

时尚百例丛书



CSS

时尚编程百例



●网冠科技 编著

100

时尚百例丛书

CSS

时尚编程百例

网冠科技 编著

光盘包含本书素材、效果文件



机械工业出版社

本书没有对 CSS 的各种属性进行简单的罗列,而是通过实例来讲解 CSS 的用法,并且运用了 JavaScript 脚本语言制作实例。

本书通过 100 个实例,全面讲解了 CSS 的定义方法和技巧。内容包括:定义文字样式、使用边框属性定义导航菜单的边框、使用定位技术对图片进行定位、使用各种属性对页面进行定义、以及滤镜变换效果的运用。

本书适用于那些比较熟悉 JavaScript 脚本语言的网页设计人员、程序设计人员,也可作为培训教材使用。

图书在版编目(CIP)数据

CSS 时尚编程百例 / 网冠科技编著.

-北京:机械工业出版社,2001.6

(时尚百例丛书)

ISBN7-111-09017-9

I. C … II. 网… III. 主页制作-软件工具, CSS IV. TP393.092

中国版本图书馆 CIP 数据核字(2001)第 033217 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策 划:胡毓坚

责任编辑:王 虹

责任印制:路 琳

北京机工印刷厂印刷·新华书店北京发行所发行

2001 年 6 月第 1 版·第 1 次印刷

787mm × 1092mm $\frac{1}{16}$ · 19.25 印张 · 477 千字

0001-6000 册

定价:35.00 元(1CD)

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换
本社购书热线电话:(010) 68993821、68326677-2527

时尚百例丛书

追求时尚 追求完美

出版说明

随着 21 世纪的到来,人们更深切地感受到了计算机在生活和工作中的作用越来越重要,越来越多的职业需要具有计算机的应用技能。掌握计算机是职业的需要,更是事业发展的需要。

目前计算机技术不但广泛地应用在办公自动化中,它还全面渗透到各行各业。如果要从事平面设计的相关行业,就应该学会平面设计软件,如 Photoshop、CorelDRAW、FreeHand 等;如果要从事三维设计的相关行业,就应该学会三维设计软件,如 3DS MAX、Maya、Poser 等;如果要从事多媒体设计的相关行业,就应该学会多媒体制作软件,如 Authorware、Director、Premiere 等;如果要从事与网络相关的行业,就应该学会 Flash、Dreamweaver、Fireworks、ASP、PHP、JavaScript 等;如果要从事建筑产品、工业产品设计的相关行业,就应该学会 AutoCAD、3DS VIZ、Protel 等;如果要从事软件开发的相关行业,就应该学会 VB、VC、VFP、Delphi、PowerBuilder 等编程。

所有与计算机相关的职业都要求工作者有很强的计算机操作技能,做到运用自如,熟练而且深入地掌握软件的应用。而要做到这一点,必须从软件的各个方面入手,通过实例演练的方式训练自己,而且要反复练习,做到举一反三。

为了让读者能深入而且熟练地掌握相关软件的应用方法,机械工业出版社特别为广大读者推出了这套时尚百例丛书。本丛书对每一个应用软件精心制作了 100 个实例,其宗旨就是让读者全方位掌握软件的应用,为广大读者提供一条快速掌握计算机应用技能的捷径。

本丛书采用新颖的版式,将知识和实例紧密结合,通过对各种实例的详细讲解,使读者不必事先学习各种软件,而从实例的制作过程中体会到每个软件每项功能的使用方法,并自己做出各种实例效果,这样既节省了大量时间,同时也使读者有身临其境的感觉,并可以反复演练,将所学知识运用到职业工作中去。

书山有路勤为径。愿广大读者能通过本丛书的学习掌握计算机技能,并应用到自己的工作和事业中去。

机械工业出版社



前 言

《CSS 时尚编程百例》是“时尚百例丛书”中的一本。

在短短的几年内，HTML 的功能有了长足的进步，但是，文档结构和样式显示的混合一直是 HTML 语言的一大缺陷。为了能够让网页更好地在各种平台上兼容，W3C 标准化组织推出了 CSS1 规范，将样式描述内容彻底地独立于结构文档。

CSS 对于设计者来说是一种非常灵活的工具，不必再把繁杂的样式定义编写在文档结构中，可以将所有有关与文档的样式指定内容全部脱离出来。CSS 还显示了强大的定义功能，它不仅可以定义通用的颜色、字体、背景等样式，还可以定义控制字符间距、填充距、大小写等 50 多个的样式。对于初始的 CSS 规范，W3C 也做了延伸和扩展，包括打印的支持、声音样式表单、框架结构、定位技术以及滤镜变换的效果等。

