

# ②

## 模块二

# CSS 语言

### 内容简介：

CSS (Cascading Style Sheet, 层叠样式表/级联样式表) 是一组格式设置规则, 用于控制 Web 页面的外观。通过使用 CSS 样式设置页面的格式, 可将页面的内容与表现形式分离。页面内容存放在 HTML 文档中, 而用于定义表现形式的 CSS 规则则存放在另一个文件或 HTML 文档的某一部分中, 通常为文件头部分。将内容与表现形式分离, 不仅可使维护站点的外观更加容易, 还可以使 HTML 文档代码更加简练, 缩短浏览器的加载时间。

## 任务一 认识 CSS



### 案例分解

- 我的第一个 CSS 文件。



### 案例资讯

- CSS 的概念。
- CSS 的使用。

网页中 CSS 是设置外观的，那么我们首先应该知道 CSS 是什么？在这个任务中，我们就来学习一下 CSS 的特点、使用方法、类型、写法及运用 CSS 时的注意事项。

### 1.1 案例一 我的第一个 CSS 文件

本案例主要学习 CSS 文件及怎样运用 CSS 文件。

#### 1.1.1 案例资讯

在学习怎样制作 CSS 文件之前，我们先来学习 CSS 的一些相关概念及使用方法。

##### 1. CSS 的概念

###### (1) CSS 的基本概念。

CSS (Cascading Style Sheet, 层叠样式表单) 是一系列格式规则，控制网页内容的外观。使用 CSS 样式可以非常灵活并更好地控制确切的网页外观，从精确的布局定位到特定的字体和样式。

1998 年 5 月 12 日，W3C 组织推出了 CSS2，使得这项技术在世界范围内得到广泛的支持。CSS2 成为了 W3C 的新标准。同时，W3C CoreStyle、CSS2 Validation Service 以及 CSS Test Suite 宣布成立。它是一组样式，样式中的属性在 HTML 元素中依次出现并显示在浏览器中。样式可以定义在 HTML 文件的标记 (Tag) 里，也可以在外部附件文件中作为外加文件。此时，一个样式表可以用于多个页面，甚至整个站点，因此具有更好的易用性和扩展性。

总的来说，CSS 可以完成下列工作：

- ① 弥补 HTML 对网页格式化功能的不足，如段落间距、行距等。
- ② 设置字体变化和大小。
- ③ 设置页面格式的动态更新。
- ④ 进行排版定位。

## (2) CSS 的特点。

- ①将格式和结构分离，减少工作量。
- ②以前所未有的能力控制页面布局。
- ③制作体积更小、下载更快的页面。
- ④将许多页面同时更新，比以前更快、更容易。
- ⑤浏览器将成为更友好的界面，是对 HTML 语言处理样式的最好补充。
- ⑥控制页面中的每一个元素（精确定位）。

## (3) 使用 CSS 的好处。

### 1) 样式解决了一个普遍的问题。

HTML 标签原本被设计为用于定义文档内容。通过使用 `<h1>`、`<p>`、`<table>` 标签，HTML 的初衷是表达“这是标题”、“这是段落”、“这是表格”的信息。同时文档布局由浏览器来完成，而不使用任何的格式化标签。

由于两种主要的浏览器（Netscape 和 Internet Explorer）不断地将新的 HTML 标签和属性（如字体标签和颜色属性）添加到 HTML 规范中，那么所创建文档的内容能清晰地独立于文档表现层的站点将变得越来越困难。

为了解决这个问题，万维网联盟（W3C）这个非营利的标准化联盟肩负起了 HTML 标准化的使命，并在 HTML 4.0 之外创造出样式（Style），所有的主流浏览器均支持层叠样式表。

### 2) 多重样式将层叠为一个。

样式表允许以多种方式规定样式信息。样式可以规定在单个的 HTML 元素中，在 HTML 页的头元素中或在一个外部的 CSS 文件中。甚至可以在同一个 HTML 文档内部引用多个外部样式表。

当同一个 HTML 元素被不只一个样式定义时，会使用哪个样式呢？

一般而言，所有的样式会根据下面的规则层叠于一个新的虚拟样式表中：

- ①浏览器默认设置。
- ②外部样式表。
- ③内部样式表（位于 `<head>` 标签内部）。
- ④内联样式（在 HTML 元素内部，也叫行内样式）。

因此，内联样式（在 HTML 元素内部）拥有最高的优先权，这意味着它将优先于 `<head>` 标签中的样式声明、外部样式表中的样式声明、浏览器中的样式声明（默认值）。

### 3) 样式表极大地提高了工作效率。

样式表定义如何显示 HTML 元素，与 HTML 3.2 的字体标签和颜色属性所起的作用一样。样式通常保存在外部的 .css 文件中。仅仅通过编辑一个简单的 CSS 文档，外部样式表就使你能力同时改变站点中所有页面的布局和外观。

由于允许同时控制多重页面的样式和布局，CSS 可以称得上 Web 设计领域的一个突破。作为网站开发者，你能够为每个 HTML 元素定义样式，并将其应用于你希望的任意多的页面中。如要进行全局的更新，只需简单地改变样式，网站中的所有元素均会自动地更新。

## 2. CSS 的使用

### (1) CSS 的类型。

CSS 层叠样式表包含以下 3 种类型：

①自定义 CSS，相应的标记中出现 class 属性。自定义样式是生成一个新的样式，制作完毕后就可以在样式面板中看到制作完成的样子。在应用时，首页面中选中对象，然后选择样式。

代码如下：

```
.bg {  
background-image: url (bg.gif);  
}
```

在 HTML 中使用<body class="bg">。

**注意：**bg 是编者自行定义的，前面一定加“.”。

②重定义标记的 CSS（将现有的标签赋上样式，应用时无须选中而直接用）选择器。

代码如下：

```
td {  
color: #000099;  
font-size:9pt  
}
```

### ③CSS 选择符。

CSS 选择符为特殊的组合标记定义 CSS，使用 ID 作为属性，以保证文档具有唯一可用的值。CSS 选择符是一种特殊类型的样式，常用的有 4 种，分别为：a:link、a:active、a:visited、a:hover。其中：a:link 设定正常状态下链接文字的样式；a:active 设定单击鼠标时链接的外观；a:visited 设定访问过的链接外观；a:hover 设定鼠标放置在链接文字之上时文字的外观。

