>标签。<head>标签中包含了所有的头部标签元素，主要包括 title、base、meta

等，还可以在其中插入样式文件（CSS）。

（2）<title>标签。<title>标签定义了 HTML 文档的标题：浏览器中工具栏的标题、收藏网页时显示设定的标题、在浏览器的搜索结果中显示的标题。

<title>在 HTML/XHTML 文档中一定是存在的。

示例如下：

```
<head>
  <title>在此处填文档的标题 </title>
</head>
```

（3）<base>标签。<base>标签描述了基本的链接地址/链接目标，该标签作为 HTML 文档中所有的链接标签的默认链接，示例如下：

```
<head>
  <base href="//www.w3cschool.cn/images/" target="_blank" />
</head>
```

提示：在 HTML 中，<base>标签中没有结束标签；在 XHTML 中，必须正确地关闭<base>标签。

（4）<link>标签。<link>标签定义了文档与外部资源间的相互关系，经常用于链接到样式表，示例如下：

```
<head>
  <link rel="stylesheet" type="text/css" href="mystyle.css" />
</head>
```

（5）<style>标签。<style>标签定义了 HTML 文档的样式文件，可以指定样式文件来渲染 HTML 文档，示例如下：

```
<head>
  <style type="text/css">
    body {
      background-color: yellow
    }

    p {
      color: blue
    }
  </style>
</head>
```

（6）<script>标签。<script>标签用于加载脚本文件，如 JavaScript，本书将在后续章节对其进行介绍。

（7）<meta>标签。<meta>标签定义了一些基本的元数据。在<meta>标签中提供了元数据。元数据不会在页面上显示，但会被浏览器解析。

<meta>标签通常用于定义网页的描述内容、关键词、文件的最后修改时间、作者和其他元数据。

元数据可以应用于浏览器（如何显示内容或重新加载页面）、搜索引擎（关键词）或其他 Web 服务，示例如下：

1) 为搜索引擎定义关键词。

```
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">
```

2) 为网页定义描述内容。

```
<meta name="description" content="Free Web tutorials on HTML and CSS">
```

3) 定义网页作者。

```
<meta name="author" content="Hege Refsnes">
```

4) 每隔 30 秒刷新当前页面。

```
<meta http-equiv="refresh" content="30">
```

5) 中文编码。在 HTML 页面的头部声明字符为 UTF-8 可以解决在浏览器中出现中文乱码的问题。

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

3. 文档主体

body 元素在文档的主体中定义，包含文档中的所有内容，如文本、超链接、图像、表格、列表等。

1.2 HTML 语法基础

可以在 HTML 代码中插入注释，以提高其可读性，即使其更易于理解。浏览器忽略注释且不显示它。

HTML 注释可以帮助网页设计者更好地理解代码。

注释写法如下：

```
<!-- 这是一个注释 -->
```

注意：感叹号应该紧跟在左括号后面，而不是在右括号之前。合理使用注释有助于将来的代码编辑。

1.2.1 文本

1. 标题

在 HTML 文档中，标题可以用来表示文档的结构。设置合理的标题有助于用户浏览网页。标题由 `<h1>`~`<h6>` 标签定义。其中，`<h1>` 定义最大的标题，`<h6>` 定义最小的标题。例如：

```
<h1>这是最大的标题。</h1>
```

```
<h2>这是第二大标题。</h2>
```

```
<h3>这是第三大标题。</h3>
```

注意：浏览器会自动在标题前后添加空行。默认情况下，HTML 会在块级元素（如段落和标题元素）前后自动添加一个额外的空行。

不能只使用标题来生成粗体或大号的文本。搜索引擎使用标题来索引网页的结构和内容，用户可以通过标题快速浏览网页，因此使用标题来表示文档结构非常重要。

h1 应该用作主标题（最重要），其次是 h2（第二重要），然后是 h3，依次类推。

2. 段落

文档可以被 HTML 分割为若干段落。段落是通过 `<p>``</p>` 标签来定义的。例如：

```
<p>这是一个段落</p>
```

```
<p>这是另一个段落</p>
```