本书以实例的形式向读者讲解如何使用 CSS。本书共分为五篇，第一篇制作一些基于文字的实例；第二篇关于导航类菜单的应用；第三篇对图片进行定位的应用；第四篇对页面进行描述；第五篇滤镜变换效果。

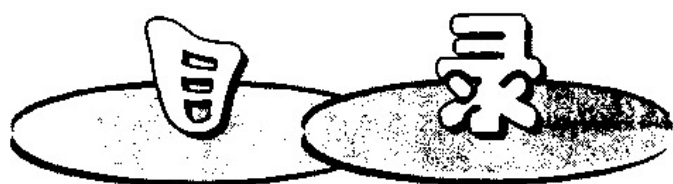
相信通过本书 100 个实例的学习，网页设计者能够更好地设计页面。



网冠科技

本书光盘含配套素材技术支持请点击网冠科技站点 <http://netking.163.com>。E-mail:

netking_@yeah.net.  是网易公司的标志。



出版说明 前言

第一篇 CSS 文本效果篇

实例 1	利用鼠标改变文字的样式	2
实例 2	不同的超链接	4
实例 3	彩色波浪文字	7
实例 4	滚动链接	10
实例 5	动态按钮	12
实例 6	无中生有的按钮	14
实例 7	垂直滚动板	16
实例 8	多变的超链接	19
实例 9	提示文字	23
实例 10	有停歇的滚动链接	28
实例 11	为控件添加样式	33
实例 12	上下弹跳的文字	35
实例 13	旋转的文字	38
实例 14	文字的不同显示	40
实例 15	具有打字效果的文字	44
实例 16	抖动的文字	47
实例 17	七彩链接	50
实例 18	总是不动的文本	52
实例 19	文字的特殊显示	54
实例 20	文字的动态显示	57
实例 21	逐个闪烁的文字	60
实例 22	自动输出的文本	63
实例 23	弹簧卷轴	65
实例 24	文字变幻特效	68
实例 25	跑马灯式样的提示信息	73
实例 26	文字慢慢出现	76
实例 27	消息公告板	81



实例 28	链接窗口说明	84
-------	--------	----

第二篇 CSS 导航效果篇

实例 29	动态菜单	87
实例 30	推拉式菜单	90
实例 31	弹出菜单	94
实例 32	自定义右键菜单	98
实例 33	循环的导航按钮	102
实例 34	可拖动的菜单	105
实例 35	图片翻滚导航	109
实例 36	可用鼠标驱动的菜单	112
实例 37	可层叠的菜单	115
实例 38	带边框的菜单	120
实例 39	隐藏的菜单	123
实例 40	卷帘菜单	125
实例 41	下拉菜单	129
实例 42	树形菜单	132

第三篇 CSS 图片效果篇

实例 43	抖动的图片	136
实例 44	图片播放器	138
实例 45	左右移动的图片	141
实例 46	移动的广告条	144
实例 47	跟随屏幕移动的图像	146
实例 48	被盖住的图片	151
实例 49	飞舞的图片链接	154
实例 50	静态导航	158
实例 51	图片扭曲变形	160
实例 52	抖动的窗口超链接	162

第四篇 CSS 页面效果篇

实例 53	为控件添加样式	165
实例 54	背景时钟	167



实例 55	滚动的背景	170
实例 56	倒计时	173
实例 57	十字准星	176
实例 58	模拟 Flash 效果	179
实例 59	陨石背景	182
实例 60	输入框的消失	186
实例 61	下雨背景	188
实例 62	水韵效果	192
实例 63	星空漫步	194
实例 64	页面打开方式(一)	197
实例 65	页面打开方式(二)	200
实例 66	页面打开方式(三)	203
实例 67	刷新页面	206
实例 68	任意改变页面样式	209
实例 69	跟随鼠标的文字	212
实例 70	动态鼠标指针	215
实例 71	流星攻击鼠标	219
实例 72	鼠标痕迹	223
实例 73	飞出来的字母	226
实例 74	跟随鼠标跳动的文字	230
实例 75	跟随鼠标的叠影文字	233
实例 76	跟随鼠标的时钟	236
实例 77	围绕鼠标旋转的小球	240
实例 78	图文混排	244
实例 79	项目列表	247
实例 80	自动重排列	249
实例 81	图片的显示和隐藏	251
实例 82	文字环绕图片	253
实例 83	文本列	255
实例 84	背景色渐变	257
实例 85	打开任意大小的窗口	260

