



HTML 基础

3.1 HTML 概述

超文本标记语言(HyperText Mark-up Language,HTML)是目前网络上应用最为广泛的语言,也是构成网页文档的主要语言。

一个 HTML 文档是由一系列的 HTML 标记和 HTML 元素组成的,标记名不区分大小写。HTML 超本文档分文档头和文档体两部分,在文档头里,对这个文档进行了一些必要的定义,文档体中才是要显示的各种文档信息。

下面是一个最简单的 HTML 文档示例。

例程 3-1 sample.html: 欢迎页面。

```
<html><!-- 开始标记 -->
<head>
  <title>一个简单的示例</title> } 文档头
</head>
<body>
  <h1><font color = red>欢迎光临!!!</font></h1> } 文档体
</body>
</html><!-- 结尾标记 -->
```

`<html>`、`</html>`在文档的最外层,标记中间包含文本和其他 HTML 标记。事实上,现在常用的 Web 浏览器都可以自动识别 HTML 文档,并不要求有 `<html>` 标记,也不对该标记进行任何操作,为了使 HTML 文档看起来更规范,建议不要省略这个标记。

`<head>`、`</head>`是 HTML 文档头标记,在浏览器窗口中,头部信息是不显示在正文中的,在此标记中可以插入其他标记,用于说明文件的标题和整个文件的一些公共属性。`<title>`、`</title>`是嵌套在`<head>`头部标记中的,标记之间的文本是文档标题,它显示在浏览器窗口的标题栏。

`<body>`、`</body>`是 HTML 文档体标记,标记之间的文本是正文,是在浏览器要显示的页面内容。`<h1>`、`</h1>`标记表示标题,标记间包含的内容会以标题形式显示;`<font>`、`</font>`标记表示字体,用于设置标记间内容显示的字体和颜色。

例程 3-1 运行效果如图 3-1 所示。



图 3-1 运行效果

使用 HTML 编写出来的文件扩展名有两种选择：第一扩展名命名为 .htm，第二扩展名命名为 .html。这两种扩展名没有本质的差别，在实践时选择其一即可。

3.2 HTML 常用标记

3.2.1 文档头

文档头 `<head>`、`</head>` 标注中可嵌套的常用标记如下：

1. 文档标题标记

`<title>` 标记用于定义文档的标题。浏览器会以特殊的方式来使用标题，并且通常把它放置在浏览器窗口的标题栏或状态栏上。同样，当把文档加入用户的链接列表或者收藏夹或书签列表时，标题将成为该文档链接的默认名称。例如：

```
<head>
<title>这是一个测试网站</title>
</head>
```

2. 链接标记

`<link>` 标记用于定义文档与外部资源的关系，最常见的用途是链接样式表。例如：

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

属性 `rel` 定义当前文档与被链接文档之间的关系，`href` 定义被链接文档的位置，`type` 规定被链接文档的 MIME 类型。

3. 样式标记

`<style>` 标记用于为 HTML 文档定义样式信息。例如：

```
<html>
<head>
<style type = "text/css">
```

```
h1 {color:red}
</style>
</head>
<body>
<h1>Header 1 </h1>
</body>
</html>
```

在<style>中,可以规定在浏览器中如何呈现 HTML 文档。type 属性是必需的,定义 style 元素的内容,唯一可能的值是 text/css。

4. 基路径标记

<base>标记用于为页面上的所有链接规定默认地址或默认目标。通常,浏览器会从当前文档的 URL 中提取相应的元素来填写相对 URL 中的空白。使用 <base> 标记可以改变这一点,浏览器随后将不再使用当前文档的 URL,而使用指定的基本 URL 来解析所有的相对 URL,其中包括 <a>、、<link>、<form> 标记中的 URL。例如:

```
<head>
<base href = "http://localhost:8080/web/" />
<base target = "_blank" />
</head>
<body>
<img src = "logo.jpg" />
<a href = "index.html">首页</a>
</body>
```

在<base>中可以指定两个属性,属性 href 规定页面中所有相对链接的基准 URL,属性 target 规定在何处打开页面中所有的链接,可选值有 _blank、_top、_parent 和 _self,如表 3-1 所示。图片 logo.jpg 的访问路径为 http://localhost:8080/web/logo.jpg,点击链接“首页”会在一个新的浏览器窗口中显示。

