

第 3 章

CSS3 的高级特性

在前面的章节已经了解到 CSS 的 3 个基本选择器，但如果仅仅依靠这 3 种选择器完成页面制作会比较烦琐。本章学习 CSS 的高级属性，在提高页面制作效率上会有很大帮助。CSS3 的高级属性包括复合选择器、CSS3 新增选择器、CSS3 的层叠特性及继承特性等。

本章要点(已掌握的在方框中打钩)

- 掌握复合选择器的使用方法
- 掌握 CSS3 新增选择器的使用方法
- 熟悉 CSS3 的继承关系
- 熟悉 CSS3 的层叠特性
- 掌握制作新闻菜单的方法

3.1 复合选择器

通过对基本选择器的组合，可以得到更多种类的选择器，从而实现更强、更方便的选择功能。这种通过基本选择器组合得到的选择器就是复合选择器。

3.1.1 全局选择器

如果想要一个页面中所有 html 标记使用同一种样式，可以使用全局选择器。全局选择器，顾名思义，就是对所有 HTML 元素都起作用的选择器。其语法格式如下：

```
*{property:value}
```

其中“*”表示对所有元素都起作用；property 表示 CSS3 属性名称；value 表示属性值。其使用示例如下：

```
*{margin:0; padding:0;}
```

【例 3.1】 设置全局选择器(案例文件：ch03\3.1.html)。

```
<!DOCTYPE html>
<html>
<head><title>全局选择器</title>
<style>
*{ color:red; font-size:30px }
</style>
</head>
<body>
<p>使用全局选择器修饰</p>
<p>林花谢了春红，太匆匆。无奈朝来寒雨晚来风。</p>
<h1>胭脂泪，相留醉，几时重。自是人生长恨水长东。</h1>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 3-1 所示，可以看到两个段落和标题都是以红色字体显示，大小为 30px。

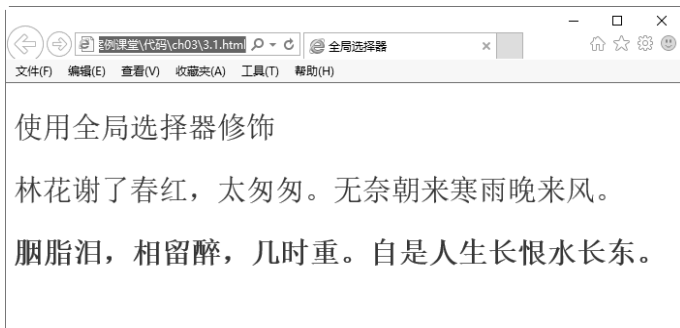


图 3-1 使用全局选择器

3.1.2 交集选择器

交集选择器由两个选择器直接连接构成，其结果是选中二者各自元素范围的交集。其中第 1 个必须是标记选择器，第 2 个必须是类选择器或 ID 选择器。这两个选择器之间不能有空格，必须连续书写，这种方式构成的选择器，将选中同时满足前后二者定义的元素，也就是前者所定义的标记类型，并且指定了后者的类别或者 id 的元素，因此被称为交集选择器。

【例 3.2】 设置交集选择器(案例文件：ch03\3.2.html)。