代码如下：

```
a:link {  
color:#FF3366;  
font-family:"宋体";  
text-decoration:none;  
}  
A:hover {  
color:#FF6600;  
font-family:"宋体";  
text-decoration:underline;  
}  
A:visited {  
color:#FF339900;  
font-family:"宋体";  
text-decoration:none;  
}
```

### (2) CSS 的基本写法。CSS 的基本写法有以下 3 种：

①在 head 内实现，即<head></head>标记内。

②在<body>内实现，使用语法<h3 style="font-size:10pt">，这样的写法虽然直观，但是无

法体现出 CSS 的优势，因此不推荐使用。

③在文件外的调用。

CSS 的定义既可以在 HTML 文档内部，也可以单独成立文件，以下代码是将 CSS 样式链接到外部的 style.css 文件（此文件是单独存在的）：

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

（3）CSS 的冲突。

CSS 是有优先级的，所谓“优先级”，就是指 CSS 样式在浏览器中被解析的先后顺序。既然有优先级，就会有一个规则来约定这个优先级。在 CSS 中的规则如下：

①统计选择符中的 ID 属性个数。

②统计选择符中的 CLASS 属性个数。

③统计选择符中的 HTML 标记名个数。

最后，按正确的顺序写出三个数字，不要加空格或逗号，得到一个三位数（css2.1 是用 4 位数表示）。相应于选择符的最终数字列表可以很容易确定较高数字特性凌驾于较低数字的。

**注意：**要把数字转换成一个以三个数字结尾的更大的数。

例如：

每个 ID 选择符（#someid）加 0,1,0,0。

每个 Class 选择符（.someclass）、属性选择符（形如[attr=value]等）、伪类（形如:hover 等）加 0,0,1,0。

每个元素或伪元素（:firstchild 等）加 0,0,0,1。

其他选择符（包括全局选择符\*）加 0,0,0,0。相当于没加，不过这也是一种 specificity，后面会解释。

表 2-1 按特性分类的选择符列表

选择符	特性值
h1 {color:blue;}	1
p em {color:purple;}	2
.apple {color:red;}	10
p.bright {color:yellow;}	11
p.bright em.dark {color:brown;}	22
#id316 {color:yellow}	100

表 2-2 选择符列表（详细）

选择符	特殊值
h1 {color:blue;}	1
p em {color:purple;}	1+1=2
.apple {color:red;}	10
p.bright {color:yellow;}	1+10=11
p.bright em.dark {color:brown;}	1+10+1+10=22
#id316 {color:yellow}	100

通过以上表格可以很简单地看出，HTML 标记的权重是 1，CLASS 的权重是 10，ID 的权重是 100，继承的权重为 0（后面会讲到）。

按这些规则将数字字符串逐位相加就得到最终的权重，然后在比较取舍时按照从左到右的顺序逐位比较。

在冲突解决时，还会了解 CSS 的继承性。

①继承的表现。继承是 CSS 的一个主要特征，它是依赖于祖先—后代的关系的。继承是一种机制，它允许样式不仅可以应用于某个特定的元素，还可以应用于它的后代。例如，一个 body 定义了的颜色值也会应用到段落的文本中。

②继承的局限性。继承是 CSS 重要的一部分，我们甚至不用去考虑它为什么能够这样，但 CSS 继承也是有限制的。有一些属性不能被继承，如 border、margin、padding、background 等。

### 1.1.2 案例步骤

#### 1. CSS 在 head 中实现

记事本里的代码如下：

```
<html>
<head>
  <title>CSS 在 head 中的实现</title>
  <style type="text/css">
    body {
      font-family: "黑体";
      font-size: 12pt;
      line-height: 16pt;
      color: #FFFFFF;
      background-color: #006699;
    }
  </style>
</head>
<body>
  主流的网页设计软件
  <p>目前，网页技术进入了一个新的阶段，现在的网页再也不是图片的堆积和枯燥无味的文本了，人们现在追求的是网页的动态效果和交互性。而 Macromedia 公司的网页设计三剑客软件 Dreamweaver、Flash、Fireworks 正是交互性网页设计的杰出代表，其最新版本 MX 继承了前期版本的优点，进行了功能的进一步整合，非常适合于网页设计和网站建设的需要。</p>
</body>
</html>
```

显示效果如图 2-1 所示。

#### 2. CSS 在 body 中实现

打开记事本后，可以在里面输入以下代码：

```
<html>
<head>
  <title>CSS 在 body 中的实现</title>
</head>
```

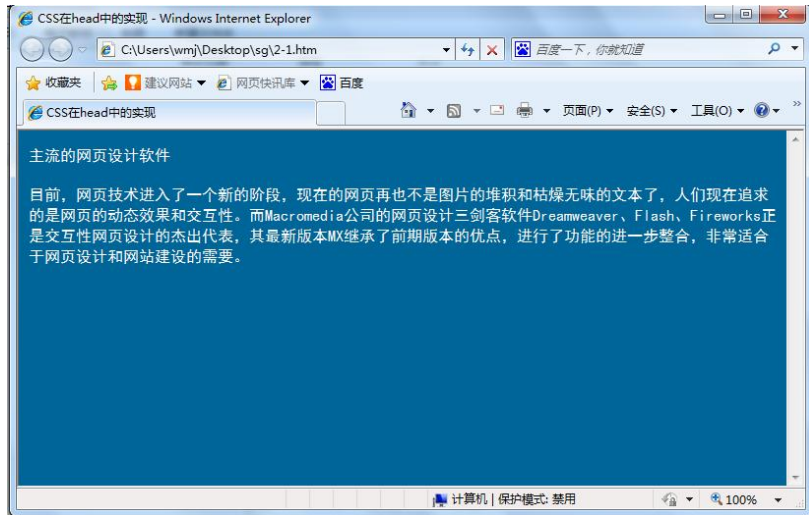


图 2-1 CSS 在 head 中实现效果

```
<body>
```

```
    主流的网页设计软件
```

```
    <p style="font-size:9pt;
```

```
line-height:12pt;
```

```
background-color:#FFCC00;
```

```
font-family:宋体">目前，网页技术进入了一个新的阶段，现在的网页再也不是图片的堆积和枯燥无味的文本了，人们现在追求的是网页的动态效果和交互性。而 Macromedia 公司的网页设计三剑客软件 Dreamweaver、Flash、Fireworks 正是交互性网页设计的杰出代表，其最新版本 MX 继承了前期版本的优点，进行了功能的进一步整合，非常适合于网页设计和网站建设的需要。
```

```
</p>
```

```
</body>
```

```
</html>
```

