

面向21世纪高等院校规划教材

CSS+DIV

网页设计开发 技术与实例应用

璐 袁 珏 主编

周芦燕 汪 晋 巴长忠 编著
耿龙龙 吕明远 李 奇 崔宝珍



国防工业出版社
National Defense Industry Press

面向 21 世纪高等院校规划教材 内容

CSS+DIV 网页设计开发 技术与实例应用

孔璐 袁珏 主编

周芦燕 汪晋 巴长忠 耿龙龙 编著
吕明远 李奇 崔宝珍

附赠 (CD) 光盘

本书可作为高等院校计算机专业及相关专业教材

国防工业出版社

地址：北京市海淀区中关村大街

开本：787mm×1092mm 1/16 印张：11 字数：260千字

2010年6月第1版 2010年6月第1次印刷

(责任编辑：王静)

• 北京 •

发行部：(010) 68432453 邮购部：(010) 6841474

发行部：(010) 68411232 邮购部：(010) 68432453

内 容 简 介

本书对 CSS+DIV 技术应用于各种网页元素的方法和过程进行了深入地剖析与介绍,系统地讲述了如何利用 CSS+DIV 技术控制文字、图片、表格、背景以及导航菜单等网页元素的方法,结合实际网页制作中可能遇到的问题,提供解决的思路、方法和技巧。书中重点讲解了几种典型的布局方式,从实际案例入手,让读者在学习 CSS+DIV 应用技术的同时掌握其精髓。各案例与实例按照由浅入深、由易到难的顺序编排,即使是初学者也可以轻松掌握 CSS+DIV 技术的布局方式,制作出精美实用的网页作品。

全书中所有的案例与实例都是编译通过的。本书所有源代码与相关素材均可以在国防工业出版社网站免费下载。本书适合从事网页设计、网站制作等相关工作的专业人员学习和使用,还可作相关专业的教材使用。

图书在版编目(CIP)数据

CSS+DIV 网页设计开发技术与实例应用/孔璐,袁珏

主编. —北京:国防工业出版社, 2010.6

ISBN 978-7-118-06919-8

I. ①C... II. ①孔... ②袁... III. ①主页制作-软件工具, CSS IV. ①TP393.092

中国版本图书馆 CIP 数据核字 (2010) 第 124333 号

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

天利华印刷装订有限公司印刷

新华书店经售

*

开本 787×1092 1/16 印张 11 字数 360 千字

2010 年 6 月第 1 版第 1 次印刷 印数 1—4000 册 定价 25.00 元

(本书如有印装错误, 我社负责调换)

国防书店: (010) 68428422

发行邮购: (010) 68414474

发行传真: (010) 68411535

发行业务: (010) 68472764

前言

CSS 是 Cascading Style Sheets (层叠样式表单) 的简称, 它可以使网页更加美观、简练和便于维护。CSS 技术已经被广泛应用于网页与网站开发, 使用任何浏览器浏览网页都要与 CSS 打交道; 无论使用什么工具制作网页, 都必须掌握 CSS 技术。CSS 是一门在网页与网站开发应用中必须要掌握和使用的重要技术。DIV 的全称为 Division 意为“区分”, 称为区隔标记, 它的作用是为了设定字、画和表格等 HTML 元素的摆放位置, 为 HTML 文档中的内容提供框架结构, 通过 CSS 样式配合来定义不同 DIV 区域中的各元素属性, 可以方便快捷地完成网页样式的定义。学会使用 CSS+DIV 网页布局, 是网页设计和制作人员必需掌握的技能之一。

本书以提高读者综合运用 CSS+DIV 技术进行网页与网站开发的应用能力为目的, 系统地讲解了 CSS 层叠样式表的基础理论和实际运用技术, 结合大量案例与实例对 CSS 进行深入浅出地分析, 详细地描述了各案例与实例的开发过程和方法, 对 CSS 的基本语法和概念有着更简明的阐述和说明, 并且通过 CSS 盒子模型详细地讲述了 CSS+DIV 布局的精髓所在。本书紧密结合经典开发案例, 系统讲述了如何利用 CSS 控制文字、图片、表格、背景以及导航菜单等网页元素方法, 并详细介绍了圆角矩形框、凹陷文字、滑动门和多级下拉菜单等常用技术。书中重点讲解了几种典型的布局方式, 分别对不同的 CSS+DIV 布局模式进行了细致地比较。此外, 书中还给出了 CSS+DIV 网站应用综合实例, 让读者进一步巩固学到的知识, 提高工程应用开发能力。本书还介绍了网页设计中常遇到的技术问题, 指导读者制作出符合 Web 标准的网页, 使从事或即将从事网站设计开发专业工作的读者提升专业技术水平和同行竞争能力。

