

本章学习 CSS 基础理论和使用方法,通过大量实例学习 CSS 的基本语法和概念,学习设置文字、图片、背景、表格等网页元素样式的方法以及如何用 DIV+CSS 进行网页布局。

3.1 CSS 概念

CSS 表示层叠样式表。通过 CSS 可以对网页的字体、颜色、背景等设置,可以结合 DIV 对网页进行布局。CSS 的出现引发了网页设计布局技术的变革。

使用 CSS 控制网页格式的方法有行内法、内嵌式、链接式和导入式等。例 3.1 使用内嵌式样式。所谓内嵌式样式,是通过< style >标记将样式定义在 HTML 文件的头部。

【例 3.1】 CSS 小试牛刀。

```
1 <html >
2   <head >
3     <title >内嵌式样式</title >
4     <style type = "text/css" >
5       p{
6         font - family:隶书;
7         font - size:20px;
8         text - decoration:underline;
9       }
10    </style >
11  </head >
12  <body >
13    <p >CSS 小试牛刀,文字隶书,20px,带下画线。
14  </body >
15 </html >
```

预览效果如图 3.1 所示。

网页文件中定义了< p >标记选择器样式,设置字体为隶书,大小为 20px,带下画线。只要此网页中出现< p >标记,系统将会自动应用新样式。

在 CSS 中,样式的定义遵循一定的语法要求。



CSS 概念



图 3.1 样式效果图

3.1.1 CSS 基本语法

```
选择器{  
    属性 1:属性值 1;  
    属性 2:属性值 2;  
    :  
}
```

说明：选择器表示样式的名称。例 3.1 定义的选择器为 p,p 在 HTML 中为段落标记，因此例 3.1 重新定义了段落标记的样式，网页中所有<p>标记的样式发生了变化。

属性是设置元素的格式，如字体、字号、颜色等。

3.1.2 CSS 的三种基本选择器

在 CSS 中，选择器分为基本选择器和复合选择器。基本选择器包括 HTML 标记选择器、class 选择器和 ID 选择器。

1. HTML 标记选择器

语法：

```
HTML 标记{  
    属性 1:属性值 1;  
    属性 2:属性值 2;  
    :  
}
```

例如：

```
div{  
    width:300px;  
    height:300px;  
}
```

代码中定义了 div 选择器,大小为 300px×300px,网页中只要出现< div >标记,则都按照这个尺寸显示。

标记选择器的特点:定义了标记选择器之后,网页中该标记的样式均发生改变。

2. class 选择器

class 选择器也称为类选择器,class 选择器的名称以英文句点开始,后面为英文单词。

(1) class 选择器定义。

类选择器定义语法:

```
.classname{
    属性 1:属性值 1;
    属性 2:属性值 2;
    ...}
```

例如:

```
.first{
    border:1px solid black;
    color:red;
}
```

类选择器 first 的样式为,边框为 1px 黑色的实线,文字颜色为红色。

(2) class 选择器应用语法:

```
<html 标记 class = "classname"></html 标记>
```

例如:

```
<div class = "first"></div>
<p class = "first"></p>
```

在< div >和< p >标记中均应用 first 类选择器,此时< div >和< p >拥有 first 中定义的属性,边框为 1px 黑色的实线,文字颜色为红色。

class 选择器的特点:当修改了类选择器之后,只有应用此类选择器的标记样式发生改变。

3. ID 选择器

ID 选择器名称是以英文#开始,选择器定义之后需要使用 id 属性应用样式。

(1) 选择器定义。

语法:

```
# IDname{
    属性 1:属性值 1;
    属性 2:属性值 2;
    :
}
```

例如：

```
#title{
    font-size:14px;
    font-weight:bold;
}
```

ID 选择器的 title,文字大小是 14px,粗体显示。

(2) 定义 ID 选择器应用。

选择器应用语法：

```
<html 标记 id = "IDname"></html 标记>
```

例如：

```
<div id = "title"></div>
```

在<div>标记中应用 ID 选择器 title,此时该<div>标记拥有 title 定义的属性,文字大小是 14px,粗体显示。

ID 选择器的特点：在网页中定义了 ID 选择器样式之后,只有应用这个 ID 选择器的标记样式发生改变。与 class 选择器不同的是,ID 选择器在网页中只能应用一次。