显示效果如图 2-2 所示。

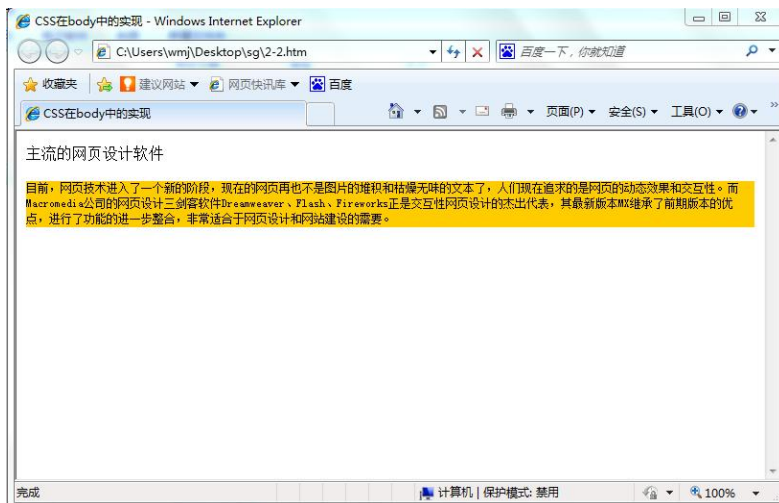


图 2-2 CSS 在 body 中实现效果

## 任务二 用 CSS 实现页面风格的统一



### 案例分解

- 用 CSS 美化字体。
- 用 CSS 实现页面的精细排版。
- 用 CSS 设置对象的边框。
- 用 CSS 设置页面的颜色和背景。



### 案例资讯

- 字体。
- 排版。
- 边框。
- 背景。

### 2.1 案例一 用 CSS 美化字体

我们在学习 HTML 时，已经接触过文字方面的设置、排版等。实际上，文字的详细设置是在 CSS 部分，这里我们就学习使这些字体更加美观的方法。

#### 2.1.1 案例资讯

在学习美化字体之前，我们先来学习 CSS 中的字体。

##### 1. 字体

字体就是网页里面需要设置的是什么样的字，如黑体、宋体等。

字体的属性如下：

**font-family:** 用一个指定的字体名或一个类的字体族科；

**font-size:** 字体显示的大小；

**font-style:** 设定字体风格；

**font-weight:** 以 **bold** 为值可以使字体加粗。

还有 **in**（英寸）、**cm**（厘米）、**mm**（毫米）、**pt**（点数）、**pc**（打字机字间距）、**em**（ems）、**ex**（x-height）、**px**（像素）。

##### 2. 字号

字号就是字体的大小。单位可以是厘米、像素、磅等，另外还有其它一些值，如 **xx-small**、**x-small**、**smaller**、**x-large**、**xx-large** 等。最常用的单位为 **pt**。



### 3. 字体样式

字体样式也就是字体的风格，属性如下：

**normal**：普通的文字；

**italic**：斜体的文字；

**oblique**：倾斜的文字，在中文文字的使用上与 **italic** 并无明显区别。

### 4. 字体重量

字体的重量即字体的粗细，字体的加粗属性值在 100~900 之间。其他的属性如下：

**normal**：普通的文字；

**bold**：加粗；

**bolder**：特粗；

**lighter**：加细。

### 5. 字体大小写

字体的大小写一般是指字符。

**text-transform**：文本转换属性，其值有 **none**、**capitalize**、**uppercase**、**lowercase** 和 **inherit**

**none**：默认值，不做任何大小写转换；

**capitalize**：将每个单词的首字变成大写；

**uppercase**：将每个字符变成大写；

**lowercase**：将每个字符变成小写。

## 2.1.2 案例步骤

### 美化字体

打开记事本后，可以在里面输入以下代码：

```
<html>
<head>
  <title>CSS 美化字体</title>
  <style type="text/css">
    H1{                                     //h1 规定字体为隶书
      font-family: "华文新魏";
      font-size:20pt;
      font-style:italic;
      font-weight:900;
      text-transform:capitalize;
    }
    .text {                                 //规定字体为宋体或仿宋体，即当客户机没
有宋体字体时，浏览器会使用仿宋体字体来显示
      font-family: "黑体, 仿宋_gb2312";
      font-size:15pt;
      font-style:oblique;
      font-weight:bolder;
      text-indent:50px;
    }
  </style>
```

```
</head>
```

```
<body>
```

```
  <H1>主流的网页设计软件</H1>
```

```
  <p class="text">目前，网页技术进入了一个新的阶段，现在的网页再也不是图片的堆积和枯燥无味的文本了，人们现在追求的是网页的动态效果和交互性。而 Macromedia 公司的网页设计三剑客软件 Dreamweaver、Flash、Fireworks 正是交互性网页设计的杰出代表，其最新版本 MX 继承了前期版本的优点，进行了功能的进一步整合，非常适合于网页设计和网站建设的需要。</P>
```

```
</body>
```

```
</html>
```