```
<!DOCTYPE html>
<head>
<title>交集选择器</title>
<style type="text/css">
P{color:blue;font-size:18px;}
p.p1{color:red;font-size:24px;} /*交集选择器*/
.p1{ color:black; font-size:30px}
</style>
</head>
<body>
<p>使用 p 标记</p>
<p class="p1">众芳摇落独暄妍，占尽风情向小园。疏影横斜水清浅，暗香浮动月黄昏。</p>
<h3 class="p1">霜禽欲下先偷眼，粉蝶如知合断魂。幸有微吟可相狎，不须檀板共金尊。</h3>
</body>
</html>
```

上面代码中定义了 p.p1 的样式，也定义了.p1 的样式，p.p1 的样式会作用在<P class="p1">标记上，p.p1 中定义的样式不会影响 h 标签使用了.p 的标记，在浏览器中预览，显示效果如图 3-2 所示。

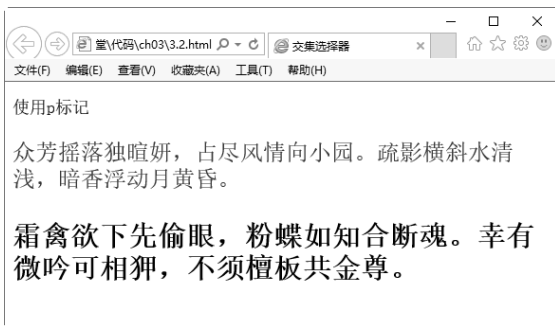


图 3-2 使用交集选择器

3.1.3 并集选择器

并集选择器的结果是同时选中各个基本选择器所选择的范围。任何形式的选择器(包括标记选择器、类选择器、ID 选择器)都可以作为并集选择器的一部分。并集选择器是多个选择器通过逗号连接而成的。如果某些选择器的风格是完全相同的，那么这时便可以利用并集选择器同时声明风格相同的 CSS 选择器。

【例 3.3】 设置并集选择器(案例文件: ch3\3.3.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>并集选择器</title>
<style type="text/css">
  h1,h2,p{ color:red; font-size:20px; font-weight:bold; }
</style></head><body>
<h1>相思似海深, 旧事如天远。</h1>
<h2>泪滴千千万万行, 更使人愁肠断。要见无因见, 了拼终难拼。</h2>
<p>若是前生未有缘, 待重结、来生愿。 </p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 3-3 示, 可以看到网页上标题 1、标题 2 和段落都以红色字体加粗显示, 并且大小为 20px。

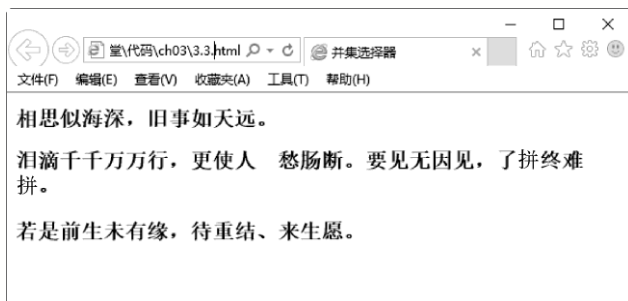


图 3-3 使用并集选择器

3.1.4 继承选择器

继承选择器也被称为后代选择器, 它的规则是子代标记在没有定义的情况下所有的样式是继承父代标记的, 当子代标记重复定义了父代标记已经定义过的声明时, 子代标记就执行后面的声明; 与父代标记不冲突的地方仍然沿用父代标记的声明。CSS 的继承是指后代元素继承祖先元素的某些属性。

使用示例如下:

```
<div class="test">
<span></span>
</div>
```

对上面层而言, 如果其修饰样式为下面代码:

```
.test span img {border:1px blue solid;}
```

则表示该选择器先找到 class 为 test 的标记, 再从它的子标记里查找 span 标记, 再从 span 的子标记中找到 img 标记。也可以采用下面的形式:

