WEB 第3章 DESIGN

CSS基础

本章学习目标

- CSS 概念、CSS 语法规则
- CSS 选择器
- CSS 常用属性与示例调试
- CSS 应用示例

本章首先向读者介绍 CSS 的概念在页面中引用 CSS 的方法以及 CSS 的语法规则,紧接着介绍 CSS 选择器的相关知识以及常用的 CSS 属性与示例调试,最后讲解三个简单的 CSS 应用实例。

3.1 CSS 概述

CSS(cascading style sheets)是层叠样式表。CSS2规范于 1998 年 2 月通过 W3C 的审核与推荐。CSS 并不是专为 XHTML 所设计的,还可以被其他标记语言拿来制作排版样本,如 HTML、XML 文件都可以用 CSS 来美化页面的设计。学会了 CSS,在 HTML、XHTML 和 XML 文件中都可以使用。CSS 的功能在于为 HTML 元素设置样式,构建布局合理、样式美观的页面。

3.1.1 CSS 简介

1. CSS 的产生缘由

HTML标签原本用于定义文档内容,但是浏览器默认的显示样式不美观,不能够满足 人们需求。Netscape和IE意图通过不断增加用于定义显示格式的标签和属性来解决显示 样式的问题,例如<center>、等格式标签及align、color等属性,结果仍不能满足美 观页面的设计需要,反而使得混杂着文档格式标签的页面结构越发复杂,难以维护。为此, W3C在HTML之外创造出了CSS,解决了上述问题。

W3C在HTML4之外创造出了CSS,并废弃了许多用于控制显示格式的标签和属性。 标签和属性被废弃就意味着在未来版本的HTML和XHTML中将不再支持这些标签和属 性,所以在网页设计中应尽量避免使用这些被废弃的标签和属性。表 3.1 列出了一些应避 免使用的标签和属性,这些标签和属性的功能将由专门应用于文档表现层的CSS替代。比 如例 2-30 中虽然起到作用了,但不建议使用,推荐使用 CSS 方 法替代。

类型	标签或属性	描 述
标签	< center >	定义居中的文本内容
标签	< font >,< basefont >	定义 HTML 字体
标签	< s >,< strike >	定义删除线文本
标签	< u >	定义下画线文本
属性	align	定义文本的对齐方式
属性	bgcolor	定义背景颜色

表 3.1 一些应避免使用的 HTML 标签和属性

2. 引用 CSS 的大体步骤

引用 CSS 的步骤如下所示。

(1) 查看没有使用 CSS 前元素的默认方式。

(2)理解要达到的效果,选择合适的 CSS 属性,必要时修改 HTML 代码。

(3)选择恰当的引入方式,检验效果,并考虑设置 CSS 后对其他相关元素的影响,限定 CSS 规则的影响范围。CSS 的设计如果仅用于一个页面较容易;如果一个 CSS 文件要在网站的 100 个页面中使用,又要这 100 个页面中的 HTML 相关元素显示效果互不影响就 很难了。

3. CSS 语法

CSS 语法的基本格式如下,属性和属性值之间以冒号隔开,每条样式规则都要以分号 结束:

属性 1:属性值 1;属性 2:属性值 2;

这里要特别强调,冒号和分号必须是英文输入法下输入的;属性名是约定好的,不能随 意书写,多或者少一个字母是不允许的,一般都是小写的;属性值也是有取值范围的。CSS 的属性值没有加双引号,在CSS前期学习中,书写错误会影响学习的进度。

3.1.2 CSS 的引用方式及特性

CSS 的引用方式有内联样式、内部样式、外部样式三种,其 CSS 语法表现形式略有不同,具体如下。

内联样式:

style = "属性名 1: 属性值 1; 属性名 2: 属性值 2;"

内部样式和外部样式:

选择器 1{	/ * 选择页面中的元素 * /
属性名 1:属性值 1;	/*一条样式规则 */
属性名 2:属性值 2;	/*又一条样式规则 */
}	

网页设计与制作实践

第3章 CSS基础

其中,选择器用于指定选择哪个或者哪些 HTML 元素,选择器选中的 HTML 元素使 用花括号中的样式规则; "/*"和"*/"之间的内容是 CSS 注释。下面通过具体示例进 行讲解。

1. 内联样式

内联样式又称为行内样式,它定义在 HTML 元素的开始标签里。使用 style 属性设置 样式规则,并且样式规则仅对当前的 HTML 元素有效果。

【例 3-1】 CSS 内联样式示例。

```
01 <! DOCTYPE html >
02 < html >
03
     < head >
04
         <title>CSS 应用示例</title>
         < meta charset = "utf - 8" />
05
     </head>
06
07
     < body >
         猴子
80
09
         <J</p>
         <牛</p>
10
11
         < span >落叶松</ span >
         < span >白杨树</ span >
12
13
         < span >菊花</ span >
         < span >荷花</ span >
14
15
     </body>
16 < /html >
```

代码第 8 行对段落"猴子"设置了样式,效果如图 3.1 所示。这里 color:red 表示设置颜 色为红色,font-size:25px 表示设置字体大小为 25px,text-align:center 表示居中对齐。从 图中可以看出,仅段落"猴子"为居中、红色,字体比其他文字大,其他文本呈现为默认显示 样式。



图 3.1 CSS 的内联样式

2. 内部样式

内部样式表是指将页面中需要应用的所有样式规则集中定义在页面头部分。使用 < style >样式标签定义,< style type="text/css">与</style >之间定义样式规则。



```
【例 3-2】 CSS 内部样式示例。
```

网页设计

与制作实践

```
01 <! DOCTYPE html >
02 < html >
03
      < head >
04
          <title>CSS 内部样式示例</title>
05
          < meta charset = "utf - 8" />
          < style type = "text/css">
06
07
          р {
80
              font - size:20px;
09
             text - align:center;
10
          }
          </stvle>
11
12
      </head>
13
      < body >
          猴子
14
15
          < a>>马</a>>
          <牛</p>
16
17
          < span >落叶松</ span >
18
          < span >白杨树</ span >
19
          < span >菊花</ span >
20
          < span >荷花</ span >
21
      </body>
22 </html>
```

代码第 6~11 行的< style >标签位于< head >与</head >之间,样式中的 p{…}表示要 对页面中所有的段落元素定义显示样式。效果如图 3.2 所示。这里 font-size:20px 表示设 置字体大小为 20px,text-align:center 表示居中对齐。



图 3.2 CSS 的内部样式

从图中可以看出,所有段落都居中。要注意此种内部样式的写法影响的范围是所有段 落。段落"猴子"没有受 font-size: 20px 的影响,采用的是 font-size: 25px。这是 CSS 的一个特点,相同的属性名称,离元素最近的起作用。

3. 外部样式

外部样式将网页的样式信息与网页内容完全分离开来,网页本身存储的是文档内容,而 内容如何呈现是在外部的样式表文件中定义的。样式文件也是普通的文本文件,它的后缀 名为".css",注意存储编码为"utf-8",使用 Notepad++新建、编辑、存储为 CSS 类型文件即 可;使用普通的文本编辑器也可以打开。由于样式文件与页面文件分开存储,我们需要在 页面文件的头部使用<link>标签将定义好的样式导入。

样式文件的组织结构如图 3.3 所示,其中 ch3_03.html 与 ch3_03.css 同在目录 ch3_03 内。



图 3.3 外部样式文件的组织结构

【例 3-3】 CSS 外部样式示例。

ch3_03.html 文件内容如下:

```
01 <! DOCTYPE html >
02 < html >
03
      < head >
          <title>CSS 外部样式示例</title>
04
05
          < meta charset = "utf - 8" />
06
      k href = "ch3 03.css" rel = "stylesheet" type = "text/css">
      <!-- 引入文件 ch3 03.css-->
          < style type = "text/css">
07
08
          р{
09
              font - size:20px;
10
              text - align:center;
11
          }
12
          </style>
      </head>
13
14
     < body >
          猴子
15
          <J</p>
16
          <牛</p>
17
          < span >落叶松</ span >
18
19
          < span >白杨树</ span >
20
          < span >菊花</ span >
21
          < span >荷花</ span >
      </body>
2.2.
23 </html>
ch3_03.css 文件内容如下:
01 p {
02
      color: blue;
03
      font - size: 12px; }
04 span {
05
      font - size: 12px;
06
      color: red; }
```

例 3-3 运行如图 3.4 所示。ch3_03. html 文件代码第 6 行使用< link >标签引入的外部 样式表 ch3_03. css,其中 href 属性指定了要引入的样式表文件的路径。ch3_03. css 文件代 码第 3 行 font-size: 12px,ch3_03. html 文件代码第 9 行 font-size: 20px,都设置字体大小, 第3章 CSS基础

从运行效果看是 font-size: 20px 在起作用。试着修改 ch3_03. html,把代码第 6 行移动到 代码第 12 行后,再观察一下运行结果,会发现 font-size: 12px 在起作用。所以在引入多个 CSS 文件时,我们要注意 CSS 文件的先后顺序。



图 3.4 外部样式

4. CSS 的特性

网页设

计与制

的作实践

CSS 有以下两种特性。

(1) 传递(继承性)原则。父元素设置的样式规则会传递给子元素。

(2) 就近原则。如果没有为 HTML 元素定义任何样式,那么元素将按照浏览器的默认 设置显示,如第2章中所有示例都是按照浏览器的缺省设置显示的。

事实上,允许同时以上述几种方式规定样式信息。也就是说,样式可以写在单个的 HTML 元素内,也可以规定在 HTML 的头部分,或者定义于一个外部的 CSS 文件中。如 果同一个 HTML 元素同时通过上述多种方式定义了样式规则,例如在文件 ch3_03. html 中,我们对段落"猴子"在内联样式中设置了 font-size: 25px,在内部样式中设置了 font-size: 20px,在外部样式 ch3_03. css 中设置了 font-size: 12px,那么最终呈现在页面上的效果到底 是使用的哪一条规则呢? 这些层叠的规则一般将会按照就近原则,按从上到下的顺序,离元 素最近的起作用。一般情况下,我们要避免在多处对相同的属性名设置不同的属性值,这也 是 CSS 设计的一个难点,需要通过文档来约定。使用 Firefox 浏览器,按 F12 键调试可以查 看调试 CSS,选中段落"猴子",其调试如图 3.5 所示,画横线的代码表示该 CSS 没有起 作用。



图 3.5 CSS 调试

3.2 CSS 定义选择器

在 CSS 语法中,"选择器"指明了花括号中所定义的样式规则的作用对象,也就是样式规则对页面中的哪些元素起作用。下面介绍一些常用的 CSS 选择器。

3.2.1 标签选择器

标签选择器是对 HTML 元素标签应用相应的 CSS 样式。假设对标签设置 CSS,则标签选择器的写法为:

在例 3-2 中, CSS 内部样式采用标签选择器对设置了样式, 具体如下:

其作用范围是所有的标签。从图 3.2 可以看出段落"马"和段落"牛"都按标签选择器设置的 CSS 显示,段落"猴子"因为设置了内联样式,覆盖了标签选择器的 显示效果。

3.2.2 id 选择器

我们可以对 HTML 元素定义一个 id 属性,然后根据属性 id 选择元素, id 属性值就是这个元素的唯一标识符。不允许在一个页面中对多个 HTML 元素定义相同的 id 属性值。

设某标签元素 id="demo",则其属性值为 demo,其 id 选择器的写法为:

```
      # demo {
      / * 选择页面中 id = "demo"的元素 * /

      属性名 1: 属性值 1;
      / * 一条样式规则 * /

      属性名 2: 属性值 2;
      / * 又一条样式规则 * /

      }
```

复制 ch3_01. html,重命名为 ch3_04. html,其内容见例 3-4。代码第 14 行设置了 id= "houzi"来唯一标识段落"猴子";代码第 6~11 行为 CSS 内部样式;第 7 行" # houzi"定义 了 id 选择器,具体 CSS 规则不再赘述。其运行效果见图 3.1,建议使用 id 选择器替换内联 样式,而不要在标签中设置 style 属性。

【例 3-4】 id 选择器示例。

```
01 <! DOCTYPE html >
02 < html >
03 < head >
```



第3章 CSS 基

```
04
          <title> id 选择器示例</title>
05
           < meta charset = "utf - 8" />
          < style type = "text/css">
06
07
           # houzi {
80
              color:red;
09
              font - size:25px;
10
               text - align:center;}
11
           </stvle>
12
      </head>
      < body >
13
          猴子
14
15
          <只
          <牛</p>
16
17
          < span >落叶松</ span >
          < span >白杨树</ span >
18
19
          < span >菊花</ span >
20
          < span >荷花</ span >
21
      </body>
22 </html>
```



网页设

计

与制作实

践

3.2.3 class 选择器

与 id 属性一样, HTML 中的元素可以定义一个 class 属性。不同的是, 多个不同的 HTML 元素可以定义相同的 class 属性值。class 属性值相同的元素可以由 CSS 类选择器 选取并定义相同的样式规则。