显示效果如图 2-3 所示。

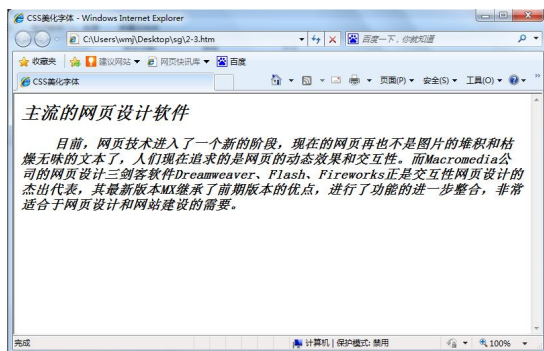


图 2-3 美化字体效果

## 2.2 案例二 用 CSS 实现页面的精细排版

在前面的案例中，我们已经学会了怎样运用 CSS 中的字体。在这个案例中，我们来用这些字体进行精细的排版。

### 2.2.1 案例资讯

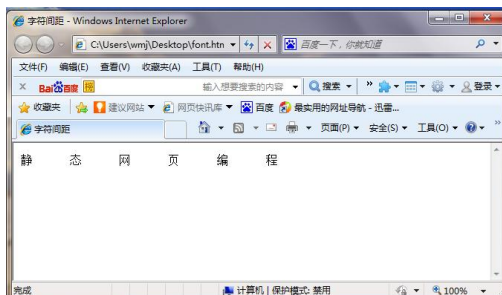
在学习排版之前，我们先来学习一下在 CSS 中都会用到哪些知识点。

#### 1. 字符间距

字符间距就是汉字与汉字之间的距离，改变段落中字符之间的间距用参数 letter-spacing。

例如：<p style="letter-spacing:50px "> 静态网页编程 </p>

显示结果如下：



50px 表示字符的间距大小。

## 2. 单词间距

单词间距是段落中字符或英文单词之间的间距，改变段落中字符之间的间距用参数 `word-spacing`。

## 3. 文字修饰

文字修饰就是在文字上加下划线或中间加条线。其基本语法如下：

```
p{ text-decoration: underline; } (P 是选择符)
```

`underline` 给文字加下划线；`overline` 给文字加上划线；`line-through` 给文字加删除线；`none` 使得上述效果都不会发生，常用于去掉超链接的下划线。

## 4. 文字对齐

文字对齐是指文本中的文字按照什么样的方式来存放。文字的对齐用 `text-align` 属性。

语法：<p align:center> 要输入的文字 </p>

`center` 可以换成 `left` 或 `right`。

## 5. 文字首行缩进

文字首行缩进类似于每段文字空两个字的位置，用 `text-indent` 属性，允许使用负值。如果使用负值，那么首行会被缩进到左边，默认值是 `not specified`。

## 6. 行高

行高是指使用 CSS 属性单词 `height`，单位为像素，参数如下：

`max-height`: CSS 最大高度；

`min-height`: CSS 最小高度；

`line-height`: CSS 上下居中。

表格里的高度可以用 `height`，文本里的用 `line-height`，自适应高度。

一般我们需要在宽度一定时，让高度随内容的增加而增高，此时我们将无须设置高度即可实现此效果，同时也无须使用 `height:auto` 来实现高度自适应。通常在默认情况下不设置高度，对象高度即自适应高度。

### 2.2.2 案例步骤

用 CSS 实现页面的精细排版

打开记事本后，可以在里面输入以下代码：

```
<html>
<head>
  <title>精细排版</title>
  <style type="text/css">
    h1{
      font-family: "华文新魏";           //h1 规定字体为隶书
      text-align: right;                  //文字对齐方式
      text-decoration: overline;         //文字修饰，与上面那个只能识别一个
    }
    .text {
      //规定字体为宋体或仿宋体，即当客户机没有宋
      font-family: "黑体,仿宋_gb2312";
    }
  </style>
</head>
<body>
  <h1>精细排版</h1>
  <div class="text">
    精细排版
  </div>
</body>
</html>
```

```
text-indent:50px; //首行缩进
line-height:5;    //行高
    }
</style>
</head>
<body>
  <H1>主流的网页设计软件</H1>
  <p class="text" >目前，网页技术进入了一个新的阶段，现在的网页再也不是图片的堆积和
枯燥无味的文本了，人们现在追求的是网页的动态效果和交互性。而 Macromedia 公司的网页设计三剑客软件
Dreamweaver、Flash、Fireworks 正是交互性网页设计的杰出代表，其最新版本 MX 继承了前期版本
的优点，进行了功能的进一步整合，非常适合于网页设计和网站建设的需要。</P>
</body>
</html>
```

显示效果如图 2-4 所示。

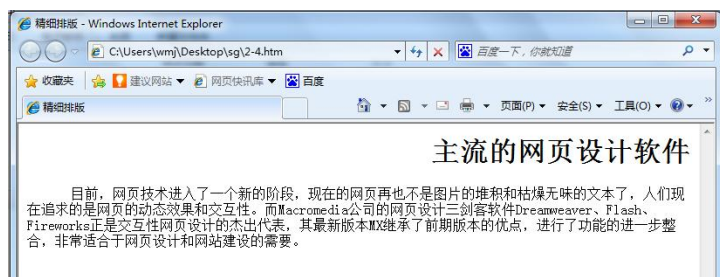


图 2-4 用 CSS 实现精细排版的效果

## 2.3 案例三 用 CSS 设置对象的边框

在前面的案例中，我们已经学会了运用 CSS 创建精细排版的网页，那么怎么设置对象的边框呢？现在我们就来学习。

### 2.3.1 案例资讯

在学习边框之前，我们先来学习一下在 CSS 中的边框都包含哪些内容。

#### 1. 边框宽度

边框宽度就是所设置的边框的粗细，边框宽度的属性为 `border-width`，其值有 `Medium`（默认值）、`thin`（比 `medium` 细）、`thick`（比 `medium` 粗）。用长度单位定值，可以用绝对长度单位（`cm`、`mm`、`in`、`pt`、`pc`）或者相对长度单位（`em`、`ex`、`px`）。

#### 2. 边框颜色

`border-color` 属性用来定义所有边框颜色，或者为四个边分别设置颜色。它可以取颜色的值或被设置为透明（`transparent`）。

注意：在 `border-color` 前最好先设置 `border-style`，否则 `border-color` 可能会不显示。

#### 3. 边框样式

`border-style` 属性用来设置元素所有边框的样式，或者单独为各边设置边框样式。它有 10

个属性值，分别如下：

- none: 没有边框，无论边框宽度设为多大；
- hidden: 同样是无样式，主要用于解决与表格的边框冲突；
- dotted: 点线式边框；
- dashed: 破折线式边框，即虚线；
- solid: 直线式边框；
- double: 双线式边框，两条线加上中间的空白等于 border-width 的取值；
- groove: 槽线式边框；
- ridge: 脊线式边框，与 groove 相反；
- inset: 内嵌效果的边框；
- outset: 突起效果的边框，与 inset 相反。

其中 groove、ridge、inset、outset 有些像 3D 效果，其效果受 border-color 的影响。

border-style 作用在四个方向时所用的方法与前面曾讲过的 padding 属性的书写方法相同，如果设置四个参数值，将按照上一右一下一左的顺序定义边框；如果只设置一个，将用于四个边框统一设置；如果设置两个值，第一个作用于上下，第二个则作用于左右；如果设置三个值，第一个作用于上边框，第二个作用于左右边框，第三个作用于下边框。