第五篇 CSS 滤镜效果篇

实例 86	花的世界	263
实例 87	动态的滤镜效果	265
实例 88	图片的淡入淡出	267
实例 89	多变的转换效果	269

实例 90	倒计时过程	271
实例 91	动态光源效果	273
实例 92	火焰字	276
实例 93	雷达演示	278
实例 94	隐藏的文字	281
实例 95	若隐若现的文字	284
实例 96	光影效果的图片	287
实例 97	文字渐变	290
实例 98	图片的循环播放	293
实例 99	虚幻的图片	296
实例 100	多变的注释	299

第一篇

CSS 文本效果篇

本篇总览

本篇主要运用 CSS 中用于控制文本的一些属性，进行应用程序的开发，本篇将制作一些基于文本的例子。

通过本篇中各个实例的制作，读者可以对 CSS 中与文本有关的属性有一个大概的了解。这些属性是网页上出现最多的也是地位最重要的属性，通过对网页上文字和段落的精确控制，可以使读者制作的网页看起来更整齐更专业。

实例 1 利用鼠标改变文字的样式

实例说明

本例运行效果如图 1-1 所示。这是一个动态改变文字样式的例子。

程序运行后，将鼠标移到蓝色文字之上就会看到样式的改变。

本例的知识点有：font family、font size、letter spacing 等属性，以及响应鼠标技术。

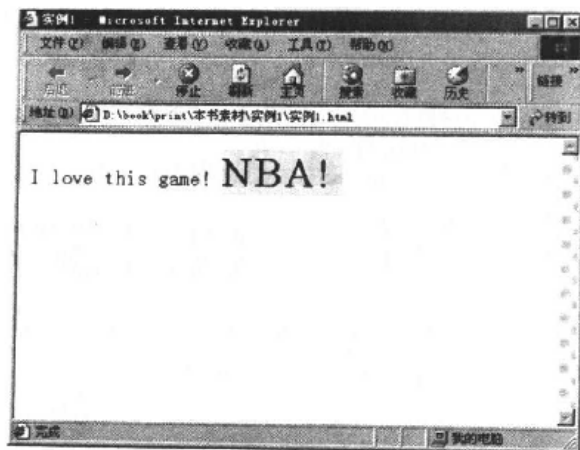


图 1-1 效果图

编程思路

本实例主要运用了 CSS 的 font family、font size、letter spacing 等属性，本例在<HEAD>标记中事先定义好所需的样式，然后给它一个 ID 标识。这样在文档中就可以随意调用，这种方法的好处是便于修改。

font family 属性的表述方法与大多数 CSS 属性有些不同，它的值用逗号分隔，用来指定可替换的字体。例如：“font family:times,Helvetica,sans serif;”如果浏览器没有找到“times”字体，那么就用“Helvetica”或者用“sans serif”字体替代。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<head>
<title>实例 1</title>
<style>
<!--
.Change
{font family: times,Helvetica,sans serif;
font size:200%; letter spacing:4px; background:yellow;
|
```

```
.Init  
{color:yellow; background:blue;  
}  
-->  
</style>  
</head>  
<body>  
<p>I love this game!  
<span onmouseover="this.className='Change'" onmouseout="this.className='Init'"  
class="start">NBA!  
</span>  
</p>  
</body>  
</html>
```

代码键入完毕，存盘为“实例 1.html”。

二、运行程序

双击文件“实例 1.html”。

实例 2 不同的超链接

实例说明

本例运行效果如图 2-1 所示。这是一个在浏览器中显示 3 种不同的超链接的例子。

程序运行后，将鼠标置于文本上时，会改变文本的样式。

本例的知识点有：text-decoration 属性、调用外部文件技术。

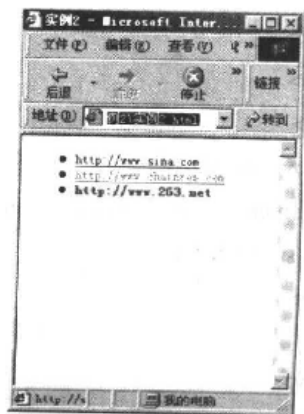


图 2-1 效果图

编程思路

本实例主要应用 text-decoration 属性对文本进行修饰。

text-decoration 属性可以方便地为网页上的文本加上一些有趣的修饰，例如下划线、上划线、中划线甚至文字的闪烁效果。

text-decoration 属性具有以下 5 种。

- (1) Underline：下划线；
- (2) Overline：上划线；
- (3) Line-through：删除线；
- (4) Blink：闪烁；
- (5) None：默认时使用表示无修饰。

创作步骤

一、创建外部文件

打开记事本，键入如下代码：

