

# 第 4 章

## 网站前台静态网页设计

### 4.1 项目导引

目前市面上见到的网站布局基本都是采用 table 和 div, 无论采用哪种方式进行布局都要根据网站本身特点、需求单位要求、网站后期维护、网站性价比等多种因素决定。如果是一些小网站, 前台一旦确定布局基本不变情况下都采用 table, 因为它开发效率较快、不用考虑浏览器兼容性问题。但如果对一些大中型网站, 企业要求较高的网站, 为了便于以后的二次开发或者维护一般都采用 div 进行布局, 这种方式设计费用较高、布局设计相对来说需要时间较多。

那么无论采用何种方式进行页面布局, 网站前台中都离不开样式 CSS 的使用, 哈尔滨众仁智业人力资源有限公司, 开发的 <http://www.heyishengya.com/360ejob> 平台, 为求职者提供一个求职平台, 这个平台的整个前台布局采用了 table 和 div 相结合的布局方式, CSS 样式在平台布局设计中得到了广泛应用。

### 4.2 项目分析

本章主要介绍完成 <http://www.heyishengya.com/360ejob> 所需要的前台静态 HTML 部分知识, 如图 4-1 所示, 是平台登录的窗口, 那么本章不是介绍图片的美观, 而是介绍这样的网页页面如何布局, 使用 table、div 都可以, 但必须达到预期的效果。那么布局完毕后, 网页的样式如何达到在 Photoshop 中的设计效果, 就需要灵活使用 CSS 样式了, 这也是本章重点。

本系统网站页面很多, 不可能一一介绍, 在这里只介绍布局设计思路、应用到的技术, 读者在实际应用中根据设计思路灵活使用技术。

图 4-2 是对图 4-1 的一个布局划分。在开发的时候, 首页由前台设计者根据用户需求设计出如图 4-1 所示的页面效果图, 网页设计者是把页面效果图转为 HTML 的静态网页。对登录界面进行布局如下:



图 4-1 平台登录页面

(1)先加一个总盒子,其余盒子可以被它包含,因为网页需要居中显示,那么设计者只要调一个盒子居中,它包括的所有子盒子就自动居中了。

(2)把一些标记样式初始化一下,例如 ul 标记在默认情况下,文字前面会空出来一部分,因此将它们进行初始化设置:margin:0px;padding:0px;

(3)先把整个网页分成两行。

(4)从上向下设计,每一个行从左向右设计。

(5)每一行可以看做一个一行一列的表格,也可以看做一个 div 标记。每一行根据实际情况再包含多个表格或者多个 div 标记。

(6)每一行根据内容复杂程度再分成多少列;每一列又根据实际再分若干行,每一行再分列,依次类推下去。本页第二行由于内容较多,那么把第二行先分成了两列,第一列需要放置一个图片,确定宽和高跟 ps 裁切图片大小相同;此处图片不需要加文字,因此设计为背景图像或者直接插入图片都可以。第二列复杂一些,需要把第二例分成两行,第一行放置图片,与左侧图片处理方式相同;第二行的内容较多,需要添加详细的样式。

对网页进行布局,可以归纳为先分行,一行一行做,从上到下,从左至右,从大到小,从简单到复杂。



图 4-2 平台登录页面布局划分

## 4.3 技术准备

### 4.3.1 CSS 简介

HTML 与 CSS 的关系就是“内容”与“形式”关系,通过 HTML 部分设计网页内容,而通过 CSS 来确定网页内容的表现形式。

CSS 是由 W3C 组织负责制定和发布的,在 1996 年发布了第一个版本 CSS 1.0,而在 1998 年发布了 2.0 版本,目前得到广泛应用的是 CSS 2.1 和 3.0 版本。

在发布 CSS 标准后,W3C 没有强制要求所有的软件公司遵循其规范,所以目前很多浏览器并没有严格遵循统一的规范进行设计网页,这样就给很多程序员在编程时带来了很大困难,需要不断调节 CSS 适合每一款浏览器。但在中国使用最广泛浏览器 IE 和 Firefox 基本上已经达到了显示一致的 CSS 布局页面。

网页中的 HTML 和 CSS 是两种不同的语言,要把两者很好的融合在一起才能让网页页面达到预期效果,那么 HTML 中引入 CSS 有三种方法:

#### 1. 行内样式

行内样式也就是在 HTML 标记中加入 style 属性,这种方式在程序开发中达不到“内容”与“形式”的分离,不值得推荐,例如:

代码:

```
<h1 style="color:red;">CSS 样式简介</h1>
```

效果如图 4-3 所示。



图 4-3 样式效果

#### 2. 嵌入样式

嵌入样式是将对页面元素的样式集中写在<head>和</head>之间,对于单个网页来说,这种格式比较合适。如果一个网站包括很多个网页,那么这种格式就失去意义了,因为在一个网站多个页中都写入样式就很难维护,如果想修改某些相同的样式效果需要逐个页面修改,所以这种方式也不值得提倡,例如:

代码:

```
<html>
<head>
  <title>嵌入样式格式</title>
  <style type="text/css">
    h1{
      color:red;
      background-color:blue;
```

```
    }  
</style>  
</head>  
<body>  
    <h1>嵌入样式效果</h1>  
</body>  
</html>
```

