

在基于有效的 XHTML 规则,且格式良好地标记网页内容的基础上,采用有效的 CSS 呈现内容,这样的网站就是符合 Web 标准的网站。本章主要介绍 CSS、CSS 的基本语法、页面中设置 CSS 的几种方式以及 CSS 样式的分类。

3.1 CSS 概述

CSS 是 Cascading Style Sheet 层叠样式表的缩写,是由 W3C 定义和维护的标准,它是一种用来给结构化文档(如 HTML、XHTML 和 XML 等)添加样式的计算机语言。CSS 有层叠和样式表两层含义,其中样式表比较好理解,就是包含许多样式的文本文件(其扩展名为 CSS),而样式是给结构化的标记(可理解为上一章所学习的各种 XHTML 标记)定义表现形式,如文字效果、边框、背景和位置等;对于层叠,由于 XHTML 文档是一种层次结构(如大部分标记中可包含其他标记),当有多个样式指向同一元素时,该元素的最终效果由针对性最强的样式决定,这就是所谓层叠规则,具体在后续的章节中介绍。

CSS 1 是 W3C 于 1996 年 12 月推出的,由于早期浏览器对 CSS 的支持不理想,使得 CSS 备受冷落。随着 2000 年年底 W3C 推荐标准 HTML 4.01 或 XHTML 1.0 的发布,其所倡导的“内容与表现分离”的理念逐步被业界认可,以及 2007 年 7 月发布的 CSS 2.1 获得大多数主流浏览器的良好支持,CSS 才重新焕发生机,并迅速流行起来,成为 Web 页面设计的新标准,目前 CSS 3 是最新的版本。

3.1.1 CSS 的基本语法

CSS 是给 Web 页的页面元素设置表现样式的 W3C 标准,一条 CSS 规则包含选择符和样式声明(列表)两部分,其中每条样式声明又包含属性和属性值两部分。其通用语法格式是:

选择符{样式声明列表}

其中样式声明列表由一组或者多组诸如“属性:属性值;”的样式声明构成。如 `p{color:blue;}` 就是让文档中的所有段落文本为蓝色,可用如图 3-1 所示的对应关系来具体化。

一条 CSS 规则可以包含多条样式声明,如想要“页面中所有

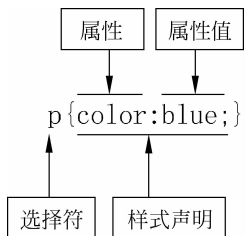


图 3-1 CSS 规则示意图

段落文本颜色是蓝色,字体大小是 15px,首行缩进 2 个字符”,那么实现这种表现形式的 CSS 规则为:

```
p{
  color:blue;
  font-size:15px;
  text-indent:2em;
}
```

CSS 规则中的选择符用于指定样式施加的对象,或者说样式的受体。正因为有了多种选择符才可以定位到页面中的任意位置,实现表现与内容的分离,否则只能使用行内的 CSS,理解和灵活使用选择符是理解 CSS 的核心内容。

样式声明(列表)则是明确应用规则后会呈现的效果,或者说受体的表现形式,每条声明的属性和属性值之间用英文的冒号隔开,而每条声明之间用英文的分号隔开。

属性是用来确定要改变元素的哪一个方面(如边框、颜色、背景等),每个元素都有很多通过 CSS 设置的属性,这些属性根据元素而不同,如可以给段落元素设置文本的 color 和 font-size 属性,但是不能给图像元素设置这些属性。

属性值则是设置属性是多少或者怎么样,CSS 规则中的属性值可分为以下 3 类。

1. 单词

如{float:left; position:absolute;color:red;}CSS 规则中的“left”、“absolute”和“red”就是属性值是单词的情形,如果一个属性对应的属性值是单词,那么这些属性值是 CSS 标准预先规定好的,只能从现有的选项中按需选取,不能自定义。

2. 数字值

数字值用于定义元素的宽度、高度或者粗细。实际中根据需要,有时使用绝对值,单位一般是像素 px(如 width:600px;);有时使用相对值,单位可以是%(如 width:50%;表示宽度是父元素的一半)或者 em(如 text-indent:2em;表示文本缩进 2 个字符的宽度),这里 em 是与字体大小有关的度量单位,其计算依据是一种字体的字符宽度,也就是说,em 的大小依据使用字体的不同而不同。