```
div span img {border:1px blue solid;}
```

可以看出其规律是从左往右, 依次细化, 最后锁定要控制的标记。

【例 3.4】设置继承选择器(案例文件: ch03\3.4.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>继承选择器</title>
<style type="text/css">
h1{color:red; text-decoration:underline;}
h1 strong{color:#004400; font-size:40px;}
</style>
</head>
<body>
<h1>测试 CSS 的<strong>继承</strong>效果</h1>
<h1>此处使用继承<font>选择器</font>了么? </h1>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 3-4 所示, 可以看到第一个段落颜色为红色, 但是“继承”两个字使用绿色显示, 并且大小为 40px, 除了这两个设置外, 其他的 CSS 样式都是继承父标记 <h1>的样式, 例如下划线设置。第二个标题中, 虽然使用了 font 标记修饰选择器, 但其样式都是继承于父类标记 h1。



图 3-4 使用继承选择器

3.2 CSS3 新增的选择器

选择器(selector)也被称为选择符。所有 HTML 语言中的标记都是通过不同的 CSS 选择器进行控制的。在 CSS3 中, 新增选择器包括属性选择器、结构伪类选择器、UI 元素状态伪类选择器等。

3.2.1 属性选择器

不通过标记名称或自定义名称, 通过直接标记属性来修饰网页, 直接使用属性控制 HTML 标记样式, 称为属性选择器。

属性选择器就是根据某个属性是否存在或属性值来寻找元素, 因此能够实现某些非常有意思和强大的效果。CSS2 中已经出现了属性选择器, 但在 CSS3 版本中, 又新加了 3 个属性选择器。也就是说, 现在 CSS3 中, 共有 7 个属性选择器, 共同构成了 CSS 的功能强大的标

记属性过滤体系。

在 CSS3 版本中，常见属性选择器如表 3-1 所示。

表 3-1 CSS3 属性选择器

属性选择器格式	说明
E[foo]	选择匹配 E 的元素，且该元素定义了 foo 属性。注意，E 选择器可以省略，表示选择定义了 foo 属性的任意类型元素
E[foo="bar"]	选择匹配 E 的元素，且该元素将 foo 属性值定义为了 bar。注意，E 选择器可以省略，用法与上一个选择器类似
E[foo~="bar"]	选择匹配 E 的元素，且该元素定义了 foo 属性，foo 属性值是一个以空格符分隔的列表，其中一个列表的值为 bar。注意，E 选择符可以省略，表示可以匹配任意类型的元素 例如，a[title~="b1"] 匹配 ，而不匹配
E[foo="en"]	选择匹配 E 的元素，且该元素定义了 foo 属性，foo 属性值是一个用连字符(-)分隔的列表，值开头的字符为 en 注意，E 选择符可以省略，表示可以匹配任意类型的元素。例如，[lang="en"] 匹配 <body lang="en-us"></body>，而不是匹配 <body lang="f-ag"></body>
E[foo^="bar"]	选择匹配 E 的元素，且该元素定义了 foo 属性，foo 属性值包含了前缀为"bar"的子字符串。注意，E 选择符可以省略，表示可以匹配任意类型的元素。例如，body[lang^="en"] 匹配 <body lang="en-us"></body>，而不匹配 <body lang="f-ag"></body>
E[foo\$="bar"]	选择匹配 E 的元素，且该元素定义了 foo 属性，foo 属性值包含后缀为 bar 的子字符串。注意 E 选择符可以省略，表示可以匹配任意类型的元素。例如，img[src\$="jpg"] 匹配 ，而不匹配
E[foo*="bar"]	选择匹配 E 的元素，且该元素定义了 foo 属性，foo 属性值包含 b 的子字符串。注意，E 选择器可以省略，表示可以匹配任意类型的元素。例如，img[src*="jpg"] 匹配 ，而不匹配

【例 3.5】 使用属性选择器(案例文件：ch03\3.5.html)。

```

<!DOCTYPE html>
<html>
<head>
<title>属性选择器</title>
<style>
[align]{color:red}
[align="left"]{font-size:20px;font-weight:bolder;}
[lang^="en"]{color:blue;text-decoration:underline;}
[src$="gif"]{border-width:5px;boder-color:#ff9900}
</style>
</head>
<body>
    
```

```

<p align=center>这是使用属性定义样式</p>
<p align=left>这是使用属性值定义样式</p>
<p lang="en-us">此处使用属性值前缀定义样式</p>
<p>下面使用了属性值后缀定义样式

</body>
</html>

```

在 IE11.0 中浏览效果如图 3-5 所示，可以看到第一个段落使用属性 `align` 定义样式，其字体颜色为红色。第二个段落使用属性值 `left` 修饰样式，并且大小为 20px，加粗显示，其字体颜色为红色，是因为该段落使用了 `align` 这个属性。第三个段落显示为红色，且带有下划线，是因为属性 `lang` 的值前缀为 `en`。最后一个图片，以边框样式显示，是因为属性值后缀为 `gif`。

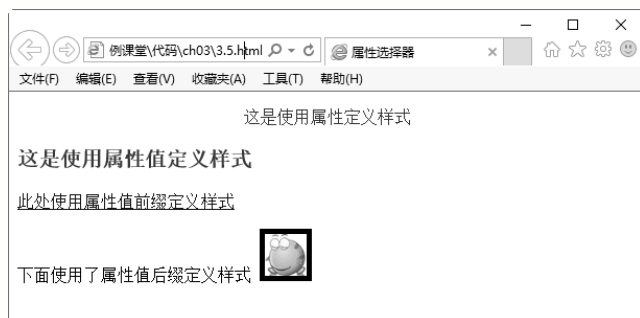


图 3-5 使用属性选择器

3.2.2 结构伪类选择器

结构伪类(structural pseudo-classes)是 CSS3 新增的类型选择器。顾名思义，结构伪类就是利用文档结构树(DOM)实现元素过滤，也就是说，通过文档结构的相互关系来匹配特定的元素，从而减少文档内对 `class` 属性和 `ID` 属性的定义，使得文档更加简洁。

在 CSS3 版本中，新增结构伪类选择器如表 3-2 所示。

表 3-2 结构伪类选择器

选择器	含义
E:root	匹配文档的根元素，对于 HTML 文档，就是 HTML 元素
E:nth-child(n)	匹配其父元素的第 n 个子元素，第一个编号为 1
E:nth-last-child(n)	匹配其父元素的倒数第 n 个子元素，第一个编号为 1
E:nth-of-type(n)	与 :nth-child() 作用类似，但是仅匹配使用同种标签的元素
E:nth-last-of-type(n)	与 :nth-last-child() 作用类似，但是仅匹配使用同种标签的元素
E:last-child	匹配父元素的最后一个子元素，等同于 :nth-last-child(1)
E:first-of-type	匹配父元素下使用同种标签的第一个子元素，等同于 :nth-of-type(1)
E:last-of-type	匹配父元素下使用同种标签的最后一个子元素，等同于 :nth-last-of-type(1)

续表

选择器	含义
E:only-child	匹配父元素下仅有的一个子元素，等同于:first-child:last-child 或 :nth-child(1):nth-last-child(1)
E:only-of-type	匹配父元素下使用同种标签的唯一一个子元素，等同于:first-of-type:last-of-type 或 :nth-of-type(1):nth-last-of-type(1)
E:empty	匹配一个不包含任何子元素的元素(注意，文本节点也被看作子元素)

【例 3.6】 使用结构伪类选择器(案例文件: ch03\3.6.html)。