3.1.3 CSS 样式表的 4 种引入方法

使用 CSS 控制网页格式的方法有行内样式、内嵌式、链接式和导入式等方法。

1. 行内样式

通过 style 属性定义在 HTML 标记内部的样式称为行内样式,行内样式只能影响该标签内的对象,无法影响另一个标签内对象样式。

语法：

```
<html 标签 style = "属性 1:属性值 1;属性 2:属性值 2;...">
```

例如：

```
<p style = " border:1px solid black;color:red">
```

说明：行内样式可以应用在<body>标记的所有子标记,包括<body>标记在内,但不能用在<head><title><meta>等标记中。

2. 内嵌式

用<style>标记设置样式的方法称为内嵌式样式。

语法：

```
<style type = "text/css">
    选择器{
```

```
        属性 1:属性值 1;  
        属性 2:属性值 2;  
        ...}  
</style>
```

说明:

- < style >标记用来声明样式;
- type 属性表示 CSS 的 MIME 编码;
- 选择器可以是 3.1.2 节中介绍的三种选择器。

例如,定义一个内嵌式标记选择器:

```
< style type = "text/css">  
div{  
    width:300px;  
    height:300px;  
}  
</style>
```

3. 链接式

链接式是指引用外部独立的 CSS 文件。定义了外部样式之后,网站中的所有网页都可以引用此样式。方法为,在网页中通过< link >标记链接 CSS。

语法:

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

说明:

- href 表示外部样式表文件的路径;
- rel 表示浏览器引用的是 CSS 文件;
- type 属性表示 CSS 的 MIME 编码。

一个外部的样式表文件可以应用于多个网页,当改变外部样式表文件时,所有的页面样式都将随之改变。

【例 3.2】 将例 3.1 中的样式定义到外部的样式文件 cssurl.css 中,CSS 文件中样式代码如下:

```
div{  
    width:300px;  
    height:300px;}
```

网页源代码如下:

```
1 <html >  
2   < head >  
3     < link href = "cssurl.css" rel = "stylesheet" type = "text/css">  
4   </head >
```

```

5 <body>
6   <div>hello world!</div>
7 </body></html>

```

4. 导入式

与<link>标记类似,使用@import 可以导入外部的样式,但是@import 只能在<style> 标记中使用,而且必须放在其他 CSS 样式之前。

语法:

```

<style type="text/css">
  @import url(外部样式 url);
</style>

```

说明:使用导入式样式注意,引用外部样式语句结束一定要写上分号。

3.2 CSS 常见样式



3.2.1 设置文字样式

CSS 可以对网页中的文字格式进行设置,常见的文字格式如表 3.1 所示。

表 3.1 文字格式属性

文本属性	说明
font-size	定义字体大小(单位有 pt、pc、px、in、cm、mm)
font-family	定义字体类型(宋体,隶书,楷体)
font-style	定义字体样式(italic 斜体,normal 正常)
color	设置文本的颜色(可以是表示颜色的英文单词,或 RGB 颜色值)
font-weight	定义文字粗细(可以给出一个数值,数值越大文字越粗)
text-decoration	定义文字特殊效果(可取 underline,overline,line-through)
text-indent	定义文字首行缩进

【例 3.3】 制作 Google 公司的 Logo,如图 3.2 所示。



图 3.2 Logo 效果图

分析:该 Logo 由 6 个字母构成,每个字母具有一定的格式,可以分别为其定义样式。

```

1 <html>
2 <head>

```

```

3     <title>Google</title>
4     <style>
5         p{
6             font-size:80px;
7             letter-spacing:-2px;
8             font-family:Arial, Helvetica, sans-serif;
9         }
10    .g1, .g2{ color:#183dc6; }
11    .o1, .e{ color:#c61800; }
12    .o2{ color:#efba00; }
13    .l{ color:#32c33a; }
14    </style>
15    </head>
16    <body>
17    <p><span class="g1">G</span>
18        <span class="o1">o</span>
19        <span class="o2">o</span>
20        <span class="g2">g</span>
21        <span class="l">l</span>
22        <span class="e">e</span>
23    </p>
24    </body></html>