本书由孔璐和袁珏作为主编, 本书的第五章是由中北大学机械工程与自动化学院崔宝珍编写。本书适合从事网页设计、网站制作、Web 程序开发和 Web 标准化等人员学习和使用, 也可作为高等院校相关专业的教学和参考用书以及各类社会培训机构的培训用书。

计算机技术的发展日新月异, 加之作者水平有限, 书中不足之处在所难免, 欢迎读者不吝指正 (Email: burning666@163.com)。

编者

2010 年 4 月

目 录

第一章 CSS 基础.....1	2.2 文本效果.....30
1.1 CSS 概述.....1	2.2.1 行距.....30
1.1.1 CSS 的定义.....1	2.2.2 文字与单词间距.....34
1.1.2 CSS 的基本语法.....2	2.2.3 首行缩进.....36
1.2 CSS 选择器.....3	2.2.4 水平对齐方式.....37
1.2.1 标签选择器.....3	2.2.5 垂直对齐方式.....39
1.2.2 选择器组.....4	2.2.6 首字下沉.....41
1.2.3 类选择器.....5	2.3 伪类.....42
1.2.4 ID 选择器.....6	2.4 文字实例.....44
1.2.5 相邻选择器.....7	第三章 背景效果.....47
1.2.6 后代选择器.....8	3.1 背景色的定义.....47
1.3 CSS 的继承.....9	3.2 自定义背景图.....48
1.4 CSS 优先级.....11	3.3 重复背景.....49
1.4.1 原则一.....11	3.4 背景位置.....54
1.4.2 原则二.....12	3.5 背景实例.....56
1.4.3 原则三.....14	第四章 边框设计.....58
1.4.4 原则四.....14	4.1 边框的定义.....58
1.5 CSS 链接.....15	4.1.1 边框样式.....58
1.5.1 内嵌式样式表.....15	4.1.2 边框颜色.....60
1.5.2 链接式样式表.....17	4.1.3 边框宽度.....61
1.5.3 输入样式表.....18	4.1.4 边框属性.....63
1.5.4 行内样式表.....18	4.1.5 独立边框.....64
第二章 文字效果.....21	4.2 表格边框.....65
2.1 文字的基本样式.....21	第五章 DIV 层.....69
2.1.1 字体样式.....21	5.1 DIV 概述.....69
2.1.2 字体颜色.....23	5.1.1 使用 DIV.....69
2.1.3 字体大小.....24	5.1.2 定义 DIV.....70
2.1.4 字体加粗.....26	5.1.3 块级标签.....71
2.1.5 字体倾斜.....28	5.1.4 行级标签.....72
2.1.6 字体下划线、顶划线、 删除线.....29	

5.2 盒子模型	73	8.5 Wave 滤镜	118
5.2.1 盒子模型概述	73	第九章 菜单导航制作	120
5.2.2 外补丁	80	9.1 横向导航菜单	120
5.2.3 内补丁	81	9.2 竖向导航菜单	122
5.2.4 浮动	82	9.3 变换颜色的导航菜单	125
5.3 网页布局模式	88	9.4 滑动门菜单	127
5.3.1 网页布局概述	88	9.5 CSS 圆角菜单	132
5.3.2 一列布局	89	9.6 下拉菜单	135
5.3.3 两列布局	93	第十章 实例应用	139
5.3.4 三列布局	98	10.1 按钮的制作	139
第六章 图片效果	101	10.2 CSS 分页效果	142
6.1 图片边框	101	10.3 CSS 相册滑动浏览效果	143
6.2 自适应大小	102	10.4 博客列表制作	145
6.3 阴影效果	105	10.5 图文混排 CSS 列表	149
第七章 溢出和剪切	108	第十一章 个人博客网站设计	155
7.1 overflow 的使用	108	11.1 整体布局构思	155
7.2 text-overflow 的使用	109	11.2 网站 TOP 制作	157
第八章 CSS 滤镜	113	11.3 网站 Content 制作	159
8.1 Alpha 滤镜	113	11.4 网站 Bottom 制作	163
8.2 Chroma 滤镜	114	附录 CSS 书写规范	165
8.3 Glow 滤镜	116	参考文献	168
8.4 Shadow 滤镜	117		