与字体大小相关的度量单位还有 ex,ex 是指给定字体小写字母 x 的高度。

3. 颜色值

在 CSS 中,经常使用颜色值来指定文本颜色、背景颜色和边框颜色。颜色值可采用诸如“red”、“blue”等表示颜色名的单词,但是这有很大的局限性,因为 W3C 标准只规定了 16 中颜色名(aqua、black、blue、fuchsia、gray、green、lime、maroon、navy、olive、purple、red、silver、teal、white、yellow),而且同一个颜色名的浏览效果还会因浏览器的不同而有些许差异。

由于计算机/电视等显示屏是采用 RGB 的颜色模式,即颜色由红色 Red、绿色 Green 和蓝色 Blue 这三原色混合而成(所有颜色混合是白色,去掉所有颜色即为黑色),实际中常使用以 # 开头的十六进制定义的颜色值 #RRGGBB,每种颜色 RR、GG、BB 均由 8 位的二进制数来描述,其取值范围是 0~255,其中 0 对应的十六进制表示为:#00,255 对

应的十六进制表示为：#FF；或者用 rgb(R,G,B)表示颜色值,这里 R、G、B 的取值范围是 0~255。如红色可以是“#ff0000”、“#FF0000”、“rgb(255,0,0)”或者“RGB(255,0,0)”，表 3-1 表示白色、黑色、三原色和三次色(由三原色中相邻两种颜色合成的颜色,共三种,分别为黄色 yellow、水绿 aqua 和紫色 purple)颜色值的几种表示方式。

表 3-1 颜色值的几种表示方式

颜色	#RRGGBB	rgb(R,G,B)	单 词
黑色	#000000	rgb(0,0,0)	black
红色	#FF0000	rgb(255,0,0)	red
绿色	#00FF00	rgb(0,255,0)	green
蓝色	#0000FF	rgb(0,0,255)	blue
黄色	#FFFF00	rgb(255,255,0)	yellow
水绿	#00FFFF	rgb(0,255,255)	aqua
紫色	#FF00FF	rgb(255,0,255)	purple
白色	#FFFFFF	rgb(255,255,255)	white

需要说明的是：当用 #RRGGBB 表示颜色值时,如果满足第 1 位与第 2 位相等、第 3 位与第 4 位相等、第 5 位与第 6 位相等这三个条件,可缩写成 3 位,如“#FF0000”、“#9988CC”可分别简写为“#F00”、“#98C”。当使用 Adobe Dreamweaver 工具制作网页时,利用其提供的智能工具选取颜色时,会弹出如图 3-2 所示的 Web 安全色面板,其特点是颜色值中只使用了 0、3、6、9、C 和 F 这 6 个数字或字母,而且都采取诸如“#33F”这样的三位缩写形式,这就是所谓的 Web 安全色(共 216 种颜色),考虑到现在的显示设备,颜色的选取完全不受这种 Web 安全色的限制,可以在设计中应用任何颜色,关于配色方面的知识请参阅其他相关文档。

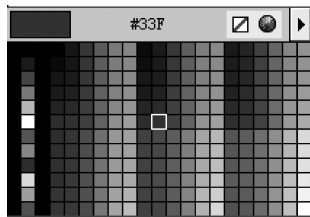


图 3-2 Web 安全色面板

3.1.2 标签选择符和组选择符

标签选择符和组选择符的介绍利用素材“ch3\exp3_1.html”来辅助,图 3-3 是素材文件的代码片段。

```
<html>
<body>
  <h1>第 1 个标题 </h1>
  <p>第 1 个段落 </p>
  <p>第 2 个段落 </p>
  <h2>第 2 个标题 </h2>
  <p>第 3 个段落 </p>
</body>
</html>
```

图 3-3 素材的代码片段

(1) 体会浏览器对各个 HTML 元素都有默认的样式(可比较 IE 和 Firefox 的浏览效果)。

查看素材“ch3\exp3_1.html”的浏览效果,留意浏览器对标题和段落默认的字体颜

色、大小、对齐和间距等样式。

(2) 利用自定义 CSS 指定元素的样式。

例 3-1 素材“ch3\exp3_1.html”，将页面中 3 个段落的文本颜色设置为红色，第 1 个、第 2 个标题的颜色为蓝色。

对应的 CSS 语句是：

```
p{color:red;}
h1,h2{color:blue;}
```

(3) 在页面中利用嵌入式引入 CSS。要使设置好的 CSS 发挥作用，需要将上述两条 CSS 语句引入页面中，引入的方式有多种，若这两条语句以独立的 CSS 文件存在，则需要用 `<link>` 标记引入，这是最常采用的方式，将在本章后续的内容中学习；若这两条语句直接放在页面文件中（即与 HTML 元素在同一个文件），则需要用 `<style>` 标记带入，这就是所谓的嵌入式引入 CSS，具体如图 3-4(b) 所示。