效果如图 4-4 所示。



图 4-4 嵌入样式效果

### 3. 外部样式

外部样式就是 CSS 代码和 HTML 代码存放在两个不同的文件中,HTML 网页中使用样式时,需要通过命令将 CSS 样式引入,分为导入式和链接式。

链接式是在<head>和</head>部分加入外部文件。

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

导入式是在<head>和</head>部分加入外部文件。

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

```
    @import "mystyle.css";
```

```
</style>
```

这两种方式还是有一定区别的,链接式是在网页主体部分装载前装载 CSS 样式文件,这种方式是网页从一开始就带有样式效果的。而导入式会在网页加载后再装载 CSS 文件,如果网页整体很大,那么开始显示时不带有样式效果,闪烁一下才看到带样式效果的页面。

值得注意的是如果开发一个较大的网站,为了便于程序员维护,开发样式时会把样式分成几类放到不同的样式文件中,如果采用链接式则需要很多句引入命令,这个时候如果在调整样式分类,那么还需要调整 HTML 文件中的引入命令,这样维护成本增加。如果采用导入式可以先导入一个总样式,在总样式中再导入分类样式,这样就解决了维护难问题,而链接式不具有这个特性。

### 4.3.2 样式选择器

选择器是 CSS 中一个很重要概念,HTML 中所有标记的样式都是通过选择器来设置 CSS 样式的。

选择器可以这样理解,例如 h1 标题用黄色文字显示,h2 标题用红色文字显示,每一个“内容”都要有一个“形式”和它对应,也就是 CSS 和 HTML 元素一一对应起来,从而实现了 CSS 对 HTML 的“选择”。

CSS 中选择器可以分为基本选择器和复合选择器。基本选择器又包括了标记选择器、类选择器、ID 选择器;复合选择器就是基本选择器的多个组合,可以分为交集选择器、并集选择器、后代选择器。

## 1. 基本选择器

基本选择器是网页开发中最常使用的,一个前台设计者必须掌握的基本技能,下面就对基本选择器的三种类型做详细介绍。

### (1) 标记选择器。

HTML 网页中有很多的元素,CSS 标记选择器就是来声明每个标记所对应的样式的,每个 HTML 标记名作为 CSS 中的标记选择器的名称。h2 选择器就是来声明网页主体中所有的 <h2> 标记的样式风格的,例如:

```
<style>
  h2{
    color:red;
    background-color:blue;
    font-size:12px;
  }
</style>
```

代码中设定了网页主体部分中所有 <h2> 标记的样式:字体为红色,背景为蓝色,字体大小为 12 像素。从这段代码可以看出,每个标记选择器的格式,含有标记选择器名称,这个标记选择器名称必须是现有的 HTML 标记名称,每个标记选择器中可以含有多个属性和值。属性和值被大括号所包含,属性和值间用冒号连接代表赋值关系,值后面使用分号结束。

值得注意的是这里的属性和值都必须是正确的 CSS 规范中存在的,否则 CSS 样式不会执行样式效果。

### (2) 类选择器。

标记选择器一旦被声明,那么 HTML 中所有对应的标记元素都会显示相应的效果。例如把 <h1> 标记声明了蓝色文本,网页中所有的 <h1> 标记都将会变成蓝色文本显示,如果想让其中一个不为蓝色,而是显示为红色,用标记选择器是解决不了的,那么就会产生另外一种选择器——类选择器。

类选择器不像标记选择器声明那么严格,类选择器的名称可以由用户自己确定,在自定义的类选择器名称前加一个“.”。类选择器中的属性和值与标记选择书写格式及含义是相同的。

例如:网页中有三个 h2 标记,每个标记的文字颜色都不相同,这个效果就不能采用标记选择器来完成,可以通过类选择来完成。

```
<html>
<head>
<title>class 选择器</title>
<style type="text/css">
.h2red{
  color:red;          /* 文字颜色:红色 */
  font-size:18px;    /* 文字大小 */
}
.h2green{
  color:green;       /* 文字颜色:绿色 */
  font-size:20px;    /* 文字大小 */
```

```
    }  
    .h2blue{  
        color:blue;        /* 文字颜色:蓝色 */  
        font-size:24px;    /* 文字大小 */  
    }  
</style>  
</head>  
<body>  
<h2 class="h2red">class 选择器 1</h2>  
<h2 class="h2green">class 选择器 2</h2>  
    <h2 class="h2blue">class 选择器 3</h2>  
<h3 class="h2blue">h3 同样适用</h3>  
</body>  
</html>
```

显示效果如图 4-5 所示,图中三个<h2>标记都显示了不同的效果,而且任何一个类选择器都适应于任何一个 HTML 标记,只需要在 HTML 标记中加入 class 属性并赋值某一个类选择器就可以,例子中的<h3>标记同样适应于类选择器“.h2blue”。