注意：浏览器会自动在段落的前后添加空行。

虽然忘记使用结束标签，大多数浏览器也会正确显示 HTML，但是忘记使用结束标签可能会导致意外的错误和结果。

在未来的 HTML 版本中，不允许省略结束标签。

3. 链接

HTML 可以使用超链接从一个页面跳转到另一个页面，或者跳到该页面上的指定位置，或者跳到另一个页面上的某个位置。

(1) 超链接（链接）。HTML 使用 `<a>` 标签来设置超链接，超链接是可以将一个页面指向到另一个目的端的链接。

语法：`链接`

举例：`<p>百度 </p>`

当将鼠标指针移到网页中的链接上时，默认的超链接样式会将鼠标指针更改为一个手形图标。

默认情况下，链接在浏览器中显示如下：

- 尚未访问的链接以蓝色字体显示并带有下划线。
- 访问过的链接以紫色字体显示并带有下划线。
- 当点击一个链接时，链接会以红色字体显示并带有下划线。

这可以通过定义样式来更改。

(2) 空链接。点击空链接后，鼠标指针变为手形图标，但点击后仍停留在当前页面上。临时添加空链接，是为了更好地看到最终效果。

语法：`链接文字`

其中，“#”表示空链接。

(3) id 属性。id 属性可用于在一个 HTML 文档中创建书签标记，用于跳转到此页或其他页的某个位置。

在 HTML 文档中插入 id：

```
<a id="tips">有用的提示部分 </a>
```

在 HTML 文档中创建一个链接到“有用的提示部分 (id="tips")”：

```
<a href="#tips">访问有用的提示部分</a>
```

或者从另一个页面创建一个链接到“有用的提示 (id="tips") 部分”：

```
<a href="//www.w3cschool.cn/html_links.html#tips">访问有用的提示部分</a>
```

提示：书签在 HTML 文档中是不显示的，所以对于读者来说是隐藏的。

超链接可以是一个字、一个词、一组词或一个图像，点击这些超链接可以跳转到新文档或当前文档的某个部分。

示例代码：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>
```

```

<p>
  图片超链接:
  <a href="http://www.baidu.com">
    
  </a>
</p>

<p>
  无边框的图片链接:
  <a href=" http://www.baidu.com ">
    
  </a>
</p>
<p>是否想要跳出此框架? </p>
<a href="http://www.baidu.com/" target="_top">点击这里! </a>
</body>
</html>

```

上面代码中用到的图片 `result.png` 请提前准备并放到和当前 HTML 文档在同一个路径下的 `images` 文件夹里。页面内容如图 1.4 所示。



图 1.4 页面内容

注意：资源的路径与网页文档所在的路径可能不一致。

(4) 链接的 `target` 属性。使用 `target` 属性可以定义链接文档的显示位置（在新窗口或现有窗口中打开）。

以下示例代码会在新窗口中打开文档。

```
<a href="//www. baidu.com /" target="_blank">访问百度! </a>
```

提示：默认的被链接文档会在原有的窗口中打开，若将 `target` 属性设置为 “`_blank`”，则文档会在新窗口中打开。`target` 属性值如表 1.3 所示。

表 1.3 `target` 属性值

值	描述
<code>_blank</code>	在新窗口/标签页中打开
<code>_self</code>	在同一框架中打开（默认）
<code>_parent</code>	在父框架中打开
<code>_top</code>	在整个窗口中打开
<code>framename</code>	在指定的框架中打开

4. 换行和水平线

如果希望不产生一个新段落进行换行（新行），可使用
标签。

在 HTML 中，
标签定义为一个换行符，它可以理解为简单地输入一个空行，而不是用来对内容进行分段。

标签是一个空的 HTML 元素，因为关闭了标签没有任何意义，所以它就没有结束标签。

<hr />标签用于在 HTML 页面中创建水平线。

hr 元素可以用于分隔内容，由该元素生成的水平线将文档直观地分为多个部分，示例如下：