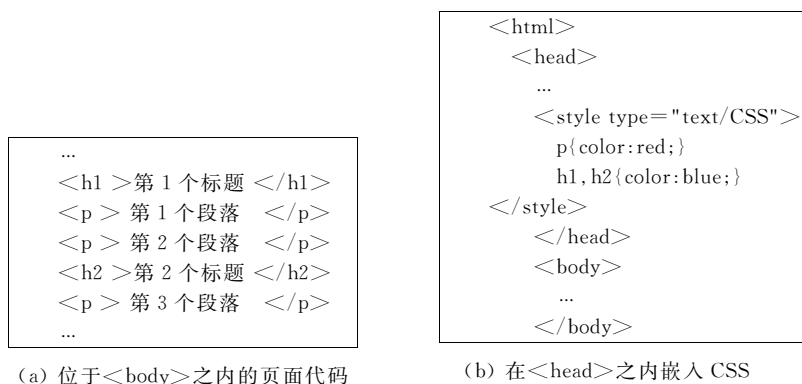


图 3-4 在页面中采用嵌入式 CSS

浏览在素材“ch3\exp3_1.html”嵌入了 CSS 样式的网页，体会 CSS 对页面效果的影响，在这一过程中，用到了标签选择符和组选择符。

① 标签选择符。标签选择符是利用标签设置样式，其形式是：标签名{样式表}，如“`p{color:red;}`”就是利用 `p` 作为选择符，实现所有 `p` 标记中的文本颜色是红色的样式效果。

② 组选择符。当不同的标记具有相同的样式需要时，可采用组选择符将这些标记看成一组，每个标记可看成组成员，组成员之间用英文逗号隔开，如“`h1,h2{color:blue;}`”的 CSS 规则实现了标题 `h1` 和 `h2` 的文本颜色均为红色的样式效果，它等价于以下两条 CSS 规则：

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

组选择符中的组成员可以是标记、后面即将学习的 `id` 选择符、`class` 选择符、伪类或者伪元素等。

3.1.3 相邻选择符、id 选择符和类选择符

例 3-2 基于素材“ch3\exp3_2.html”，将 HTML 代码片段第 3 个段落的字体颜色设置为绿色。

该例要求更细化，可采用以下几种方式实现。

方式 1: `h2+p{color:green;}`

方式 2: 给元素添加 id 属性定位。

(1) 给第 3 个 `<p>` 添加 id 属性，即 `<p id="p3">` 第 3 个段落 `</p>`。

(2) 添加 CSS 规则: `p#p3{color:green;}`

方式 3: 给元素添加 class 属性定位。

(1) 给第 3 个 `<p>` 添加 class 属性，即 `<p class="pg">` 第 3 个段落 `</p>`。

(2) 添加 CSS 规则: `p.pg{color:green;}`

为了依次单独测试以上三种方式，可利用 CSS 语法注释语句的语法，即“`/* 被注释的 CSS 样式 */`”。如图 3-5 所示的 CSS 片段，两条被注释的 CSS 样式就不起作用了。在编写 CSS 样式时，不管是调试还是加强代码可读性会经常用到代码的注释（注意，要区分注释 HTML 代码的“`<!-- 被注释的 HTML 代码 -->`”）。

```
<html>
  <head>
    ...
    <style type="text/CSS">
      p{color:red;}
      h1,h2{color:blue;}
      /* h2+p{color:green;}
      p#p3{color:green;} */
      p.pg{color:green;}
    </style>
  </head>
  <body>
    ...
  </body>
```

图 3-5 CSS 中的代码注释

方式 1 用到的选择符叫相邻选择符（IE 6 和 IE 7 不支持），这里的 `h2+p` 是指紧邻 `<h2>` 标记且与 `<h2>` 同级的第一个段落。

方式 2 和方式 3 涉及一个非常重要且常见的现象，即在 HTML 代码中定义 id 和 class 属性和在 CSS 中引用其属性值，要遵守 XHTML 如下规定。