#### 4. 单边边框

设定上边框属性 border-top，它的设置格式与 border 相同，依次设置宽度、样式、颜色，border-top: border-width、border-style、border-color；

border-top 是将宽度、样式、颜色三种属性值放在一起而设置的属性，如果要单独设置其中的任意一项也可以使用以下属性：border-top-width（单独设置上边框宽度）、border-top-style（单独设置上边框样式）、border-top-color 属性（单独设置上边框颜色）。

```
{border-top-width: 1px;  
border-top-style: dashed;  
border-top-color: #FF0000;}
```

与 {border-top: 1px dashed #FF0000;} 效果是相同的。

设定下边框属性：border-bottom、border-bottom-width、border-bottom-style、border-bottom-color，设置方法同 border-top。

设定右边框属性：border-right、border-right-width、border-right-style、border-right-color，设置方法同 border-top。

设定左边框属性：border-left、border-left-width、border-left-style、border-left-color，设置方法同 border-top。

### 2.3.2 案例步骤

#### 我的对象边框

打开记事本后，可以在里面输入以下代码：

```
<html>  
<head>  
  <title>我的表格边框</title>
```

```
<style type="text/css">
  .tableborder{
    border-right-width:3px;
    border-right-color:red;
    border-right-style:dashed;
    border-bottom-width:4px;
    border-bottom-color:green;
    border-bottom-style:solid;
    padding-top:20px;
    padding-left:10px;
  }
</style>
</head>
<body>
<table border="0">
  <tr>
    <td class="tableborder">手机充值</td>
    <td class="tableborder">电子彩票</td>
  </tr>
  <tr>
    <td class="tableborder">电脑硬件</td>
    <td class="tableborder">数码相机</td>
  </tr>
</table>
</body>
</html>
```

显示效果如图 2-5 所示。

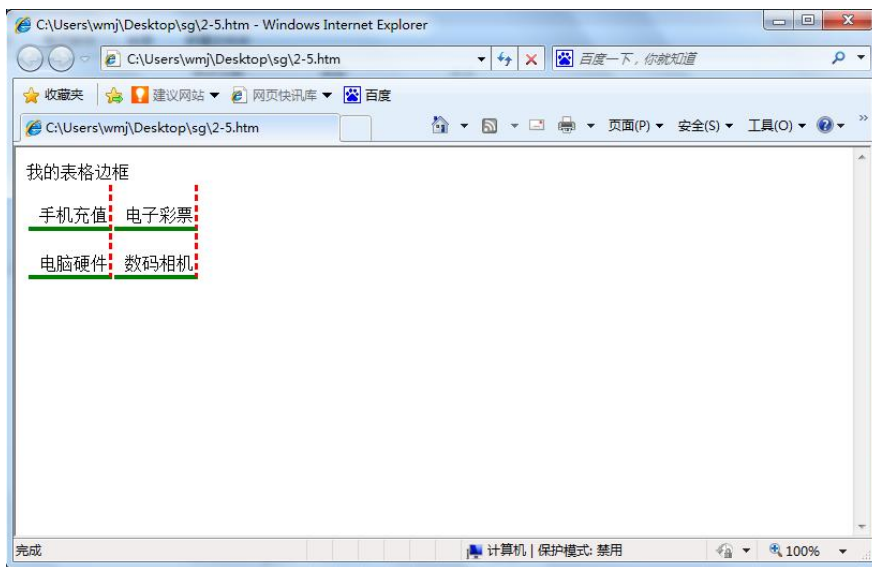


图 2-5 对象边框效果

## 2.4 案例四 用 CSS 进行页面的颜色和背景设置

在前面的案例中，我们已经学会了运用 CSS 进行对象边框的设置，那么怎么设置页面的颜色和背景呢？现在我们就来学习。

### 2.4.1 案例资讯

在学习 CSS 页面背景的设置之前，我们先来学习一下在 CSS 中有关背景方面的知识。

#### 1. 颜色

CSS 中的颜色表示法很多。

(1) CSS 预定义颜色表示法（就是使用表示颜色的英文单词预定义颜色表示法）。

如 `color:red`; `color:blue`;

(2) RGB 颜色表示法。

如 `color:rgb(255,0,0)`;`color:rgb(0,255,0)`;

RGB 颜色表示法就是红（R: Red）、绿（G: Green）、蓝（B: Blue）这三原色混合后呈现出的颜色，其中每种颜色的取值为 0~255。

(3) 十六进制颜色表示法

如 `color:#ff0000`; `color:#00ff00`; `color:#1199ff`

十六进制颜色表示法就是使用三对十六进制数分别表示 RGB 中的三原色。如上面例子的最后一个 `color:#1199ff`，其中 11 代表 R 的颜色（十六进制的 11 就等于十进制中的 17），99 代表 G 的颜色（十六进制的 99 就等于十进制中的 153），ff 代表 B 的颜色（十六进制的 ff 就等于十进制中的 255），前面再加一个 # 号（#1199ff 等价于 `rgb(17,153,255)`）。

(4) 短十六进制颜色表示法（属于网络安全色）。

如 `color:#f00`; `color:#0f0`; `color:#00f`

短十六进制颜色表示法就是当十六进制颜色表示法中的两个表示颜色值的数字一样时的简写，比如 `color:#ff0000`；就可以简写为 `color:#f00`。

(5) RGBA 颜色表示法。

如 `color:rgba(255,0,0,1)`; `color:rgba(0,255,0,1)`; `color:rgba(0,0,255,1)`;

RGBA 颜色表示法就是在 RGB 颜色的基础上增加了 Alpha 通道。

(6) HSL 颜色表示法。

如 `color:hsl(120,100%,75%)`; `color:hsl(360,100%,75%)`; `color:hsl(240,100%,75%)`;

HSL 颜色表示法就是使用色相（hue）、饱和度（saturation），亮度（lightness）表示颜色的一种方法。

(7) HSLA 颜色表示法。

如 `color:hsla(120,100%,75%,1)`; `color:hsla(360,100%,75%,1)`; `color:hsla(240,100%,75%,1)`;