```
<p>段落 1</p>
<br />
<p>段落 2</p>
<br />
<p>段落 3</p>
```

5. 文本格式化

HTML 中存在一些格式化文本的标签，它们可以被直接使用，而不用对样式进行调整，如、、<i>、<small>、、<sub>、<sup>、<ins>、和<pre>等，示例如下：

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>标题</title>
</head>
<body>

  <b>加粗文本</b><br><br>
  <i>斜体文本</i><br><br>
  <code>表示计算机源代码</code><br><br>
  这是 <sub> 下标</sub> 和 <sup> 上标</sup>

</body>
</html>
```

HTML 使用标签和<i>对输出的文本进行格式化，这些 HTML 标签被称为格式化标签。

通常标签是替换加粗标签来使用的，是替换<i>标签来使用的。然而这些标签的含义是不同的：和<i>分别定义粗体和斜体文本，或意味着要呈现的文本是极其重要的，所以要突出显示。

主流的浏览器都能渲染各种效果的字体。不过，未来浏览器可能会支持更好的渲染效果。文本格式化示例：

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

```

<body>
  <b>加粗文本</b>
  <br />
  <strong>这个文本是加粗的</strong>
  <br />
  <big>放大文本字体</big>
  <br />
  <em>强调文本</em>
  <br />
  <i>设置文本字体为斜体</i>
</body>
</html>

```

以下示例演示如何使用<pre>标签对空行和空格进行控制。

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>

<pre>
此例演示如何使用<pre>标签
对空行和   空格
进行控制
</pre>

</body>
</html>

```

以下示例演示如何使用标签和<ins>标签来删除文本和插入文本。

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>

  <p>最喜欢的颜色是 <del>蓝色</del> <ins>红色</ins>!</p>

</body>
</html>

```

6. 计算机代码元素

通常 HTML 使用可变的字母尺寸和可变的字母间距，但是在显示计算机代码示例时无须如此，可以使用<kbd>、<samp>、<code>、<var>等标签固定字母尺寸和间距，如表 1.4 所示。

表 1.4 HTML 计算机代码元素

标签	描述
<code><code></code>	定义计算机代码文本
<code><kbd></code>	定义键盘文本
<code><samp></code>	定义计算机输出示例
<code><var></code>	定义变量

(1) HTML 键盘格式。HTML `<kbd>` 标签定义键盘输入，例如：

```
<p>To open a file, select:</p>
<p><kbd>File | Open...</kbd></p>
```

(2) HTML 样本格式。HTML `<samp>` 标签定义计算机输出示例：

```
<samp>
demo.example.com login: Apr 12 09:10:17
Linux 2.6.10-grsec+gg3+e+fhs6b+nfs+gr0501+++p3+c4a+gr2b-reslog-v6.189
</samp>
```

(3) HTML 代码格式。HTML `<code>` 标签定义计算机代码文本示例：

```
<code>
var person = { firstName:"Bill", lastName:"Gates", age:50, eyeColor:"blue" }
</code>
```

`<code>` 标签不保留多余的空格和换行，例如：

```
<p>Coding Example:</p>
```

```
<code>
var person = {
  firstName:"名字",
  lastName:"姓氏",
  age:年龄,
}
</code>
```

如果需要保留一定的空格和换行，必须使用 `<pre>` 标签包围代码，例如：

```
<p>Coding Example:</p>
```

```
<code>
<pre>
var person = {
  firstName:"名字",
  lastName:"姓氏",
  age:年龄,
}
</pre>
</code>
```

(4) HTML 变量格式化。HTML `<var>` 标签用于定义数学变量，例如：

```
<p>Einstein wrote:</p>
<p><var>E = m c<sup>2</sup></var></p>
```

7. 引用

HTML 引文、引用和定义元素如表 1.5 所示。

表 1.5 HTML 引文、引用和定义元素

标签	描述
<abbr>	指缩写或首字母缩略语
<address>	指作者的联系地址信息
<bdo>	指文本方向
<blockquote>	指从其他资源引用的节
<dfn>	指项目或缩略词的定义
<q>	指行内引用
<cite>	指著作的标题