图 4-5 class 选择器

### (3) ID 选择器。

ID 选择器的使用和类选择器基本一致,选择器的名称也是可以由用户自己定义,在自定义的 ID 选择器名称前加一个“#”,属性与值的格式和类选择器是一样的。不同之处是同一个类选择器可以在 HTML 网页中重复使用,而同一个 ID 选择器在 HTML 网页中只能被使用一次,所以它的使用针对性较强。在使用时只需要在 HTML 标记中加入 ID 属性并赋值某一个 ID 选择器就可以,例如:

```
<html>  
<head>  
<title>ID 选择器</title>  
<style type="text/css">
```

```
#first{
    font-weight:bold;    /* 粗体 */
}
#second{
    font-size:30px;    /* 字体大小 */
    color:#009900;    /* 颜色 */
}
</style>
</head>
<body>
<p id="first">ID 选择器 1</p>
<p id="second">ID 选择器 2</p>
<p id="second">ID 选择器 3</p>
</body>
</html>
```

显示结果如图 4-6 所示,这里第 2 行和 3 行文字都调用了 # second 选择器,虽然效果正常显示了,但在 JavaScript 编程用到一个命令 document.getElementById("ID 名称"),要用到 ID 的名称,上述代码中两行文字的 ID 相同,在编程时就会出现这个问题,所以不要把同一个 ID 选择器用到多个 HTML 标记中。

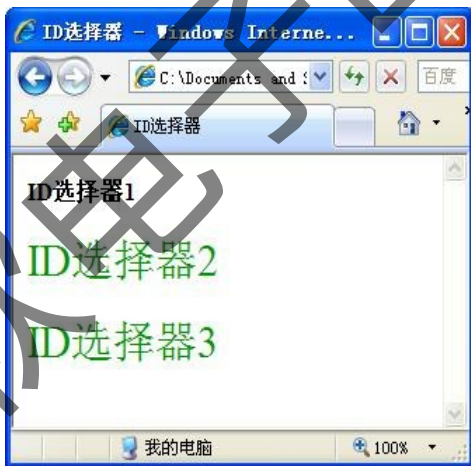


图 4-6 ID 选择器

## 2. 复合选择器

复合选择器就是把多个基本选择器通过不同种方式连接起来的一种形式。

### (1) 交集选择器。

交集复合选择器由两个基本选择器组成,其中第一个为标记选择器,第二个为类选择器或者 ID 选择器,这两个选择器之间不能有空格,必须连续书写。例如 h2.first、p#one。

这种方式组合的选择器,将对应同时满足前后选择器的 HTML 元素,即元素名与第一个标记选择器名称对应,元素中加入 class 或者 id 属性即可。例如:

```
<html>
<head>
```

```
<title>交集复合选择器</title>
<style type="text/css">
p{          /* 标记选择器 */
  color:blue;
}
p.ys{      /* 标记.类别选择器 */
  color:red;    /* 红色 */
}
.ys{       /* 类别选择器 */
  color:green;  /* 绿色 */
}
</style>
</head>
<body>
<p>正常段落文本</p>
<h3>正常 h3 标题文本</h3>
<p class="ys">指定了.ys类别的段落文本</p>
<h3 class="ys">指定了.ys类别的标题文本</h3>
</body>
</html>
```

运行结果如图 4-7 所示,代码中定义了 p 标记选择器、.red 类选择器,还定义了一个交集复合选择器 p.ys,这个 p.ys 定义的选择器只能控制元素格式为<p calss="ys">的样式,而不会影响使用了.ys的其他 HTML 标记。



图 4-7 交集选择器

(2) 并集复合选择器。

并集选择器是多个选择器通过逗号连接而成的,在使用并集选择器时,说明各类基本选择器的效果完全相同或者部分相同时才使用,也可以理解为集体声明,例如:

```
<html>
<head>
```



```
<title>并集选择器</title>
<style type="text/css">
h1, h2, p{ /* 并集选择器 */
  color:purple; /* 文字颜色紫色 */
  font-size:15px; /* 字体大小 15 像素 */
}
h2.special, .special, #one{ /* 并集选择器 */
  text-decoration:underline; /* 下划线 */
}
</style>
</head>
<body>
<h1>示例文字 h1</h1>
<h2 class="special">示例文字 h2</h2>
<p>示例文字 p1</p>
<p class="special">示例文字 p2</p>
<p id="one">示例文字 p3</p>
</body>
</html>
```

效果如图 4-8 所示,可以看出所有行都是紫色文字,大小是 15 像素,这种并集选择器的书写与分开单独书写标记选择器效果是相同的。后面声明的 `h2.special`, `.special`, `#one` 并不会影响到前面声明的样式效果,而是在前面基础上又添加了新效果,这就是一种并集效果,即两部分样式和为最后样式效果。



图 4-8 并集选择器

### (3) 后代复合选择器。

在 CSS 样式中,存在一种嵌套格式的选择器,产生这个选择器的原因是由于 HTML 中存在元素间嵌套,例如 `<h1><b>文字</b></h1>`,这里的 `<b>` 标记嵌套在 `<h1>` 标记中,也可以说 `<h1>` 标记包含 `<b>` 标记。这个时候如果想控制 `<b>` 标记的样式效果,就可以使用后代选择器了。