```

第7行代码设置字母间距,第10~第13行代码为Google中的6个字母分别定义了样式,第17~第22行代码应用样式。



设置图片样式

3.2.2 设置图片样式

可以为图片设置边框、位置等属性,具体属性如表3.2所示。

表 3.2 图片属性表

图片属性	说明
border-style	设置图片边框的线型,可选 dashed、dotted、groove、solid
border-color	设置图片边框颜色
border-width	设置边框粗细
border	将线型、颜色、粗细属性合并成一条语句,如 border: 5px double #ff00ff
text-align	设置水平对齐方式,可选 left、right、center
vertical-align	设置文字垂直对齐方式,可取 bottom、super、sub、middle 等
float	设置图片环绕文字方式,可选 left 或 right
margin	设置图片与文字之间距离
width	定义图片宽度,可以给一个相对值(百分比),表示相对于父元素的宽度
height	定义图片高度

【例 3.4】 设置图片边框效果。

利用 CSS 为图片的 4 个边框设置不同的样式风格。

```
1 <html >
2 < head >
3 < title>分别设置 3 边框</title >
4 < style >
5     img{
6         border - left:5px dotted #FF9900;
7         border - right:2px dashed #33CC33 ;
8         border - top: 10px #CC00FF solid;
9         border - bottom: 15px groove #666;
10    }
11 </style >
12 </head >
13 < body >
14 < img src = "grape. jpg">
15 </body ></html >
```

代码第 6~第 9 行分别定义了 img 标记的上右下左边框样式。网页效果图如图 3.3 所示。



图 3.3 img 样式应用

3.2.3 设置背景样式

利用 CSS 为网页元素设置背景颜色和背景图片。在 CSS 中背景的属性如表 3.3 所示。



设置背景样式

表 3.3 背景属性表

背景属性	说明
background-color	设置背景颜色
background-image	设置背景图像 URL(图片路径)
background-repeat	设置背景图像是否重复出现以及以哪种方式重复显示,可取值 repeat-x、repeat、no-repeat、repeat-y
background-position	设置背景图片的位置,如 bottom right(右下角位置)
background-attachment	设置背景图片是否固定,默认值为 fixed(固定背景图片)

【例 3.5】 为网页添加背景样式。

```

1 <html >
2   <head >
3     <style >
4       body{
5         padding:0px;
6         margin:0px;
7         background-image:url(jiuzg.jpg);
8         background-repeat:no-repeat;
9         background-position:bottom right;
10        background-color:# eeeee8;}
11     span{
12       font-size:70px;
13       float:left;
14       font-family:黑体;
15       font-weight:bold;}
16     p{
17       margin:0px; font-size:13px;
18       padding-top:10px;
19       padding-left:6px;
20       padding-right:8px;}
21   </style >
22 </head >
23 <body >
24   <p><span>九</span>寨沟大多数湖泊形成源于水中所含碳酸钙。远古时代,地球处于冰期时,
25   水中所含碳酸钙质无法凝结,只能随水漂流。到距今约 12000 年前,气候转暖后流水中的碳酸
26   钙质活跃起来,一旦遇到障碍物便附着其上,逐渐积累,形成今天九寨沟中一条条乳白色的钙质
27   堤埂,这些堤埂堆积起来形成堰塞湖,也就是所谓的"海子"…</p>
28 </body >
29 </html >