HSLA 颜色表示法就是在 HSL 颜色的基础上增加了 Alpha 通道。

#### 2. 背景颜色

背景颜色使用 `background-color` 属性，这个属性接受任何合法的颜色值。

`background-color` 不能继承，其默认值是 `transparent`（透明）。也就是说，如果一个元素没

有指定背景色，那么背景就是透明的，这样其祖先元素的背景才能可见。

### 3. 背景图片

把图片放入背景需要使用 `background-image` 属性，默认值是 `none`，表示背景上没有放置任何图片。如果需要设置一个背景图片，必须为这个属性设置一个 URL 值。

语法：`{background-image: url (*.jpg);}`

大多数背景都应用到 `body` 里面，但是也可以用到某一个段落或某一个行内元素。

### 4. 背景图片平铺

背景图片平铺使用 `background-repeat` 属性，与 `background-image` 属性连在一起使用，决定背景图片是否重复。如果只设置 `background-image` 属性而没有设置 `background-repeat` 属性，在默认状态下，图片既横向重复，又纵向重复。

`repeat-x`：背景图片横向平铺；

`repeat-y`：背景图片竖向平铺；

`no-repeat`：背景图片不平铺。

默认背景图片从一个元素的左上角开始。

### 5. 背景图片位置

背景图片位置使用 `background-position` 属性，与 `background-image` 属性连在一起使用，决定了背景图片的最初位置。

为 `background-position` 属性提供值有很多方法。①可以使用一些关键字，如 `top`、`bottom`、`left`、`right` 和 `center`。通常这些关键字会成对出现，不过也不总是这样；②还可以使用长度值，如 `100px` 或 `5cm`；③也可以使用百分数值。不同类型的值对于背景图像的放置稍有差异。

## 2.4.2 案例步骤

### CSS 关于背景的例子

打开记事本后，可以在里面输入以下代码：

```
<html>
<head>
  <title>背景</title>
</head>
<body style="background-color:#cccccc">
  <p>
    
    <h2>静夜思</h2>
    <h3>作者：李白</a></h3>
    <p style="color:#ff0000;font-size:18px;
      font-family:隶书;border-bottom-style:dashed">
      床前明月光，<br>
      疑似地上霜。<br>
      举头望明月，<br>
      低头思故乡。<br> </p>
  <p>注释：
    静夜思：宁静的夜晚所引起的乡思。疑：怀疑，以为。举：抬、仰。
```



```
</p>  
</body>  
</html>
```

显示效果如图 2-6 所示。

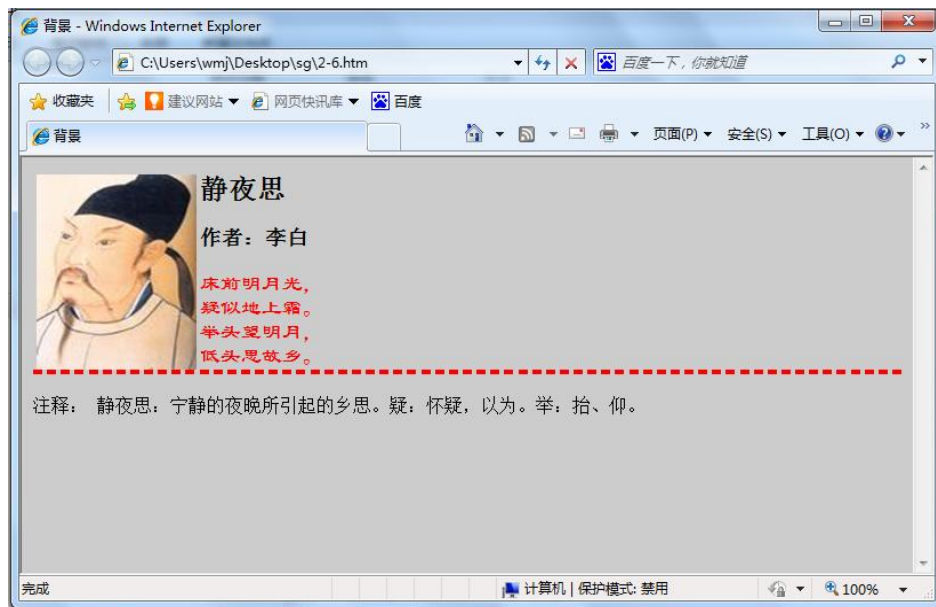


图 2-6 CSS 背景效果

## 任务三 学会 CSS 的高级应用技巧



### 案例分解

- 用 CSS 实现层的排版。
- 用 CSS 的滤镜实现各种特效。



### 案例资讯

- 定位。
- 区块。
- 特效。

通过前面的学习，我们已经可以设置页面基本的效果了，那么怎样才能把网页布置得更美观大方呢？在这个任务中，我们主要学习如何应用 CSS 高级设置，如定位和特效。

### 3.1 案例一 用 CSS 实现层的排版

本案例主要学习用 CSS 怎样实现层（div）的排版。我们在学习 Photoshop 时接触的层是会叠加在一起的，显示的效果更好。网页里面的层虽然可以叠加，但是显示的时候只能看到一个层的内容，即每个层是单独来运行的。

#### 3.1.1 案例资讯

在学习 CSS 实现层的排版之前，我们先来学习一下 CSS 排版的一些相关概念及使用方法。

##### 1. 定位

定位（position）允许用户精确定义元素框出现的相对位置，可以相对于它通常出现的位置、上级元素、另一个元素或浏览器视窗本身。每个显示元素都可以用定位的方法来描述，而其位置由此元素的包含块来决定。

常用定位属性如下：

（1）position 检索对象的定位方式。

语法：position : static | absolute | relative

static: 静态（默认）无特殊定位，对象遵循 HTML 定位规则。

absolute: 绝对，将对象从文档流中拖出，使用 width、height、left、right、top、bottom 等属性与 margin、padding、border 进行绝对定位。绝对定位的元素可以有边界，但这些边界不