(1) 用于短引用的<q>。浏览器通常会为<q>元素包围引号，就是将<q>与</q>中间的内容自动加上一对引号。示例：

```
<p>WWF 的目标是：<q>构建人与自然和谐共存的世界。</q></p>
```

(2) 用于长引用的<blockquote>。

<blockquote>元素定义被引用的节。浏览器通常会对<blockquote>元素进行缩进处理。示例：

```
<p>以下内容引用自 WWF 的网站：</p>
<blockquote cite="http://www.worldwildlife.org/who/index.html">
  五十年来，WWF 一直致力于保护自然的未来。
  世界领先的环保组织，WWF 工作于 100 个国家，
  并得到美国一百二十万会员及全球近五百万会员的支持。
</blockquote>
```

(3) 用于缩略词的<abbr>。<abbr>元素定义缩写或首字母缩略语。

对缩写进行标记能够为浏览器、翻译系统和搜索引擎提供有用的信息。示例：

```
<p><abbr title="World Health Organization">WHO</abbr> 成立于 1948 年。</p>
```

(4) 用于定义的<dfn>。<dfn>元素定义项目或缩写的定义。

<dfn>的用法，按照 HTML5 标准中的描述，有点复杂，具体如下：

1) 若设置了<dfn>元素的 title 属性，则定义项目，比如定义项目 WHO：

```
<p><dfn title="World Health Organization">WHO</dfn> 成立于 1948 年。</p>
```

2) 若<dfn>元素包含具有标题的<abbr>元素，则 title 定义项目。示例：

```
<p>
  <dfn><abbr title="World Health Organization">WHO</abbr></dfn> 成立于 1948 年。
</p>
```

3) <dfn>文本内容就是项目，并且父元素包含定义。示例：

```
<p><dfn>WHO</dfn> World Health Organization 成立于 1948 年。</p>
```

如果希望简而化之，请使用第一条，或者使用<abbr>代替。

(5) 用于定义联系信息的<address>。<address>元素定义文档或文章的联系信息（作者/拥有者），此元素通常以斜体显示，大多数浏览器会在此元素前后添加换行。示例：

```
<address>
  Written by Donald Duck.<br>
  Visit us at:<br>
  Example.com<br>
  Box 564, Disneyland<br>
  USA
</address>
```

(6) 用于定义著作标题的<cite>。<cite>元素定义著作的标题。浏览器通常会以斜体显示<cite>元素。示例：

```
<p><cite>The Scream</cite> by Edward Munch. Painted in 1893.</p>
```

(7) 用于双向重写的<bdo>。<bdo>元素定义双流向覆盖，用于覆盖当前文本方向。示例：

```
<bdo dir="rtl">My Bolg</bdo> //对应为 gloB yM
```

1.2.2 表格与布局

1. 表格

表格由<table>标签定义。每个表格都有几行（由<tr>标签定义），每一行被划分为多个单元格（由<td>标签定义）。字母 td 表示表数据，即数据单元格的内容。数据单元格可以包含文本、图片、列表、段落、表单、水平线、表格等。表格标签如表 1.6 所示。

表 1.6 表格标签

标签	描述
<table>	指表格
<th>	指表头
<tr>	指行
<td>	指表格单元格
<caption>	指表格标题
<colgroup>	指表格列的组
<col>	指表格列的属性
<thead>	指表格的页眉
<tbody>	指表格的主体
<tfoot>	指表格的页脚

<table>...</table>：定义表格。

<th>...</th>：定义表格的标题栏（文字加粗）。

<tr>...</tr>：定义表格的行。

<td>...</td>：定义表格的单元格。

示例：

```
<table border="1">
```

```

<tr><th>Header 1</th><th>Header 2</th></tr>
<tr><td>row 1, cell 1</td><td>row 1, cell 2</td></tr>
<tr><td>row 2, cell 1</td><td>row 2, cell 2</td></tr>
</table>