(1) id 和 class 的属性值只能是以字母开头的字母数字下划线系列。否则就是无效的属性值，予以忽略。

(2) 对应的引用：在 id 和 class 的属性值前分别加上英文的井号 # 和点号 .，即：“#id 属性值”和“. class 属性值”。

(3) 同一页面中，id 的属性值具有唯一性（即只能用于定义一个元素），class 的属性

值表示一类(即可用于定义多个元素)。如图 3-6(a)所示的 id 定义是违规的。

<pre>... <h1>第 1 个标题 </h1> <p id="p3">第 1 个段落 </p> <p>第 2 个段落 </p> <h2>第 2 个标题 </h2> <p id="p3">第 3 个段落 </p> ...</pre>	<pre>... <h1>第 1 个标题 </h1> <p class="pg">第 1 个段落 </p> <p>第 2 个段落 </p> <h2>第 2 个标题 </h2> <p class="pg">第 3 个段落 </p> ...</pre>
--	--

(a) 错:因 id 的属性值具有唯一性

(b) 对:因 class 的属性值表示一类

图 3-6 id 和 class 属性对比

若多个选择符同时都指向一个位置,如图 3-7(b)所示三个选择符“h2+p”、“p#p3”和“p.pg”都是指向图 3-7(a)中的第 3 个段落,若再考虑浏览器对<p>标记默认的样式,那么就有 4 个字体大小施加于第 3 个段落<p>标记中的文字,这 4 个选择符中“p#p3”的级别最高。所以最终起作用的声明是“p#p3{font-size:35px;}”,即第 3 段落的字体大小是 35px。这里就是 CSS 层叠性规则在起作用了,总的原则是:针对性越强,级别越高。CSS 层叠性规则具体体现如下。

(1) id 的针对性高于类。如“p#p3{font-size:35px;}”优先于“p.pg{font-size:50px;}”。

(2) 类的针对性高于位置标记。如“p.pg{font-size:50px;}”优先于“h2+p{font-size:10px;}”。

(3) 位置标记针对性高于普通标记。如“h2+p{font-size:10px;}”优先于浏览器对 p 标记的默认字体大小(绝大多数浏览器默认的 p 元素文本大小约为 16px)。

(4) 显示定义标记的针对性高于默认样式。如图 3-7(b)中声明“p{color:red;}”产生的效果是:浏览时图 3-7(a)中第 1 段、第 2 段落的文本是红色,覆盖了浏览器默认的所有文本是黑色的设置。

<pre>... <h1>第 1 个标题 </h1> <p>第 1 个段落 </p> <p>第 2 个段落 </p> <h2>第 2 个标题 </h2> <p id="p3" class="pg">第 3 个段落</p> ...</pre>	<pre><html> <head> ... <style type="text/CSS"> p{color:red;} h1,h2{color:blue;} h2+p{font-size:10px;color:blue;} p#p3{font-size:35px;} p.pg{font-size:50px;} </style> </head> <body> ... </body></pre>
--	--

(a) 位于<body>内的页面代码

(b) 在<head>内嵌入 CSS

图 3-7 层叠现象:多个 CSS 样式声明施加于同一个 html 元素

对于初学者,在实际的体验中可通过依次注释优先级别较高的 CSS 样式声明来测试上述规则,如图 3-7(b)所示的样式声明中出现了典型的层叠现象,最终发挥作用的样式声明是针对性最高的选择符后面的样式声明,即:“p#p3{font-size:35px;}”。

3.1.4 后代选择符

后代选择符的讲解基于图 3-8 所示的页面代码片段。

```

...
<body>
  <h1>第 1 个标题 </h1>
  <p class="p1">第<span>1</span>个段落 </p>
  <p>第 2 个段落 </p>
  <h2>第 2 个标题 </h2>
  <p id="p3">第<span>3</span>个段落 </p>
</body>

```

图 3-8 代码片段

图 3-8 代码片段的层次结构可用图 3-9 表示。

说明: <body> 标记包含 5 个子标记,分别是 <h1>、<p>、<p>、<h2> 和 <p>,这 5 个元素之间是同级(即兄弟关系),都是 <body> 的子标记(或子元素或后代元素),或者说 <body> 是它们的父标记(或父元素),这些标记又都是根元素 <html> 的后代元素。