第一章 CSS 基础

在网页上使用 HTML 定位文字和图像比较复杂，必须用表格标签和隐式 GIF 图像，但即使这样也不能准确定位，因为不同的浏览器和操作平台的显示结果也不同。通过学习 CSS 的属性，可以精确地设定要素位置，并将定位的要素进行叠加。CSS 可以同时控制多重页面的样式和布局，能够为每个 HTML 元素定义样式，并将其应用于页面设计中。

1.1 CSS 概述

网页的主要组成可分为三个部分：结构（Structure）、表现（Presentation）和行为（Behavior），对应的标准也分三个方面：结构标准、表现标准和行为标准。结构标准的语言主要是指 XHTML 和 XML；表现标准的语言主要是指 CSS；行为标准的语言主要是指对象模型如 W3C DOM，ECMAScript 等。

1.1.1 CSS 的定义

网页设计最初是用 HTML 标记来定义页面文档及格式，例如标题<h1>、段落<p>、表格<table>和链接<a>等，但这些标记不能满足更多的文档样式需求。为了解决这个问题，在 1997 年 W3C（The World Wide Web Consortium）颁布 HTML4 标准的同时也发布了有关样式表的第一个标准 CSS1 之后，又在 1998 年 5 月发布了 CSS2 版本，样式表得到了更多的充实。W3C 把 DHTML（Dynamic HTML）分为三个部分来实现：脚本语言（包括 JavaScript 和 Vbscript 等）、支持动态效果的浏览器（包括 Internet Explorer 和 Netscape Navigator 等）和 CSS 样式表。

CSS（Cascading Style Sheet，可译为“层叠样式表”或“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 规则存放在另一个文件中或作为 HTML 文档的某一部分，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便捷，而且还可以使 HTML 文档代码更加简练，缩短浏览器的加载时间。

在 CSS 出现之前，虽然 HTML 为网页设计者提供了强大的格式设置功能，但必须为每个需要设置的地方使用格式设置标记，而不能为具有一定逻辑含义的内容设置统一的格式。这对设计和维护一个网页数量众多的网站来说，将增加许多工作量。此外，每个网页设计者按照自己的喜好设计和制作网页，不同设计者开发的网页作品很难统一在一个网络中。现在，运用 CSS 技术，可以克服 HTML 的这些缺陷，方便地为所有网页设置一种风格。如果将原来安排在网页文件中的格式化元素和属性提取到网页外部，将这些样式规则定义到一个样式表文件中，则可以为所有需要使用该样式的网页所链接。总之，

CSS 是一种格式化网页的标准方式，它对颜色、字体、间隔、定位和边距等方面提供了多种属性，这些属性均可用于 HTML 标记符。

CSS 层叠样式表不但可以做出外形优美的页面，还可以减少设计人员大量的重复工作，既省时又省力，对网站设计者来说是不可多得的好帮手。它的作用就是为了让网页更加美观，在网络发展迅速的今天，单单是 HTML 标记已经不够用了，通过 CSS 样式的学习，可以更加自由灵活地设计出需要的网页来。

1. CSS 样式表可以制作出体积更小、下载更快的网页
样式表只是简单的文本，它不需要图像、执行程序、插件和流式。以前必须求助于 GIF 的工作现在可以通过 CSS 来实现。使用样式表可以减少表格标签，加大 HTML 体积的代码并减少图像用量与文件尺寸。

2. 快速、便捷地维护及更新大量的网页
没有样式表时，如果想更新整个网站中所有主体文本的字体，必须逐页修改网页。即便站点用数据库提供服务，仍然需要更新所有的模板和每一模板中每一个实例的样式。样式表的主旨就是将格式和结构分离。利用样式表，可以将网站上所有的网页都指向单一的一个 CSS 文件，只要修改 CSS 文件中某一行，整个网站都会随之发生变动。

