

学习目的与要求

本章简要介绍 HTML,包括 HTML 网页文件的整体结构和常用的 HTML 标签,同时介绍 CSS (Cascading Style Sheet)。通过本章的学习,理解 HTML 页面和 CSS 基本原理,能够开发静态 Web 页面。

本章主要内容

- HTML 网页文件结构。
- HTML 基本标签。
- HTML 其他常用标签。
- CSS。

HTML(Hyper Text Markup Language,超文本标记语言)是 Web 应用中用于编写网页的语言。HTML 中每个用来作为标记的符号都可以被看作一条命令,它告诉浏览器应该如何显示文件的内容。

3.1 HTML 网页文件结构

一个完整的 HTML 文件由标题、段落、表格和文本等各种嵌入的对象组成,这些对象统称为元素。HTML 使用标记来分隔并描述这些元素。实际上,整个 HTML 文件就是由元素与标记组成的。

下面是一个 HTML 文件的基本结构。

```
<html>文件开始标记  
<head>文件头开始的标记  
...文件头的内容  
</head>文件头结束的标记
```

```
<body>文件主体开始的标记  
...文件主体的内容  
</body>文件主体结束的标记  
</html>文件结束标记
```

从上面的代码可以看出,HTML 文件包含文件头和文件体两部分,其中各标记含义如下。

`<html>...</html>`: `<html>` 在最外层,表示这对标记间的内容是 HTML 文件,其中包括 `<head>` 和 `<body>` 标记。HTML 文件中所有的内容都应该在这两个标记之间,一个 HTML 文件总是以 `<html>` 开始,以 `</html>` 结束。

`<head>...</head>`: HTML 文件的头部标记,通常将这两个标签之间的内容统称为 HTML 的头部。文件的头部描述了文件的各种属性和信息,包括文件的标题、在 Web 中的位置以及和其他文件的关系等。绝大多数文件头部包含的数据都不会作为内容显示给读者。

`<body>...</body>`: 用来指明文件的主体区域,网页所要显示的内容都放在这个标记内,其结束标记 `</body>` 指明主体区域的结束。

HTML 文件(包括 CSS 和 JavaScript)可以使用操作系统自带的文本编辑器进行编辑,也可以使用下列专业的工具来编辑。推荐以下几款常用的编辑器。

```
Dreamweaver: https://www.adobe.com/products/dreamweaver.html  
Notepad++: https://notepad-plus-plus.org/  
VS Code: https://code.visualstudio.com/
```

(1) 新建 HTML 文件,输入以下代码。

```
<html>  
<head>  
<meta charset="utf-8">  
<title>Web 页面开发</title>  
</head>  
<body>  
  <h1>HTML 开发基础</h1>  
  <p>HTML 是一种用于创建网页的标准标记语言。</p>  
</body>  
</html>
```

(2) 另存为 HTML 文件,文件名为 `html_test.html`。当保存 HTML 文件时,既可以使用 `htm` 扩展名,也可以使用 `html` 扩展名。两者没有区别,完全根据自己的喜好。

(3) 在浏览器中运行这个 HTML 文件。

启动系统的浏览器,然后选择“文件”菜单中的“打开文件”命令,或者直接在文件夹中双击创建的 HTML 文件,显示结果如图 3-1 所示。

HTML开发基础

HTML是一种用于创建网页的标准标记语言。

图 3-1 显示效果

3.2 HTML 基本标签

HTML 文件是由各种 HTML 元素组成的,如 html(HTML 文件的根)元素、head(HTML 头部)元素、body(HTML 主体)元素、title(HTML 标题)元素和 p(段落)元素等。如果一个元素包含另一个元素,它就被包含元素的父元素,被包含元素称为子元素。这些元素都是通过由尖括号“<>”组成的标签形式来表现的。

HTML 标签通常是由一对尖括号“<>”及标签名组成的。标签分为起始标签(start tag)和结束标签(end tag)两种,两者的标签名称是相同的,只是结束标签多了一个斜杠“/”。例如,<p>...</p>段落标签。<p>为起始标签,</p>为结束标签,p 为标签名称,它是英文 paragraph(段落)的缩写。HTML 元素指的是从起始标签到结束标签的所有代码。起始标签和结束标签之间是 HTML 元素的内容,它既可以是需要显示在网页中的文字内容,也可以是其他元素。空元素既不包含文本也不包含其他元素。它看起来像是起始标签和结束标签的结合,由左尖括号开头,接着是元素的名和可能包含的属性,然后是一个可选的空格和一个可选的斜杠,最后是必有的右尖括号。如定义换行的
就是空元素。