第 1 个和第 3 个 <p> 标记又都包含子元素 标记,是 的父元素; <body> 是 的祖元素, 是 <body> 的孙元素,所以除根元素 <html> 外,其余的 html 标记很多时候具有后代元素或父元素、祖元素等多重身份,取决于具体的参照标记。

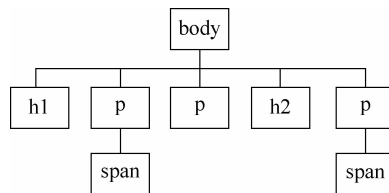


图 3-9 代码片段的层次结构

例 3-3 基于素材“ch3\exp3_3.html”,将页面中第 1 段和第 3 段的数字 1 和 3 的文本背景颜色分别设置为绿色和蓝色。

可采取以下两种方式。

(1) 采用后代选择符定位,并为其设置背景样式,即在 <style> 和 </style> 标记之间增加以下两行样式声明:

```

p. p1 span{background-color:green;}
p#p3 span{background-color:blue;}

```

从上述代码可以看出,后代选择符的要点是标记之间用空格隔开(前面学到的组选择符是标记之间用英文逗号隔开)。

(2) 分别给 标记添加不同的 id 或者 class 属性,然后引用属性值设置背景样式。

3.1.5 通用选择符

前面反复提及每个浏览器都有一个默认的样式表,不同浏览器之间的样式表存在些许不同,自定义样式就是希望让浏览的效果在主流浏览器上显示一致,要实现这个目标有很多技术细节需要处理,使用通用选择符 * 是最常采用的方式之一,这里的通用选择符 * 是指 HTML 文档中的所用标记。

浏览器给块元素定义了一个默认的外边距 margin 和内边距 padding,多数情形需要重新定义这些内、外边距,为消除浏览器默认的内、外边距,可以使用 * 进行样式设置,有时页面中绝大多数的文本大小相同,也可使用 * 来设置,总之用 * 来设置页面中共性比较多的基础样式。

例 3-4 基于素材“ch3\exp3_4.html”,利用通用选择符 * 实现:使页面中所有元素的内、外边距均设置为 0,所有文本颜色是红色,字体大小是 13px。

实施过程如下。

(1) 先浏览 exp3_4.html 的页面效果。

(2) 使用记事本编辑页面文档,在<style>和</style>之间增加如下样式:

```
* {  
    margin:0;                /* 取消所有元素四周的外边距 */  
    padding:0;              /* 取消所有元素四周的内边距 */  
    color:red;  
    font-size:13px;  
}
```

保存,浏览效果。

(3) 若继续添加样式: h2{color:blue;} ,那么<h2>标记的内容就会显示蓝色。这还是层叠规则的效用,因为 h2 的针对性比 * 高,所以一般在样式声明设置的最开始处使用通用选择符 * 来设置样式。

综上所述,本小节学习的 CSS 选择符有以下几种。

- (1) 通用选择符 *。
- (2) 标记选择符。
- (3) 后代选择符。
- (4) 类选择符。
- (5) id 选择符。
- (6) 相邻选择符。

对于上述选择符按照“针对性越强级别越高”原则,具体如下。

- (1) 选择高于继承。
- (2) ID 选择符高于类选择符,类选择符高于标签选择符。
- (3) 标签 #id 高于 #id,标签. class 高于. class。

3.1.6 伪类

类是在标记的开始部分添加 class 属性定义的,伪类(pseudo-class)是能够实现好像

给元素添加了新的类但实际上并没有给元素添加类的效果。通过伪类可以在某个事件发生时,将样式应用到元素上,最常见的就是当鼠标指向或者悬停于链接元素上所发生的变化,伪类还能基于标记中某些匹配的条件来应用样式。

1. 对<a>标记的某些事件应用样式的伪类

- (1) a:link: 设置未被访问的超链接的样式。
- (2) a:visited: 设置已被访问的超链接的样式。
- (3) a:hover: 设置悬停时的超链接的样式,使用频率最高。
- (4) a:active: 设置单击鼠标但未释放时的超链接的样式。

如以下样式可实现<a>标记在悬停状态时实现取消下划线和颜色变蓝的效果:

```
a:visited{
    text-decoration:none;
    color:blue;
}
```

注意:目前主流浏览器(若为IE,需8.0以上)不仅支持<a>标记的悬停效果,也支持其他任何元素的悬停效果的实现,如p:hover和img:hover等,这在后面章节的实例中会多次用到。