3. 浏览器的界面将更加友好
不像其他的网络技术，样式表的代码有很好的兼容性，也就是说，如果用户丢失了某个插件时不会发生中断，在使用老版本的浏览器时代码不会出现杂乱无章的情况。只要是可识别样式表的浏览器就可以应用 CSS 样式。

4. 内容和结构独立性强
CSS 语言定义了网页的结构和各要素的功能，从而让浏览器自己决定如何让各要素显示。样式表通过将定义结构的部分和定义格式的部分分离使设计者对页面的布局施加更多的控制。当 Netscape 公司推出新的可以控制网站外观的 HTML 标签时，受到了广大网页设计师的欢迎。

1.1.2 CSS 的基本语法

CSS 语法由三个基本部分构成：选择符 (Selector)、属性 (Properties) 和属性的取值 (Value)。

格式：`selector {property: value}` 而 (选择符 {属性:值})
选择符可以是多种形式，一般是需要定义样式的 HTML 标记，例如 BODY、P 和 TABLE 等，可以通过此方法定义它的属性和值，属性和值要用冒号隔开。例如：

```
p {
    font-family: "宋体";
}
```

/*定义段落字体为宋体*/

如果需要对一个选择符指定多个属性时，使用分号将所有的属性和值分开。例如：

```
p {
    font-size: 14px;
    text-align: center;
```

```

}
/*字体大小为 14 像素，段落居中排列*/
.center {
    font-size: 12px;
    text-align: center;
}

```

这里定义了所有的 p 标记，字体大小为 12 像素，并且让 p 标记中的所有文字居中排列。效果如图 1.1 所示。

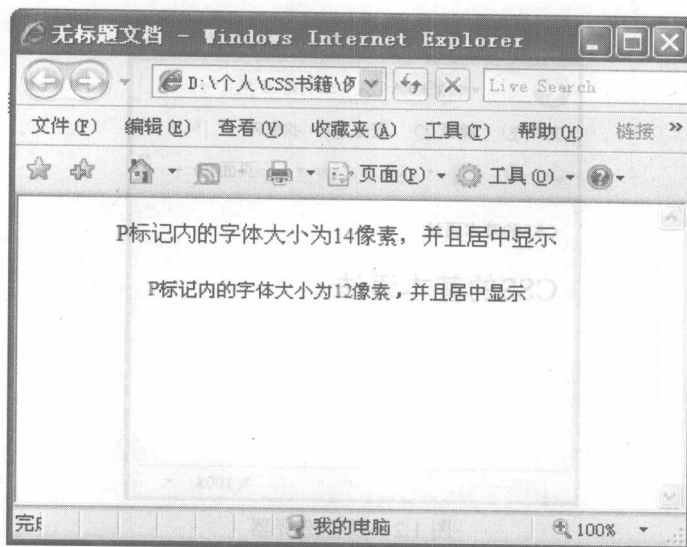


图 1.1 通过 CSS 定义字体的大小

注意：被/* */包含的内容为 CSS 注释内容，它的作用就是为代码做解释说明，让二次开发人员更清楚的了解代码的作用范围以及功能结构，便于日后的维护工作。它的写法非常简单，以/*开始，以*/结束，中间为代码的解释内容。对于 CSS 初学者来说，适当的注释可以加深对 CSS 的理解和记忆。

1.2 CSS 选择器

一幅地图只有标记了位置才可以让人们找到目的地，一篇结构良好的 XHTML 文档需要为 CSS 样式提供相应的接口，才能让 CSS 样式得到用武之地。想要发挥 CSS 的自身价值，就需正确地掌握 CSS 选择器，合理地使用它，将使页面代码不再繁杂。

1.2.1 标签选择器

一个完整的 HTML 页面是由很多不同的标签组成的，标签选择器就是直接将 HTML 标签作为选择器，决定哪些标签采用相应的 CSS 样式。例如：

```
h1 {
    font-size:12px;/*定义 h1 标签内的字体都为 12 像素*/
}

p {
    font-size: 22px;/*定义 p 标签内的字体都为 22 像素*/
}
```

这里直接使用了 `h1` 和 `p` 作为选择器,它们定义了 HTML 文档中所有的 `h1` 和 `p` 元素。在上面的例子中所有的 `h1` 元素的字体大小都为 12 像素, `p` 元素中的所有文字大小都为 22 像素。效果如图 1.2 所示。