后代选择器的书写格式为,把外层标记写在前面,内层标记写在后面,之间用空格分开,内层标记就是外层标记的后代了,例如:

```
<html>
<head>
<title>后代选择器</title>
<style type="text/css">
p span{ /* 嵌套声明 */
    color:red; /* 颜色 */
}
span{
    color:blue; /* 颜色 */
}
</style>
</head>
<body>
<p>嵌套使<span>用 span</span>标记</p>
嵌套之外的<span>span 标记</span>不生效
</body>
</html>
```

运行结果如图 4-9 所示,上面代码可以看出后代选择器 p span 声明的效果不会影响单独声明的 span 标记效果。而外部的单独声明的 span 效果也不会影响到后代选择器 p span 的效果。

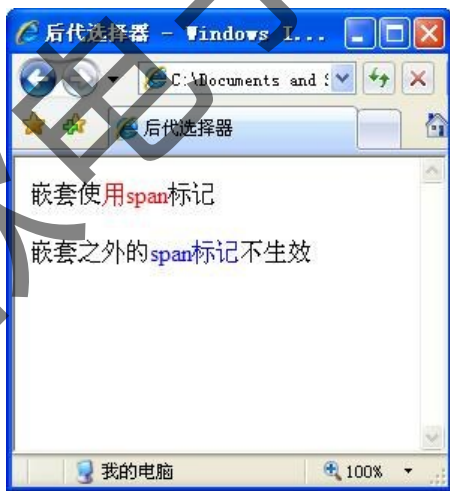


图 4-9 后代选择器

### 4.3.3 边框(border)

什么是 CSS 的盒子模式呢?为什么叫它是盒子?先说说我们在网页设计中常听的属性名:内容(content)、填充(padding)、边框(border)、边界(margin),CSS 盒子模式都具备这些属性,如图 4-10 所示。

可以把这些属性转移到日常生活中的盒子(箱子)上来理解,日常生活中所见的盒子也具有

这些属性,所以叫它盒子模式。那么内容就是盒子里装的东西;而填充就是怕盒子里装的东西(贵重的)损坏而添加的泡沫或者其他抗震的辅料;边框就是盒子本身了;至于边界则说明盒子摆放的时候不能全部堆在一起,要留一定空隙保持通风,同时也为了方便取出。在网页设计上,内容常指文字、图片等元素,但是也可以是小盒子(div 嵌套),与现实生活中盒子不同的是,现实生活中的东西一般不能大于盒子,否则盒子会被撑坏的,而 CSS 盒子具有弹性,里面的东西大过盒子本身时最多把盒子撑大,但它不会损坏。填充只有宽度属性,可以理解为生活中盒子里的抗震辅料厚度,而边框有大小和颜色之分,我们又可以理解为生活中所见盒子的厚度以及这个盒子是用什么颜色材料做成的,边界就是该盒子与其他东西要保留多大距离。