设某标签元素 class="demo",则其属性值为 demo,其 class 选择器的写法为:

```
. demo { /*选择页面中 class = "demo"的元素 * / 
属性名 1: 属性值 1; /* 一条样式规则 * / 
属性名 2: 属性值 2; /* 又一条样式规则 * / 
}
```

假如我们的目标是设置< span >落叶松、< span >白杨树的字体大小为 25px,则可以使用 class="shu",把"落叶松"和"白杨树"归到 shu 类,并进一步设置 CSS 规则。

复制 ch3_04. html,重命名为 ch3_05. html,其内容见例 3-5。代码第 18、19 行设置了 class="shu"来标识"落叶松"和"白杨树",代码第 11、12 行". shu"定义了 class 选择器,具体 CSS 规则不再赘述,其运行效果见图 3.6,这里 class 选择器选中了两个< span >元素,另外 两个< span >元素"菊花"和"荷花"不受影响。class 选择器还可以和标签选择器结合使用,用于更精确地选择元素。例如选择器" span. shu"表示选择所有 class 属性值为 shu 的 < span >元素。

【例 3-5】 class 选择器示例。

```
01 <!DOCTYPE html>
02 < html>
03 < head>
04 < title> class 选择器示例</title>
```

```
< meta charset = "utf - 8" />
05
06
           < style type = "text/css">
07
           # houzi {
08
               color:red;
               font - size:25px;
09
10
               text - align:center;}
11
           .shu {
12
               font - size:25px;
                                   }
13
           </style>
14
      </head>
      < body >
15
           猴子
16
           <只
17
           <牛</p>
18
           < span class = "shu">落叶松</span>
19
20
           < span class = "shu">白杨树</span>
           < span >菊花</ span >
21
22
           < span >荷花</ span >
23
      </bodv>
24 </html>
```

	class选择器示例	N X	+	-		×
$\langle \epsilon \rangle$	→ C' @	[] file:///D:/书稿	》/第三章/代码/ch3_05.html		»	≡
		X	卖子			î
马						
4						
落	叶松白杨	杨树菊花荷花				~

图 3.6 class 选择器

3.2.4 后代选择器

后代选择器也称为包含选择器,用来选择特定元素或元素组的后代。后代选择器用两 个常用选择器中间加一个空格表示,其中前面的常用选择器用于选择父元素,后面的常用选 择器用于选择子元素,样式最终会应用于选定的子元素。一般可结合 DOM 来表示后代选 择器。

复制 ch3_05. html,重命名为 ch3_06. html,其内容见例 3-6。在 Dreamweaver CC 中打 开,ch3_06. html 对应的 DOM 见图 3.7。结合代码第 7 行,可知 body ♯ houzi 的 body 为父 标签,♯ houzi 是在 body 的子标签层选择的。body span 是典型的写法,是使用 Dreamweaver CC 设计 CSS 时的默认写法,表示选择 body 下的所有 span 标签。因为 body 是默认的父标签,所以经常省略。需要特别注意的是父子标签之间有空格,书写时容易出 错。例 3-6 的运行效果见图 3.6。



第3章 CSS 基础

WEB 网页设计与制作实践

【例 3-6】 后代选择器示例。

01	DOCTYPE html
02	< html >
03	< head >
04	<title>class 选择器示例</title>
05	<meta charset="utf-8"/>
06	< style type = "text/css">
07	body #houzi {
08	<pre>color:red;</pre>
09	<pre>font - size:25px;</pre>
10	<pre>text - align:center;}</pre>
11	body span. shu {
12	<pre>font - size:25px; }</pre>
13	body span { }
14	
15	
16	< body >
17	猴子
18	马
19	牛
20	< span class="shu">落叶松
21	< span class="shu">白杨树
22	< span > 菊 花
23	< span >荷花
24	

25 </html>

DOM	资源	代码片断		
~ [h	tml			
~	head	D		
	tit	le		
	m	eta		
	st	yle		
~	body	D		
	P	> #houzi		
	P	>		
	P	>		
	sp	oan).shu		
	sp	oan).shu		
	sp	oan 🔪		
t .	sp	ban		

图 3.7 ch3_06.html 对应的 DOM

3.2.5 分组选择器

当几种元素样式属性相同时,可以共同调用一个声明,元素之间用逗号分隔。 ch3_03.css 内容如下:

```
01 p {
02     color: blue;
03     font - size: 12px; }
04     span {
05         font - size: 12px;
06         color: red; }
```

可以改写为:

```
01 p{
02     color: blue;}
03 p,span { font - size: 12px; }
04     span{
05     color: red;}
```

把 font-size: 12px 放在"p, span"选择器中,表示和< span>都使用该样式规则。要 注意与后代选择器的区分,用逗号分隔的是分组选择器,表示并列;用空格分隔的是后代选 择器,表示父子关系,会进一步限定选择元素的范围。

3.2.6 通用选择器

通用选择器为"*",表示选择页面中所有元素。例如:

* { /*选择页面中所有元素 * / font-size: 12px; /* 设置字体大小为 12px * / }

表示选择页面中所有元素,将字体大小都设置为12px。

通用选择器"*"的用法一般是设置盒模型的 margin、padding 属性,以避开不同浏览器的默认值不同,提升显示的一致性。其写法如下:

* { /*选择页面中所有元素 * / margin:0px;padding:0px; }

结合 DOM 图,通用选择器"*"要放在其他选择器的上面,在 CSS 文件中一般放在最前面。打开 ch3_06_bak. html 文件,可以观看通用选择器"*"的效果。

3.2.7 伪类选择器

有时候需要用文档以外的其他条件来应用元素的样式,例如鼠标悬停、鼠标经过某个 I HTML 元素等,这时我们就需要用到伪类选择器。超链接伪类选择器的语法如下:

```
      a:link {...}
      /*选择未被访问的超链接*/

      a:visited {...}
      /*选择已访问的超链接*/

      a:hover {...}
      /*当鼠标移动到超链接上*/

      a:active {...}
      /*已选定的超链接*/
```

以上代码分别定义了未被访问的超链接样式、已访问过的超链接样式、鼠标移动到超链 接上的样式和选定的超链接样式。之所以称之为伪类,就是说它不是一个真实的类。正常

075

的类是以点开始,后边跟一个类属性值;而它是以 a 开始后边跟个冒号,再跟状态限定字符。例如在"a:hover"选择器中定义的样式,只有当鼠标移动到该链接上时它才生效,而 "a:visited"只对已访问过的超链接生效。使用伪类选择器可以让用户体验大大提高,例如 我们可以设置鼠标移动到超链接上时改变颜色或字体大小等属性来告知用户这个是可以单 击的。例 3-7 设置了鼠标移动到超链接上的样式,在 Firefox 浏览器中打开文件 ch3_07. html,按 F12 键调试,单击椭圆处的":hov"切换到伪类选择器调试,页面效果如图 3.8 所 示。当鼠标放在超链接上时,超链接文本的样式发生了相应的变化。

【例 3-7】 超链接伪类选择器示例。

```
01 <! DOCTYPE html >
02 < html >
03
      < head >
04
          <meta charset = "utf - 8">
05
          <title>伪类选择器示例</title>
06
          < style type = "text/css">
07
              a:hover {
                                /*当鼠标移动到超链接上*/
              font-size: 20px; /*设置字体大小 20px */
80
09
              color:red;
                                /*设置文字颜色为红色*/
10
              }
11
          </style>
      </head>
12
      < body >
13
          <a href = "https://www.baidu.com">百度</a>
14
15
          <a href = "http://www.tjzhic.edu.cn">天津理工大学中环信息学院</a>
16
      </body>
17 </html>
```

伪美选择器示例 ×	+
↔ ở ŵ	[] file:///D:/书稿/第三章/代码/ch3_07.html
百度天津理工大学中环信息	急学院
www.tjzhic.edu.cn	
↓ 查看器 ▷ 控制台 ▷ 调制	【器 1↓ 网络 {} 样式编辑器 ① 性能 ① 内
Q,搜索 HTML +	✓ ∀过滤样式 :hov).cls + 13
<pre>clDOCTYPE html> @ chtml> b chead> m c/head> @ w cbody></pre>	♥:hover □:active □:focus □:focus-visible □:focus-within □:visited 元录 Q (内联
<pre> </pre> (空日) (如本) fittp://www.tjchic.edu.c 天津理工大学中环信息学院	目後 } ehover ❹ { 内联:2 font-size: 20px; color: ● red; }

图 3.8 伪类选择器调试

读者可以参考 https://www.w3school.com.cn/css/css_pseudo_classes.asp 了解其他的伪类选择器,例如(:focus)。有些实现也可以借助 JavaScript 的事件来完成。

3.2.8 高级选择器

下面介绍的一些选择器在某些低版本的浏览器中可能无效,因为浏览器目前对这些选

网页设计与制作实践

择器的支持还不太完善,所以网站中功能很重要的元素应该避免使用这些高级选择器。

1. 子选择器

子选择器与后代选择器不同,子选择器仅指它的直接后代,或者你可以理解为作用于子 元素的直接后代。而后代选择器作用于所有后代元素。后代选择器通过空格来连接两个选 择器,而子选择器通过">"连接两个选择器,即"选择器>选择器"。读者可以搜索"子选择 器"来了解更多的内容。

2. 相邻同胞选择器

除了上面的子选择器与后代选择器,有时我们可能还希望找到相邻的两个元素当中的 下一个。读者可以搜索"相邻同胞选择器"来了解更多的内容。

例如,一个标题 h1 元素后面紧跟了两个段落元素,想选择位于 h1 元素之后的第一个段 落元素,并对它应用样式,此时就可以使用相邻同胞选择器,代码如下:

```
h1 + p{ /*选择紧邻 h1 元素的 p 元素 * /
font - weight: bold; /* 设置字体加粗显示 * /
}
```

3. 属性选择器

属性选择器根据元素的属性来匹配选择元素。例如通过判断 HTML 标签的某个属性 是否存在,或者通过判断 HTML 标签的某个属性是否和某个值相等来选择元素。读者可 以搜索"属性选择器"来了解更多的内容。

例如,选择具有 title 属性的超链接和选择 title 属性值为 pic 的超链接,代码如下:

a[title] {…} /*选择具有 title 属性的超链接 */ a[title = "pic"] {…} /*选择 title 属性值为"pic"的超链接 */

3.3 CSS 的字体属性

学习 CSS 的属性需要查看 CSS 参考手册,读者可以参考网页 http://css.doyoe.com/ 和 https://www.w3school.com.cn/css/css_reference.asp 来在线了解有哪些属性名称及 属性值的范围,并学习具体案例。这里仅给出常见的属性及其使用方法。

3.3.1 字体属性名称及属性值

1. 字体属性概述

我们想要学习了解图 3.9 所示的文字效果是如何形成的,就需要学习字体属性。 字体对应 font。CSS 中常用的字体属性及部分取值如下。

1) 字体属性 font-family

字体属性 font-family 用于设置文本显示时使用的字体名称,可设置多个字体名称,按优先顺序排列,以逗号隔开。

(1) 语法: font-family: 字体名称 1, 字体名称 2, …;