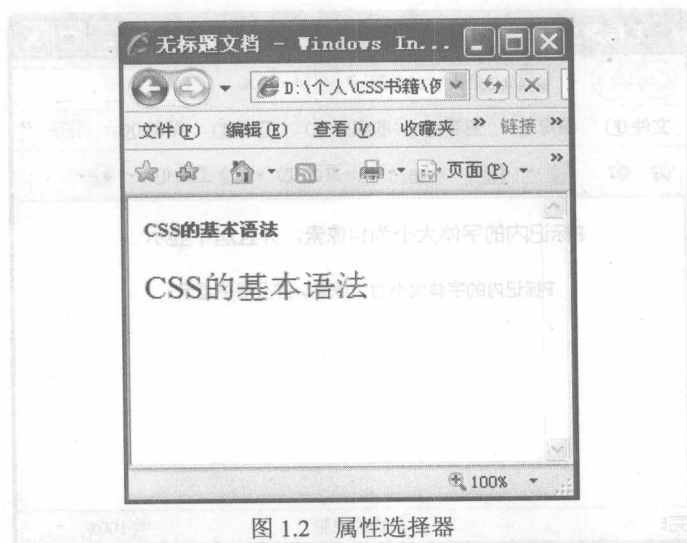


图 1.2 属性选择器

1.2.2 选择器组

选择器组就是将相同属性的选择器组合起来书写,用逗号将不同的选择器分开,用以减少 CSS 样式的重复定义。

以下代码可以更好地理解选择器组的书写形式。例如:

```
<style type="text/css">
h1, h2, h3, h4 {
    color:#000000;
}/*这个组中所有标题元素的文字颜色都为黑色。*/
</style>
<body>
<h1>CSS 选择器组</h1>
<h2>CSS 选择器组</h2>
<h3>CSS 选择器组</h3>
<h4>CSS 选择器组</h4>
</body>
```

以上代码中，利用 CSS 选择器组将 h1, h2, h3, h4 字体颜色定义为了黑色。效果如图 1.3 所示。

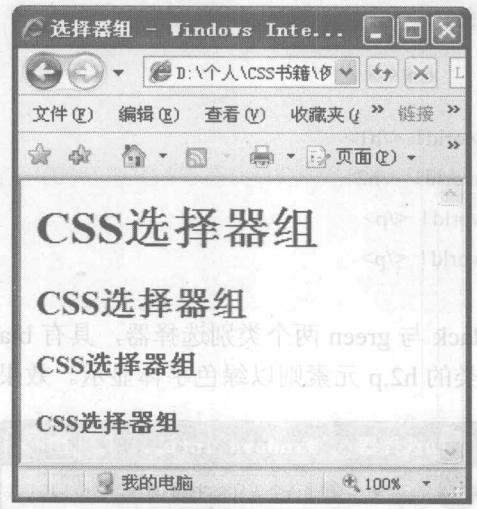


图 1.3 选择器组的使用

利用 CSS 选择器组，可以减少大量重复的代码。例如：

```
body {
font-size: 12px;
color: red;
}/*所有元素的文字大小都为 12 像素，字体为红色*/

p {
font-size: 12px;
color: red;
}/*所有 p 标记内的文字大小都为 12 像素，字体为红色*/
```

以上代码中 body 与 p 两部分代码可以合并为一个部分。代码如下：

```
body,p {
font-size: 12px;
color: red;
}/*这个组中所有标题元素的文字大小都为 12 像素，字体为红色*/
```

1.2.3 类选择器

class 类选择器就是将相同的元素分类，再进行 CSS 样式的定义。在网页中允许 class 类选择器多次使用，定义类选择器时，需要在自定义类的名称前加点名“.”。例如：

```
<style type="text/css">
.black {
color:#000000;
}/*设置字体的颜色为黑色*/
```

```

.green {
color: green;
}/*设置字体的颜色为绿色*/
</style>
<body>
<h1 class="black">hello world! </h1>
<h2 class="green">hello world! </h2>
<p class="black"> hello world! </p>
<p class="green"> hello world! </p>
</body>

```

上面的例子定义了 black 与 green 两个类别选择器，具有 black 类的 h1,p 元素均以黑色字体显示，具有 green 类的 h2,p 元素则以绿色字体显示。效果如图 1.4 所示。