表 3-1 target 属性值说明

属 性 值	说 明
_blank	在新窗口打开
_top	在浏览器的整个窗口中打开,忽略任何框架
_parent	在上一级窗口中打开,常用于分帧的框架页面
_self	在同一帧或窗口打开,也是该属性的默认值

5. 脚本标记

<script>标记用于定义客户端脚本,如 JavaScript。script 元素既可以包含脚本语句,也可以通过 src 属性指向外部脚本文件。例如:

```
<script type = "text/javascript">
document.write("Hello World!")
</script>
```

或

```
<script type = "text/javascript" src = "sample.js"></script>
```

属性 type 规定脚本的 MIME 类型,必须指定。

6. 网页信息标记

<meta>标记可提供有关页面的元信息(meta-information),如针对搜索引擎和更新频度的描述和关键词。在<meta>中有两个属性,name 和 http-equiv:

属性 name 主要用于描述网页,与之对应的属性值为 content,content 中的内容主要是便于搜索引擎机器人查找信息和分类信息用的。meta 标记的 name 属性语法格式是<meta name = "参数" content = "具体的参数值">。其中,参数主要有 keywords、description、robots 和 author:

keywords: 关键字,即告诉搜索引擎,这个网页的主题是什么,例如:

```
<meta name = "keywords" content = "test, sample">
```

description: 简介,告诉搜索引擎,这个网页的大致内容,例如:

```
<meta name = "description" content = "This is my page">
```

robots: 用来告诉搜索机器人哪些页面需要索引,哪些页面不需要索引。content 的参数有 all、none、index、noindex、follow、nofollow,默认是 all,例如:

```
<meta name = "robots" content = "none">
```

author: 作者,告诉搜索引擎,这个页面的作者是谁,例如:

```
<meta name = "author" content = "david. yang">
```

属性 http-equiv,顾名思义,相当于 http 的文件头作用,它可以向浏览器传回一些有用的信息,以帮助正确和精确地显示网页内容,与之对应的属性值为 content,content 中的内容其实就是各个参数的变量值。http-equiv 属性语法格式是<meta http-equiv = "参数" content = "参数变量值">,其中参数主要有 Expires、Pragma、Refresh、Set-Cookie、Window-target 和 content-Type:

Expires: 可以用于设定网页的到期时间。一旦网页过期,必须到服务器上重新传输,例如:

```
<meta http - equiv = "Expires" Content = "Fri, 15 Jun 2007 01:13:13 GMT">
```

```
<meta http - equiv = "Expires" Content = "0">
```

时间格式必须为 GMT 标准时间,Content 设置为 0,则代表该网页永不过期。

Pragma: 禁止浏览器从本地计算机的缓存中访问页面内容,例如:

```
<meta http - equiv = "Pragma" Content = "no - cach">
```

Refresh: 自动刷新并指向新页面,例如:

```
<meta http - equiv = "Refresh" Content = "60">
```

Content = "60"的意思是说,60 秒后,页面刷新

```
<meta http - equiv = "Refresh" Content = "10; Url = http://www.czbin.cn">
```

Content = "10" 的意思是说,10 秒后,网页将自动跳转。

Set-Cookie: 设置 cookie,例如:

```
<meta http-equiv = "Set - Cookie"
Content = "cookievalue = xxx; expires = Wednesday, 21 - Oct - 98 16:14:21 GMT; path = /">
```

Window-target: 强制页面在当前窗口以独立页面显示,例如:

```
<meta http-equiv = "Window - target" Content = "_top">
```

这个属性是用来防止别人在框架里调用你的页面。

content-Type: 设定页面使用的字符集,例如:

```
<meta http-equiv = "Content - Type" Content = "text/html; Charset = gb2312">
```

告诉浏览器,该页面为 HTML 类文档,并且用 gb2312(中文)作为默认字符。

3.2.2 图片

标记用于定义一幅图像。语法格式为:

```
<img src = "" alt = "" width = "" height = "" />
```

属性解释如下:

- src 属性: 指定图像的 URL。
- alt 属性: 指定图像找不到时显示的替代文本。
- width 属性: 指定图像的宽度。
- height 属性: 指定图像的高度。

例如,显示网站图标。

```
<img src = "logo.jpg" alt = "这是一个测试网站" width = 300 height = 200/>
```