```

<!DOCTYPE html>
<html>
<head><title>结构伪类</title>
<style>
tr:nth-child(even){
background-color:#f5fafa
}
tr:last-child{font-size:20px;}
</style>
</head>
<body>
<table border=1 width=80%>
<th>编号 </th><th>名称</th><th>价格</th>
<tr><td>001</td><td>芹菜</td><td>1.2 元/kg </td></tr>
<tr><td>002</td><td>白菜</td><td>0.65 元/kg </td></tr>
<tr><td>003</td><td>西红柿</td><td>1.8 元/kg </td></tr>
<tr><td>004</td><td>萝卜</td><td>0.78 元/kg </td></tr>
</table>
</body>
</html>
    
```

在 IE11.0 中浏览效果如图 3-6 所示，可以看到表格中奇数行显示指定颜色，并且最后一行字体以 20px 显示，其原因就是采用了结构伪类选择器。



图 3-6 使用结构伪类选择器

3.2.3 UI 元素状态伪类选择器

UI 元素状态伪类(The UI element states pseudo-classes)也是 CSS3 新增选择器。其中 UI 即

User Interface(用户界面)的简称。UI 设计则是指对软件的人机交互、操作逻辑、界面美观的整体设计。好的 UI 设计不仅要让软件变得有个性有品位，还要让软件的操作变得舒适、简单、自由、充分体现软件的定位和特点。

UI 元素的状态一般包括：可用、不可用、选中、未选中、获取焦点、失去焦点、锁定、待机等。CSS 3 定义了 3 种常用的 UI 元素状态伪类选择器，详细说明如表 3-3 所示。

表 3-3 UI 元素状态伪类

选择器	说明
E:enabled	选择匹配 E 的所有可用 UI 元素。注意，在网页中，UI 元素一般是指包含在 form 元素内的表单元素。例如 <code>input:enabled</code> 匹配 <code><form><input type=text/><input type=button disabled=disabled/></form></code> 代码中的文本框，而不匹配代码中的按钮
E:disabled	选择匹配 E 的所有不可用元素，注意，在网页中，UI 元素一般是指包含在 form 元素内的表单元素。例如 <code>input:disabled</code> 匹配 <code><form><input type=text/><input type=button disabled=disabled/></form></code> 代码中的按钮，而不匹配代码中的文本框
E:checked	选择匹配 E 的所有可用 UI 元素。注意在网页中，UI 元素一般是指包含在 form 元素内的表单元素。例如 <code>input:checked</code> 匹配 <code><form><input type=checkbox/><input type=radio checked=checked/></form></code> 代码中的单选按钮，但不匹配该代码中的复选框

【例 3.7】 使用链接 CSS 文件(案例文件：ch03\3.7.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>UI 元素状态伪类选择器</title>
<style>
input:enabled { border:1px dotted #666; background:#ff9900; }
input:disabled { border:1px dotted #999; background:#F2F2F2; }
</style>
</head>
<body>
<center>
<h3 align=center>用户登录</h3>
<form method="post" action="">
用户名: <input type=text name=name><br>
密 码: <input type=password name=pass disabled="disabled"><br>
<input type=submit value=提交>
<input type=reset value=重置>
</form>
<center>
</body>
</html>
```