第3章 CSS 基

CSS字体	属性	×	+								
← → C ⁴	<u>۵</u> (] file:///D:/	/书稿/第3	三章/作	℃码/ch	13_08	/ch3_0	8.html			
振動・ 干	在古镇林	ふ柳吉谷	ነው								
					- 11.41						
協柳書镇历史	沉积久远.	文化底蕴深	夏。 明清	「日日」	1. 是·	云河	。 唐沄重	要枢纽.	. 成 3	为中国	北方
是作山休礼	了士镇佳绪	网络的旱	「古法派	TR.	50	U.IX	· =	结田河	र सन्द	¥ 1	भाग
取朱丁仲少	「日東マシレ		用 と /	-4+			· ·			=	<i>中パッ</i> リ 由
后古镇带为	、」 系木加	1#0 <u>1</u> , 7	*月 J 1	ŦSC.	XH4	和氏	卿之.	木, 走		見ロソ4	150
	く」 京木 の 	D 调试器	*月」! ↑↓ 网络	{}	又化 初 样式编辑	相氏	同之の曲	不,走 11 ① 内	存	実の が	#5₀ ≋ ††
第日項第3 日 日 日 日 日 日 第 第 3 日 県 常 メ の 査 着 常 れ の 、 の う 査 着 常 の 、 の う う 香 常 の 、 の の 、 の の の 、 の の 、 の の の の の の の の の の の の の	▲ ● 控制台	口调试器	*月」1 ↑↓ 网络 +	{) /		御氏 編器 	り 之。 へ 皓 ^{布局}	不,定 影心内 计算值	存更	東山 日 存 改 日	₩ 5 。 着 † 科体
な で、 位 章都 の、 建業 HTML biaoti:	ない ない ない ない ない ない ない ない	hen yang liu q	*月」1 ↑↓ 网络 + qing jian	{} {} *		初氏 編器 	り 之。 へ 性静 布局	不,左 ◎ 内 计算值	三白 ね 明存 更近	東ロン 日 存 改 う	があ。 清 下休
G 日 す の 世 査 器 へ 地 変 HTML biaoti: jie (/bia)	ない 空間 「 空間 合 gian nian gu zł	hen yang liu q	¥ 月」1 ↑ 网络 + qing jian	₹ £%. {} ∦	又北北 样式編 □□ 「 マ 过 元素 白	和氏 編器 規则 詳式	り 乙 つ 性的 布局	不, 左 影 ① 内 计算值	また。 明存 更近	会ロジ 日 存 改 う	储 清 ≍体
G 石 現 守 の 全 査器 の 建素 HTML biaoti: jie 、 (p id=""	マノ 泉木的 第) 控制台 qian nian gu zł	hen yang liu g	←月 J 1 ↑↓ 网络 + qing jian	() /	又1434 样式編集 □□ 第 マ 过端 元素 白	和氏 編器 親则 詳式) (周之 の 性能 布局	不, 左 影 ① 内 计算值	百日 明存 更近	毎 のの 日 存 改 う	786 0 清 〒 ₩
G 古 現 伊 3 Q 捜索 HTML biaoti: jie 	、 京 京 校制台 qian nian gu z 和 、	hen yang liu q 文化底道深厚。	▶ 网络 ↓ 网络 ↓ 1 ↓ 网络 ↓ + qing jian 明清时联	王 式に、 () オ		和氏 編器 規则 詳式) {	(月)之。 ① 性謝 布局	不, 左 影 ① 内 计算值	府 更	身用34 日 研 改 号	₩5 0 ≋ † ₽ά
G 古 損 伊 オ Q		(中ユエ、子) () 调试器 () 调试器 () () () () () () () () () () () () () (● J 1 ↑ 网络 + qing jian 明清时期 通、文化	王式に 。 () メ へ 発	 	和氏 編器 规则 詳式)((t-fami	(月)之 (7) 性能 布局 illy: *束	不, 左 影 ① 内 计算值 书","宋体	游 游 更	复用34 〒 存i 改 う	786 0 唐 〒 17 ☆
日 古 積 (部分)	(1) 秋米的 (ian nian gu z) (ig) (ig) (ig) (ig) (ig) (ig) (ig) (ig	(中日ユエ、デー) 调试器 hen yang liu q 文化底道深厚。 9中国北方向资流	● J 1 ↑ 网络 + qing jian 明清时期 [通、又化]	() () ・ た	样式編 構式編 可 可 可 可 可 可 可 可 可 可 可 可 可	和EG 編器 规则 詳样式) { (t-fami t-size	(月)之 。 介性館 布局 Lly: *衆 :: 50%;	不, 章 ① 内 计算值 书","宋体			痞。 ⋷ † ≍体
G 古 現 常 /	(二) 秋米の (1) 控制台 (1) 控制台 (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	(中日ユニ, デ) 调试器 hen yang liu q 文化底道深厚。 b中国北方高気流	F月」「 ↑↓ 网络 ↓ qing jian 明清时郑 谜。文化	モゴに 。 () パ へ 一 是 定	样式编辑 「「」」 「」」 「」」 「」」 「」」 「」」 「」」 「	相民 編器 规则 詳样式) { t-fami t-size	(月)之 。 介性論 布局 ily: *ま :: 50%;	不, 资 ① 内 计算值 书","宋体			786 0 諸 †↑ ₽体

图 3.9 CSS 的字体属性

body { font - family: helvetica, verdana, sans - serif; }

(2) 属性取值说明。如果字体名称包含空格或中文,则应使用引号括起。

(3) 设置的字体浏览器未必支持,在不影响效果的情况下,要选常见的字体。

2) 字体大小属性 font-size

字体大小属性 font-size 用于设置文本字体大小。

(1) 语法: font-size: xx-small | x-small | small | medium | large | x-large | xx-large| 长度值 | 百分比;

(2) 属性取值说明

① xx-small 等固定的值:类比于衣服的尺码。

②长度值:不允许出现负值,长度常用的单位是 px(像素)。

③ 百分比:百分比取值是基于父对象中字体的尺寸,不允许为负值。

3) 字体样式属性 font-style

字体样式属性 font-style 用于设置文本显示样式。

(1) 语法: font-style: normal | italic;

(2) 属性取值说明。

① normal:用于指定文本字体样式为正常的字体,为默认值。

② italic: 用于指定文本字体样式为斜体。

4) 字体粗细属性 font-weight

字体粗细属性 font-weight 用于设置文本的粗细程度。

(1) 语法: font-weight: normal | bold | bolder | lighter | 整数值(100 | 200 | ... | 900);

(2) 属性取值说明:

① normal: 表示正常的字体,相当于数字值 400。

② bold: 表示粗体,相当于数字值 700。

网页设计与制作实践

③ bolder:用于定义比继承值更重的值。

④ lighter:用于定义比继承值更轻的值。

⑤ 整数值:用数字表示字体粗细,取值为100 200 300 400 500 600 700 800 900。

5) 属性 font-variant

属性 font-variant 用于设置文本是否为小型的大写字母。

(1) 语法: font-weight: normal | small-caps;

(2) 属性取值说明。

① normal: 表示正常的字体。

② small-caps: 表示小型的大写字母字体。

其他更多的文本类属性请查阅 CSS 手册。

2. 字体样式示例

假设 ch3_08. html 与 demo. css 同时存放在 ch3_08 目录下,其样式示例见例 3-8。

【例 3-8】 CSS 字体样式示例。

ch3_08. html 内容如下:

```
01 <! DOCTYPE html >
02 < html >
03
   < head >
     < meta charset = "utf - 8" />
04
     k type = "text/css" href = "./demo.css" rel = "stylesheet" />
05
     <title>CSS 字体属性</title>
06
07
    </head>
80
   < body >
     < div >
09
         <h2>标题:千年古镇杨柳青简介</h2>
10
         <h3>biaoti:qian nian gu zhen yang liu qing jian jie </h3>
11
         杨柳青镇历史沉积久远,文化底蕴深厚.明清时期,是运河漕运重要
12
                   枢纽,成为中国北方商贸流通、文化交流集散地.
         最集中体现古镇传统风格的是南运河民俗文化区.古镇因河而兴,御
13
                   河(京杭大运河杨柳青河段)给古镇带来了繁荣的商业,孕育了传统
                   文化和民间艺术,是古镇的魂.
     </div>
14
15 \ll body >
16 </html>
demo. css 内容如下:
01 * { margin:0px; }
02 h3 {
         font - variant:small - caps;}
03 #p1{
     font - family:"隶书","宋体", sans - serif;
04
     font - size:50 %;}
05
font - size:20px;
07
80
     font - style:italic;
     font - weight: bold; }
09
```

第3章 CSS 基础

ch3_08.html 文件代码第 5 行引入了外部样式表 demo.css,ch3_08.html 第 11 行与 demo.css 第 2 行通过标签选择器 h3 关联起来,ch3_08.html 第 12 行与 demo.css 第 3 行通 过 id 选择器" # p1"关联起来。在 Firefox 浏览器中打开 ch3_08.html,按 F12 键,选中第一 个段落< pid="p1">,见图 3.9。注意图中椭圆处的方框,这里没有选中 font-size:50%,通 过单击方框,可以改变选中状态,通过变化我们能加深对该属性的理解。也可在 Firefox 浏览器调试窗口增删改查属性及取值,观看属性的作用。

3.3.2 使用 Dreamweaver CC 编辑字体属性

使用 Dreamweaver CC 编辑字体属性时,首先复制文件夹 ch3_08,重命名为 ch3_08_ bak,重命名 ch3_08. html 为 ch3_08_bak. html; 修改 ch3_08_bak. html,删除第 5 行; 删除 demo. css。步骤如下。

(1) 打开 Dreamweaver CC, 打开 ch3_08_bak. html, 单击"拆分", 上面为设计区, 下面 为代码区, 见图 3.10。



图 3.10 CSS 设计器创建内部样式

(2)单击椭圆处的"CSS设计器",打开 CSS设计面板;单击"源"处的按钮,选择"在页面中定义",在页面中添加内部样式;此时会出现"源< style >"显示,观察 ch3_08_bak. html 的变化,发现添加了内部样式的框架代码。

(3) 按图 3.11 的步骤,添加 font 属性:第1步选中"源< style>";第2步在设计区选择 要添加 CSS 的元素;第3步单击按钮 •,会自动出现 body div h3选择器;第4步单击 T 按 钮,切换到文本面板;第5步设置 font-variant:small-caps。注意观察代码的变化,使用 Dreamweaver CC 的 CSS 设计器可以避免书写错误,适于初学者。

ch3_08_bak.html* × 原代码 demo.css*	文件 CC Libraries 插入 CS	SS 设计器
标题:千年古镇杨柳青 简介	+ - 寮: <style></style>	

图 3.11 CSS 设计器添加 font 属性

网页设计

与制作实

践



(4) 重复上面的步骤,添加其他元素的 font 属性。有些属性需要手动录入属性值,如 font-family。另外,*{margin:0px;}需要在代码区直接录入,注意录入位置放在其他选择器前面。

3.4 CSS 的文本属性

3.4.1 文本属性名称及属性值

1. 文本属性概述

我们想要学习了解图 3.12 所示的文字效果是如何形成的,就需要学习文本属性。

标题:干年古镇杨柳青简介 biaoti:gian nian gu zhen yang liu ging jian jie 杨柳青镇历史沉积久远,文化底蕴深厚。明清时期,是运河漕运重要枢纽,成为中国北 商贸流通、 文化交流集散地。 方商贸流诵、 最集中体现古镇传统风格的是南运河民俗文化区。古镇因河而兴,御河(京杭大运河杨柳青河段)给古镇带来了繁荣的商业,孕育 了传统文化和民间艺术,是古镇的魂。 □ 查看器 □ 控制台 □ 调试器 1 网络 {} 样式编辑器 ① 性能 ① 内存 日存储》 п… × Q 搜索 HTML + ■ 規则 布局 计算值 字体 ▼ <!DOCTYPE htmL> 7 过滤样式 :hoy .cls + <html> 元素白{ 内联 <head> stody> p1 0 { demo.css:14 = cdiva color: grgb(100%,0%,0%); <h2>标题:千年古镇杨柳青简介</h2> ✓ font-size: 18px; <h3>biaoti:qian nian gu zhen yang liu qing jian jie</h3> ✓ text-indent: 24px; ▼ ✓ line-height: 18px; 杨柳青镇历史沉积久远,文化底蕴深厚。明清时期,是运河漕运重要检 ✓ opacity: 0.5; 成为中国北方商贸流通、 文化交流集散地。 demo.css:1 0 (▶ ··· margin: ▶ 0px; </div> 3

图 3.12 CSS 的文本属性

文本对应 text。CSS 中常用的文本属性及部分取值如下。

1) 颜色属性 color

颜色属性 color 用于设置文本颜色。

(1) 语法: color: 颜色值。

(2) 属性取值说明。

颜色值的表示方法有很多,常用的有 Color Name(颜色名称)、HEX(十六进制数)、 RGB等形式。

① 颜色名称表示方法:用 red、blue、pink 等表示颜色的英文单词表示,颜色单词可以 查阅 CSS 手册。

② 十六进制数颜色表示方法:即 # RRGGBB 或者 # RGB,其中 RR 表示红色值,GG 表示绿色值,BB 表示蓝色值,三种颜色的取值范围是 00~FF。如果每种颜色在两位上的数 字都相同,则可以简写为 # RGB,例如 # FF8800 可以缩写为 # F80。

③ RGB 颜色表示方法:即 rgb(R,G,B),其中 R 表示红色值,G 表示绿色值,B 表示

蓝色值。三者可以取正整数(范围为 0~255),或者取百分数(范围为 0~100.0%)。

④ 与颜色密切相关的透明度属性 opacity 用于设置对象的不透明度,其语法为 opacity:数值;数值被约束在[0.0-1.0]范围内,如果超过了这个范围,其计算结果将截取 到与之最相近的值。