HTML 标签名称对大小写不敏感,如<P>...</p>和<P>...</P>的效果是一样的。不过,HTML 规范推荐使用小写字母表示标签。

HTML 标签可以拥有属性。属性总是以名称/值对的形式出现,例如 name="value" 并且总是在 HTML 元素的开始标签中规定,如图 3-2 所示。

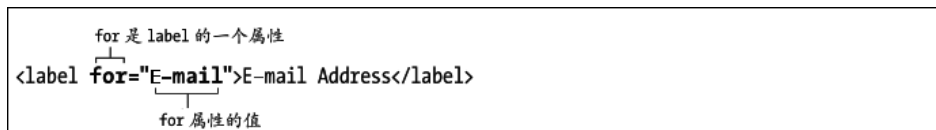


图 3-2 属性和值

下面介绍基本的 HTML 标签。

1. HTML 标题

HTML 标题(Heading)是通过 <h1>~<h6> 等标签进行定义的,此标签只能在 head 标签内出现。例如:

```

<h1>最大字号标题</h1>
<h2>中字号标题</h2>
  
```

```
<h6>最小字号标题</h6>
```

说明：浏览器会自动在标题的前后添加空行。默认情况下,HTML 会自动在块级元素前后添加一个额外的空行,例如段落、标题元素前后。

2. HTML 段落

HTML 段落是通过 `<p>` 标签进行定义的。例如:

```
<p>HTML 的英文全称是 Hyper Text Markup Language,它是超文本置标语言的缩写,是 Internet 上用于编写网页的主要语言。</p>
```

```
<p>HTML 中每个用来作为标记的符号都可以被看作一条命令,它告诉浏览器应该如何显示文件的内容。</p>
```

说明：浏览器会自动在段落的前后添加空行。(`<p>` 是块级元素)

3. HTML 注释

HTML 注释是通过 `<!-->` 标签进行定义的。例如:

```
<!--这是我的注释内容-->
```

说明：用于注释 HTML 源码,只能在源码中看到,不在 HTML 效果中展示。

4. HTML 图像

HTML 图像是通过 `<img>` 标签进行定义的。例如:

```

```

说明：图像的名称和宽、高是以属性的形式提供的。

5. 表格

表格由 `<table>` 标签来定义。每个表格均有若干行(由 `<tr>` 标签定义),每行被分割为若干单元格(由 `<td>` 标签定义)。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。例如:

```
<table border="1">
<tr>
<td>row 1,column 1</td>
<td>row 1, column 2</td>
</tr>
<tr>
<td>row 2,column 1</td>
<td>row 2,column 2</td>
</tr>
</table>
```

浏览器显示结果:

row 1,column 1	row 1,column 2
row 2,column 1	row 2,column 2

6. 表单

表单是一个包含表单元素的区域。表单元素允许用户在表单中(如文本域、下拉列表、单选按钮、复选框等)输入信息,然后服务器端可以获得用户输入的数据。表单使用表单标签(<form>)定义。

```
<form>
...
    input 元素
...
</form>
```

多数情况下被用到的表单标签是输入标签(<input>)。输入类型是由类型属性(type)定义的。大多数经常被用到的输入类型如下。

1) 文本域

当用户要在表单中键入字母、数字等内容时,就会用到文本域。

```
<form>
姓名:
<input type="text" name="name" />
<br />
年龄:
<input type="text" name="age" />
</form>
```

浏览器显示结果:

姓名:

年龄:

注意: 表单本身并不可见。同时,在大多数浏览器中,文本域的默认宽度是 20 个字符。

2) 单选按钮

当用户从若干给定的选择中选取其一,就会用到单选按钮。

```
<form>
<input type="radio" name="sex" value="male" />男
<br />
<input type="radio" name="sex" value="female" />女
</form>
```

浏览器显示结果:

☐ 男

☐ 女

注意: 选项只能从中选取其一。

3) 复选框

当用户需要从若干给定的选择中选取一个或若干选项时,就会用到复选框。

```
<form>
<input type="checkbox" name="apple" />
我喜欢苹果
<br />
<input type="checkbox" name="banana" />
我喜欢香蕉
</form>
```