3.2.3 超链接

<a>标记用于定义一个链接。超链接是 HTML 中最重要的特性之一,是网站的灵魂,超链接实现文档的非线性组织。它可以从一个页面跳转到其他页面、图像或服务器。例如:

```
<a href = "" target = "" >xxx</a>
```

属性 href 指定链接的目标 URL,属性 target 指定打开链接的目标窗口,该属性的默认值为_self,即在原窗口打开,该属性的属性值参见表 3-1。

href 属性的值在不同情况下可以有不同的写法,当要链接到的目标文件是同一个站点下的某个文件,通常会给出目标文件相对于当前文件的相对路径或根路径。当要链接到的目标文件是其他站点的文件时,通常会给出绝对路径。下面将分别具体介绍相对路径、根路径、绝对路径。

1. 相对路径

相对路径是以当前文件所在的路径为基准,进行查找。如果目标文件和当前文件位于相同目录下,则该路径仅需要给出目标文件名。例如,目标文件名为 targetFile. html,该文件和当前文件位于同一目录下,作为 href 属性值写作 targetFile. html。如果目标文件位于当前文件的下一级文件夹中,则该路径为: 下一级文件夹名/目标文件名。例如: 目标文件名为 targetFile. html,该文件位于当前文件同一目录 abc 下,作为 href 属性值写作 abc/targetFile. html。如果目标文件位于当前文件的上一级目录时,则该路径为“../文件名”。例如,目标文件名为 targetFile. html,该文件位于当前文件上一级目录时,作为 href 属性值写作../targetFile. html,如此类推,如果位于当前文件上两级目录,则 href 属性值写作../../targetFile. html。

2. 根路径

根路径用于链接相同站点内的文件。根路径与相对路径比较,其差别在于根路径以一斜杠(/)开头,而相对路径不能以斜杠(/)开头。根路径是以 Web 服务器上项目的根目录为基准点进行查找。使用根目录的优点是如果当前文件的路径发生变化不会影响到文件内的链接。

3. 绝对路径

绝对路径包括了标识 Internet 上的文件所需要的所有信息。一个完整的绝对路径包括协议名称、Web 服务器的 ip 或名称、端口号、文件夹名称和文件名称。例如,http://localhost: 8887/xy/gk. htm 就是一个完整的路径。其中,http 为协议名,localhost 是 Web 服务器的主机名,8887 是端口号,xy 是文件夹名称,gk. htm 是文件名称。

HTML 的超链接应用可以分为内部链接、外部链接和书签链接 3 大类。所谓内部链接是指链接到同一个网站内部,不同的资源之间的链接关系。外部链接是指跳转到其他网站或其他元素的连接,href 的值在这种情况下都应设置为绝对路径,绝对路径上面已经详细介绍了。大多数读者都看过电子小说,因为小说很长,一次难以阅读完,所以前面通常会给出一些链接。通过这些链接,可以链接到小说内部的特定位置,这样的连接就是书签链接。实现该链接需要两步操作,第一需要在目标位置加上锚点即,第二需要在超链接的位置加入超链接标记,但是 href 属性值必须在定位标记名称前加上 # 号,即。如果该书签链接需要链接到其他文件或其他站点某页面的特定位置,该超链接的写法为。

通过使用 name 或 id 属性,可以创建一个文档内部的书签,例如:

```
<html>
<body>
<a href="#minnong">悯农</a><br>
<a href="#jingyesi">静夜思</a><br>
<a name="minnong">悯农</a><br>
锄禾日当午,<br>
汗滴禾下土.<br>
```

```

谁知盘中餐,<br>
粒粒皆辛苦! <br>
<a name = "jingyesi">静夜思</a><br>
床前明月光,<br>
疑是地上霜.<br>
举头望明月,<br>
低头思故乡! <br>
</body>
</html>

```

3.2.4 表格

HTML 表格对于 HTML 有着巨大的作用,可以用表格控制页面布局,也可以利用表格展示数据。HTML 表格和人们日常生活中见到的表格没有区别,都包含表格标题、表头、行、列和脚注,如图 3-2 所示。