2) 文本对齐属性 text-align

文本对齐属性 text-align 用于设置文本显示时的对齐方式。

(1) 语法: text-align: left | right | center | justify。

(2) 属性取值说明。

网页设计

与制作实践

① left: 表示文本内容左对齐,为默认取值。

② right: 表示文本内容右对齐。

③ center: 表示文本内容居中对齐。

④ justify: 表示文本内容两端对齐。

3) 文本装饰属性 text-decoration

文本装饰属性 text-decoration 用于设置文本装饰,它属于复合属性,可分拆为三个独立属性,介绍如下。

(1) 属性 text-decoration-line。

属性 text-decoration-line 用于设置装饰线条,主要体现在位置上。

① 语法: text-decoration-line: none | underline | overline | line-through | blink。

② 属性取值说明。

• none: 指定文字无装饰,为默认取值。

• underline: 指定文字的装饰是下画线。

• overline: 指定文字的装饰是上画线。

• line-through: 指定文字的装饰是贯穿线。

• blink: 指定文字的装饰是闪烁。

(2) 属性 text-decoration-style。

属性 text-decoration-style 用于设置装饰线条的线型。

(1) 语法: text-decoration-style: solid | double | dotted | dashed | wavy。

(2) 属性取值说明。

• solid: 表示实线。

• double: 表示双线。

• dotted: 表示点状线条。

• dashed: 表示虚线。

• wavy: 表示波浪线。

4) 属性 text-decoration-color

属性 text-decoration-color 用于设置装饰线条的颜色。

(1) 语法: text-decoration-color: 颜色值。

(2) 属性取值说明。颜色值见属性 color 的取值说明。

5) 文本缩进属性 text-indent

文本缩进属性 text-indent 用于设置文本的首行缩进。

(1) 语法: text-indent: 长度值 | 百分比。

(2) 属性取值说明。

① 长度值:用长度值指定文本的首行缩进距离,允许为负值。

② 百分比:用百分比指定文本的首行缩进距离,允许为负值。

6) 行高属性 line-height

行高属性 line-height 用于设置文本行的行高。

(1) 语法: line-height: normal | 长度值 | 百分比。

(2) 属性取值说明。

① normal:允许内容顶开或溢出指定的容器边界,为默认值。

②长度值:用长度值指定行高,不允许为负值。

③ 百分比:用百分比指定行高,其百分比取值是基于字体的高度尺寸,不允许为负值。 其他更多的文本类属性请查阅 CSS 手册。

2. 文本样式示例

假设 ch3_09. html 与 demo. css 同时存放在 ch3_09 目录下,其样式示例见例 3-9。

【例 3-9】 CSS 文本样式示例。

ch3_09. html 与 ch3_08. html 内容相同,见示例 3-8,这里不再赘述。demo. css 内容 如下:

```
01 * { margin:0px; }
02 h2 {
03
       text - align:center;
       text - decoration:underline double red;
04
05 }
06 / * 注意:
07
       text - decoration:underline double red;
       等价于以下三条样式:
80
09
       text - decoration - line:underline;
       text - decoration - style:double;
10
       text - decoration - color:red;
11
12 */
13 h3 {
           color:rgb(255,0,0);}
14 #p1{
       color:rqb(100%,0%,0%);
15
16
       font - size:18px;
17
       text - indent:36px;
18
       line - height:18px;
19
       opacity:0.5;
20
       }
21 #p2{
22
       font - size:12px;
23
       text - indent:24px;
24
       line - height:150 %; }
```

ch3_09.html 文件代码第 5 行引入了外部样式表 demo. css, ch3_09.html 第 10 行与 demo. css 第 2 行通过标签选择器 h2 关联起来, text-align: center 使标题文本居中, text-

083

第3章 CSS 基础

decoration 设置下画线(underline)为线形双线(double),颜色为红色(red)。ch3_09.html 第 11 行与 demo.css 第 13 行通过标签选择器 h3 关联起来,设置颜色为红色。ch3_09.html 第 12 行与 demo.css 第 14 行通过 id 选择器" # p1"关联起来,颜色设置为红色,使用了百分 比的方式,字体大小设置为 18px,首行缩进(text-indent:36px)是字体大小 18px 的 2 倍,表 示缩进 2 个字符;行高(line-height)设置为 18px,恰好装下文字(默认设置情况也是如此)。 如果设置的值比 18px 大,则行边框与文字之间有缝隙,表现为行间距大;如果设置的值比 18px 小,则文字会超出行边框,表现为行间距小,2 行文字有重叠。文字透明度(opacity)设 置为 0.5,则文字看起来有点淡。ch3_09.html 第 13 行与 demo.css 第 21 行通过 id 选择器 " # p2"关联起来,这里行高(line-height:150%)使用百分比方式,相当于行高的值为 12× 150%=18px。在 Firefox 浏览器中打开 ch3_09.html,按 F12 键,选中第一个段落" p1">,见图 3.12。注意图中椭圆处的方框,通过单击方框,可以改变选中状态,通过变化我 们能加深对该属性的理解。也可在 Firefox 浏览器调试窗口增删改查属性及取值,观看属 性的作用。

3.4.2 使用 Dreamweaver CC 编辑文本属性

使用 Dreamweaver CC 编辑文本属性时,首先复制文件夹 ch3_09,重命名为 ch3_09_ bak,重命名 ch3_09. html 为 ch3_09_bak. html; 修改 ch3_09_bak. html,删除第 5 行; 删除 demo.css。步骤如下。

(1) 打开 Dreamweaver CC,打开 ch3_09_bak. html,单击"拆分",上面为设计区,下面 为代码区,与图 3.10 相似。

(2)单击椭圆处的 CSS 设计器,打开 CSS 设计面板;单击"源"处的按钮,选择"在页面中定义",在页面中添加内部样式;此时会出现"源< style >"显示,观察 ch3_09_bak. html 的变化,发现添加了内部样式的框架代码。

(3) 按类似图 3.11 的步骤,添加 text 属性。 * {margin:0px;}需要在代码区直接录入,注意录入位置放在其他选择器前面。

3.5 CSS 的背景属性

3.5.1 背景属性名称及属性值

1. 背景属性概述

我们想要学习了解图 3.13 所示的页面背景效果是如何形成的,就需要学习背景属性。 背景属性 background 用于设置对象的背景特性。它属于复合属性,相对比较复杂,可 分拆为多个独立属性。此处讲解常用的四个独立属性,参见语法:

background: background - color || background - image || background - repeat || background position;

1) 属性 background-color

属性 background-color 用于设置背景颜色。



网页设计与制作实践



图 3.13 CSS 的背景属性

(1) 语法: background-color: 颜色值。

(2) 属性取值说明。颜色值见属性 color 的取值说明。

2) 属性 background-image

属性 background-image 用于设置背景图像。

(1) 语法: background-image: 图像 | none。

(2) 属性取值说明

① 图像:可以使用 url(图像路径)的方式引用背景图像。当同时定义了背景颜色和背景图像时,背景图像覆盖在背景颜色之上。

② none: 无背景图像。

3) 属性 background-position

属性 background-position 用于设置背景图像的位置。

(1) 语法: background-position: 水平方向位置 垂直方向位置。

① 水平方向位置可取值: left | center | right |百分比 | 长度值。

② 垂直方向位置可取值: top | center | bottom |百分比|长度值。

(2) 属性取值说明。

① left: 表示背景图像在横向上填充从最左边开始,为默认值。

② right: 表示背景图像在横向上填充从最右边开始。

③ center: 表示背景图像在横向或者纵向上居中。

④ top: 表示背景图像在纵向上填充从最顶部开始,为默认值。

⑤ bottom: 表示背景图像在纵向上填充从最底部开始。

⑥ 百分比:表示用百分比指定背景图像填充的位置,可以为负值,其参考的尺寸为容器大小减去背景图片大小。容器的 width、height 对百分比的效果影响较大。

⑦ 长度值:表示用长度值指定背景图像填充的位置,可以为负值。

4) 属性 background-repeat

属性 background-repeat 用于设置背景图像如何铺排填充。

第3章 CSS 基

础

DESIGN B 网页设计与制作实践

(1) 语法: background-repeat: repeat | no-repeat | repeat-x | repeat-y。

(2) 属性取值说明。

① repeat: 表示背景图像在横向和纵向平铺,为默认值。

② no-repeat: 表示背景图像不平铺, 仅显示一次。

③ repeat-x: 表示背景图像在横向上平铺。

④ repeat-y: 表示背景图像在纵向上平铺。

更多的背景属性请查阅 CSS 手册。

2. 背景样式示例

假设 ch3_10. html 与 demo. css 同时存放在 ch3_10 目录下,其样式示例见例 3-10。 【例 3-10】 CSS 背景样式示例。

```
ch3_10.html 内容如下:
```

```
01 <! DOCTYPE html >
02 < html >
03
      < head >
04
          < meta charset = "utf - 8" />
          k href = "demo.css" type = "text/css" rel = "stylesheet" />
05
          <title>CSS 背景样式属性</title>
06
07
      </head>
      < body >
80
          <div id = "d1">< img src = "./shu.png">< span >图像</span ></div >
09
10
          <div id = "d2">< span >背景</span ></div >
          <div id = "d3">d3 </div>
11
12
          < div id = "d4">d4 </div>
          13
14
      </body>
15 </html>
```

}

}

demo.css 内容如下:

```
01 * { margin:0px; padding:0px;
                                    }
02 div {
             border: 1px solid black;}
03 #d1 {
             background - color: # FFFFFF}
05
       background - image: url(shu.png);
06
       background - repeat: no - repeat;
07 #d3 {
       background - image: url(shu.png);
08
       background - repeat: repeat - x;
09
10 #d4 {
11
       background - image: url(shu.png);
12
       background - repeat: repeat - y;
13
       height: 40px;
       background - position: 50 % 0 %;}
14
15 #d5{
16
       background - color: # FF0000;
       background - image: url(dy.png);
17
       text - indent: 20px;
18
```

19 background - repeat: no - repeat;

20 background - position: 0 % 50 %; }

代码说明如下。

(1) ch3_10. html 文件代码第5行引入了外部样式表 demo. css,运行效果见图 3.13。

(2) ch3_10. html 第 9 行与 demo. css 第 3 行通过 id 选择器" # d1"关联起来,第 10 行 与 demo. css 第 4 行通过 id 选择器" # d2"关联起来。" # d1"是把 shu. png 作为图像引入, " # d2"是把 shu. png 作为背景引入,背景图像不占据正常的 HTML 文档位置,所以文字 "背景"显示在背景图像上,而使用< img >标签插入的图像会占据页面的位置,文字"图像" 在侧面显示。

(3) ch3_10. html 第 11 行与 demo. css 第 7 行通过 id 选择器" # d3"关联起来,第 12 行 与 demo. css 第 10 行通过 id 选择器" # d4"关联起来。" # d3"与" # d4"都引入了 shu. png, 不同之处在于" # d3"的 background-repeat 为 repeat-x,是横向平铺; " # d4"的 background-repeat 为 repeat-y,是纵向平铺。

(4) 仔细观察会发现"♯d3"的图片 shu. png 没有显示全,这里也是图像用作背景与一般图像不同的地方,一般图像会膨胀出来,而用作背景的图像不会,超出范围会截断。这点对我们调试盒子模型以及观察一些属性非常有用。

(5) ch3_10. html 第 13 行与 demo. css 第 15 行通过 id 选择器" # d5"关联起来," # d5" 引入了 dy. png,同时用 text-indent: 20px 使文字偏移开背景图像, background-repeat 为 no-repeat,不平铺。background-position: 0% 50%使得背景图像靠左垂直居中。

(6) demo. css 第 2 行设置了 border 属性, 画了个边框, 使得各元素区分明显, 便于观察。具体含义参考后面边框属性的讲解。

(7) 在 Firefox 浏览器中打开 ch3_10. html,按 F12 键,选中" # d5",见图 3.13。注意图 中椭圆处的方框,通过单击方框,可以改变选中状态,通过变化我们能加深对该属性的理解。 也可在 Firefox 浏览器调试窗口增删改查属性及取值,观看属性的作用。

CSS 的高效还在于允许我们在书写样式规则时使用简写方案,通过简写可以让 CSS 文件更小。在设置 CSS 背景时就可以使用以下的简写方案,将原本定义在多条规则中的语句 合并成为一条语句:

background: color url repeat attachment position;

按照上面的简写方案, demo. css 第 15~20 行的 background 相关部分可以简写如下:

background: #FF0000 url(dy.png) no-repeat 0% 50%;

3.5.2 使用 Dreamweaver CC 编辑背景属性

使用 Dreamweaver CC 编辑背景属性时,首先复制文件夹 ch3_10,重命名为 ch3_10_ bak,重命名 ch3_10.html为 ch3_10_bak.html;修改 ch3_10_bak.html,删除第5行;删除 demo.css。步骤如下。

(1) 打开 Dreamweaver CC,打开 ch3_10_bak. html,单击"拆分",上面为设计区,下面 为代码区,与图 3.10 相似。



(2) 单击椭圆处的 CSS 设计器,打开 CSS 设计面板;单击"源"处的按钮 ,选择"在页面中定义",在页面中添加内部样式;此时会出现"源< style >"显示,观察 ch3_10_bak. html 的变化,发现添加了内部样式的框架代码。

(3) 按图 3.14 的步骤,添加背景属性:第1步选中"源< style>";第2步在设计区选择 要添加 CSS 的元素,注意此时选择的是整个 div;第3步单击按钮,会自动出现"♯d1"选 择器;第4步单击斜纹"背景"按钮,切换到背景面板;第5步设置相应的背景属性,注意观 察代码的变化,使用 Dreamweaver CC 的 CSS 设计器可以避开书写错误,适于初学者。

15		代码 拆分 设计	•
ch3_10_bak.html ×	文件 插入 CSS 设计器		
半 图像			全部当前
背景 2	+ - 浙	+ - 属性	
d3	所有源		
104 化单 作京	<style></style>		

图 3.14 CSS 设计器添加背景属性

(4) * { margin:0px; padding:0px; }与 div { border: 1px solid black; } 需要在代码 区直接录入,注意录入位置放在其他选择器前面。