在 IE11.0 中浏览效果如图 3-7 所示，可以看到表格中可用的表单元素都显示为浅黄色，而不可用元素显示为灰色。

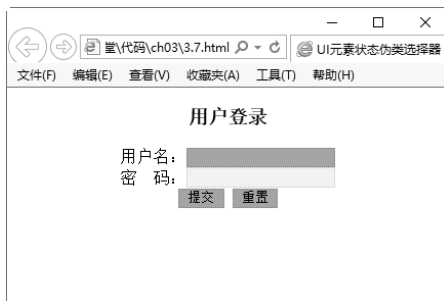


图 3-7 使用 UI 元素状态伪类选择器

3.2.4 伪类选择器

伪类也是选择器的一种，但是用伪类定义的 CSS 样式并不是作用在标记上的。伪类作用在标记的状态上。由于很多浏览器支持不同类型的伪类，没有一个统一标准，所以很多伪类都不常用。伪类包括 :first-child、:link、:visited、:hover、:active、:focus 和 :lang 等。其中有一组伪类是主流浏览器都支持的，就是超链接的伪类，包括 :link、:visited、:hover 和 active。

伪类选择符定义的样式最常应用在标记 <a> 上，它表示链接 4 种不同的状态：未访问链接 (link)、已访问链接 (visited)、激活链接 (active) 和鼠标指针停留在链接上 (hover)。要注意的是，a 可以只具有一种状态 (:link)，或者同时具有 2 种或者 3 种状态。例如，任何一个有 HREF 属性的 a 标签，在未有任何操作时都已经具备了 :link 的条件，也就是满足了有链接属性这个条件；如果访问过的 a 标记，同时会具备 :link :visited 2 种状态。把鼠标指针移到访问过的 a 标记上时，a 标记就同时具备了 :link :visited :hover 3 种状态。

使用示例如下：

```
a:link{color:#FF0000; text-decoration:none}
a:visited{color:#00FF00; text-decoration:none}
a:hover{color:#0000FF; text-decoration:underline}
a:active{color:#FF00FF; text-decoration:underline}
```



上面的样式表示该链接未访问时颜色指针为红色且无下划线，访问后是绿色且无下划线，激活链接时为蓝色且有下划线，鼠标指针放在链接上时为紫色且有下划线。

【例 3.8】 设置伪类(案例文件：ch03\3.8.html)。

```
<!DOCTYPE html>
<html>
<head>
<title>伪类</title>
<style>
a:link {color: red}           /* 未访问的链接 */
a:visited {color: green}     /* 已访问的链接 */
a:hover {color:blue}        /* 鼠标移动到链接上 */
a:active {color: orange}    /* 选定的链接 */
</style>
</head>
```

```
<body>
<a href="">链接到本页</a>
<a href="http://www.sohu.com">搜狐</a>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 3-8 所示，可以看到两个超级链接，第一个超级链接是鼠标指针停留在上方时，显示颜色为蓝色，另一个是访问过后，显示颜色为绿色。



图 3-8 伪类显示

3.3 CSS 的继承特性

继承是一种机制，它允许样式不仅可以应用于某个特定的元素，还可以应用于它的后代。从表现形式上说它是被包含的标记具有其外层标签的样式性质。在 CSS3 中，继承相应比较简单，具体来说就是指定的 CSS 属性向下传递给子孙元素的过程。

3.3.1 继承关系

在 CSS 中也不是所有的属性都支持继承。如果每个属性都支持继承的话，对于开发者来说有时候带来的方便可能没有带来的麻烦多。开发者需要把不需要的 CSS 属性一个一个的关掉。CSS 研制者考虑得很周到，只有那些能带来轻松书写的属性才可以被继承。

以下属性是可以被继承的。

(1) 文本相关的属性是可以被继承的，比如：

```
font-family,font-size,font-style,font-weight,font,line-height,text-align,text-indent,word-spacing
```

(2) 列表属性相关的事可以被继承的，比如：

```
list-style-image,list-style-position,list-style-type,list-style
```

(3) 颜色属性相关的是可以继承的，比如：

```
color
```

【例 3.9】 设置继承关系(案例文件：ch03\3.9.html)。

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

```

<title>继承关系</title>
<style type="text/css">
p{color:red;}
</style>
</head>
<body>
<p>嵌套使<span>用 CSS</span>标记的方法</p>
</body>
</html>
    