```

上面示例代码对应显示结果如图 1.5 所示。

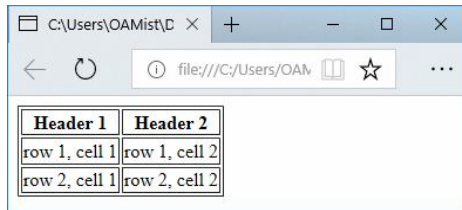


图 1.5 表格示例



表格单元格如何实现跨行或跨列

表格单元格如何实现跨行或跨列呢？请扫描右侧二维码查看示例代码。

2. 列表

HTML 支持无序列表、有序列表和自定义列表。HTML 列表标签如表 1.7 所示。

表 1.7 HTML 列表标签

标签	描述
	定义了有序列表
	定义了无序列表
	定义了列表项
<dl>	定义了自定义列表
<dt>	自定义了列表项
<dd>	定义了自定义列表项的描述

(1) 无序列表。无序列表是用粗点（通常是小黑圆圈）标记的项目列表，适用于成员之间没有级别顺序关系的情况。

无序列表使用标签定义，示例如下：

```

<ul>
  <li>第一条</li>
  <li>第二条</li>
</ul>

```

上面示例代码对应显示结果如图 1.6 所示。

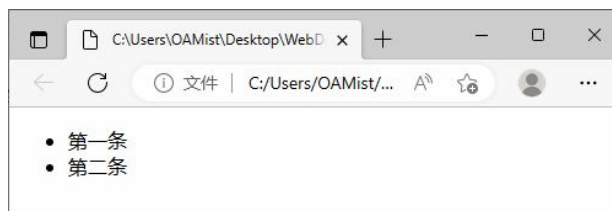


图 1.6 无序列表

(2) 有序列表。有序列表是用数字标记的项目列表，使用标签定义，每个列表项都以标签开始。

有序列表适用于项目之间存在顺序关系的情况，列表项用数字标记，示例如下：

```
<ol>
  <li>第一条</li>
  <li>第二条</li>
</ol>
```

上面示例代码对应显示结果如图 1.7 所示。

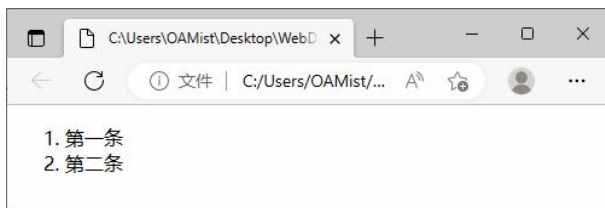


图 1.7 有序列表

(3) 自定义列表。自定义列表不仅是包含列表项的列表，还是列表项及其描述的组合。

自定义列表以<dl>标签开始，每个自定义列表项以<dt>开头。每个自定义列表项的描述都以<dd>开头。

自定义列表中的术语名称可以对应多个定义，也可以对应同一个定义。如果只有术语名称或定义是可用的，那么<dt>和<dd>的数量是无限的，且对应关系也是无限的。自定义列表示例代码如下：

```
<dl>
  <dt>第一条</dt>
  <dd>- 首先</dd>
  <dt>第二条</dt>
  <dd>- 其次</dd>
</dl>
```

上面示例代码对应显示结果如图 1.8 所示。

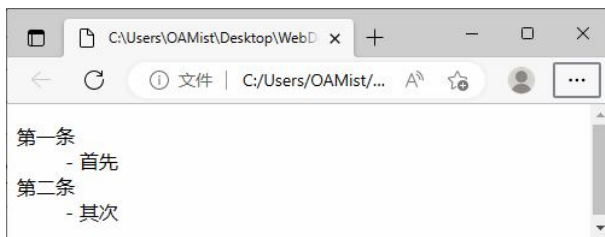


图 1.8 自定义列表

提示：列表项内部可以使用段落、换行符、图片、链接和其他列表等。

3. 块级元素、内联元素与布局

HTML 分组标签如表 1.8 所示，可以通过<div>和把 HTML 元素组合起来。