3.6 CSS 的列表属性

3.6.1 列表属性名称及属性值

1. 列表属性概述

想要学习了解图 3.15 所示的页面列表效果是如何形成的,就需要学习列表属性。

列表属性 list-style 用于设置列表项目相关内容。它属于复合属性,可分拆为3个独立属性,参见语法:

```
list - style: list - style - type || list - style - position || list - style - image;
```

1) 属性 list-style-type

属性 list-style-type 用于设置列表项目符号。

(1) 语法: list-style-type: none | disc | circle | square | decimal。

(2) 属性取值说明。

① none: 表示无项目标记。

② disc: 表示项目标记是实心圆点,为默认值。

网页设计

与制作实践

	CSS列表属性			×	+						-	-		×
¢) → C' @	٥	file:///D:	/书和	\$/第三	章/代码	/ch3_11	/ch3_11	©	7 ☆	١١١/	۵	۲	≡
	> 计算机和 > 软件工和 > 网络安全	科学与技 星 全	术											Î
R	○ 查看器 [>	〕控制台	D 调試講	# 1	↓ 网络	{} #	抗编辑器	の度	e ()	内存	日存儲	»	പം	•• ×
Q搜	索 HTML		11-	+ /	* 🗉	規则	布局	计算值	更改	字体	动画			
10</td <td>DOCTYPE htmL> tml> 滚动 運出 (head) ···· </td> <td></td> <td></td> <td></td> <td>へ ⊽ 元課</td> <td>过滤样式</td> <td>_</td> <td>_</td> <td>_</td> <td></td> <td></td> <td>:hov</td> <td>.cls +</td> <td>• 🗈 内联 🔺</td>	DOCTYPE htmL> tml> 滚动 運出 (head) ····				へ ⊽ 元課	过滤样式	_	_	_			:hov	.cls +	• 🗈 内联 🔺
htm	<pre>coosys v(u)s v(lis) v(li</pre>	lī			bod V	y ul li list-st list-st list-st		: url(<u>dy.</u> square; ion: insi	ong); de;	>			demo.c:	55:1

图 3.15 CSS 的列表属性

- ③ circle: 表示项目标记是空心圆圈。
- ④ square: 表示项目标记是实心方块。
- ⑤ decimal: 表示项目标记是阿拉伯数字。
- ⑥ 其他更多的属性取值请查阅 CSS 手册。
- 2) 属性 list-style-position
- 属性 list-style-position 用于设置列表项符号的位置。
- (1) 语法: list-style-position:outside | inside。
- (2) 属性取值说明

① outside: 表示列表项目标记放置在文本以外,且环绕文本不根据标记对齐,为默认值。

② inside: 表示列表项目标记放置在文本以内,且环绕文本根据标记对齐。

3) 属性 list-style-image

属性 list-style-image 用于设置列表项标记的图像。

(1) 语法: list-style-image: none | 图像。

(2) 属性取值说明。

① none: 表示列表项标记不指定图像,为默认值。

② 图像:表示可以使用绝对或相对地址指定列表项标记图像。如果图像地址无效,默认列表项标记将被 list-style-type 属性值代替;如果图像地址有效,list-style-type 属性值设置失效。

列表属性也可以简写,写法如下:

list-style: url(./xjt.png) outside none; /* 属性顺序依次是 image、position 和 type */

2. 列表样式示例

ch3_11_from_ch2_13. html 来自 ch2_13. html,只是引入了 demo. css,略做修改。假设 与 demo. css 同时存放在 ch3_11 目录下,其列表样式示例见例 3-11。

【例 3-11】 CSS 列表样式示例。

ch3_11_from_ch2_13. html 内容如下:

第3章 CSS 基础

```
01 <! DOCTYPE html >
02 < html >
03
      < head >
04
          < meta charset = "utf - 8">
05
         k href = "demo.css" rel = "stylesheet" type = "text/css">
06
          <title>CSS 列表属性</title>
07
      </head>
80
      < body >
09
          10
              <1i>计算机科学与技术</1i>
11
              $软件工程
12
              >网络安全
          13
14
      </body>
15 </html>
demo.css 内容如下:
01 body ul li {
02
      list - style - image: url(dy.png);
03
      list - style - type: square;
04
       list - style - position: inside; }
```

ch3_11_from_ch2_13. html 文件代码第 5 行引入了外部样式表 demo. css,第 10 行与 demo. css 第 1 行通过标签选择器 body ul li 关联起来,我们也可以使用标签选择器 body ul 来关联起来。在 Firefox 浏览器中打开 ch3_11_from_ch2_13. html,按 F12 键,选中第一个 列表项,见图 3.15。注意图中椭圆处的方框,通过单击方框,可以改变选中状态,通过变化 我们能加深对该属性的理解。也可在 Firefox 浏览器调试窗口增删改查属性及取值,观看 属性的作用。list-style-type:none 无标记在菜单栏中经常使用,可以试着修改一下赋值,观 看效果。



网页设

计

与制

作实

践

3.6.2 使用 Dreamweaver CC 编辑列表属性

使用 Dreamweaver CC 编辑列表属性时,首先复制文件夹 ch3_11,重命名为 ch3_11_ bak,重命名 ch3_11_from_ch2_13. html 为 ch3_11_from_ch2_13_bak. html; 修改 ch3_11_ from_ch2_13_bak. html,删除第 5 行; 删除 demo. css。步骤如下。

(1) 打开 Dreamweaver CC,打开 ch3_11_from_ch2_13_bak. html,单击"拆分",上面为 设计区,下面为代码区,与图 3.10 相似。

(2)单击椭圆处的 CSS 设计器,打开 CSS 设计面板;单击"源"处的按钮,选择"在页面中定义",在页面中添加内部样式;此时会出现"源< style >"显示,观察代码区的变化,发现添加了内部样式的框架代码。

(3) 按图 3.16 的步骤,添加列表属性:第1步选中"源< style>";第2步在设计区选择 要添加 CSS 的元素;第3步单击按钮 ,会自动出现"body ul li"选择器;第4步单击"边 框"按钮,切换到边框面板,向上稍拖动鼠标,找到 list-style-type 属性,设置相应的值;第5 步设置相应列表属性,注意观察代码的变化。



图 3.16 CSS 设计器添加列表属性

3.7 CSS 的边框属性

3.7.1 边框属性名称及属性值

1. 边框属性概述

我们想要学习了解图 3.17 所示的页面边框效果是如何形成的,就需要学习边框属性。



图 3.17 CSS 的边框属性

边框属性 border 用于设置对象的边框特性。它属于复合属性,可分拆为3个独立属性。参见语法:

border: border - width || border - style || border - color;

1) 属性 border-width

属性 border-width 用于设置边框的宽度。



第3章 CSS 基

础

(1) 语法: border-width: 长度值{1,4}。

(2) 属性取值说明。

网页设计

与制作实践

{1,4}表示边框宽度值可以在四个方向上分别进行设置,可以写 1~4 个宽度值。其设置规则如下所示(外边距、内边距等属性也涉及 4 个方向上的相关属性值设置,同样适用本规则)。

① 如果提供全部四个参数值,将按上、右、下、左(顺时针)的顺序作用于四边。

② 如果只提供一个,将用于全部的四边。

③ 如果提供两个,第一个用于上、下,第二个用于左、右。

④ 如果提供三个,第一个用于上,第二个用于左、右,第三个用于下。

代码如下所示:

border - width: 1px 5px 10px 15px;	/*分别设置上、右、下、左边的宽度*/
border-width: 1px;	/*设置四条边宽都为1px */
border – width: 1px 5px;	/*设置上下边宽 1px,左右边宽 5px */
border – width: 1px 5px 10px;	/*设置上边宽1px,左右边宽5px,下边宽10px *

/

2) 属性 border-style

属性 border-style 用于设置边框的样式。

- (1) 语法: border-style: [none | solid | dashed | dotted | double] {1, 4}。
- (2) 属性取值说明。
- ① none: 表示无轮廓。
- ② solid: 表示实线轮廓。
- ③ dashed: 表示虚线轮廓。
- ④ dotted: 表示点状轮廓。
- ⑤ double: 表示双线轮廓。
- ⑥ {1,4}表示边框样式可以在四个方向上分别进行设置。
- ⑦其他更多属性取值请查阅参考手册。
- 3) 属性 border-color

属性 border-color 用于设置边框的颜色。

- (1) 语法: border-color: 颜色值 {1, 4}。
- (2) 属性取值说明。
- {1,4}表示边框颜色可以在四个方向上分别进行设置。

可以对上下左右其中一个边框进行单独设置,此类属性共有4个,如下所示。

- ① border-top: border-width || border-style || border-color。
- 2 border-bottom: border-width || border-style || border-color.
- ③ border-left: border-width || border-style || border-color.
- ④ border-right: border-width || border-style || border-color。

可以对上下左右其中一个边框的每个独立属性进行单独设置,此类属性按照以下方式 组合: border-[top | bottom | left | right]-[width | style | color],共 12 个属性,分 别为。

① border-top-width | border-top-style | border-top-color。

2 border-bottom-width | border-bottom-style | border-bottom-color。

③ border-left-width | border-left-style | border-left-color。

④ border-right-width | border-right-style | border-right-color。

如果单独对一条边设置,以上边为例,可以使用 border-top-width、border-top-style 和 border-top-color 属性。同样,只要将属性中的 top 替换为 left、bottom 和 right 就可以对另 外三条边进行单独设置。

CSS 边框的宽度和边框类型必须设置,否则边框不能被显示。如果边框四条边的宽度、线型和颜色属性一样,则可以使用简写方案,代码如下所示:

border: 1px solid black; /* 依次设置宽度、线型和颜色 */

2. 边框样式示例

假设 ch3_12. html 与 demo. css 同时存放在 ch3_12 目录下,其样式示例见例 3-12。 【例 3-12】 CSS 边框样式示例。

ch3_12.html 内容如下:

```
01 <! DOCTYPE html >
02 < html >
03
       < head >
       <meta charset = "utf - 8" />
04
       link href = "demo.css" rel = "stylesheet" type = "text/css">
05
       <title>CSS 边框属性</title>
06
07
       </head>
       < body >
08
           < div id = "d1">d1 </div>
09
           <div id = "d2">d2 </div>
10
11
       </body>
12 </html>
```

demo.css 内容如下:

```
01 div {
02
           width:280px;
03
           height:40px;
04
           margin:10px; }
05 #d1 { border: 2px solid #FD070B; }
06 \pm d2 {
07
       border - top: 3px solid #FB0509;
08
       border - right: 4px dotted # 2C03FC;
09
       border - bottom: medium dashed # 000000;
       border - left: 7px double #FB797B; }
10
```

ch3_12.html 文件代码第 5 行引入了外部样式表 demo.css,ch3_12.html 第 10 行与 demo.css 第 6 行通过 id 选择器" # d2"关联起来。为了使方框 4 边显示明显,设置了元素 div 的宽度、高度和页边距。在 Firefox 浏览器中打开 ch3_12.html,按 F12 键,选中" # d2", 见图 3.17。注意图中椭圆处的方框,通过单击方框,可以改变选中状态,通过变化能加深对该 属性的理解。也可在 Firefox 浏览器调试窗口增删改查属性及取值,观看属性的作用。

第3章 CSS 基础



使用 Dreamweaver CC 编辑边框属性时,首先复制文件夹 ch3_12,重命名为 ch3_12_ bak,重命名 ch3_12. html为 ch3_12_bak. html; 修改 ch3_12_bak. html,删除第5行; 删除 demo. css。步骤如下。

(1) 打开 Dreamweaver CC,打开 ch3_12_bak. html,单击"拆分",上面为设计区,下面 为代码区,与图 3.10 相似。

(2)单击椭圆处的 CSS 设计器,打开 CSS 设计面板;单击"源"处的按钮,选择"在页面中定义",在页面中添加内部样式;此时会出现"源< style >"显示,观察代码区的变化,发现添加了内部样式的框架代码。

(3) 按图 3.18 的步骤,添加边框属性:选择" # d2",第1步单击边框,调出边框面板; 第2步单击"上边框",设置相应的 width、style、color 值。重复这个过程,完成其他元素的 CSS 设置。

ch3_12_bak.html ×	文件 (Clibrarian)新入 (200 沿计第
d1	XIT CC LIDIANCS 18/1 C	JJ ICH BE
	+ - 10	+ - 殿性
d2	所有源	
	<style></style>	

图 3.18 CSS 设计器添加边框属性

3.8 CSS 的区块属性

3.8.1 区块属性名称及属性值

1. 区块属性概述

我们想要学习了解图 3.19 所示的页面效果是如何形成的,就需要学习区块属性。

CSS 边距属性主要是指 margin(外边距)属性和 padding(内边距)属性,属性取值为一 个宽度值,需要结合 CSS 盒模型讲解它们的具体含义。CSS 盒模型规定了处理元素内容 (element)、内边距(padding)、边框(border)和外边距(margin)的方式。如图 3.20 所示,元 素最里边的部分是实际的内容;直接包围内容的是内边距;内边距的边缘是边框;边框以 外是外边距,外边距默认是透明的,因此不会遮挡其后的任何元素。元素的背景 (background)应用于由内容和内边距、边框组成的区域。