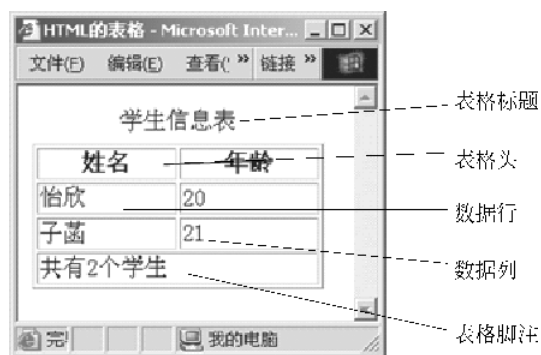


图 3-2 HTML 表格

图 3-1 的实现代码如下:

```

<html>
<head>
  <title>HTML 的表格</title>
</head>
<body>
  <table width = 200 border = 1>
    <caption>学生信息表</caption>
    <thead>
      <tr><th>姓名</th><th>年龄</th></tr>
    </thead>
    <tbody>
      <tr><td>怡欣</td><td>20</td></tr>
      <tr><td>子涵</td><td>21</td></tr>
    </tbody>
    <tfoot>
      <tr><td colspan = 2>共有 2 个学生</td></tr>
    </tfoot>
  </table>

```

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

上述代码中使用到了表格相关的标记有<table>、<caption>、<thead>、<tbody>、<tfoot>、<tr>、<th>和<td>，下面将逐一介绍各标记的使用方法。

1. 表格标记

<table>标记用于定义 HTML 表格。语法格式为：

```
<table width = "" cellpadding = "" cellspacing = "" border = "" bgcolor = "">
:
</table>
```

属性解释：

- width 属性：指定表格的宽度。
- cellpadding 属性：指定单元边沿与其内容之间的空白。
- cellspacing 属性：指定单元格之间的空白。
- border 属性：指定表格边框的宽度。
- bgcolor 属性：指定表格背景色。

2. 表格标题标记

<caption>标记用于定义 HTML 表格的标题。caption 标记必须紧随 table 标记之后，只能对每个表格定义一个标题，通常这个标题会居中于表格之上。语法格式为：

```
<table border = "1">
<caption>标题</caption>
</table>
```

3. 表格头标记

<thead>标记用于组合 HTML 表格的表头内容。<thead>元素应该与<tbody>和<tfoot>元素结合起来使用。语法格式为：

```
<table>
  <thead align = "" valign = "">
    <tr><th>xxx</th><th>xxx</th></tr>
  </thead>
</table>
```

属性解释：

- align 属性：指定 thead 元素中内容的水平对齐方式；常用属性值有 left(左对齐)、right(右对齐)、center(居中)、justify(两端对齐)和 char(指定向哪个字符看齐)，如果对齐方式为 char，那么需要通过 char 属性指定将要看齐的字符。
- valign 属性：指定 thead 元素中内容的垂直对齐方式；常用属性值有 top(上对齐)、middle(居中对齐)、bottom(下对齐)和 baseline(与基线对齐)。

4. 表格体标记

`<tbody>` 标记用于表格主体(正文)。该标记用于组合 HTML 表格的主体内容。`<tbody>` 元素应该与 `<thead>` 和 `<tfoot>` 元素结合起来使用。语法格式为:

```
<table>
  <tbody align = "" valign = "">
    <tr><td>xxx</td><td>xxx</td></tr>
  </tbody>
</table>
```

属性解释:

- align 属性: 指定 tbody 元素中内容的水平对齐方式。
- valign 属性: 指定 tbody 元素中内容的垂直对齐方式。

5. 表格脚注标记

`<tfoot>` 标记用于定义表格的页脚(脚注或表注)。该标记用于组合 HTML 表格中的表注内容。`<tfoot>` 元素应该与 `<thead>` 和 `<tbody>` 元素结合起来使用。语法格式为:

```
<table>
  <tfoot align = "" valign = "">
    <tr><td>xxx</td></tr>
  </tfoot>
</table>
```

属性解释:

- align 属性: 指定 tfoot 元素中内容的水平对齐方式。
- valign 属性: 指定 tfoot 元素中内容的垂直对齐方式。

6. 行标记

`<tr>` 标记用于定义 HTML 表格中的行。tr 元素包含一个或多个 th 或 td 元素。语法格式为:

```
<tr align = "" valign = "" bgcolor = "">
  <th>姓名</th>
  <th>年龄</th>
</tr>
```