```
A:link  
{color:blue;  
}  
A:visited  
{color: blue;
```



```
}
A:hover
{color:blue; font-weight: bold; text-decoration: none;
}
A.title:link
{color: #808080;
}
A.title:visited
{color: #808080;
}
A.title:hover
{color: #000000; text-decoration: none; font-weight: bold;
}
A.red:link
{color:red;
background-color:#808080;
}
A.red:visited
{color:red;
background-color:808080;
}
A.red:hover
{color: red;
text-decoration: none; font-weight: bold; background-color: yellow;
}
```

代码键入完毕，存盘为“mystyle.css”。

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<head>
<title>实例 2</title>
<link rel="stylesheet" href="mystyle.css">
<style>
body {font family: "宋体", "Arial Narrow", "Times New Roman"; font size: 9pt}
</style>
</head>
<body>
<ul>
```

```
<li><a href="http://www.sina.com">http://www.sina.com</a></li>
<li><a class="title"href="http://www.chinaren.com">http://www.chinaren.com</a></li>
<li><a class="red"href="http://www.263.net">http://www.263.net</a></li>
</ul>
</body>
</html>
```

代码键入完毕，存盘为“实例 2.html”。

三、运行程序

双击文件“实例 2.html”。

实例 3 彩色波浪文字

实例说明

本例运行效果如图 3-1 所示。这是一个显示波浪文字的实例。

程序运行后，6 种不同的样式在 6 个文字上循环使用，产生波浪效果。

本例的知识点有：font family、font size、color 等属性以及一些 VBScript 知识。



图 3-1 效果图

编程思路

本实例对 6 个文字定义了 6 种不同的样式，通过循环使 6 种不同的样式在 6 个文字上顺序地移动，以达到波浪的效果。其中，用到了 VBScript 脚本语言，有关这方面的内容请参看有关 VBScript 的参考书。

创作步骤

一、创建脚本代码

```
dim styles(5) //样式数组
styles(0)="style1"
styles(1)="style2"
styles(2)="style3"
styles(3)="style4"
styles(4)="style5"
styles(5)="style6"
dim colors(5) //颜色数组
colors(0)="red"
colors(1)="black"
colors(2)="green"
colors(3)="aqua"
colors(4)="lime"
```




```
colors(5)="blue"
sub change
dim i,temp
temp=styles(0)
for i=0 to 4 //循环改变样式
styles(i)=styles(i+1)
next
styles(5)=temp
for i=0 to 5
document.all("font"&(i+1)).className=styles(i)
next
randomize
for i=0 to 5
document.all("font"&(i+1)).style.color=colors(round(rnd()*5))
next
tid=window.setInterval("change",500)
end sub
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<head>
<title>实例 3</title>
<style>
<!--
.style1
{font family:宋体; font size:10pt; color:rgb(0,128,0);
}
.style2
{font family:宋体; font size:20pt; color:rgb(0,128,0);
}
.style3
{font family:宋体; font size:30pt; color:rgb(0,128,0);
}
.style4
{font family:宋体; font size:30pt; color:rgb(0,128,0);
}
.style5
{font family:宋体; font size:20pt; color:rgb(0,128,0);
```



```
}  
.style6  
{font family:宋体;font size:10pt;color:rgb(0,128,0);  
}  
-->  
</style>  
</head>  
<script language="VBScript">  
<!--  
键入以上脚本代码。  
-->  
</script>  
<body onclick="change">  
<p align="center">  
<span class="style1" id="font1">彩</span>  
<span class="style2" id="font2">色</span>  
<span class="style3" id="font3">波</span>  
<span class="style4" id="font4">浪</span>  
<span class="style5" id="font5">文</span>  
<span class="style6" id="font6">字</span>  
</p>  
</body>  
</html>
```