```

在案例中 p 标签里面嵌套了一个 span 标签，可以说 p 是 span 的父标签，在样式的定义中只定义 p 标签的样式，运行结果如图 3-9 所示。

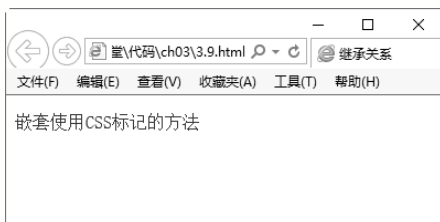


图 3-9 继承关系预览效果

可以看见，span 标签中的字也变成了红色，这就是由于 span 继承了 p 的样式。

3.3.2 CSS 继承的运用

运用继承，可以让开发者更方便轻松地书写 CSS 样式，否则就需要对每个内嵌标签都要书写样式；使用继承同时减少了 CSS 文件的大小，提高下载速度。下面通过一个例子深入理解继承的应用。

【例 3.10】 继承关系的运用(案例文件：ch03\3.10.html)。

```

<!DOCTYPE html>
<head>
<title>继承关系的运用</title>
<style>
h1{color:blue; text-decoration:underline; }
em{ color:red; }
li{ font-weight:bold; }
</style>
</head>
<body>
<h1>继承<em>关系的</em>运用</h1>
<ul>
<li>第一层行一
<ul>
<li>第二层行一</li>
<li>第二层行二
<ul>
<li>第二层行二下第三层行一</li>
<li>第二层行二下第三层行二</li>
<li>第二层行二下第三层行三</li>
    
```

```

</ul>
</li>
<li>第二层行三</li>
</ul>
</li>
<li>第一层行二:
<ol>
<li>第一层行二下第二层行一</li>
<li>第一层行二下第二层行二</li>
<li>第一层行二下第二层行三</li>
</ol>
</li>
</ul>
</body>
</html>

```

在 IE 11.0 中浏览效果如图 3-10 所示，从图中可以知道，em 标签继承了 h1 的下划线，所有 li 都继承了加粗属性。

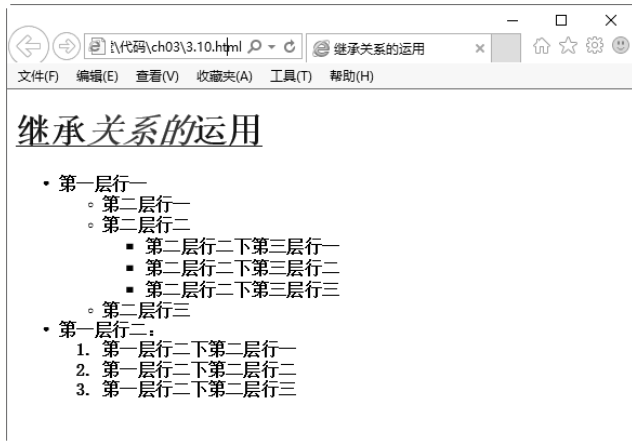


图 3-10 继承关系的应用

3.4 CSS 的层叠特性

CSS 意思本身就是层叠样式表，所以“层叠”是 CSS 的一个最为重要的特征。“层叠”可以被理解为覆盖的意思，是 CSS 中样式冲突的一种解决方法。

3.4.1 同一选择器被多次定义的处理

当同一选择器被多次定义后，就需要 CSS 的层叠特性来进行处理了。下面给出一个具体的案例，来看一下这种情况的处理方式。

【例 3.11】 设置层叠特性(案例文件：ch03\3.11.html)。

```

<!DOCTYPE html>
<head>

```

```
<title>层叠特性</title>
<style>
h1{color:blue;}
  h1{color:red;}
  h1{color:green;}
</style>
</head>
<body>
<h1>江汉曾为客，相逢每醉还。浮云一别后，流水十年间。</h1>
</body>
</html>
```

在代码中，为 h1 标签定义了三次颜色：蓝、红、绿，这时候就产生了冲突，在 CSS 规则中最后有效的样式将覆盖前面的样式，具体到本例就是最后的绿色生效，在 IE 11.0 中浏览效果如图 3-11 所示。

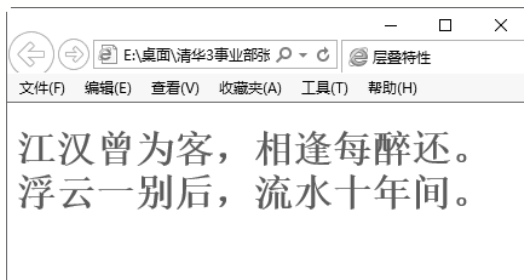


图 3-11 层叠特性的应用

3.4.2 同一标签运用不同类型选择器的处理

当遇到同一标签运用不同类型选择器时，也需要利用 CSS 的层叠特性进行处理。下面给出一个具体的案例。

【例 3.12】应用层叠特性(案例文件：ch03\3.12.html)。