浏览器显示结果:

☐ 我喜欢苹果
☐ 我喜欢香蕉

4) 表单的动作属性(Action)和确认按钮

当用户单击“确认”按钮时,表单的内容会被传送到另一个文件。表单的动作属性定义了目的文件的文件名。由动作属性定义的这个文件通常会对接收到的输入数据进行相关的处理。

```
<form name="input" action="login.jsp" method="get">
用户名:
<input type="text" name="user" />
密码:
<input type="password" name="password">
<input type="submit" value="提交" />
</form>
```

浏览器显示结果:

用户名: 密码:

在上面的文本框内输入用户名和密码,然后单击“确认”按钮,那么输入数据会传送到“login.jsp”的页面。该页面将验证用户是否为合法用户。

3.3 HTML 其他常用标签

1. 超链接

Web 上的网页都是互相链接的,通过超链接可以链接到其他页面,这里的超链接就是具有链接能力的文字或者图片。可以链接文本、媒体等,了解更多内容的接口。

1) 建立超链接标签<A>

建立超链接的标签为<A>,基本格式为:

```
<A href="链接地址" target="超链接窗口打开方式">超链接名称</A>
```

例如：

```
<a href="http://www.ascenttech.com.cn/ruanjianjishufuwu.htm" target="_blank" >  
软件技术服务</a>
```

属性 href 指定了链接到的目标地址,该地址可以是文件所在位置,也可以是一个 URL,只有正确指定目标地址,才能正确访问需要的资源。

属性 target 用于指定打开链接的目标窗口,其默认方式是原窗口。其他的取值如表 3-1 所示。

表 3-1 超链接标签<A>的属性

属 性 值	描 述
parent	在上一级窗口打开,一般在分帧的框架页面中使用
blank	在新窗口打开
self	在同一帧或窗口中打开,和默认值相同
top	在浏览器的整个窗口中打开,忽略任何框架

其中,超链接名称可以是文字,也可以是图像或其他网页元素。文本会带有下画线并且与其他文字的颜色有区别,图形通常带有边框显示。当用鼠标单击该元素时,就会跳转到链接地址所指定的位置。

2) 链接的路径

在超链接中,文件的存放位置和路径以及 URL 必须指定清楚才能正确链接,这种链接路径分为绝对路径、相对路径两种。

绝对路径是主页上的文件或目录在硬盘上的真正路径,使用绝对路径作为链接路径比较清晰,但是也存在一定的缺陷,比如把文件夹改名或者移动以后,相关的链接都将失效,这样就必须对所有 HTML 文件的链接重新编写,会很麻烦。比如有个页面 index.htm,该页面的绝对路径为 D:\AcesysHtml\index.htm,页面中有一个图片位置为 D:\AcesysHtml\img\buy.jpg,如果在这台机器上可以很顺利地通过页面访问这个图片,但是如果将这些文件移动到其他位置进行发布,没有放到 D 盘中,那么就会因为这个路径的指定而找不到该图片。典型的绝对路径格式为:通信协议://服务器地址:通信端口/文件位置……/文件名。

为了解决绝对路径不能随便移动或修改文件的问题,可以使用相对路径来指定链接的位置。相对路径是以当前文件所在的路径为基准,进行相对文件的查找。一个相对的 URL 不包括协议和主机信息,表示它的路径与当前文档的访问协议和主机名相通。相对路径的好处是只要相关的文件相对位置没有发生改变就可以访问到,比如前面提到的图片和页面之间的相对路径就是 img\buy.jpg,也就是可以忽略二者共同的路径部分,只写相对的部分,这样即使它们被移动了位置,只要它们是同时被移动,相对位置没有改变,就不会出现访问出错。

在描述相对路径中,在路径前加../表示上一级目录中的文件,如果上两级则表示为../../,那么相对路径的几种写法如下。

- product.htm,表示跟本文件在同一目录下。

- ../product.htm,表示是上一级目录下的文件。
- product/product.htm,表示当前目录下的文件夹 product 下的文件。

3) 超链接的应用

假设有 3 个文件,分别为 index.html、itservice.html、products.html。其中,index.html 是起始页面,itservice.html 和 products.html 在 product 文件夹下,product 文件夹与 index.html 是在同一级别,在 index.html 中可以链接到后面两个页面上,index.html 的代码如下。

```
<html>
<head>
<title>AscentWeb 电子商务</title>
</head>

<body>
<div>
    <h1>加拿大·亚思晟科技 </h1>
    <h2>Quality is everything!</h2>
</div>

    <div>
        <ul>
            <li><a href="index.html">首页</a></li>
            <li><a href="product/itservice.html">IT 服务</a></li>
            <li><a href="product/products.html">电子商务系统</a></li>
        </ul>
    </div>
</body>
</html>
```