```

程序第 7 行定义网页主体部分背景图片,第 8 和第 9 行代码定义背景图像不重复及图像位置,第 13 行代码定义 span 标记中的文字首字放大。

浏览器运行效果如图 3.4 所示。

在网页中“九”字有首字下沉效果,图片作为背景出现在网页的右下位置。



图 3.4 首字下沉效果图

3.2.4 设置项目列表

项目列表中应用了 CSS 后，可以将列表制作成导航条形式。项目符号中常用的 CSS 属性如表 3.4 所示。



设置项目符号

表 3.4 列表样式表

列表属性	说明
list-style-type(用于 ul 标记)	定义是否显示项目符号
list-style-image	设置图片符号
float(用于 li 标记)	定义水平或垂直显示项目

【例 3.6】 项目符号样式应用。

```

1 <html >
2 < head >
3 < title >菜单的横竖转换</title >
4 < style >
5     ul{
6         list-style-type:none;
7     }
8     li{
9         float:left;
10        margin:2px; }
    
```

```

11     a{
12         display:block;
13         text-decoration:none;
14         margin:3px;}
15     </style>
16 </head>
17 <body>
18 <div>
19     <ul>
20         <li><a href = "#"> Home</a></li>
21         <li><a href = "#"> News</a></li>
22         <li><a href = "#"> Suggestions</a></li>
23         <li><a href = "#"> BBs</a></li>
24         <li><a href = "#"> Contact us</a></li>
25     </ul>
26 </div>
27 </body></html>

```

程序中第 6 行代码定义不显示项目符号,第 9 行代码将项目列表转换成水平列表,第 11 行代码将 a 标记转换成区块元素。程序运行效果如图 3.5 所示。



图 3.5 项目符号预览效果图

此外还可以为超链接设置不同的伪类别(分别代表不同的状态)。

3.2.5 设置超链接样式

超链接有 4 种伪状态:

- a: link 表示超链接普通样式;
- a: visited 表示被点击过超链接样式;
- a: hover 表示鼠标指针经过超链接的样式;
- a: active 表示在超链接上单击时超链接的样式。

注意:一般激活状态(a: active)很少用。在定义时,最好按照上述顺序进行描述(先定义 link 状态,然后 visited,最后 hover)。

【例 3.7】 在例 3.6 的基础上,增加超链接的三种伪状态,做出动态变化的超链效果。

```

1  a:link{
2      color:#002255;
3      text-decoration:none;}
4  a:visited{
5      color:#003399;

```



设置超链接

```

6     text-decoration:underline;}
7  a:hover{
8     color:#ffff00;
9     text-decoration:none;}

```

3.3 DIV+CSS 布局



盒子模型

3.3.1 盒子模型

1. 盒子标记

在网站开发中,经常将网页元素放置在<div>或盒子中。通过控制盒子的位置达到网页布局的目的。

1) <div>标记

<div>是区块标记,区块元素会自动换行。在<div>标记中可以容纳段落、标题、表格等多种 HTML 元素。

语法:

```
<div></div>
```

在网页中,块级元素还有<table>标记、<p>标记等元素。

2) 标记

标记表示行内元素,在行内元素前后不会自动换行,同时没有结构意义。标记也是一个容器,可以放置段落、标题、表格、图片等网页元素。

语法:

```
<span></span>
```

【例 3.8】 <div>和标记应用。

```

1 <html >
2   <head >
3     <style type = "text/css">
4       img{
5         width:200px;}
6     </style >
7   </head >
8   <body >
9     div 标记应用
10    <div ><img src = "img1. jpg"></div >
11    <div ><img src = "img1. jpg"></div >
12    span 标记应用
13    <span ><img src = "img1. jpg"></span >
14    <span ><img src = "img1. jpg"></span >
15  </body ></html >