```
<!DOCTYPE html >
<head>
<title>层叠特性</title>
<style type="text/css">
p{color:black;}
.red{color:red;}
.purple {color:purple;}
#p1{color:blue;}
</style>
</head>
<body>
<p >这是第 1 行文本</p>
<p class="red">这是第 2 行文本</p>
<p id="p1" class="red">这是第 3 行文本</p>
<p style="color:green;" id="p1">这是第 4 行文本</p>
<p class=" purple red">这是第 5 行文本</p>
</body>
</html>
```

在 IE 11.0 中浏览效果如图 3-12 所示。



图 3-12 层叠特性的应用

在代码中，有 5 个 p 标签和声明了 4 个选择器，第一行 p 标签没有使用类别选择器或者 ID 选择器，所以第一行的颜色就是 p 标记选择器确定的颜色黑色。第二行使用了类别选择器，这就与 P 标记选择器产生了冲突，这将根据优先级的先后确定到底显示谁的颜色。由于类别选择器优先于标记选择器，所以第二行的颜色就是红色。第三行由于 ID 选择器优先于类别选择器，所以显示蓝色。第四行由于行内样式优先于 ID 选择器，所以显示绿色。在第五行，是两个类选择器，它们的优先级是一样的，这时候就按照层叠覆盖处理，颜色是样式表中最后定义的那个选择器，所以显示紫色。

3.5 综合案例——制作新闻菜单

网上浏览新闻是每个上网者都喜欢做的事情。一个布局合理、样式美观大方的新闻菜单是吸引人的主要途径之一。本案例使用 CSS 控制 HTML 标记，创建新闻菜单。具体步骤如下。

step 01 分析需求。

创建一个新闻菜单，需要包含两个部分：一是父菜单，用来表明新闻类别；二是子菜单，介绍具体的新闻消息。菜单的创建方式很多，可以 table 创建，也可以用列表创建，同样也可以使用段落 p 创建。本案例采用 p 标记结合 div 创建。案例完成后，效果如图 3-13 所示。

step 02 分析局部和整体，构建 HTML 网页。

在一个新闻菜单中，可以分为三个层次：一是新闻父菜单；二是新闻焦点；三是新闻子菜单。它们分别使用 div 创建。其 HTML 代码如下：

```
<!DOCTYPE html>
<html >
<head><title>导航菜单</title>
</head><body>
  <div class="big">
    <h2>时事热点 </h2>
    <div class="up">
      <a href="#">7 月周周爬房团报名</a>
```

```

</div> <div class="down">
  <p>• 50 万买下两居会员优惠 全世界大学排名 工薪阶层留学美国</p><p>
• 家电 | 买房上焦点打电话送礼 楼市松动百余项目打折</p><p>
• 财经 | 油价大跌 CPI 新高 </p>
</div> </div>
</body>
</html>

```

在 IE 11.0 中显示效果如图 3-14 所示。会看到 1 个标题、1 个超级链接和 3 个段落，以普通样式显示，其布局只存在上下层次。

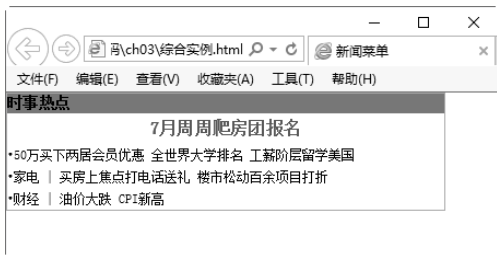


图 3-13 新闻菜单显示

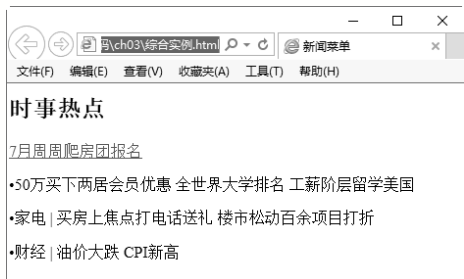


图 3-14 没有添加样式的效果

step 03 添加 CSS 代码，修饰整体样式。

对于 HTML 页面，需要有一个整体样式，其代码如下：