图 4-10 盒子模型

盒子模型有两种,分别是 IE 盒子模型和标准 W3C 盒子模型。他们对盒子模型的解释各不相同,

标准盒子模型,如图 4-11 所示。

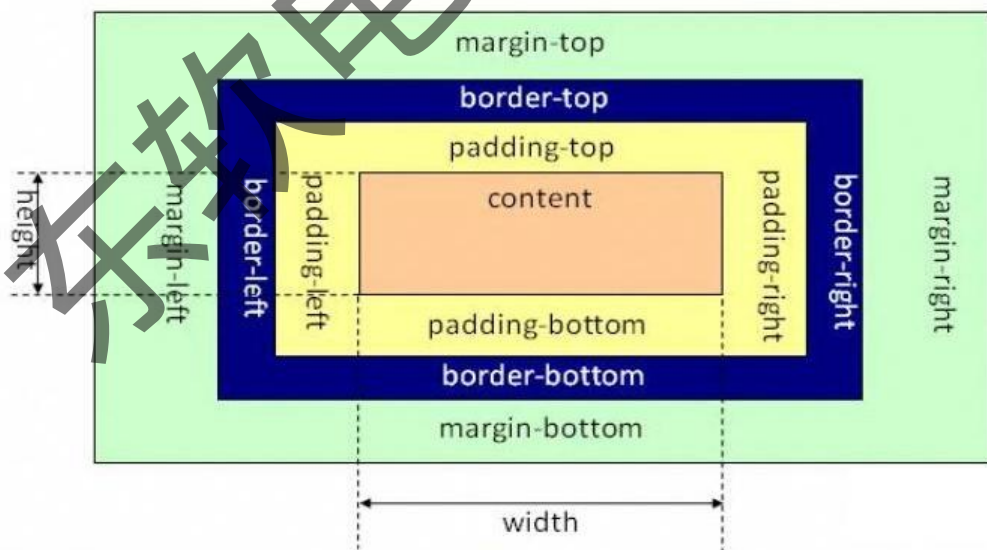


图 4-11 标准盒子模型

可以看出标准 W3C 盒子模型包括 margin、border、padding、content,并且 content 部分不包含其他部分。

IE 盒子模型如图 4-12 所示。

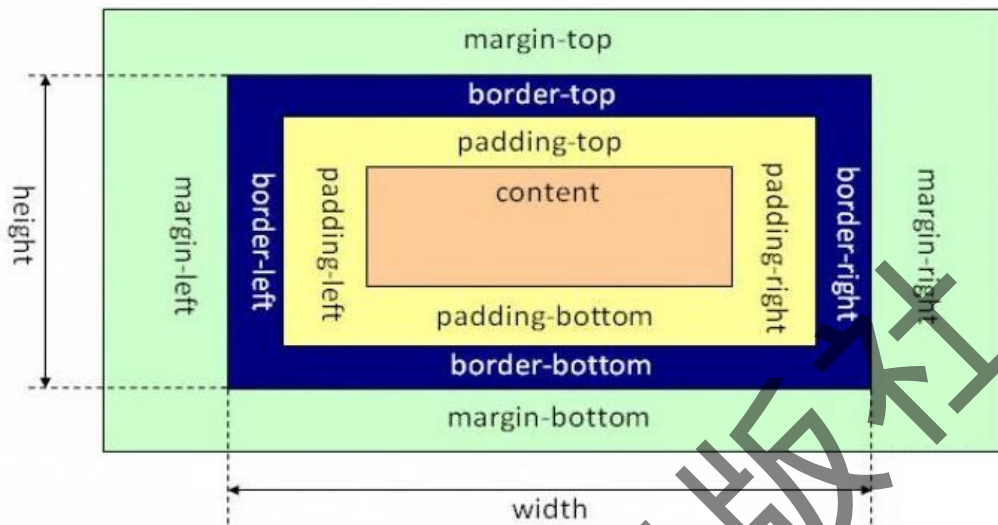


图 4-12 IE 盒子模型

可以看到 IE 盒子模型的范围也包括 margin、border、padding、content，和标准 W3C 盒子模型不同的是：IE 盒子模型的 content 部分包含了 border 和 padding。

元素外边距内就是元素的边框（border）。元素的边框就是元素内容和内边距之间的区域。每个边框有 3 个属性：宽度、样式以及颜色。

### 1. 边框的宽度

可以通过 border-width 属性为边框指定宽度。为边框指定宽度有两种方法：可以指定长度值，比如 2px 或 0.1em；或者使用 3 个关键字之一，它们分别是 thin、medium（默认值）和 thick。通过下列属性分别设置边框各边的宽度：border-top-width、border-right-width、border-bottom-width、border-left-width。

```
p {
  border-style: solid;
  border-top-width: 15px;
  border-right-width: 5px;
  border-bottom-width: 15px;
  border-left-width: 5px;
}
```

### 2. 边框的颜色

CSS 使用 border-color 属性对边框颜色进行设置，可以分别为每一条边框设置颜色属性，它们的原理与单边样式和宽度属性相同。

```
border-top-color
border-right-color
border-bottom-color
border-left-color
```

### 3. 边框样式

盒子的边框有很多种样式，例如 none、hidden、dotted、dashed、solid、double、groove、ridge、

inset,outset。

```
<html>
<head>
<title>边框样式</title>
</head>
<body>
  <p style=border-style:none>无边框</p>
  <p style=border-style:dotted>点线式边框</p>
  <p style=border-style:dashed>破折线式边框</p>
  <p style=border-style:solid>直线式边框</p>
  <p style=border-style:double>双线式边框</p>
  <p style=border-style:groove>槽线式边框</p>
  <p style=border-style:ridge>脊线式边框</p>
  <p style=border-style:inset>内嵌效果的边框</p>
  <p style=border-style:outset>突起效果的边框</p>
</body>
</html>
```

代码效果如图 4-13 所示。

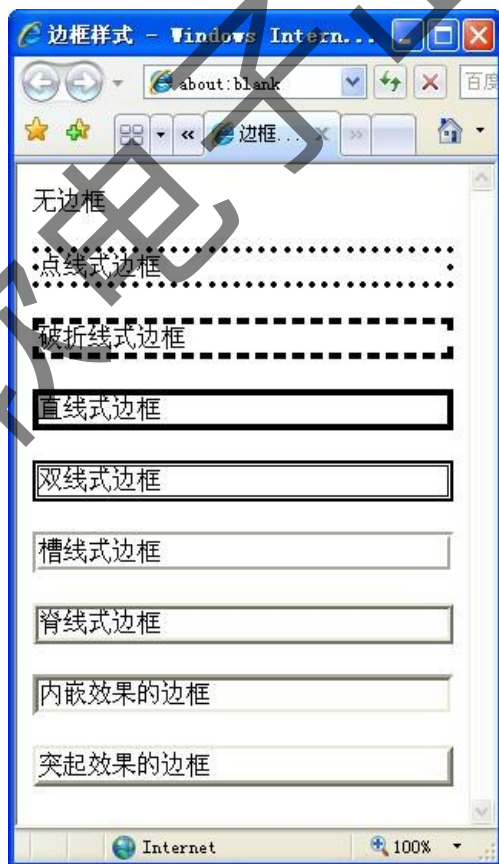


图 4-13 边框样式

如果一个盒子的四个边框样式都相同,那么样式代码可以把宽度、颜色、样式合起来写: `border:1px red solid`。如果四个边框样式不相同,需要对四个边框单独书写样式: `border-left:1px red solid`。