```

上述代码在浏览器中的运行效果如图 3.6 所示。

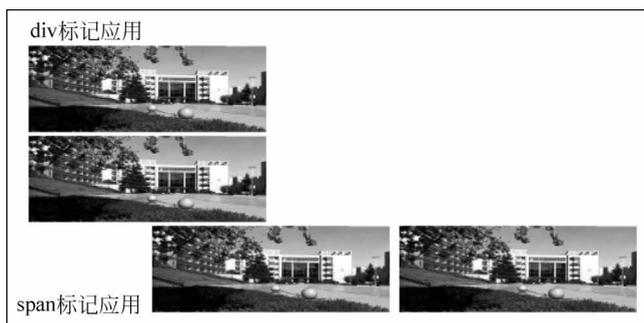


图 3.6 <div>标记和标记运行效果图

元素在使用时,如果一行没有占满就不会自动换行。元素没有结构上的意义,纯粹是为了应用样式。如果网页中其他标记都不适合时,可以试试元素。

2. 盒子模型

网页中可以将很多标记(如<p>标记、标记等)都看作盒子,盒子都具有边框,有一定的尺寸,占据着页面的一定的空间。通过调整盒子的边框、距离、内边距和外边距参数可以控制盒子的位置。

盒子模型常用的 CSS 属性如表 3.5 所示。

表 3.5 盒子模型的样式表

属 性	CSS 名称	说 明
边界属性	margin-top	设置盒子距外部其他盒子的上边距
	margin-right	设置盒子距外部其他盒子的右边距
	margin-bottom	设置盒子距外部其他盒子的下边距
	margin-left	设置盒子距外部其他盒子的左边距
边框属性	border-style	设置盒子边框的样式
	border-width	设置盒子边框的宽度
	border-color	设置盒子边框的颜色
填充属性	padding-top	设置内容与盒子上边框之间的距离
	padding-right	设置内容与盒子右边框之间的距离
	padding-bottom	设置内容与盒子下边框之间的距离
	padding-left	设置内容与盒子左边框之间的距离

3. 盒子元素的定位

在 CSS 中,可以通过下面两种方式对网页中的盒子进行定位。

1) float 定位

float 定位是设置元素相对于其他网页元素的定位方式,可以设置为 left、right 或默认 none。

在标准流中,一个块元素(独占一行的元素,如<div>、<p>、<table>等元素)在水平方向上会自动伸展,在垂直方向上和其他块级元素依次排列。如果希望块级元素并排显示的话,可以通过浮动方式实现。当设置了浮动属性,此时元素将脱离标准流,后面盒子将移动

到元素所在位置。当 float 取值为 left 或 right, 元素就会向父元素的左侧或右侧紧靠。设置了浮动, 盒子的宽度不再延伸, 会根据盒子中的内容决定宽度。当 float 设置为 none, 表示盒子不浮动。

float 属性的参数:

- left: 对象浮动在父元素的左边。
- right: 对象浮动在父元素的右边。

【例 3.9】 浮动定位应用。

```
1 <html >
2   <head >
3     <style type = "text/css">
4       .main{
5         width: 440px;
6         height: 230px;
7         border: 1px solid black; }
8     .leftdiv{
9       float: left;
10      width: 200px;
11      height: 200px;
12      border: 1px solid black; }
13    .rightdiv{
14      float: right;
15      width: 200px;
16      height: 200px;
17      border: 1px solid black; }
18  </style >
19 </head >
20 <body >
21   <div class = "main">
22     <div class = "leftdiv"> left </div >
23     <div class = "rightdiv"> right </div >
24   </div >
25 </body ></html >
```

2) position 定位

position 定位用来指定块的位置, 可以取 static、absolute、relative 和 fixed。

- static: 静态定位, 是 position 默认的属性值, 表示盒子按照标准流进行布局。
- absolute: 绝对定位, 使用标准流的排版方式, 盒子的位置以父盒子为基准进行偏移。
- relative: 相对定位, 使用标准流的排版方式, 表示盒子相对于它原来标准位置偏移的位置。
- fixed: 固定定位, 和绝对定位相似, 但是以浏览器窗口为基准进行定位。当点击浏览器窗口的垂直滚动条时固定定位的盒子位置保持不变。