压缩。其层叠通过 `z-index` 属性定义。

**relative:** 相对，对象不可层叠，但将依据 `left`、`right`、`top`、`bottom` 等属性在正常文档流中偏移位置。

**fixed:** 悬浮，使元素固定在屏幕的某个位置，其包含块是可视区域本身，因此它不随滚动条的滚动而滚动（IE 5.5+不支持此属性）。

**inherit:** 这个值从其上级元素继承得到。

(2) `z-index` 检索或设置对象的层叠顺序，默认为 0，向前为正，向后为负，步长为 1。

语法：`z-index:auto`

**auto:** 遵从父对象的定位。

(3) 在本文流中，任何一个元素都被文本流限制了自身的位置，但是通过 CSS，我们依然可以使这些元素改变自己的位置。我们可以通过 `float` 来让元素浮动，通过 `margin` 来让元素产生位置移动。但事实上那并非是真的位移，因为那只是通过加大 `margin` 值来实现的相对定位。而真正意义上的位移是通过 `top`、`right`、`bottom`、`left` 针对一个相对定位的元素所产生的。

## 2. 区块

包含块（**containing block**）是格式编排发生的关联场景。例如，一个加粗元素的包含块可以是该元素出现的段落，如图 2-7 所示。

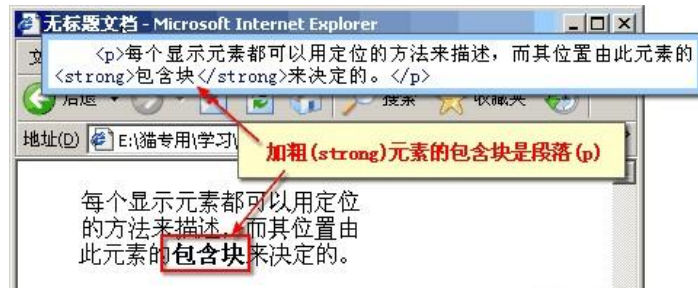


图 2-7 包含块效果

### (1) 区块基本信息。

建立包含块的规则如下：

1) 根元素的包含块（也叫初始包含块）由用户代理生成。在 HTML 中，根元素是 HTML 元素，尽管有的浏览器会不正确地使用 `body` 元素。

2) 对于那些未绝对定位的非根元素来说，元素的包含块设置为最近的块级祖先元素的内容区边沿。

3) 对那些使用绝对（`absolute`）作为定位（`position`）的非根元素，包含块设为最近的定位（`position`）不是静止（`static`）的祖先元素（任何类型），有两种情况：①如果祖先元素是块级（`block`）元素，包含块设为祖先元素的填充（`padding`）边沿，也就是被边框（`border`）约束的区域；②如果祖先元素是内联（`inline`）元素，包含块设为祖先元素的内容边沿。

因此，绝对定位的元素往往以浏览器可视区域的左上为坐标原点来进行定位。

### (2) 区块边缘信息。

**区块宽度：**区块宽度是整个包含边缘在内的宽度，而元素宽度指的只是内容元素的宽度。

区块性质指令如下：

**margin-top:** 设定上边缘宽度；

**margin-right:** 设定右边缘宽度；

**margin-bottom:** 设定下边缘宽度；

**margin-left:** 设定左边缘宽度。

**注意：**如果指定了四个合法设定值，则会依次套用于四个边框；如果只指定一个合法设定值，则会统一套用于四个边框；如果指定两个或三个合法设定值，则未指定的边框会套用对边的颜色设定值；如果没有指定此性质，则套用 color 性质的设定值。

### 3.1.2 案例步骤

定位

记事本中的代码如下：

```
<html>
<head>
  <style type="text/css">
    img.x
    {
      position:absolute;
      left:0px;
      top:0px;
      z-index:-1
    }
  </style>
</head>
<body>
  <h1> 定位</h1>
  
  <p>这个是简单的定位例子。</p>
</body>
</html>
```

显示效果如图 2-8 所示。

或者

```
<html>
<head>
  <style type="text/css">
    img
    {
      position:absolute;
      clip:rect(0px 50px 200px 0px)
    }
  </style>
</head>
```

```
<body>
  <p>这个是简单的定位。</p>
  <p></p>
</body>
</html>
```