2. HTML 框架结构标签(<frameset>)

- (1) 框架结构标签(<frameset>)定义如何将窗口分割为框架。
- (2) 每个 frameset 定义了一系列行或列。
- (3) rows/columns 的值规定了每行或每列占据屏幕的面积。

通过使用框架,可以在同一个浏览器窗口中显示不只一个页面。每份 HTML 文件称为一个框架,并且每个框架都独立于其他框架。

框架结构 HTML 文件 frameset.html 如下。

```
<!doctype html>
<html>
<head>
<title>HTML Frameset</title>
</head>
<frameset cols="30%,70%">
    <frame src="frame_a.html">
    <frame src="frame_b.html">
</frameset>
```

```
</html>
```

HTML 文件 frame_a.html 如下。

```
<!DOCTYPE html>
<html>
<body bgcolor="e9967a">
<h1>我是第一列</h1>
</body>
</html>
```

HTML 文件 frame_b.html 如下。

```
<!DOCTYPE html>
<html>
<body bgcolor="#00ccff">
<h1>我是第二列</h1>
</body>
</html>
```

在上面的这个例子中设置了一个两列的框架集。第一列被设置为占据浏览器窗口的 30%；第二列被设置为占据浏览器窗口的 70%。HTML 文件 frame_a.htm 被置于第一个列中,而 HTML 文件 frame_b.htm 被置于第二列中,效果如图 3-3 所示。



图 3-3 框架使用效果

3. HTML 区块

HTML 区块元素：大多数 HTML 元素为块级元素或内联元素。

(1) 块级元素：通常在浏览器显示时会以新行开始(和结束),例如： <p> <h1> <table>等。

(2) 内联元素：在显示时不会以新的一行开始,例如： <a> 等文本的容器。

1) <div>元素

HTML <div>元素是块级元素,它可用于组合其他 HTML 元素的容器。

<div>元素没有特定的含义。除此之外,由于它属于块级元素,浏览器会在其前后显示折行;如果和 CSS 一同使用,<div>元素可用于对大的内容块设置样式属性。

`<div>`元素的另一个常见的用途是文档布局。它取代了使用表格定义布局的老式方法。

例如,div_test.html 代码如下。

```
<html>
<head>
<meta charset="utf-8">
<title>div 块级元素</title>
</head>
<body>
<h3>这是标题 3</h3>
  <p>这是段落</p>
  <div style="color:#00FF00">
    <h3>这是标题三</h3>
    <p>这是段落</p>
  </div>
</body>
</html>
```

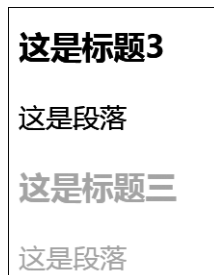


图 3-4 块级元素效果

显示效果如图 3-4 所示。

2) span 元素

`<span>`属于内联元素。`<span>`可作为文本的容器,与 CSS 一同使用时可作为部分文本设置样式属性。

例如,span_test.html 代码如下。

```
<!DOCTYPE html>
<html>
<head>
<title>span 内联元素</title>
</head>
<body>
<p>基于 Web 技术的物联网应用与开发 <span style="color:blue;font-weight:bold">智云平台</span><span style="color:red;font-weight:bold">硬件设备</span>线上资源</p>
</body>
</html>
```

显示效果如图 3-5 所示。



图 3-5 内联元素效果

3.4 CSS

前面了解了如何通过 HTML 创建文件结构和内容,接下来介绍 CSS(Cascading Style Sheet)怎么为 HTML 添加表现样式。

每个 HTML 元素都有一组样式属性,可以通过 CSS 设定。这些属性涉及元素在屏幕上显示时的不同方面,比如在屏幕上位置、边框的宽度,文本内容的字体、字号和颜色,等等。CSS 就是一种先选择 HTML 元素,然后再设定选中元素 CSS 属性的机制。CSS 选择器和要应用的样式构成了一条 CSS 规则。