1) 外边距属性 margin

外边距属性 margin 用于设置元素对象的外边距。

页设计

与制

作实

践

bb		
梅子金黄杏子肥	梅子金黄杏子dived1 742×63 菜花稀。日长篱落无人过,	惟有蜻蜓蛱蝶飞。
梅子金黄杏子肥, 麦花雪白菜花稀。 日长篱落无人过, 惟有蜻蜓蛱蝶飞。	+ ↑↓ 网络{}祥式编辑器 ○ 性能 ①:内存 日 存储 ≫	
L 搜索 HTML 十	✓ マ过滤样式 :hov.ds + 3 回 布局 计算值	更改 字体 动画
<html> (進出)</html>	^ 元素 ϕ { 内联	
▶ <head> ····· </head> ▼ <bodys (滅死)<="" td=""><td>} #d1 /h / dama ccc+5</td><td></td></bodys>	} #d1 /h / dama ccc+5	
<div id="d1">hh</div>	padding: ▶ 20px; margin	10
<div id="d2">梅子金黄杏子肥</div>) border	28
(div 1d="d3") 将子金黄杏子肥,麦花雪白亮花稀。日长5 茶子人, 他友情知些好了。	div Q { demo.css:1 products margin: > 10px; border: > 1px solid ● #F06;	28 1 19

图 3.19 CSS 的区块属性



图 3.20 CSS 盒模型

(1) 语法: margin: [长度值 | 百分比] {1,4}。

(2) 属性取值说明。

① 长度值:用长度值来定义外边距,可以为负值。

② 百分比:用百分比来定义外边距,水平方向上参照其包含块 width 进行计算,垂直方向上参照 height,可以为负值。

③ {1,4}表示外边距可以在四个方向上分别进行设置,可以写 1~4 个长度值。

(3)可以对上下左右其中一个方向的外边距进行单独设置,此类属性共有4个,如下所示。

① margin-top:长度值 | 百分比。

② margin-bottom:长度值 | 百分比。

③ margin-left:长度值 | 百分比。

第3章 CSS 基础

```
DESIGN
WEB 网页设计与制作实践
```

- ④ margin-right:长度值 | 百分比。
- 2) 内边距属性 padding

内边距属性 padding 用于设置元素对象的内边距。

(1) 语法: padding: [长度值 | 百分比] {1, 4}。

(2) 属性取值说明。

① 长度值:用长度值来定义内边距,不允许为负值。

② 百分比:用百分比来定义内边距,水平方向上参照其包含块 width 进行计算,垂直方向上参照 height,不允许为负值。

③ {1,4}表示内边距可以在四个方向上分别进行设置,可以写 1~4个长度值。

(3)可以对上下左右其中一个方向的内边距进行单独设置,此类属性共有4个,如下所示。

① padding-top:长度值 | 百分比。

② padding-bottom:长度值 | 百分比。

③ padding-left:长度值 | 百分比。

④ padding-right:长度值 | 百分比。

margin 属性和 padding 属性也有类似于 border 属性的简写方法,如下所示:

margin: 5px;	/*设置外边距都为5px */
margin: 5px 10px;	/*设置上下外边距 5px,左右外边距 10px */
margin: 5px 10px 15px;	/*设置上外边距 5px,左右外边距 10px,下外边距 15px */
margin: 5px 15px 20px 25px;	/*分别设置上、右、下、左外边距*/
padding: 5px;	/*设置内边距都为 5px */
padding: 5px 10px;	/*设置上下内边距 5px,左右内边距 10px */
padding: 5px 10px 15px;	/*设置上内边距 5px,左右内边距 10px,下内边距 15px */
padding: 5px 15px 20px 25px;	/*分别设置上、右、下、左内边距*/

3) 尺寸属性

尺寸属性用于定义块级元素的大小,即宽度和高度。属性如表 3.2 所示。

属性名称	属性值	说 明
height	像素值/百分比	定义元素的高度
width	像素值/百分比	定义元素的宽度
max-height	像素值/百分比	定义元素的最大高度
min-height	像素值/百分比	定义元素的最小高度
max-width	像素值/百分比	定义元素的最大宽度
min-width	像素值/百分比	定义元素的最小宽度

表 3.2 CSS 尺寸属性

宽度属性 width、高度属性 height 仅设定元素盒模型中的灰色部分,元素在页面中的实际尺寸还要加上内边距、方框、外边距的尺寸。min-width 定义了最小宽度,当元素自身尺寸小于 min-width 时,min-width 也会占据设定大小的宽度值;当元素自身尺寸大于 min-width 时,min-width 不起作用,元素会自动外展,不受 min-width 的限定。网页 https://blog.csdn.net/m0_38102188/article/details/80612233 对 min-width、width 和 max-width

之间的区别与联系作了详尽的解释,供参考。实际中可以通过比较 min-height、height 和 max-height 来观察理解效果,更容易设计示例。

2. 区块样式示例

【例 3-13】 CSS 区块样式示例。

ch3_13.html 内容如下:

```
01 <! DOCTYPE html >
02 < html >
03
    < head >
      <meta charset = "utf - 8" />
04
    link href = "./demo.css" type = "text/css" rel = "stylesheet" />
05
    <title>CSS 尺寸属性</title>
06
07
     </head>
     < body >
80
   < div id = "d1"> hh </div >
09
    <div id = "d2">梅子金黄杏子肥</div>
10
    < div id = "d3">梅子金黄杏子肥,麦花雪白菜花稀.日长篱落无人过,惟有蜻蜓蛱蝶飞.</div>
11
12
    < div id = "d4">梅子金黄杏子肥,麦花雪白菜花稀.日长篱落无人过,惟有蜻蜓蛱蝶飞.</div>
13
    <div id = "d5"></div>
     </body>
14
15 </html>
demo.css 内容如下:
01 div {
02
      margin:10px;
03
      border:1px solid #F06;}
04 #d1 {
      padding:20px;
05
06
      max - height:40px;
07
      }
08 #d2{
09
      min - width:200px;
      height:40px;
10
11
      background - color: # FF0000;
12
      display:inline - block;}
13 #d3 {
      min - width:200px;
14
15
      min - height:40px;
16
      background - color: #F5FF0C;
17
      display:inline - block;}
18 #d4 {
19
      max - width:140px;
20
      max - height:40px;
21
      background - color: #F5FF0C; }
23
      min - height: 40px;
      max - width: 200px; }
24
```

ch3_13. html 文件代码第5行引入了外部样式表 demo. css。为了使得区块区域明显,

第3章 CSS 基础

我们设置了元素 div 的边框,根据需要设置了不同 div 的背景色。ch3_13.html 第 9 行与 demo.css 第 4 行通过 id 选择器" # d1"关联起来。在 Firefox 浏览器中打开 ch3_13.html, 按 F12 键,选中" # d1",见图 3.19。我们发现外边距 margin 对应黄色,内边距对应紫色,元 素自身对应浅蓝色;内容"hh"没有设置宽度和高度,此时宽度为页面宽度,高度为 hh 文字 元素的默认高度。id 选择器" # d2"设置了最小宽度 min-width:200px,虽然其元素内容"梅 子金黄杏子肥"没有那么宽,但是" # d2"仍然在页面上占据了 200px 的宽度。通过使用 display:inline-block 使得" # d2"与" # d3"同行显示,当我们进一步缩小页面窗口时,因为 " # d2"与" # d3"的宽度和超出了页面宽度的大小,会使得" # d3"换行显示,这也是我们设 定元素尺寸时需要考虑的问题。" # d4"通过设定 max-width:140px 迫使元素内容换行显 示,设定" # d4"的背景色为黄色,同时设定了 max-height:40px,这样" # d4"的内容就会溢 出,超出的部分占据" # d5"的位置来显示。如果" # d5"处有文字等内容,则会产生文字重 叠,这也是我们在访问页面时常见的显示错误。注意背景并没有跟着内容同样溢出去占据 " # d5"的位置。

3.8.2 使用 Dreamweaver CC 编辑区块属性

使用 Dreamweaver CC 编辑区块属性时,首先复制文件夹 ch3_13,重命名为 ch3_13_ bak,重命名 ch3_13. html为 ch3_13_bak. html;修改 ch3_13_bak. html,删除第5行;删除 demo.css。步骤如下。

(1) 打开 Dreamweaver CC,打开 ch3_13_bak. html,单击"拆分",上面为设计区,下面 为代码区,与图 3.10 相似。

(2)单击椭圆处的 CSS 设计器,打开 CSS 设计面板;单击"源"处的按钮,选择"在页面中定义",在页面中添加内部样式;此时会出现"源< style >"显示,观察代码区的变化,发现添加了内部样式的框架代码。

(3) 按图 3.21 的步骤,添加区块属性:选择" # d1",第 1 步单击布局,调出布局面板; 第 2 步找到 max-height,设置相应的值。重复这个过程,完成其他元素的 CSS 设置。



图 3.21 CSS 设计器添加区块属性

网

页设计

与制作实践

3.9 CSS 的布局属性

3.9.1 布局属性名称及属性值

如果没有 CSS 的作用,则在默认的情况下,块级元素会按顺序、每次另起一行的方式一 直往下排。我们可以通过 CSS 来改变这种 HTML 的默认布局模式,把块级元素摆放到想 要的任何位置上去,也可以通过 CSS 来改变块级元素的宽度和高度。

布局属性包括溢出属性 overflow、显示属性 display、浮动属性 float 和清除浮动属性 clear,下面分别加以介绍。

1. 溢出属性 overflow

溢出属性 overflow 用于设置元素对象处理溢出内容的方式。

(1) 语法: overflow: visible | hidden | scroll | auto。

(2) 属性取值说明。

① visible: 对溢出内容不做处理,内容可能会超出容器,为默认值(除 body 对象和 textarea)。

② hidden: 隐藏溢出容器的内容且不出现滚动条。

③ scroll: 隐藏溢出容器的内容,溢出的内容将以卷动滚动条的方式呈现。

④ auto:当内容没有溢出容器时不出现滚动条,当内容溢出容器时出现滚动条,按需出现滚动条,此为 body 对象和 textarea 的默认值。

(3) 可以对水平或者垂直方向的其中一个方向进行单独设置,包含以下两个属性。

① overflow-x: visible | hidden | scroll | auto.

② overflow-y: visible | hidden | scroll | auto.

2. 显示属性 display

显示属性 display 用于设置元素对象是否显示以及如何显示。

(1) 语法: display: none | inline | block | inline-block。

(2) 属性取值说明。

① none: 表示元素对象被隐藏。

② inline: 指定对象转换为内联元素显示。

③ block: 指定对象转换为块级元素显示。

④ inline-block: 指定对象转换为内联块元素显示。

⑤ 其他更多的属性取值请查阅 CSS 参考手册。

3. 浮动属性 float

浮动属性 float 用于设置元素对象是否浮动以及如何浮动。

(1) 语法: float: none | left | right。

(2) 属性取值说明。

① none: 设置对象为不浮动,为默认值。

② left:设置对象浮动在左侧。

- ③ right:设置对象浮动在右侧。
- 4. 清除浮动属性 clear

清除浮动属性 clear,用于设置元素对象不允许有浮动对象的边(左边或右边)。

- (1) 语法: clear: none | left | right | both。
- (2) 属性取值说明。
- ① none: 允许两边(左边和右边)都可以有浮动对象,为默认值。
- ② left: 不允许左边有浮动对象,即清除左侧存在浮动元素时产生的影响。
- ③ right:不允许右边有浮动对象,即清除右侧存在浮动元素时产生的影响。
- ④ both:两边都不允许有浮动对象,即清除两侧存在浮动元素时产生的影响。

3.9.2 布局属性示例

1. 溢出属性 overflow 示例

【例 3-14】 CSS 溢出属性 overflow 示例。

ch3_14. html 内容与 ch3_13. html 相同,这里不再列出。demo. css 内容如下:

```
01 div {
02
       margin:10px;
0.3
       border:1px solid #F06;}
04 #d1 {
05
       padding:20px;
06
       max - height:40px;
                            }
07 #d2{
08
       min - width:200px;
       height:40px;
09
       background - color: # FF0000;
10
11
       display:inline - block;}
12 #d3 {
13
       min - width:200px;
       min - height:40px;
14
15 background - color: # F5FF0C;
       display:inline - block; }
16
17 #d4 {
18
       max - width:140px;
19
       max - height:40px;
20 background - color: #F5FF0C;
21
       overflow:scroll;}
22 #d5 {
23
       min - height: 40px;
2.4
       max - width: 200px; }
```

ch3_14.html 文件代码第 5 行引入了外部样式表 demo.css。" # d4"通过设定 maxwidth:140px 迫使元素内容换行显示,设定" # d4"的背景色为黄色,同时设定了 maxheight:40px,这样" # d4"的内容就会溢出,超出的部分占据" # d5"的位置来显示。现在增 加了第 21 行代码 overflow:scroll,溢出部分就隐藏在" # d4"设定的范围内,并可以通过滚 动条来查看隐藏的内容,如图 3.22 所示。