显示效果如图 2-9 所示。

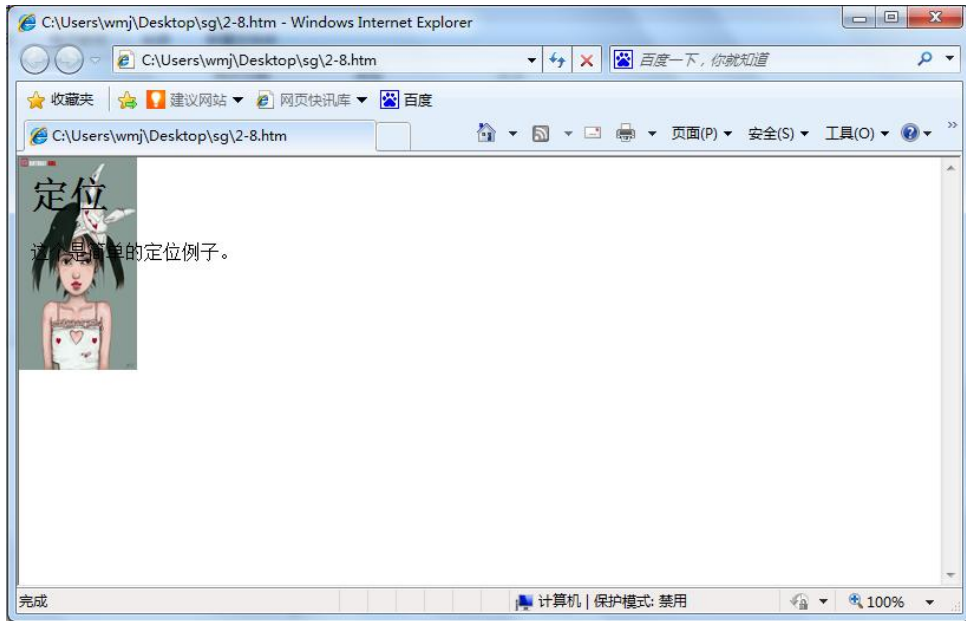


图 2-8 定位效果 1

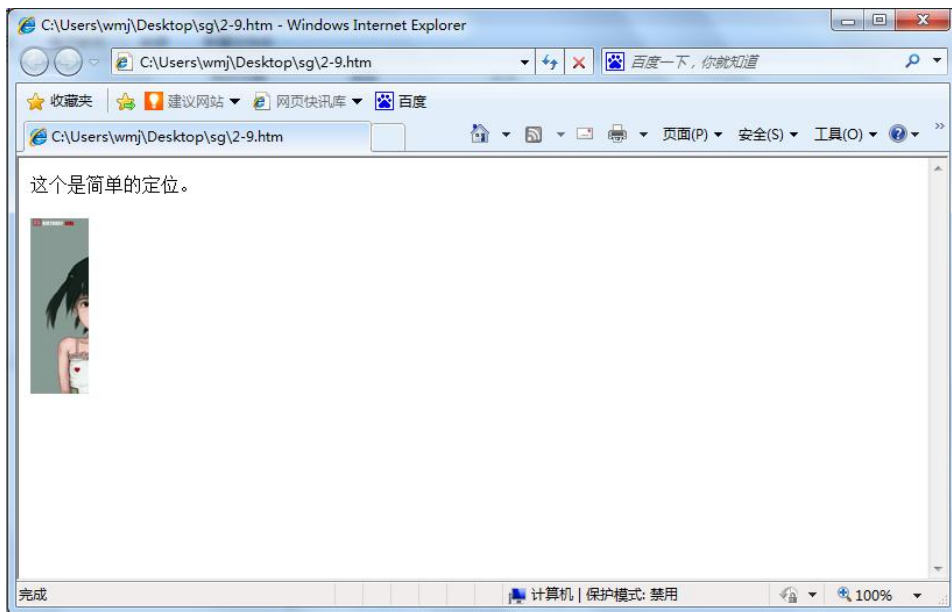


图 2-9 定位效果 2

## 3.2 案例二 用 CSS 的滤镜实现各种特效

在前面的学习中，我们已经会运用 CSS 的设置了，本案例主要学习 CSS 在页面中增加的各种特效。

### 3.2.1 案例资讯

有时我们在做网页时，需要一些特别的效果，如让某个图片或文字透明、翻转等，现在我们就来学习这些功能。

#### 1. 透明

**alpha** 滤镜的作用主要是对图片的透明度进行处理。

语法：

```
{ filter: alpha(
  opacity=value1,
  finishopacity=value2,
  style=value3,
  startx=startx,
  starty=starty,
  finishx=finish, f
  inishy=finish) }
```

或者

```

```

说明：

(1) **value1** 为图片的透明值，范围是 0（完全透明）~100（完全不透明）；

(2) **value2** 为图片透明度变换结束时的透明值，范围是 0（完全透明）~100（完全不透明）；

(3) **value3** 为图片透明度变换方向。取值为 1 时，图片透明度按从左到右线性变化；取值为 2 时，图片透明度从内到外沿半径变化；取值为 3 时，图片透明度从内到外呈矩形变化；

(4) **startx** 和 **starty** 代表渐变透明效果的开始 x 和 y 坐标；

(5) **finishx** 和 **finishy** 代表渐变透明效果的结束 x 和 y 坐标。

注意：**value2** 只有在 **value3** 设定时才有效。

#### 2. 模糊

**blur** 滤镜的作用主要是对图片的透明度进行处理

语法：

```
{
  filter: blur (
    add=add,
```

```
direction=direction,  
strength=strength ) }
```

说明:

(1) **add** 参数是一个布尔判断, 值为 **true** (默认) 或 **false**, 它指定图片是否被改变成印象派的模糊效果;

(2) 模糊效果是按顺时针方向进行的, **direction** 参数用来设置模糊的方向, 其中  $0^\circ$  表示垂直向上, 每  $45^\circ$  为一个单位, 默认值是向左的  $270^\circ$  ;

(3) **strength** 值只能使用整数来指定, 它代表有多少像素的宽度将受到模糊影响, 默认是 5 像素。

### 3. 水平翻转

**flipH** 滤镜的作用是产生水平翻转效果。

语法: { **filter:flipH** }

### 4. 垂直翻转

**flipV** 滤镜的作用是产生垂直翻转效果。

语法: { **filter:flipV** }

### 5. 灰度

**gray** 滤镜的作用是设置灰度。

语法: {**filter:gray, img=src="\*.jpg"**}

## 3.2.2 案例步骤

### CSS 特效

在这里我们只做了模糊和垂直翻转。

记事本中的代码如下:

```
<html>  
<head>  
    <title>CSS 中的特效</title>  
    <style type="text/css">  
        img{  
            filter: flipv;  
            blur (  
                strength=10);  
        }  
    </style>  
</head>  
<body>  
    <h2>强大的 CSS 滤镜</h2>  
    <img src=xuexiang.jpg>  
</body>  
</html>
```

显示效果如图 2-10 所示。

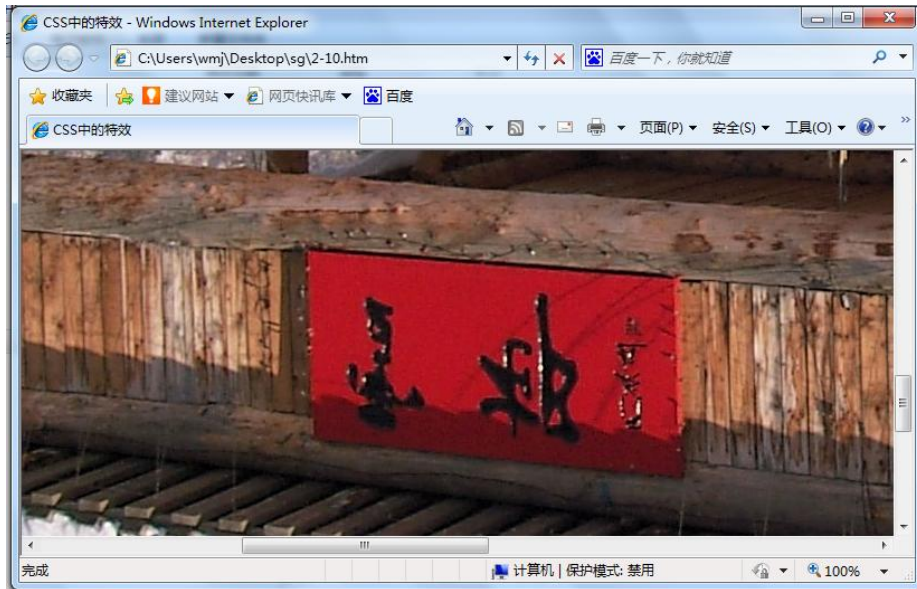


图 2-10 CSS 特效效果