CSS 的主要特点有:能精确定位,控制页面中的每一个元素;能对 HTML 处理的样式做最好的补充;能把内容和格式处理相分离,大大减少工作量。

3.4.1 创建 CSS

CSS 的编写很容易,可以是文本编辑器、样式表编辑器或者专业开发工具。无论用哪种方式,都是要创建一个以 css 为扩展名的文件,该文件中包含了要设定的规则。通常,把它保存在一个特定的目录下。

样式表的基础就是组成它的 CSS 规则,每一条规则都是单独的语句。CSS 规则由选择器(selector)和声明两部分组成。声明由属性(property)和属性的值组成。其语法格式为:

```
选择器{声明 1; 声明 2; ...; 声明 N}
```

CSS 会自动忽略掉附加的空白,所以为了方便阅读,可以添加一些空格。

1. 选择器

用来指定针对哪一个 HTML 标签应用样式表的部分,任何 HTML 元素都可以是一个 CSS 的选择器。例如:

```
P {color: red}
```

中的选择器是 P,规则就会选择所有<p>标签的样式,该规则表示在段落标签里的内容为红色。

2. 声明

声明是包含在大括号中的内容,首先给出属性名,可以包括颜色、边界和字体等;然后是属性值,给一个属性能够接受的值,加上分号之后可以指定多个样式。例如:

```
H1{font-size: 24pt; color: green}
```

上述语句表示标题中的字体大小为 24pt,颜色为绿色。

3. 组合规则

用户通过大括号内列出声明,有时候对于同一个选择器列出了几个规则,这时候规则是可以合并起来写的,同时很多不同的选择器却具有重复的样式表声明,为了减少样式表

的重复声明,组合的选择器声明是允许的。例如:

```
H1, H2, H3, H4, H5, H6 {
    color: red;
    font-family: sans-serif}
```

该规则标明 H1、H2、H3、H4、H5、H6 的颜色都为红色,并且字体为 sans-serif。

4. 继承

实际上,所有在选择器中嵌套的选择器都会继承外层选择器指定的属性值,除非另外更改。例如,一个 BODY 定义了的颜色值也会应用到段落的文本中。

有些情况是内部选择器不继承周围的选择器的值,但理论上这些都是特殊的。例如,上边界属性是不会继承的;直觉上,一个段落不会有同文件 BODY 一样的上边界值。

5. 注释

注释是用户嵌入 CSS 代码中的专用位,浏览器会忽略注释。样式表里面的注解使用与 C 语言编程中一样的约定方法去指定。CSS 的注释以字符/* 开始,以 */结束,两者之间的内容浏览器都会忽略,注解的例子如以下格式。

```
/* 这是注释内容 */
```

注释可以出现在 CSS 文件中的任何位置。

下面创建一个简单的样式表:

```
/****** Body and tag styles *****/

body{
font:76%Verdana,Tahoma,Arial,sans-serif;
line-height:1.4em;
text-align:center;
color:#303030;
background:#e8eaec;
}

a{
color:#467aa7;
font-weight:bold;
text-decoration:none;
background-color:inherit;
}

p{padding:0 0 1.6em 0;}
p form{margin-top:0; margin-bottom:20px;}
```

3.4.2 添加 CSS 的方法

为了使 CSS 所设置的样式能够在网页中产生作用,必须通过一些方法将 CSS 和

HTML 挂接在一起,才可以正常工作。

在 HTML 中,添加的样式表有 4 种:行内样式表、嵌入样式表、链接样式表和输入样式表。

1. 行内样式表

行内样式表(Inline Styles)是最简单的一种使用方式,由标签 style 属性支持,CSS 规则直接写在标签内。例如:

```
<p style="font-size:10px; bgcolor:#fff000; ">
```

这种方法的不便之处在于必须在每行中都加入样式表规则,否则到下一行时浏览器将转回到文件的默认设置。虽然加入行内的样式表相比较来说不如嵌入样式表、链接样式表及输入样式表的功能强大,但是有时候它非常有用。

2. 嵌入样式表

嵌入样式表(Embedded Styles)是一个样式集,它是网页代码的一部分,在带有<style>标签的 HTML 文件内直接嵌入 CSS,它与行内样式表有相似的地方,但是又不同,行内样式表的作用域只有一行,而嵌入式样式表可以作用于整个文件。

对于嵌入样式表,使用<style>标签,它也只能在<head>内使用,语法格式为:

```
<style type="text/css" media="media-type">
```