2. 基于某些匹配的条件应用样式的伪类

(1) :first-child: 选取作为其他元素的第1个子元素。如设置列表中的第1个选项为红色,可直接添加“li:first-child{color:red;}”的CSS规则。

(2) :focus: 选取获取焦点的元素。如用户单击输入框时需要给输入框添加蓝色边框的效果,添加“input:focus{border:1px blue solid;}”的CSS规则。

3.1.7 伪元素

伪元素是能够实现好像在文档中添加了额外元素但其实并没有添加额外元素的效果。以::first-letter为例,假如要将h2中的首字符设置成其他字符的2倍,只要简单地添加“h2::first-letter{font-size:2em;}”CSS规则即可,就好像是添加了如下情形的span元素和CSS规则(这就是对“伪元素”的诠释):

```
h2::first-letter{font-size:2em;}
<h2><span class="first-letter" >R</span>oad to success!</h2>
```

即好像在文档中添加了一个类似span的元素,将首字符“R”单独标识,其实不需要,这就是伪元素所产生的效果。

还可以将伪元素理解为通过它来选取文本的部分内容,如文本的首字符或者首行,但是对诸如或<a>的行内元素不能使用伪元素。

常用的伪元素如下。

(1) ::first-letter: 如p::first-letter{font-size:2em;float:left;}可以实现如图3-10所示段落首字下沉的效果。在实际应用时,无须在伪元素使用双冒号,使用单冒号即可。

(2) ::first-line: 如“p::first-line{font-style:italic;font-size:1.5em;}”的CSS规则

可以实现段落首行文本产生倾斜,字号大 150%的效果,如图 3-11 所示。

山高水远,草长鹰飞,辽远的牧歌伴随着白云在悠远的晴空自由地飘飞。当岁月随着古老的日落日出而流走,在这里,炊烟依然安详,羊群依旧平静,仿佛所有的静穆都是一种隐预和暗示,这就是美丽富饶的黄河大草原。

图 3-10 伪元素: `first-letter` 的效果

山高水远,草长鹰飞,辽远的牧歌伴随着白云在悠远的晴空自由地飘飞。当岁月随着古老的日落日出而流走,在这里,炊烟依然安详,羊群依旧平静,仿佛所有的静穆都是一种隐预和暗示,这就是美丽富饶的黄河大草原。

图 3-11 伪元素: `first-line` 的效果

3.1.8 多类

多类在语法上是 `class` 的属性值中包含多个词,词之间以空格隔开,如图 3-12 所示。

上述代码中每个 `<p>` 标记都定义了两个类,这样做的好处是这 3 个段落相同的样式通过引用类名 `pp` 来设置,不同的样式引用各自不同的类名来实现。

例如,要求设置 3 个段落的字体颜色都是红色,字体大小分别是 16px、18px 和 14px,可添加如图 3-13 所示的 CSS 规则。

```
...
<body>
...
<p class="pp p1"> 第 1 个段落 </p>
<p class="pp p2"> 第 2 个段落 </p>
<p class="pp p3"> 第 3 个段落 </p>
</body>
...
```

图 3-12 多类的定义

```
<style type="text/CSS">
.pp{color:red;}
.p1{font-size:16px;}
.p2{font-size:18px;}
.p3{font-size:14px;}
</style>
```

图 3-13 多类的引用

注意: `class` 中所定义的多个类名用空格隔开,类名前后顺序不影响类名的引用。

多类只是提供了定位元素的另一种选择方式,图 3-12 中利用多类的定位方式也可以用 `id` 和类共用的方式实现,如图 3-14 所示。

对于图 3-14 的属性变更,要实现相同的效果对应的 CSS 规则如图 3-15 所示,请再次留意对 `class` 名和 `id` 名的引用。

```
...
<body>
...
<p class="pp" id="p1"> 第 1 个段落 </p>
<p class="pp" id="p2"> 第 2 个段落 </p>
<p class="pp" id="p3"> 第 3 个段落 </p>
</body>
...
```

图 3-14 `id` 和类共用

```
<style type="text/CSS">
.pp{color:red;}
#p1{font-size:16px;}
#p2{font-size:18px;}
#p3{font-size:14px;}
</style>
```

图 3-15 CSS 规则

一个元素可以使用多个类,一个类可以用于多个元素,听起来有点拗口,但在实际中频繁使用。