```

<style>
*{padding:0px;margin:0px;}
body{font-family:"宋体";font-size:12px;}
.big{width:400px;border:#33CCCC 1px solid;}
</style>

```

在 IE11.0 中显示效果如图 3-15 所示。可以看到全局层 div 会以边框显示，宽度为 400px，其颜色为浅绿色，body 文档内容中字体采用宋体，大小为 12px，并且定义内容和层之间空隙为零，层和层之间空隙为零。

step 04 添加 CSS 代码，修饰新闻父菜单。

对新闻父类菜单，进行 CSS 控制，其代码如下：

```

h2{background-color:olive;
display:block;
width:400px;
height:18px;
line-height:18px;
font-size:14px;}

```

在 IE11.0 中显示效果如图 3-16 所示。可以看到超级链接“时事热点”会以矩形方框显示，其背景色为橄榄色，字体大小为 14px，行高为 18px。

step 05 添加 CSS 菜单，修饰子菜单。代码如下：

```

.up{padding-bottom:5px;text-align:center;}
p{ line-height:20px;}

```

在 IE11.0 中显示效果如图 3-17 所示。可以看到“7 月周周爬房团报名”居中显示，即在

第二层 div 中使用类标记 up 修饰。所有段落之间间隙增大，即为 p 标记设置行高。

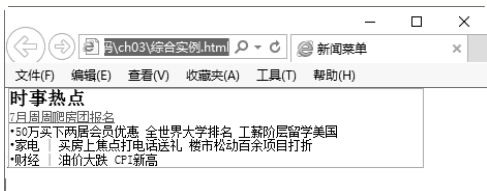


图 3-15 整体添加样式

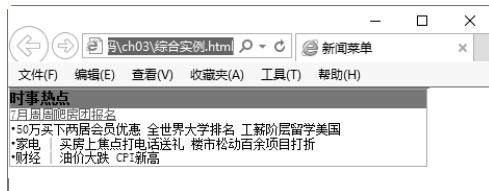


图 3-16 修饰超级链接

step 06 添加 CSS 菜单，修饰超级链接。代码如下：

```
a{font-size:16px; font-weight:800; text-decoration:none; margin-top:5px; display:block;}
a:hover{color:#FF0000; text-decoration:underline;}
```

在 IE11.0 中显示效果如图 3-18 所示。可以看到“7 月周周爬房团报名”字体变大，并且加粗，无下划线显示，当鼠标指针放在此超级链接上，以红色字体显示，并且下面带有下划线。

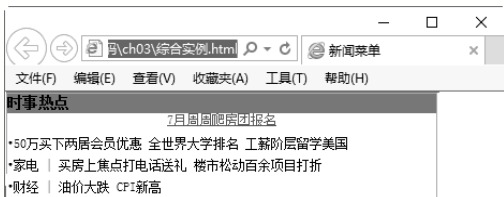


图 3-17 子菜单样式显示

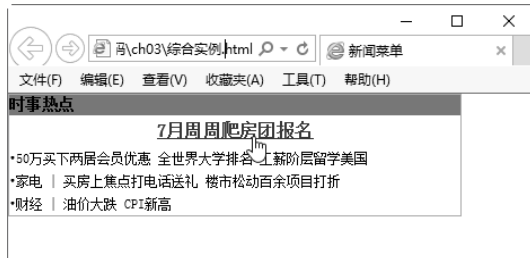


图 3-18 超级链接修饰显示

3.6 大神解惑

小白：如何解决继承带来的错误？

大神：有时候继承也会带来些错误，比如说下面这条 CSS 定义。

```
Body{color:red;}
```

在有些浏览器中这句定义会使除表格之外的文本变成红色。从技术上来说，这是不正确的，但是它确实存在。所以经常需要借助于某些技巧，比如将 CSS 定义成这样：

```
Body,table,th,td{color:red;}
```

这样表格内的文字也会变成红色。

小白：在 Firefox 浏览器下，多层嵌套时内层设置了浮动、外层设置了背景时，背景不显示？

大神：这主要是内层设置浮动后，外层高度在 Firefox 下变为 0，所以应该在外层与内层间再嵌一层，设置浮动和宽度，然后再给这个层设置背景。



3.7 跟我练练手

- 练习 1: 使用全局选择器控制文本的样式。
- 练习 2: 使用交集选择器控制文本的样式。
- 练习 3: 使用并集选择器控制文本的样式。
- 练习 4: 使用属性选择器控制文本的样式。
- 练习 5: 使用 CSS 继承关系来控制文本的样式。
- 练习 6: 制作一个新闻菜单。