【例 3.10】 静态定位实例。

```

1 <html >
2   < head >
3     < style type = "text/css">
4       .main{
5         width: 440px;
6         height: 230px;
7         border: 1px solid black; }
8       .son{
9         position: static;
10        width: 200px;
11        height: 200px;
12        border: 1px solid black; }
13     </style >
14   </head >
15   < body >
16     < div class = "main">
17       < div class = "son">静态定位</div >
18     </div >
19   </body >
20 </html >

```

上述代码在浏览器中的运行效果如图 3.7 所示。

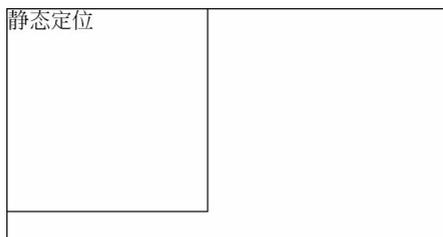


图 3.7 静态定位效果图

在静态定位中,元素保持原来位置,没有发生位移。

【例 3.11】 绝对定位实例。

```

1 <html >
2   < head >
3     < style type = "text/css">
4       .main{
5         width: 440px;
6         height: 230px;
7         border: 1px solid black;
8         position: absolute;    }
9       .son{
10        position: absolute;
11        width: 200px;
12        height: 200px;
13        border: 1px solid black;

```

```
14         top:20px;
15         left:10px;}
16     </style>
17 </head>
18 <body>
19     <div class = "main">
20     <div class = "son">静态定位</div>
21     <div>
22 </body>
23 </html>
```

采用绝对定位 son 盒子相对于 main 盒子向右和向下移动。上述代码在浏览器中的运行效果如图 3.8 所示。

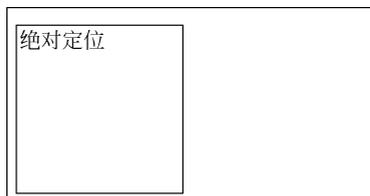


图 3.8 绝对定位效果图

【例 3.12】 相对定位实例。

```
1 <html>
2 <head>
3   <style type = "text/css">
4     .main{
5       width: 440px;
6       height: 230px;
7       border: 1px solid black;
8       position: absolute;    }
9     div{
10      border: 1px solid black;  }
11    .son{
12      position:relative;
13      width: 200px;
14      height: 200px;
15      border: 1px solid black;
16      top:10px;
17      left:10px;    }
18  </style>
19 </head>
20 <body>
21   <div class = "main">
22   <div>普通盒子</div>
```

```
23     <div class = "son">相对定位</div >
24   </body >
25 </html >
```

采用相对定位 son 盒子相对自身位置向后、向下移动。上述代码在浏览器中的运行效果如图 3.9 所示。

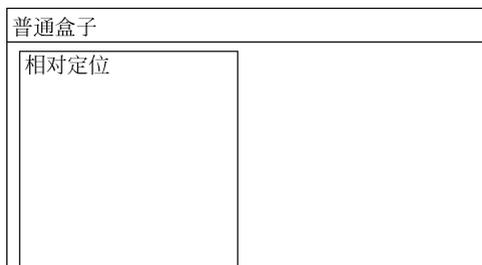


图 3.9 相对定位效果图

3.3.2 DIV+CSS 布局

DIV+CSS 布局步骤：对网页整体分区；对每个分区通过 CSS 定位。使用 CSS 技术排版的页面，可以更新 CSS 属性重新定义板块的位置，因此这种排版方式比表格布局要灵活。

3.3.3 常见布局结构

1. 宽度固定且居中

这种布局方式页面容器宽度固定，页面元素相对于浏览器窗口水平居中对齐。这是网页开发中常用的布局方式。

【例 3.13】 宽度固定且居中的布局实例。

制作方法：

(1) 定义 container 父容器。

```
1 <html >
2   <head ><title >固定宽度并居中的例子</title ></head >
3   <body >
4     <div id = "container">页面内容</div >
5   </body >
6 </html >
```