### 4.3.4 内边距(padding)

`padding` 称为内边距,用来控制内容与边框之间的距离,`padding` 的属性值有以下几种形式:

- 1 个值:代表上、下、左、右四个方向都是该值。
- 2 个值:第一个代表上、下值,第二个值代表左右值。
- 3 个值:第一个代表上面距离值,第二个代表左、右两个方向值,第三个代表下面距离值。
- 4 个值:依次是上、右、下、左四个方向的值。

如果单独想设置某个方向值,可以使用代码: `padding-left`、`padding-right`、`padding-top`、`padding-bottom`。

例如:

```
<html>
<head>
<title>内边距</title>
<style>
p {
border-style:dashed;
border-top-width: 3px;
border-right-width: 1px;
border-bottom-width: 5px;
border-left-width: 1px;
padding:5px 2px 20px;
}
</style>
</head>
<body>
<p>内边距</p>
</body>
</html>
```

运行效果如图 4-14 所示。



图 4-14 内边距

### 4.3.5 外边距(margin)

margin 指的是元素与元素之间的距离,元素之间的距离也存在四个方向,给 margin 赋值时也有四种方式,与 padding 四种方式相同,这里不做介绍。

例如:

```
<html>
<head>
<title>外边距</title>
<style>
.box1 {
border:1px solid red;
margin-bottom:30px;
height:50px;
}
.box2 {
border:1px solid blue;
height:50px;
}
</style>
</head>
<body>
<div class="box1"></div>
<div class="box2"></div>
</body>
</html>
```

运行效果如图 4-15 所示。

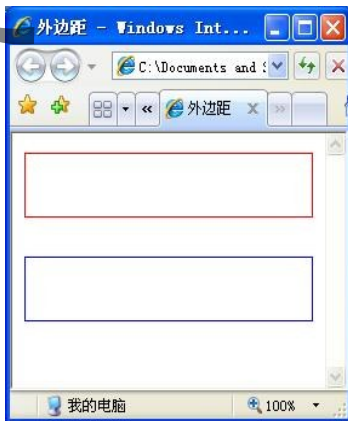


图 4-15 外边距

### 4.3.6 盒子的定位

在 CSS 样式中控制盒子的布局方式称为定位,使用的属性是 position,其属性值共有四个:

static(静态)、relative(相对)、absolute(绝对)、fixed(固定)。

### 1. static(静态定位)

static 为盒子默认值,效果是盒子保持在原来应该在的位置,前面所有实例中的效果都是采用这个布局方式,这里不再做说明。

例如:

```
<style>
#parent{
    background-color: #e8f3fd;
    border: 1px dotted #000000;
    padding: 20px;
}
#son{
    background-color: #CCCCCC;
    border: 1px dashed #000000;
    padding: 10px;
}
</style>
<div id="parent">
    <div id="son">Box-2</div>
</div>
```

运行效果如图 4-16 所示。

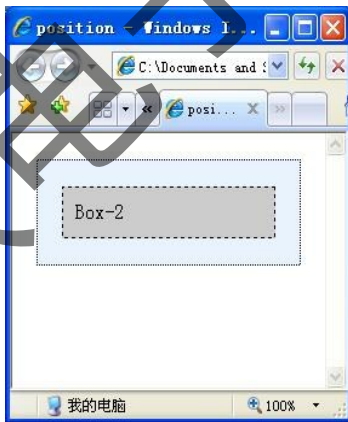


图 4-16 静态定位

### 2. relative(相对定位)

相对定位的含义是盒子从原来出现的位置偏移了多少,偏移量使用 left、top、right、bottom 四个属性控制,这四个属性只有 position 属性值为 absolute、relative、fixed 时才有效,而且属性值不相同代表的含义也不相同,在这里 position 值为 relative 时,代表是距离原来位置偏移了多少距离。

例如:

```
<style>
#parent{
```



```
background-color: #e8f3fd;
border: 1px dotted #000000;
padding: 20px;
}
#son{
background-color: #CCCCCC;
border: 1px dashed #000000;
padding: 10px;
position: relative; /* relative 相对定位 */
left: 35px; top: 35px;
}
</style>
<div id="parent">
    <div id="son">Box-2</div>
</div>
```

运行效果如图 4-17 所示,通过浏览效果可以看出 Box-2 虽然偏离了父盒子,但并没有把父盒子充大,由此得出,相对定位只是盒子的位置相对原来的位置发生了变化,而对其盒子本身、兄弟及父盒子没有影响。

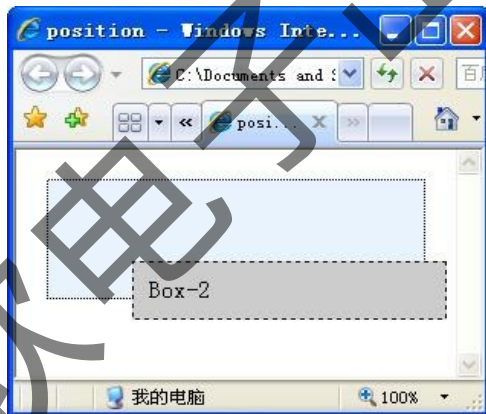


图 4-17 相对定位

### 3. absolute(绝对定位)

绝对定位与相对定位有些类似,只是偏移量的基础线不同,如果盒子的父级别元素没有设置定位属性,那么绝对定位偏移值是以浏览器窗口为基准线,如果父级别元素设置了定位属性,那么绝对定位偏离是以父级别元素为基准线的。