其他 CSS 规则为:

```
</style>
```

例如:

```
<HTML>
<STYLE TYPE="text/css">
<!--
H1{color:green;font-family:impact}
P{background:yellow;font-family:counter}
-->
</STYLE>
<HEAD>
<TITLE>My First Stylesheet</TITLE>
</HEAD>
<BODY>
...
```

在嵌入样式表规则下,浏览器在整个 HTML 页面中都执行该规则。这里的<style>标签是 HTML,它告诉网页浏览器:包含在标签内的是 CSS 代码。

使用嵌入样式表的好处是在初始时编辑容易,并给 HTML 提供一个即时的视觉提升,而且当不需要规则应用到所有网页,而只在特定页面应用时比较方便,但是嵌入样式表维护和更新网站比较困难,当需要设计包含多页面的网站时,需要复制和粘贴样式表到每个页面,而且当进行修改的时候必须编辑每一张网页,这时候用链接样式表是最适合的。

3. 链接样式表

链接样式表(Linked Style)也称外部样式表,它首先定义一个扩展名为 css 的文件,该文件包含所有需要用到的 CSS 规则,不包含任何 HTML 代码,比如 blue.css。创建样式表文件后,需要将其与要进行格式设置的 HTML 文件进行关联,这种添加样式表的方式是通过 HTML 中<link>标签来实现的,Link 标签只在 HTML 页面的<head>部分出现。链接样式表的方法就是在 HTML 文件的<head>部分添加如下代码。

```
<head>
<title>line</title>
<link rel="stylesheet" href="./css/andreas08 (blue).css" type="text/css"
media=" screen, projection " />
</head>
```

由于 Link 标签只有一个开始标签,没有相匹配的关闭标签,所以在结尾处添加一个斜杠,来结束标签,Link 有很多属性,但是这里最重要的就是上面例子中的几个属性。

- Rel 属性: 表示链接类型,定义连接的文件和 HTML 文档之间的关系时就设为 stylesheet。
- Href 属性: 指出了样式表的位置,它只是个普通的 URL 地址,可以是相对地址,也可以是绝对地址。
- Type 属性: 指明了链接样式表的样式语言,对于级联样式表,它的取值为 text/css。
- Media 属性: 用于指定样式表被接受的介质或媒体,允许的值有:
 - lscreen(默认值),提交到计算机屏幕;
 - lprint,输出到打印机;
 - lprojection,提交到投影机;
 - laural,扬声器;
 - lbraille,提交到凸字触觉感知设备;
 - ltty,电传打字机(使用固定的字体);
 - ltv,电视机;
 - lall,所有输出设备。

如果有多样的媒体,则可以通过逗号隔开的列表或 all 指定。

4. 输入样式表

输入样式表(Imported Styles)方法同链接样式表类似,不同之处在于链接样式表不能同其他方法结合使用,但输入样式表则可以。注意,使用@import 导入外部的样式表文件时,需要在<style>标签内,例如,StyleSheetDemo.html 代码如下。

```
<html>
<style type="text/css">
<!--
@ import url(company.css);
H1{color:orange;font-family:impact}
-->
```

```
</style>
<head>
<title>My First Stylesheet
</title>
</head>
<body>
<h1>Stylesheet:The Tool of the Web Design Gods
</h1>
<p>Amaze your friends!Squash your enemies!
</p>
</body>
</html>
```

其中输入的 `company.css` 文件内容为:

```
H1{color:green;font-family:times}
P{background:yellow;font-family:counier}
```

3.4.3 CSS 选择器

选择器决定样式规则应用于哪些元素。最简单的选择器可以对给定类型的所有元素(如所有的 `h2` 标题)进行样式化,有的选择器允许人们根据元素的类、上下文、状态等来应用格式化规则。

选择器可以定义以下 5 个不同的标准来选择要进行格式化的元素。

- 元素的类型或名称。
- 元素所在的上下文。
- 元素的类或 `id`。
- 元素的伪类或伪元素。
- 元素是否有某些属性和值。

1. 元素选择器

最常见的 CSS 选择器是元素选择器。换句话说,文档的元素就是最基本的选择器。如果设置 HTML 的样式,选择器通常将是某个 HTML 元素,比如 `p`、`h1`、`em`、`a`,甚至可以是 HTML 本身。

```
h1 {color:blue;}
h2 {color:silver;}
```