网页设

计

与制作实践

hh				
梅子金黄杏子肥梅子金黄杏子。	梅子金黄杏子肥,麦花雪白菜花稀。	日长篱落无人过,惟有蜻蜓蛱蝶飞。		
2 (〕 查看器) 控制台 〇 调试器 違素 HTML + グ < <body> ^</body>	N 网络 {) 样式編編器 ○ 性能 〔 マ 过滤样式 :hov .cls + 〕 元素 ♀ { 内联	〕 内存 目 存储 ≫ □ ··· □ 布局 计算值 更改 字体 动画 ▼ 弹性盒		
<pre> <div id="d1">hh</div> <div id="d2">梅子金黄杏子肥</div> 空白 cdiv id="d3"></pre>	} #d4 ф { demo.css:18 ☑ max-width: 140px; ☑ max-height: 40px; ☑ background-color: 圖 #F5FF0C;	8 选择一个弹性 (Flex) 容器或项目以继续。 ▼ 网格 此页面 上沿有使用 CSS 网络		
长篇落无人过。惟有靖建铁線飞。 div ide="d4"> 梅子金赏音子肥。麦花雪白菜花稀。日 长篇落无人过。惟有靖建铁線飞。 [雅田] 	<pre>\[verflow: ▶ scroll; } div { demo.css:1 margin: ▶ 10px; border: ▶ 1px solid ● #F06; }</pre>	应负回上没有使用 CSS 网络 ▼ 盒模型 margin 10 border 0 padding 0		

图 3.22 溢出属性示例

2. 显示属性 display 示例

【例 3-15】 CSS 显示属性 display 示例。

新建文件夹 ch3_15,复制第 2 章 ch2_18 的内容,重命名 ch2_18.html 为 ch3_15.html, 在 ch3_15.html 中通过< link >标签引入外部样式 demo.css。ch3_15.html 与 ch2_18.html 基本相同,仅多了< link href="demo.css" type="text/css" rel="stylesheet" />一句,这 里不再列出。demo.css 内容如下:

```
01 div ul li {
02 display: inline - block;
03 width: 70px;}
```

ch3_15.html 文件代码第 5 行引入了外部样式表 demo.css。设置 li 标签为 display: inline-block,这样使得 li 标签为行内块元素,去掉了其前后的换行符,li 都放在了一行内显 示。效果如图 3.23 所示。

3. 浮动属性 float 示例

【例 3-16】 CSS 浮动属性 float 示例。 ch3_16. html 内容如下:

```
01 <! DOCTYPE html >
02 < html >
03 < head >
04 < meta charset = "utf - 8" />
05 < link href = "./demo.css" type = "text/css" rel = "stylesheet" />
```



第3章 CSS 基



```
<title>CSS 浮动属性</title>
06
07
     </head>
80
     < body >
09
      < div id = "head">网页顶部分</div>
10
       <div id = "content">
11
           <div id = "left">左侧部分</div>
12
           <div id = "right">右侧部分</div>
           <div id = "clf"></div>
13
14
       </div>
       < div id = "foot">网页底部</div>
15
16
    </body>
17 </html>
```



图 3.23 显示属性 display 示例

demo.css 内容如下:

```
01 * {
02
       font - size:30px;}
03 div {
04
       border:1px solid red;}
float:left;
06
       width:150px;
07
80
       height:80px;
09
       background: yellow;}
10 # right {
11
       float:right;
12
       width:200px;
       height:100px;
13
14
       background: green; }
15 #clf {
       clear:both;}
16
```

网页设

计

与制

作实

践

ch3_16.html 文件代码第 5 行引入了外部样式表 demo.css。ch3_16.html 由头部 (head)、中部(content)、尾部(foot)上下排列组成,默认情况下就可以达到自上而下的排列。 默认情况下,中部的左侧部分(left)、右侧部分(right)也是上下排列的,而我们想要将左侧 部分、右侧部分放在一行显示,只需将 # left 设置为 float:left, # right 设置为 float:right,即 可达到同行显示的效果,如图 3.24 所示。因为 # left 设置为 width:150px, # right 设置为 width:200px,两者的和 350px 小于页面宽度,所以在 # left 与 # right 会有一段空白没有背 景颜色的部分。浮动的元素自动被转换为一个块级元素。一个元素设置了浮动,显示的层 级就提高了,会影响它后面没有设置浮动的元素,这些被影响的元素会跑到浮动层的下面 去,所以下面的元素要使用 clear 属性来清除浮动,保证正确的显示效果,故需要在 # clf 中 设置 clear:both,以清除浮动属性 float 对后面元素的影响。



图 3.24 浮动属性 float 示例

3.10 CSS 的定位属性

3.10.1 元素定位机制

元素定位方式有以下四种。

(1)常规文档流(普通流)定位:元素在页面中的显示位置由元素在 HTML 文档中出现的位置决定,元素按照自己原本的显示特征在页面中从左到右、从上到下一个接一个地排列显示。

(2) 浮动定位:元素设置了浮动属性。元素显示时就脱离了常规文档流,按照浮动的 特性去显示,它可能会影响常规文档流中其他元素的显示。

(3)相对定位:元素对象遵循常规流,但是它会参照自身在常规流中的位置,通过定义

103

基

偏移属性进行偏移。偏移时不会影响常规文档流中的任何元素。

(4)绝对定位:元素对象脱离了常规文档流,通过定义偏移属性进行偏移,此时偏移属性 参照的是离自身最近的定位祖先元素。如果没有定位的祖先元素,则一直回溯到 body 元素。

3.10.2 定位属性名称及属性值

1. 定位属性概述

网页设

计与制

。作实践

1) 定位属性 position

定位属性 position 用于定义元素对象的定位方式,需要结合 4 个偏移量属性 top、 bottom、left、right 共同确定元素具体的定位位置。

(1) 语法: position: static | relative | absolute | fixed。

(2) 属性取值说明。

① static: 元素对象遵循常规文档流,为默认值,此时 4 个偏移量属性无效。

② relative:用于设置相对定位。元素对象也会遵循常规文档流,但是它将参照自身在 常规流中的位置,根据 top、bottom、left、right 这 4 个偏移量属性的取值进行一定的位置偏 移,偏移时不会影响常规文档流中的任何元素。

③ absolute:用于设置绝对定位。元素对象脱离了常规文档流,此时该元素在页面中 定位时的参照物是离它自身最近的定位祖先元素,即从其父元素开始逐层向上查找,直到找 到一个祖先元素设置了定位属性为止,那么该祖先元素就是参照物;如果没有找到定位的 祖先元素,则一直回溯到 body 元素,body 元素就是参照物。它的偏移位置不影响仍然处在 常规文档流中的任何元素。

④ fixed: 与绝对定位 absolute 一致,但是其偏移量定位始终是以页面窗口为参考,并 且当出现滚动条时,对象也不会随着滚动。

2) 偏移量属性

偏移量属性共有4个,用于配合定位属性计算元素的偏移量。

(1) 语法如下:

① top: auto | 长度值 | 百分比。

② bottom: auto | 长度值 | 百分比。

③ left: auto | 长度值 | 百分比。

④ right: auto | 长度值 | 百分比。

(2) 属性取值说明

auto: 无特殊定位,根据 HTML 定位规则在文档流中分配。

(3) 层叠属性 z-index

层叠属性 z-index 用于设置对象的层叠顺序。非常规文档流元素之间、非常规文档流元 素与常规文档流元素之间都有可能存在位置重叠的情况,可以通过层叠属性来设置元素的 显示优先级。

(1) 语法: z-index: auto | 整数值。

(2) 属性取值说明。

① auto: 元素在当前层叠上下文中的层叠级别是 0,为默认值。

② 整数值:用整数值来定义堆叠级别,可以为负值。

③ 整数值越大的元素,其层叠级别越高,层叠级别大的显示在上面,层叠级别小的显示在下面。

2. CSS 定位示例

【例 3-17】 CSS 定位示例。

ch3_17.html 内容如下:

```
01 <! DOCTYPE html >
02 < html >
03 < head >
    < meta charset = "utf - 8" />
04
    k href = "./demo.css" type = "text/css" rel = "stylesheet" />
05
    <title>CSS 定位属性</title>
06
  </head>
07
  < body >
80
09
    我是段落1
10
    我是段落 2 
11
    我是段落 3 
12
    我是段落4
    我是段落 5, 脱离了普通流
13
    我是段落 6,普通流
14
15
    <div id = "d1">
       我是段落 7,普通流
16
17
       < div >
18
          我是段落 8, 脱离了普通流
       </div>
19
       我是段落 9,普通流
2.0
    </div>
21
22
   </body>
23 </html>
demo.css 内容如下:
01 * {
    font - size:25px;
02
```

```
03
      margin:0px;}
04 p{
05
      border:1px solid # 3A8;}
07
      color:red;
      position:relative;
80
09
      top:20px;
      left:40px;}
10
12
      color:blue;
13
      position:absolute;
14
      top:20px;
15
      left:100px;}
16 #d1 {
```

第3章 CSS 基础

```
17
       position:relative;
18
       background: # FD9; }
19 # p8{
20
       color:#F6F;
       position:absolute;
21
22
       top:5px;
23
       left:15px;
24
       z - index: -1;
25 #d3 {
                         /*在 body 中嵌套一个子元素(id = "d3")即可看到效果*/
26
       width:100px;
27
       height:200px;
28
       position:fixed;
29
       top:50px;
30
       left:50px;
31
       background:red;}
```

ch3_17.html 文件代码第 5 行引入了外部样式表 demo.css。本例中段落 2(♯p2)、段落 5(♯p5)、段落 8(♯p8)使用了定位属性 position。图 3.25 显示了段落 2(♯p2)被选中时的调试状态,改变 position 属性选中或不选中的状态,通过观察前后变化可知 relative 是相对当前常规文档流做的偏移(top:20px 是相对其本该出现的位置往下偏移了 20px),而且也没有影响段落 3 的位置。段落 5(♯p5)是绝对偏移(position:absolute,见代码第 13 行),它脱离了当前常规文档流,定位偏移(top:5px)相对于窗口,所以离页面顶部较近。段落 8(♯p8)也是绝对偏移(position:absolute),但是其父元素(♯d1)设置了 position:relative,见代码第 17 行,所以段落 8(♯p8)的偏移是相对于父元素(♯d1)的。我们在调试状态下不选中 z-index:-1,即可观察到效果。

CSS定位属性	×	+			- 🗆 ×
← → ⊂ ☆	🗋 file:///D:/书	稿/第三章/代码/ch3_17	//ch3_17. ••• 🔇	J ↓ II	•
我是段落1 我是印	<u> 没落5, </u>	离了普通流			
我是段落3					
我是段落4					
我是段落6,普	通流				
我是段落7, 普	通流				
我是段落9,普	通流				
<					>
日本 白 査看器 回 控制	台 D 调试器	↑↓ 网络 {} 样式编辑器	の性能の	内存 🖯 存储	» 0 ··· >
Q 搜索 HTML	+ / 5	了过滤样式 :hov	.ds + 🗈 🗉	布局计算的	直 更改 字体
htmL	^ 元	潇 ф {	内联 🔺 🔻	盒模型	ton
<ntml> (派詞)</ntml>	3	2 0 1	demp.css:6	osition	28
₩ <body></body>		color: • red;		margin	0
拔是改语1< <o id="02">我是没语2<</o>	/p> /p> 器名	<pre>/ position: relative; / top: 20px;</pre>	定	border	e
id Bart Balling	In Y	<pre>✓ left: 40px;</pre>	位	e 1 e : 696	6×33 0 0 0
num - body - p=pz	-		V +0		

图 3.25 CSS 的定位属性

网页设计与制作实践

3.11 CSS 应用实例

3.11.1 CSS 实现的翻页

【例 3-18】 CSS 实现的翻页。

ch3_18.html页面源代码如下:

```
01
    <! DOCTYPE html >
02
    < html >
03
       < head >
           <title>实用的翻页样式</title>
04
05
           k href = "demo.css" rel = "stylesheet" type = "text/css" />
       </head>
06
07
    < body >
80
       class = "first">< a href = " # ">1 </a>
09
10
           < a href = " # ">2 </a>
           < a href = " # "> 3 </a> 
11
           < a href = " # ">4 </a>
12
           < a href = " # ">5 </a>
13
           < a href = " # ">6 </a>
14
15
           < a href = " # ">7 </a>
           < a href = " # ">8 </a>
16
           < a href = " # ">9 </a>
17
       18
19
    </body>
20
    </html>
demo.css 文件内容如下:
```

```
01
    ul {
                               /*设置列表背景颜色*/
02
    background:white;
03
    width:400px;
                               /*设置列表宽度为400px*/
04
    height:25px;
                               /*设置列表高度为25px*/
05
    list - style - type:none;
                               /* 去掉列表项目符号*/
    padding:10px;
                               /*设置内边距为10px*/
06
07
    }
08
    ul li {
                               /*设置所有的列表项向左浮动*/
09
    float:left;
    margin - right:5px;
                               /*设置右外边距*/
10
11
    }
12
    ul li a {
13
    display:block;
                               /*元素转换为块级元素*/
    padding:3px 8px;
                               /*设置内边距 */
14
    border:1px solid #f60;
                               /*设置边框*/
15
    text - decoration:none;
                               /*去掉超链接的下画线*/
16
17
    color:black;
                               /*设置超链接颜色为黑色*/
18
    }
```