什么是设置了定位值呢,也就是 position 属性设置了除 static 之外的值。

例如:

```
<style>
#parent1{
background-color: #e8f3fd;
border: 1px dotted #000000;
padding: 20px;
```

```
    position:relative;
}
# parent2{
    background-color: # e8f3fd;
    border:1px dotted # 000000;
    padding:20px;
}
# son1{
    background-color: # CCCCCC;
    border:1px dashed # 000000;
    padding:10px;
    position:absolute; /* absolute 绝对定位 */
    left:35px;
    top:35px;
}
# son2{
    background-color: # CCCCCC;
    border:1px dashed # 000000;
    padding:10px;
    position:absolute; /* absolute 绝对定位 */
    left:15px;
    top:15px;
}
</style>
<div id="parent1">
    <div id="son1">Box-2</div>
</div>
<div id="parent2">
    <div id="son2">Box-4</div>
</div>
```

运行结果如图 4-18 所示,通过浏览器效果可以看出,Box-2 偏移基准线是 parent1 这个父元素,因为在父元素中使用了定位,而 Box-4 偏移基准线是浏览器窗口,因为在其父元素 parent2 中没有使用定位。

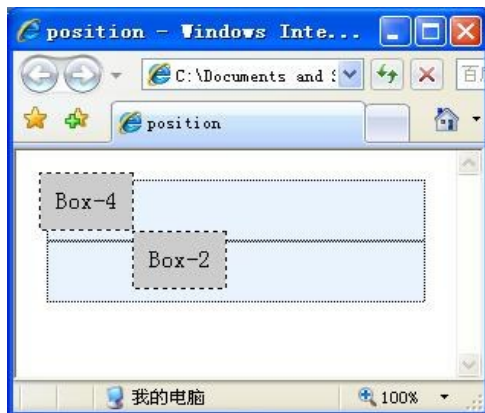


图 4-18 绝对定位



效果如图 4-19 所示,可以看出,Box-1、Box-2 加入了 float:left 属性,两个盒子宽度都自适应文字宽度,而且都是向左侧对齐;而 box-3 没有这个属性,宽度还是向右侧自动延伸。由于 p 元素也是没有 float 属性也会另换行自动延伸宽度。这里值得注意的是 float 属性在火狐浏览器按照上面书写的样式达不到这个效果,所以要注意浏览器特点。

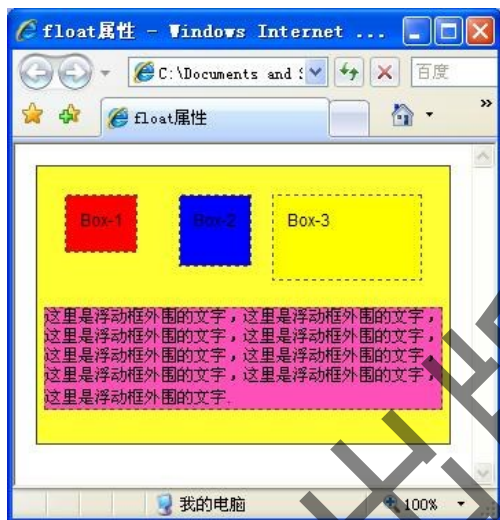


图 4-19 float 效果

### 4.3.8 盒子的 display 属性

盒子有两种类型,分别是块级元素、行内元素,对应的标记名称分别为<div>和<span>,块级元素的宽度、高度属性起作用,而行内元素宽度、高度不起作用。使用 display 属性可以把块级元素和行内元素互相转换。常用的 display 属性值有:

none 代表此盒子消失,block 代表盒子为块级元素,inline 代表盒子为行内元素。

块级元素宽度会自动延伸到父级元素,而行内元素宽度自适应文字宽度。

例如:

```
<div style="display:inline; width:200px; border:1px solid red;">Box-1</div>
<div style="display:inline">Box-2</div>
<div style="display:inline">Box-3</div>
<span style="display:block;">Box-4</span>
<span style="display:block; height:30px; width:40px; border:1px solid red;">Box-5</span>
<span style="display:block">Box-6</span>
<div style="display:none">Box-7</div>
<span style="display:none">Box-8</span>
```

效果如图 4-20 所示,可以看出 span 行内元素可以使用 display 属性转为块级元素,转为块级元素后就具有了块级元素的特点;而把 div 转为了行内元素之后,虽然设置宽度和高度但是不起作用。



图 4-20 display 属性

### 4.3.9 CSS 背景样式

背景图像在网站样式设计中广泛使用,属性及赋值格式为:background-image:url(图片地址)。图像显示默认效果是水平和垂直都重复显示,也可以根据以下四种情况显示出不同效果的背景:

repeat:水平和垂直两个方向都重复。

no-repeat:只显示一次图片。

repeat-x:只水平方向重复显示图像。

repeat-y:只垂直方向重复显示图像。

这里值得注意的是,有的时候背景颜色和背景图像同时出现,可以设计一个渐进色变换的显示效果,例如某个新闻内容显示很长,网页上半部分是蓝色,下半部分是绿色的这样的渐进效果。在设计时候,就可以制作一张图片是渐进色的,由蓝过度到绿。代码格式简写为background:green url(图片地址)。