图 1.4 类选择器的定义方法

在一些具有相同属性的样式中使用类别选择器，可以节省大量重复的代码。

1.2.4 ID 选择器

ID 选择器与类选择器类似，但它们的表示方法不同，ID 选择器由一个#字符表示。在 HTML 文档中只需要把 class 换成 id 即可。例如：

```

<style type="text/css">
#size {
font-size:22px;
}/*设置字体的大小为 22 像素*/
#size_two{
font-size:16px;

```

```

        }/*设置字体的大小为 16 像素*/
</style>
<body>
<p id="size">hello world! </p>
<p id="size_two">hello world! </p>
</body>

```

以上代码定义了 size 和 size_two 两个 ID 选择器，ID 为 size 选择器的字体大小为 22 像素，ID 为 size_two 选择器的字体大小为 16 像素。效果如图 1.5 所示。



图 1.5 ID 选择器

注意：ID 选择器用于寻找指定 ID 名称的元素，并且在同一篇 HTML 文档中同一个 ID 只能出现一次。

1.2.5 相邻选择器

如果一个标题 h1 元素后面紧跟了两个段落 p 元素，找到其中一个段落 p 元素进行定位，就可以使用相邻同胞选择器。例如：

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/
xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>相邻选择器</title>
<style type="text/css">
h1 + p {
font-size:22px;

```

```
/*设置字体的大小为 22 像素*/
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Div css 布局</h1>
```

```
<p>Div CSS 教程</p>
```

```
<p> Div CSS 布局实例</p>
```

```
</body>
```

```
</html>
```

上面例子很好地说明了相邻选择器的用法和书写规范,通过定义 h1 标题后的第一个段落字体大小为 22 像素,并且不会影响到其他段落。效果如图 1.6 所示。



图 1.6 相邻选择器的使用

1.2.6 后代选择器

后代选择器作用于所有子后代元素,通过空格来进行选择。通常把外层的标记写在前面,内层的标记写在后面,之间用空格分隔,当标记发生嵌套时,内层的标记就成为外层标记的后代。例如:

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

```
#links a {
```

```
color:#000000;/*设置链接颜色为黑色*/
```

```
font-size:20px;/*字体大小为 20 像素*/
```

```
}
```

```
a span {
```

```
color:#000000;
```

```
font-weight:700; /*加粗显示 span 标签内的字体宽度*/
```

```
}
```

```
</style>
```

```
<body>
```

```
<p id="links">
```

```
  <a href="#">Div CSS 教程</a>
```

```
</p>
```

```
<a href="#">CSS 布局<span>实例</span></a>
```

```
</body>
```

嵌套在 p 元素下的 a 标记字体大小为 20 像素，而嵌套在 a 标记下的 span 字体宽度值设为 700，这就是后代选择器的用法。效果如图 1.7 所示。

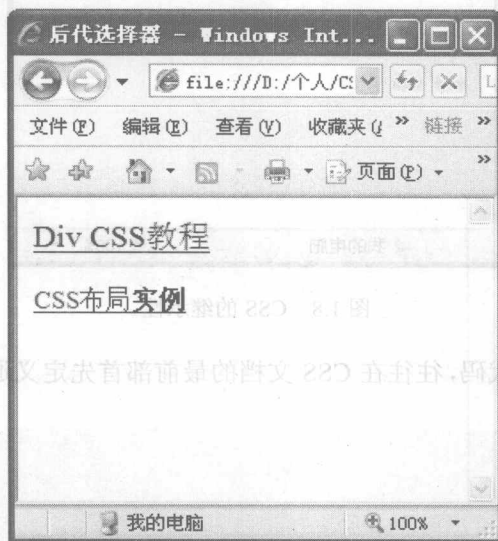


图 1.7 后代选择器的使用

1.3 CSS 的继承

CSS 的继承是指被包在内部的标签将拥有外部标签的样式性质。继承特性最典型的应用通常是在整个网页的样式预设，即整体布局声明方面。在继承原有的样式基础上再指定其他样式即可达到不同的效果，这项特性可以给网页设计者提供更理想的发挥空间。CSS 的主要特征就是继承，它是依赖于祖先与后代的关系。继承是一种机制，它不仅允许样式应用于某个特定的元素，还可以应用于该元素的后代。例如一个 BODY 定义了字体大小也会应用到段落的文本中。下面举例说明：

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