2. 派生选择器

在 CSS 中,可以根据元素的祖先、父元素或同胞元素来定位它们。派生选择器允许根据文档的上下文关系来确定某个标签的样式。通过合理地使用派生选择器,可以使 HTML 代码变得更加整洁。

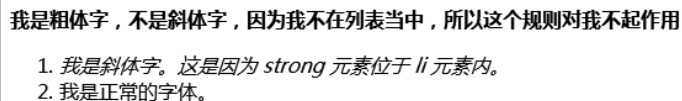
如果希望列表中的 `strong` 元素变为斜体字,而不是通常的粗体字,可以这样定义一个派生选择器,例如 `strong_test.html` 代码如下。

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Web 页面开发</title>
<style type="text/css">
li strong {
    font-style: italic;
    font-weight: normal;
}
</style>
</head>
<body>
    <p><strong>我是粗体字,不是斜体字,因为我不在列表当中,所以这个规则对我不起作用
</strong></p>
    <ol>
    <li><strong>我是斜体字。这是因为 strong 元素位于 li 元素内。</strong></li>
    <li>我是正常的字体。</li>
    </ol>
</body>
</html>

```

页面显示效果如图 3-6 所示。



我是粗体字，不是斜体字，因为我不在列表当中，所以这个规则对我不起作用

1. 我是斜体字。这是因为 strong 元素位于 li 元素内。
2. 我是正常的字体。

图 3-6 派生选择器显示效果

派生类选择器的格式为：

标签 1 标签 2 {声明}

说明：标签 2 就是我们想要选择的目标，而且只有在标签 1 是其祖先元素（不一定是父元素）的情况下才会被选中。

派生类选择器，严格地讲（也就是 CSS 规范里）应该叫作后代组合式选择器（descendant combinator selector），就是一组以空格分隔的标签名。用于选择作为指定祖先元素后代的标签。

3. id 选择器

并非所有的选择器都需要指定元素的名称。如果对某一类的元素进行格式化，而不管属于这个类的元素的类型，就可以从选择器中去掉元素名称。id 选择器可以为标有特定 id 的 HTML 元素指定特定的样式。

id 选择器：以 "#" 来定义，不加空格。下面的两个 id 选择器，第一个可以定义包含 id="red" 属性的元素颜色为红色，第二个定义包含 id="green" 属性的元素颜色为绿色。

```
#red {color:red;}
#green {color:green;}
```

下面的 HTML 代码中, id 属性为 red 的 p 元素显示为红色, 而 id 属性为 green 的 p 元素显示为绿色。

```
<p id="red">电子商务</p>
<p id="green">购物管理界面</p>
```

注意: id 属性只能在每个 HTML 文档中出现一次。

在现代布局中, id 选择器常常用于建立派生选择器。例如:

```
#sidebar p {
    font-style: italic;
    text-align: right;
    margin-top: 0.5em;
}
```

上面的样式只会应用于出现在 id 是 sidebar 的元素内的段落。这个元素很可能是 div 或者是表格单元, 尽管它也可能是一个表格或者其他块级元素。它甚至可以是一个内联元素, 比如 `<em></em>` 或者 `<span></span>`, 不过这样的用法是非法的, 因为不可以在内联元素 `<span>` 中嵌入 `<p>`。

即使被标注为 sidebar 的元素只能在文档中出现一次, 这个 id 选择器作为派生选择器也可以被使用很多次。例如:

```
#sidebar p {
    font-style: italic;
    text-align: right;
    margin-top: 0.5em;
}
#sidebar h2 {
    font-size: 1em;
    font-weight: normal;
    font-style: italic;
    margin: 0;
    line-height: 1.5;
    text-align: right;
}
```

在这里, 与页面中的其他 p 元素明显不同的是, sidebar 内的 p 元素得到了特殊的处理; 与页面中其他所有 h2 元素明显不同的是, sidebar 中的 h2 元素也得到了不同的特殊处理。

id 选择器即使不被用来创建派生选择器, 它也可以独立发挥作用。

```
#sidebar {
    border: 1px dotted #000;
    padding: 10px;
}
```

根据这条规则, id 为 sidebar 的元素将拥有一个像素宽的黑色点状边框, 同时其周围会有 10 像素宽的内边距(padding, 内部空白)。老版本的 Windows/IE 浏览器可能忽略这条规则, 除非特别地定义这个选择器所属的元素。例如:

```
div#sidebar {  
border: 1px dotted #000;  
padding: 10px;  
}
```

4. class 选择器

类选择器的语法格式为:

.类名

例如:

```
.center {text-align: center}
```

在上面的例子中, 所有拥有 center 类的 HTML 元素均为居中。

在下面的 HTML 代码中, h1 和 p 元素都有 center 类。这意味着两者都将遵守 .center 选择器中的规则。

```
<h1 class="center">标题一将会居中显示。</h1>  
<p class="center">段落将会居中显示。</p>
```

注意: 类名的第一个字符不能使用数字。它无法在 Mozilla 或 Firefox 中起作用, 且类选择器前面是点“.”紧跟着类名, 两者之间没有空格。

和 id 一样, class 也可被用作派生选择器。例如:

```
.fancy td {  
color: #f60;  
background: #666;  
}
```

在上面这个例子中, 类名为 fancy 的更大的元素内部的表格单元都会以灰色背景显示橙色文字(名为 fancy 的更大的元素可能是一个表格或者一个 div)。

元素也可以基于它们的类而被选择。例如:

```
td.fancy {  
color: #f60;  
background: #666;  
}
```

在上面的例子中, 类名为 fancy 的表格单元将是带有灰色背景的橙色。

```
<td class="fancy">
```

可以将类 fancy 分配给任何一个表格元素任意多的次数。那些以 fancy 标注的单元格

都会是带有灰色背景的橙色。那些没有被分配名为 fancy 的类的单元格不会受这条规则的影响。还有一点值得注意, class 为 fancy 的段落也不会是带有灰色背景的橙色, 当然任何其他被标注为 fancy 的元素也不会受这条规则的影响。这都是书写这条规则的方式, 这个效果被限制于被标注为 fancy 的表格单元(即使用 td 元素来选择 fancy 类)。

要在 class 选择器和 id 选择器之间做出选择的时候, 建议尽可能地使用 class 选择器。这主要是因为我们可以通过复用 class 选择器。id 选择器会引入下面两个问题。

(1) 在一个页面中, 一个 id 只能出现在一个元素上。这会导致在其他元素上重复样式, 而不是通过 class 共享样式。

(2) 它们的特殊性比 class 选择器要强得多。这意味着如果要覆盖使用 id 选择器定义的样式, 就要编写特殊性更强的 CSS 规则。如果数量不多, 可能还不难管理; 如果处理规模较大的网站, 其 CSS 就会变得比实际所需的更长、更复杂。

5. 属性选择器

对带有指定属性的 HTML 元素设置样式。可以为拥有指定属性的 HTML 元素设置样式, 而不仅限于 class 和 id 属性。

说明: 只有在规定了!DOCTYPE 时, IE7 和 IE8 才支持属性选择器。在 IE6 及更低的版本中, 不支持属性选择。

下面的程序示例为带有 title 属性(不要与 title 元素弄混淆)的所有元素设置样式, 例如, title_test.html 代码如下。

```
<!doctype html>
<html>
<head>
  <style type="text/css">
    [title]
    {
      color:green;
    }
  </style>
</head>
<body>
  <h1>属性选择器应用</h1>
  <p title="智能家居">智能家居</p>
</body>
</html>
```

页面显示效果如图 3-7 所示。



图 3-7 属性选择器显示效果

下面的例子为 title="智能家居" 的所有元素设置样式。

```
[title="智能家居"]
{
border:5px solid green;
}
```

下面的例子为包含指定值的 title 属性的所有元素设置样式。适用于由空格分隔的属性值。

```
[title~ =hello] {color:red;}
```

下面的例子为带有包含指定值的 lang 属性的所有元素设置样式。适用于由连字符分隔的属性值。

```
[lang|=en] {color:red;}
```

属性选择器在为不带有 class 或 id 的表单设置样式时特别有用。例如, title_test2.html 代码如下。

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Web 页面开发</title>
<style>
input[type="text"]
{
width:150px;
display:block;
margin-bottom:10px;
background-color:yellow;
}
input[type="button"]
{
width:120px;
margin-left:35px;
display:block;
}
</style>
</head>
<body>

<form name="input" action="demo-form.php" method="get">
传感器:<input type="text" name="sensor" value="控制" size="20">
节点:<input type="text" name="dev" value="602" size="20">
<input type="button" value="提交">
```