## 4.4 项目实施

整个项目网页界面很多,每个页面设计实施不在这里逐个讲解,只以其中一个页面制作为例,其余页面制作代码请参考提供的网站。

登录页面的制作过程如下:

(1)启动 Dreamweaver cs3,首先新建站点,在 c 盘新建一个文件夹 ejob360,这个就是站点根目录,然后在 ejob360 文件夹中建立一个 images 文件夹,用来存放网站使用的图片文件。

(2)把从登录页面效果图中裁切出的图片存储在 images 文件夹中。

(3)在站点中新建 login.html 和 style.css 两个文件。

(4)首先在样式表 style.css 中初始化 body、ul、li、h3、h2、h1 等标记。

```
body {  
margin-left: 0px;  
margin-top: 0px;  
margin-right: 0px;
```

```
margin-bottom: 0px;
background-image: url(images/index_08.gif);
background-repeat: repeat;
}
ul,li,h3,h2,h1{ margin:0px;}
```

(5)在网页中加入一个最外层 div,它包含了其他所有网页元素,把这个 div 设置为水平方向居中,如果这个 div 居中,那么它里面所有内容都居中了。

```
.box{ width:1000px; margin:0px auto;}
```

(6)网页中引入样式表,添加 div 标记。

```
<html>
<head>
  <title>ejob 登陆窗口</title>
  <link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
<div class="box">
</div>
</body>
</html>
```

(7)然后把网页分成上下两行,第一行中包含左右两部分,左侧是网站 logo,右侧是三类用户登录。这里注意的问题是水平居中样式 text-align:center,设置 line-height 属性值与 height 属性值相同,即垂直居中。

```
.header{ background:url(images/index_02.gif); height:55px;} /* 第一行头部 */
.header .headerLeft{ height:55px; width:203px; float: left; margin-left:100px;} /* 第一行头部
左侧列 */
.header .headerRight{ height:55px; width:200px; float: right; margin-left:100px; line-height:
55px;} /* 第一行头部 右侧列 */
.wenzi { font-family: "宋体"; font-size: 12px; font-style: normal; line-height: 22px; color: #
000000;text-decoration: none;}/* 超链接文字效果 */
```

Html 代码:

```
<div class="header">
  <div class="headerLeft">
    
  </div>
  <div class="headerRight">
    <a href="#" class="wenzi">企业会员</a> |
    <a href="#" class="wenzi">个人会员</a> |
    <a href="#" class="wenzi">游客登录</a>
  </div>
</div>
```

效果如图 4-21 所示。



图 4-21 header 运行效果

(8)第二行里,主要是针对右侧的登录框。这里值得注意的是,在设置登录框出现在第二行右侧时,采用相对定位。

```
.content{ background:url(images/index_06-2.gif) # fffffff no-repeat; width:1000px; height:
535px;}/* 第二行头部 右侧列 */
.content .login{ background:url(images/index_06-3-2.gif); width:375px; height:243px; position:
relative; left:570px; top:200px;}/* 登录框 */
.login ul{ list-style:none;}
.login .con{ height:29px;}
.con li{ float:left; width:120px; text-align:center; height:29px; line-height:29px;}/* 登录框 标
题头样式 */
.login .con1{ margin-top:50px; height:200px; margin-left:50px;}
.con1 li{ line-height:38px; height:38px; font-size:14px;}/* 登录框 文本框 按钮行距 */
```

Html 代码:

```
<div class="content">
  <div class="login">
    <ul class="con">
      <li><a href="#" class="wenzi">个人会员登录</a></li>
      <li><a href="#" class="wenzi">企业会员登录</a></li>
    </ul>
    <ul class="con1">
      <li>账 &nbsp;号<input name="textfield" type="text" id="textfield" size="
18" maxlength="20" /></li>
      <li>密 &nbsp;码<input name="textfield2" type="password" id="textfield2"
size="18" maxlength="20" /><a href="#" class="wenzi">忘记密码? </a></li>
      <li>验证码<input name="textfield3" type="text" id="textfield3" size="8"
maxlength="4" />  <a href="#"
class="wenzi">看不清</a></li>
      <li></li>
    </ul>
  </div>
</div>
```

效果如图 4-22 所示。



图 4-22 登录框

## 4.5 项目拓展

佳木斯龙采网是一家从事信息产业的公司,公司急需一套网站做宣传,在本书提供资料里面提供了 psd 格式的图片文件,同学们可以自己切图实现网站转化为 HTML 网页。随书光盘中提供了已经切好的图片和网页供读者参考,网站效果图如图 4-23 所示。



图 4-23 佳木斯龙采网首页

## 4.6 本章小结

本章通过实际项目介绍了网站前台布局思路,其中在技术要点中详细介绍了样式表的使用,希望读者在网站前台布局设计中能够充分理解和掌握。