第 4 行代码中，div 盒子应用 container 样式。

定义 container 容器样式。

```
#container{
    position:relative;
```



常见布局结构

```
margin:0 auto;
width:680px;}
```

说明：父容器的宽度是 680px，父容器应用相对定位方式，左右外边距为自动。

(2) 定义 body 样式。

```
body{
margin:0px;
text-align:center;
}
```

上述代码定义 body 距页面上下左右的外边距为 0，同时定义网页中的所有元素都居中。

2. 川字结构

川字结构也称为左中右结构，这种结构也是常见的排版模式，如图 3.10 所示。

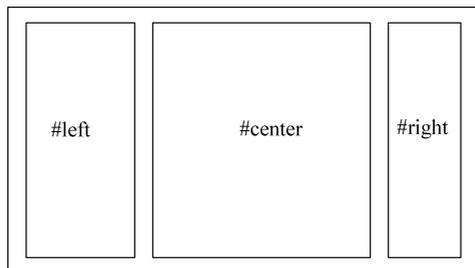


图 3.10 川字结构图

川字结构网页布局的方法，先确定左边、右边块以及中间块的大小，使用绝对或相对定位方式定位，在设置盒子大小时注意左中右盒子的大小之和不能超过页面的宽度，否则右侧盒子将会换行。

【例 3.14】 使用绝对定位方式实现左中右布局实例。

制作方法：

(1) 搭建 HTML 的结构框架，定义三个 <div> 块。代码如下：

```
<body>
<div id="left"> left </div>
<div id="center"> center </div>
<div id="right"> right </div>
</body>
```

(2) 设置左边 <div> 块的样式，绝对定位，距 body 左侧 0px，<div> 块的宽度 190px。

```
#left{
position:absolute;
```

```
top:0px;
left:0px;
width:190px;}
```

(3) 设置中间块的样式,中间<div>块到左右块的距离为 190px。

```
#middle{
margin:0px 190px 0px 190px;
}
```

(4) 设置右侧块的样式。

```
#right{
position:absolute;
top:0px;
right:0px;
width:190px;
}
```

设置右边块的位置为绝对定位(与左边块一样),距离浏览器右边框为 0px,上边框为 0px,此块的宽度为 190px。

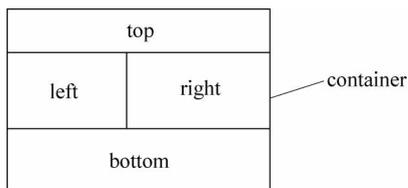


图 3.11 国字形结构构图

3.3 行 2 列式国字形布局

国字形结构包括顶部区域(top),中间主体部分(container)和下方版尾(bottom)三个部分,主体部分又可以细分为左侧(left)和右侧(right),如图 3.11 所示。

【例 3.15】 通过 float 方式实现国字形布局实例。

制作方法:

(1) 定义 5 个区块,分别是 container、top、left、right 和 bottom。

```
<body>
<div id="container">
<div id="top"> top </div>
<div id="left"> left </div>
<div id="right"> right </div>
<div id="bottom"> bottom </div>
</div>
</body>
```

(2) 定义 body 和 container 的样式。设置 body 中元素居中对齐。

```
body{
margin:0px;
text-align:center;}
```

```
# container{  
    position:relative;  
    width:780px;}
```

(3) 定义 top、left 和 right 样式。设置 top 区块绝对定位, left 盒子左侧浮动定位, right 盒子右侧浮动定位。

```
# top{  
    position:absolute;  
    top:0px;  
    left:0px;  
    height:90px;  
    width:780px;}  
  
# left{  
    float:left;  
    width:300px;}  
  
# right{  
    width:379px;  
    float:right;}
```

(4) 定义 bottom 样式。bottom 盒子中清除左右浮动的影响(clear: both)

```
# bottom{  
    clear:both;  
    height:60px;  
    width:780px;}
```