```
body {font-size:22px;}
```

```
</style>
```

```
<body>
```

<p>CSS继承性之深入剖析</p>

</body>

由于“继承性”既包括 body 中的样式属性，也使用了 strong 元素，应使用粗体。运行上面的代码。效果如图 1.8 所示。

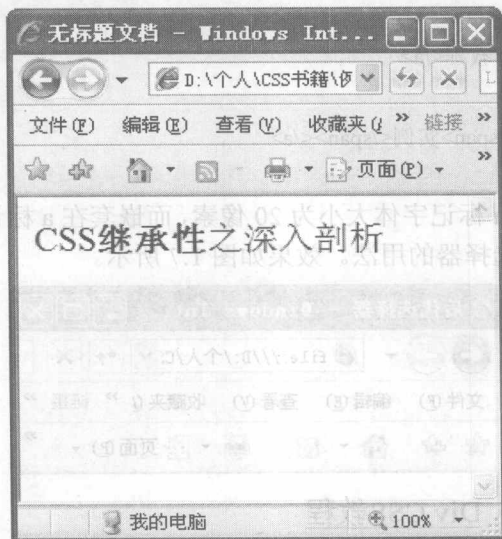


图 1.8 CSS 的继承性

在实际工作中编写代码，往往在 CSS 文档的最前部首先定义通篇文档中的所有元素，例如：

```
*{  
margin: 0;  
padding: 0;  
border: 0;  
/*设置所有的标签内外边距为 0，边框为 0*/
```

以上代码在缺省定义的情况下，所有元素的 margin、padding 和 border 的值都为零。当需要应用不同样式的时候，再单独对某元素进行定义即可，也就是整个网页的样式预设和整体布局声明。如需要指定其他样式的部分，设定在个别元素里即可达到效果。

在 CSS 中，继承是一种非常自然的行为，但是继承也有其局限性：有些属性是不能继承的。举个例子来说：border 属性的作用是设置元素的边框，它没有继承性。如果继承了边框属性，那么文档看起来就会很奇怪。例如定义容器 DIV 的边框为 1 像素，而在此容器内的“”在正常情况下都不希望它有边框，如果 border 有继承性，就要再去掉它们的边框。这样显然是不合理的。

多数边框类的属性，如：Border（边框）、Padding（内边距）、Margin（外边距）和 Background（背景）等，都是没有继承性的。在某些时候继承也会带来一些错误，比如下面这条 CSS 定义：

```
body {
```

```
color: blue
}
```

这里定义了 `body` 中的文本颜色为蓝色，如果 `body` 中含有表格，在有些浏览器中这个定义会使除表格之外的文本变成蓝色，而表格内部的文本颜色并不是蓝色。从技术上来说，这是不正确的，但它确实存在。所以需要经常借助于某些技巧，比如将 CSS 定义成这样：

```
body,table,th,td{
color: blue
}
```

这样表格内的文字也会变成蓝色了。

1.4 CSS 优先级

所谓 CSS 优先级，是指 CSS 样式在浏览器中被解析的先后顺序。优先级问题是一个冲突解决的问题，当同一个元素（内容）被 CSS 选择符选中时，就要按照优先级取舍不同的 CSS 规则，这其中涉及到的问题很多。其规则如下：

- 原则一 继承不如指定。
- 原则二 ID 选择器优先级大于类别选择器，类别选择器大于标签选择器。
- 原则三 越具体级别越高。
- 原则四 标签`#id` 优先级大于`#id`，标签`.class` 大于`.class`。

1.4.1 原则一

如果某样式是继承来的，其优先级不如具体指定的优先级高。例如：

```
<style type="text/css">
<!--
*{
font-size: 20px
}
.class3 {
font-size: 12px ;
}
-->
</style>
<body>
  <span class="class3">我是多大字号？ </span>
</body>
```

这里通过`*`对整篇文档进行了定义，定义所有的字体大小都为 20 像素，还定义`.class3`中字体大小为 12 像素，按照继承不如指定的原则，最终运行结果为：`.class3 {font-size: 12px ;}`。效果如图 1.9 所示。