础

```
19 ul li a: hover, ul li.first a { /* 鼠标经过时的超链接样式 */
20 background:green; /* 设置背景颜色为绿色 */
21 color:white; /* 设置字体颜色为白色 */
22 font - weight:bold; /* 设置字体加粗显示 */
23}
```

例 3-18 的运行效果如图 3.26 所示。

	实用	的翻页	样式	×	+		-		\times
\leftarrow	\rightarrow	С	۵	D file:///D:/	书稿/第三章/代码	3/ch3_18/ch3_18	B.html ☆	${\times}$	≡
_									^
1	2	3	4	5 6 7 8	9				~

图 3.26 实用的翻页样式



网页设计

与制作实践

3.11.2 CSS 实现的新闻列表样式

【例 3-19】 CSS 实现的新闻列表样式。

ch3_19.html页面源代码如下:

```
01
   <! DOCTYPE html >
02
   < html >
0.3
   < head >
       <title>新闻列表样式</title>
04
       k href = "demo.css" rel = "stylesheet" type = "text/css" />
05
06
   </head>
07
   < body >
      <div id = "news">
80
       < h1 >新闻列表</ h1 >
09
       <div id = "left">
10
          <img src = "./news.jpg" />
11
          中环学院经济与管理系代表队荣获全国一等奖
12
13
       </div>
14
       <div id = "right">
          15
             < a href = " # ">中环学院召开 2021 年 4 月份纪委例会</a>
16
17
             < a href = " # ">举办 2018 级学生简历制作培训会 </a>
             <a href = " # ">与天津环博科技公司举行签约授牌仪式</a>
18
             < a href = " # ">蓝桥杯大赛(天津赛区)中荣获多项佳绩</a>
19
             < a href = " # ">中环学院召开 2021 年 4 月份安全工作会</a>
20
21
          2.2.
          < a href = " # ">更多 ... </a>
       </div>
23
24
       </div>
25
    </body>
26
    </html>
demo. css 文件内容如下:
```

01 * { /*设置外边距为 0px*/ 02 margin: 0px; 03 padding: 0px; /*设置内边距为 0px */ 04 } 05 img { 06 width:190px; 07 } h1 { 80 09 height: 30px; /*设置高度为 30px*/ /*设置行高为30px*/ 10 line-height: 30px; font - size: 17px; /*设置字体大小为17px*/ 11 12 border - bottom: 2px solid green; /*设置底边边框*/ 13 } 14 ≠news { /*设置宽度为 500px*/ 15 width: 550px; 16 } 17 #left{ 18 width: 200px; /*设置宽度为200px*/ 19 float: left; /*设置左浮动*/ 20 padding: 15px 10px; /*设置上下内边距为15px,设置左右内边距为10px*/ 21 } 22 # right { /*设置宽度为300px*/ 23 width: 300px; /*设置右浮动*/ 24 float: right; 25 /*设置内边距为8px*/ padding: 8px; 26 } 27 # right ul { /*设置列表无项目符号*/ 28 list - style - type: none; 29 line-height: 180 %; /*设置行距为180%*/ 30 padding - left:10px; /*设置左内边距为10px*/ border - left: 1px dashed ♯aaa; /*设置左侧边框*/ 31 32 } 33 # right ul li { /*设置字体大小为12px*/ 34 font-size: 12px; 35 padding - left: 20px; /*设置左内边距为 20px */ 36 background: url(./xjt.png) no-repeat center left; /*设置背景图像*/ 37 } # right ul li a, p.more a{ 38 39 text - decoration: none; / * 去掉超链接的下画线 * / 40 } # right ul li a:hover { 41 42 color: #f60; /*设置鼠标经过超链接时的颜色*/ 43 } 44 p.desc { /*设置字体大小为13px*/ 45 font-size: 13px; /*设置文本对齐方式为居中*/ 46 text - align: center; 47 padding - top: 10px; /*设置上内边距为10px*/ 48 } 49 p.more { / * 设置字体大小为 12px * / 50 font - size: 12px; /*设置文本对齐方式为居右*/ 51 text - align: right; /*设置文本加粗显示*/ 52 font - weight: bold; 53 }

第3章 CSS 基

例 3-19 的页面运行效果如图 3.27 所示。



图 3.27 新闻列表样式



网页设

计

与制作实践

3.11.3 学习目标1的CSS模仿

如图 3.28 所示,目录 ch3_20 的结构如下: mubiaol 文件夹下存放的是第 2 章提到的学 习目标网页 http://www.tjzhic.edu.cn/export/sites/tjzhic/zhxw/zhyw/news/1125.html 在本地的副本。mofang 文件夹下存放的 ch3_20.html 与 ch2_25.html 的内容基本相同,只 是引入了自编的 demo.css,并添加了 div 标签来划分区域。

```
ch3_20
   mofang
 (Ba)
       CSS
    {}
          demo.css
                     OKB
       images
       ch3_20.html
 <>
                      2KB
   mubiao1
 1
       中环要闻_files
 <>>
       中环要闻.html 6KB
```

图 3.28 目录 ch3_20 结构图

【例 3-20】 学习目标 1 的 CSS 模仿示例。 ch3 20. html 的内容如下:

```
01 <! DOCTYPE html >
   < html >
02
03
        < head >
            <meta charset = "utf - 8">
04
05
            <title>仿制 mubiao1 菜单超链接</title>
06
       k href = "css/demo.css" rel = "stylesheet" type = "text/css">
        </head>
07
80
        < body >
            <div id = "head">< img src = "./images/news_logo.jpg" /></div>
09
10
            <div id = "nav">
11
                <a href = "http://www.tjzhic.edu.cn/export/sites/</li>
12
                           tjzhic/">首页</a>
            <a href = "http://www.tjzhic.edu.cn/export/sites/
13
```

```
tjzhic/zhxw/zhyw/">中环要闻</a>
14
          < a href = "http://www.tjzhic.edu.cn/export/sites/tjzhic/</li>
                            zhxw/xwdt/">新闻动态</a>
          < a href = "http://www.tjzhic.edu.cn/export/sites/tjzhic/</li>
15
                            zhxw/jxky/">教学科研</a>
16
          < a href = " # ">专题人物</a>
17
          < a href = " # ">招生就业</a>
          < a href = " # ">杨柳菁菁</a>
18
19
          < a href = "http://www.tjzhic.edu.cn/export/sites/tjzhic/</li>
                      zhxw/dzxb/">学院报刊</a>
          < a href = "http://www.tjzhic.edu.cn/export/sites/tjzhic/</li>
20
                      zhxw/gyls/">光影中环</a>
          21
22
          </div>
23
          <div id = "main">
          <h2>中环学院三星电机第二期订制班顺利入职</h2>
2.4
25
          <hr />
          < span > 作者: managercui 审核: 发布时间: [2020 - 07 - 02] 
26
27
          < span >近日,中环学院三星电机第二期订制班 37 名优秀毕业生……
             </span>
          <div><img src = "./images/hezhao.jpg" /></div>
28
          < span>自中环学院与三星电机有限公司开展校企合作以来……</span>
29
30
          <div><imq src = "./images/quli.jpg" /></div>
31
         </div>
32 </body>
33 </html>
demo.css 的内容如下:
01
   * {
```

```
02
        padding: 0px;
03
         margin:0px;}
04
   body {
05
         text - align: center;
06
         margin: 0 auto;
07
   }
    # nav {
80
09
         height: 45px;
10
        margin: 0 auto;
   }
11
    # nav ul li {
12
13
         display: inline - block;
         list - style - type: none;
14
15
         min-width: 100px;
16
         line - height: 45px;
   }
17
18 ullia {
19
         text - decoration: none;
20
         font - size: 14px;
21
         font - weight: bold;
22 }
23
   # main {
24
         max - width: 1000px;
```

第3章 CSS 基础

```
25
         margin: 0 auto;
26 }
27
    \# main h2 {
28
         margin - top: 15px;
29
        margin - bottom: 5px;
         color: #232D82;
30
31
         font - size: 20px;
32
    }
33
    # main hr {
34
         width: 900px;
35
         margin: 10px auto;
36
    }
    # main span {
37
38
        font - size: 12px;
        color: #333;
39
40
    }
41
    # main p {
42
        line - height: 180 %;
43
        text - align: justify;
        padding - top: 5px;
44
45
        text - indent: 28px;
46
    }
47
    # main p span {
48
         font - size: 14px;
49
         color: #222;
50
    }
51
    # main div img {
52
         margin: 0 auto;
         padding - top: 10px;
53
54
         padding - bottom: 15px;
55 }
```

例 3-20 的页面运行效果如图 3.29 所示。



图 3.29 添加 CSS 样式后学习目标 1 的显示图

网页设计

与制作实践

3.12 Bootstrap 简介

3.12.1 Bootstrap 概述

Bootstrap 是美国 Twitter 公司的设计师 Mark Otto 和 Jacob Thornton 基于 HTML、 CSS、JavaScript 开发的简洁、直观、强悍的前端开发框架,使得 Web 开发更加快捷。 Bootstrap 提供了优雅的 HTML 和 CSS 规范,由动态 CSS 语言 Less 写成。Bootstrap 容易 上手,移动设备优先,支持多个常见的浏览器及响应式设计,用户通过各种尺寸的设备浏览 网站均可获得良好的视觉效果。与先写好 HTML 代码再去设计 CSS 不同,我们在使用 Bootstrap 时,需要先查看 Bootstrap 的文档,知道 Bootstrap 已经为我们提供了哪些样式, 然后再使用 HTML 代码给标签设置相应的 id、class 等 CSS 选择器,就可以让网页呈现出 一个较好的效果。

3.12.2 使用字形图标

【例 3-21】 使用字形图标。 ch3 21. html的内容如下:

```
01 <! DOCTYPE html >
02 < html >
03
    < head >
04
        < meta charset = "utf - 8">
        <title>Bootstrap 实例 - 如何使用字形图标(Glyphicons)</title>
05
        < link rel = "stylesheet" href = "https://cdn.staticfile.org/</li>
06
                      twitter - bootstrap/3.3.7/css/bootstrap.min.css">
07
        < script src = "https://cdn.staticfile.org/
                      jquery/2.1.1/jquery.min.js"></script>
    < script src = "https://cdn.staticfile.org/twitter - bootstrap/</pre>
08
                      3.3.7/js/bootstrap.min.js"></script>
09
        < style type = "text/css">
10
             span {
                  font - size:25px;
11
12
                 color:red;}
13
        </style>
   </head>
14
    < body >
15
    < button type = "button" class = "btn btn - default btn - lq">
16
        < span class = "glyphicon glyphicon - user"></span > User
17
18
   </button>
19
    <br/>button type = "button" class = "btn btn - default btn - sm">
        < span class = "glyphicon glyphicon - user"></span > User
2.0
21
   </button>
    < button type = "button" class = "btn btn - default btn - xs">
22
        < span class = "glyphicon glyphicon - user"></span > User
23
24 </button>
```



25 </body> 26 </html>

网页设计与制作实践

例 3-21 的页面运行效果如图 3.30 所示。代码第 6~8 行引入了相应的 CSS、JavaScript 文件。class="btn btn-default btn-lg"设置了 3 个类属性值,通过它们来引入 Bootstrap 已 经为我们设定好的 CSS 样式,注意这里不是 CSS 的后代选择器。class="glyphicon glyphicon-user"设置了两个类属性值,来引入字形图标的 CSS 样式。网页 https://www. runoob.com/try/demo_source/bootstrap3-glyph-icons.htm 给出了字形图标对应的 class 属性值,在需要使用字形图标的 HTML 标签设置对应的 class 属性值即可。

	Bootstrap 实例 - 如何使用字形图标		+		x
÷	\rightarrow G	D file:///E:/sd/第3章	\$/代码/ch3_2 公	>>	=
1	User	LUser LUser			

图 3.30 使用字形图标

习题 3 🖉

一、简答题

- 1. 什么是 CSS? 页面中如何引用 CSS 文件?
- 2. 为什么要使用 CSS 样式文件?
- 3. CSS 样式的三种调用方式有什么不同?优先级有什么不同?
- 4. CSS 样式的语法结构是什么?如何定义一条样式规则?
- 5. 如何选择页面中不同的 HTML 元素? CSS 都有哪些选择器?

二、操作题

- 1. 使用 CSS 设计一个美观的文章显示页面。
- 2. 使用 CSS 实现一个二级导航菜单页面。