为每个块设置了边框属性(border: 3px solid #113366)。浏览器中运行效果如图 3.12 所示。

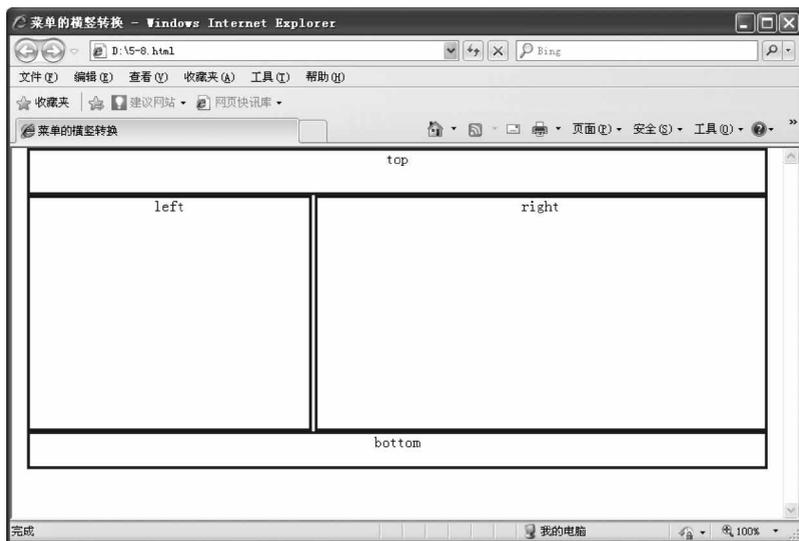


图 3.12 国字形布局效果图

可以在每个块中填充具体的网页元素,并且为其设置属性。

本书常见问题:

(1) 网页中定义的样式为何没有起作用?

【答】 造成这种结果的原因很多,这里列出几种常见的问题:

① 定义的样式没有在网页中应用,如定义了 #title 样式,但是在网页中没有使用样式。

② 样式定义错误,如给 font 样式定义了背景属性,背景对于文字不起作用。

(2) 利用表格也可以对网页布局,那么用 DIV+CSS 布局和用表格布局有什么不同?

【答】 采用表格进行布局是网页设计早期使用的排版方法,由于存在诸多不足,所以现在基本上已经被淘汰了。表格布局和 DIV+CSS 布局的区别如下:

① 使用表格布局:设计复杂,改版时工作量大;同时格式代码与内容混合,可读性差,而且也不利于数据调用分析;另外网页文件量大,浏览器解析速度慢。

② 使用 DIV+CSS 布局:高效率的开发与维护简单,网页解析速度快,用户体验好。

本章小结

CSS 是层叠样式表的缩写,它的作用是定义网页的外观和布局,可以使网页的内容和格式分离。

CSS 样式有标记选择器、类选择器和 ID 选择器,3 种基本选择器。标记选择器定义之后会自动对网页中的多个相同标记的元素产生影响(不需要应用样式),类选择器和 ID 选择器在定义之后,需要手动应用。

CSS 样式有行内式、内嵌式、链接式和导入式 4 种样式设置方法。其中链接式和导入式样式是针对外部样式表文件。行内式样式的应用面最小,仅仅为某一个标记所使用。内嵌式样式作用域在一个网页中,而外部样式(链接式和导入式)可以为网站中所有网页所使用。

使用 CSS 可以设置文字的大小、颜色、字体、字号、粗细、段间距、字间距等属性;设置图像的边框、浮动方式等属性;设置网页的背景图片并且指定背景图片的位置、大小、是否重复出现;让项目列表大变身,变为水平的列表。

通过 CSS 和 DIV 结合,对网页元素进行布局。利用 CSS 可以制作宽度固定且居中的网页布局、川字结构和 3 行 2 列式国字形等复杂布局。

课后习题

(1) CSS 选择器有_____、_____和_____。

(2) 在外部编写 CSS 文件,需要在< head ></ head >之间写上_____标记链接样式表文件,实现 HTML 与 CSS 分离。

(3) 盒子模型由_____、_____、_____和_____组成。

(4) 盒子的 position 定位有_____、_____、_____和_____。