属性解释:

- align 属性: 指定 tr 元素中内容的水平对齐方式。
- valign 属性: 指定 tr 元素中内容的垂直对齐方式。
- bgcolor 属性: 指定 tr 元素背景色。

7. 表头单元标记

`<th>` 标记用于定义表格内的表头单元格。th 元素内部的文本通常会呈现为粗体。

语法格式为：

```
<tr>
  <th align = "" valign = "" bgcolor = "" width = "" height = "" rowspan = "" colspan = "">
    姓名
  </th>
</tr>
```

属性解释：

- align 属性：指定 th 元素中内容的水平对齐方式。
- valign 属性：指定 th 元素中内容的垂直对齐方式。
- bgcolor 属性：指定 th 元素的背景色。
- width 属性：规定单元格的宽度。
- height 属性：规定单元格的高度。
- rowspan 属性：指明单元格可横跨的行数。
- colspan 属性：指明单元格可横跨的列数。

8. 标准单元标记

<td> 标记用于定义 HTML 表格中的标准单元格。HTML 表格有两类单元格,表头单元,包含头部信息(由 th 元素创建);标准单元,包含数据(由 td 元素创建),td 元素中的文本一般显示为正常字体且左对齐。语法格式为：

```
<tr>
  <td align = "" valign = "" bgcolor = "" width = "" height = "" rowspan = "" colspan = "">
    子菡
  </td>
</tr>
```

属性解释：

- align 属性：指定 td 元素中内容的水平对齐方式。
- valign 属性：指定 td 元素中内容的垂直对齐方式。
- bgcolor 属性：指定 td 元素的背景色。
- width 属性：规定单元格的宽度。
- height 属性：规定单元格的高度。
- rowspan 属性：指明单元格可横跨的行数。
- colspan 属性：指明单元格可横跨的列数。

3.2.5 表单

HTML 表单是 HTML 的一个重要部分,主要用于采集和提交用户输入的信息。

一个表单有 3 个基本组成部分：

- (1) 表单标记：包括处理表单数据组件的 URL 以及数据提交到服务器的方法。
- (2) 表单域：包括文本框、密码框、隐藏域、多行文本框、复选框、单选框、下拉选择框和文件上传框等。

(3) 表单按钮：包括提交按钮、复位按钮和一般按钮，用于将数据传送给服务器或取消输入。

登录表单示例：

```
<form action = "login.jsp" method = "post">  
    用户名: <input type = "text" name = "username" /><br>  
    密码: <input type = "password" name = "password" /><br>  
    <input type = "submit" value = "登录" />  
</form>
```

上述代码定义了一个采集登录信息的表单，包括两个输入信息项：用户名和密码；一个提交按钮：登录。

1. 表单标记

<form> 标记用于为用户输入创建 HTML 表单。语法格式为：

```
<form action = "处理程序 URL" method = "GET|POST" enctype = "编码类型">  
    其他表单控件  
</form>
```

该标记的作用类似于容器，将所有的表单控件包含在该标记中，当提交时可以将该标记内部所有表单控件的信息提交到指定的处理程序上。该标记需要用户设定的属性主要有 action、method 和 enctype：

action 属性：表示当前表单中的数据提交时交给哪个程序来处理，即指定目标程序的路径。

method 属性：表示数据提交方式，可以设置为 get 或 post，默认属性值为 get，两者的区别参见 2.3.3 节。

enctype 属性：指定表单数据在发送到服务器之前应该如何编码，可以设置为“application/x-www-form-urlencoded”或“multipart/form-data”或“text/plain”。其中，“application/x-www-form-urlencoded”表示在发送前编码所有字符（默认）；“multipart/form-data”表示不对字符编码，在使用包含文件上传控件的表单时，必须使用该值。

2. 文本框

文本框是一种让访问者输入内容的表单对象，该组件的特点是只能容纳一行文本，通常用于信息较短的输入。语法格式为：

```
<input type = "text" name = " " size = " " maxlength = " " value = " " readonly = " ">
```

属性解释：

- type 属性：值为“text”，定义单行文本输入框。
- name 属性：定义文本框的名称，要保证数据的准确采集，必须定义一个唯一的名称。
- size 属性：定义文本框的宽度，单位是单个字符宽度。
- maxlength 属性：定义最多输入的字符数。