代码键入完毕，存盘为“实例 3.html”。

三、运行程序

双击文件“实例 3.html”。

实例 4 滚动链接

实例说明

本例运行效果如图 4-1 所示。这是一个不停滚动的链接的例子。

程序运行后,在浏览器中显示循环一组链接,并且该循环不会停止。

本例的知识点有:CSS 的基本属性,以及 JavaScript 脚本语言知识。

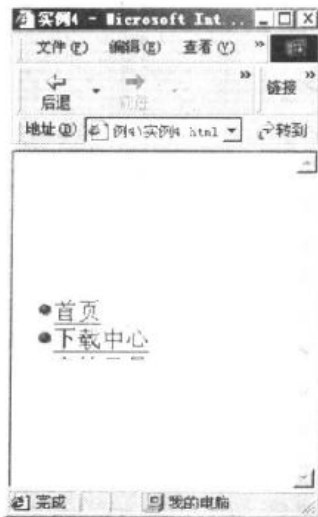


图 4-1 效果图

编程思路

本实例使用 CSS 中的 Width、Height 属性定义图片的大小,使用 Marquee 控制链接的滚动。

创作步骤

一、创建源代码

打开记事本,键入如下代码:

```
<html>
<title>实例 4</title>
<body>
<script language=javascript>
<!--
var index = 9
link = new Array(8); //链接数组
text = new Array(8); //链接文字数组
link[0] = 'time1.htm'
link[1] = 'time2.htm'
link[2] = 'time3.htm'
```

```
link[3]='time1.htm'  
link[4]='time2.htm'  
link[5]='time3.htm'  
link[6]='time1.htm'  
link[7]='time2.htm'  
link[8]='time3.htm'  
text[0]='首页'  
text[1]='下载中心'  
text[2]='竞技风暴'  
text[3]='友情链接'  
text[4]='娱乐中心'  
text[5]='技术中心'  
text[6]='网上赚钱'  
text[7]='免费资源'  
text[8]='本站论坛'  
document.write("<marquee scrollamount='1' scrolldelay='50' direction='up' width='150' height='150'>");  
for(i=0;i<index;i++)  
{ document.write("&nbsp;<img src='dotnew.gif' width='12' height='12'><a href="+link[i]+"  
target='_blank'>"); document.write(text[i] + "</A><br>");  
}  
document.write("</marquee>")  
// -->  
</script>  
</body>  
</html>
```

代码键入完毕，存盘为“实例4.html”。

二、运行程序

双击文件“实例4.html”。

实例 5 动态按钮

实例说明

本例运行效果如图 5-1 所示。这是一个动态按钮的实例。图 5-1 是鼠标置于按钮上的显示图，图 5-2 是鼠标离开按钮时的显示图。

程序运行后，单击按钮会看到按钮文字的变化。

本例的知识点主要是一些鼠标响应事件以及一些基本的属性。



图 5-1 鼠标置于按钮上



图 5-2 鼠标离开后

编程思路

本实例主要通过鼠标事件的触发改变按钮的样式。

创作步骤

一、创建源代码

打开记事本，键入如下代码。

```
<html>
<head>
<title>实例 5</title>
<style type="text/css">
<!--
.over
{color:yellow;
```

```
background:black;
}
.down
{color:yellow;background:black;font-style:italic;
}
-->
</style>
</head>
<body>
<input
type="Button"
onMouseOver="this.className='over';"
onMouseOut="this.className='';"
onMouseDown="this.className='down';"
onMouseUp="this.className='over';"
value="未被点击的按钮"
onClick="this.value='被点击以后的按钮'" name="Button">
</body>
</html>
```

代码键入完毕，存盘为“实例 5.html”。

二、运行程序

双击文件“实例 5.html”。

实例 6 无中生有的按钮

实例说明

本例运行效果如图 6-1 所示。这是一个产生动态按钮的例子。

程序运行后，将鼠标置于文字上时，就会出现按钮。

本例的知识点有：CSS 中的一些容器属性。



图 6-1 效果图

编程思路

本实例主要应用边框属性和填充距属性来制作按钮。

边框属性如下：border right、border left、border top 和 border bottom 4 种属性。它们都有 3 个参数，依次为边框的宽度、样式、颜色。

填充距属性如下：padding right、padding left、padding bottom 和 padding top 4 种属性。它们只有一个参数，用来控制元素内部的周边空间。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<head>
<title>实例 6</title>
<style>
<!--
A.mystyle
{color: black, text-decoration:none
}
A.mystyle:hover
```

```

    {color: blue
    }
    .mouseup
    {border right: medium ridge blue; border top: medium ridge blue; border left: medium ridge blue; border
bottom: medium ridge blue;
    padding right: 3px; padding left: 3px; padding bottom: 0px; padding top: 0px;
    font-weight: bold; font size: 14px;
    margin: 2px 0px; cursor: hand; color: blue; line-height: 14pt; font family: 宋体; letter spacing: -0.5pt;
    background-color:white;
    text-decoration: none
    }
-->
</style>
</head>
<body bgcolor=#ffffff>
<table cellspacing=0 cellpadding=0 width="33%" border=0>
<tbody>
<tr>
<td width="100%"></td></tr>
<tr>
<td onmouseover="this.className='mouseup'" onmouseout="this.className=''" >
<a class=mystyle href="">国内足坛</a></td></tr>
<tr>
<td width="100%"></td></tr>
<tr>
<td onmouseover="this.className='mouseup'" onmouseout="this.className=''" width="100%">
<a class=mystyle href="">国际足坛</a></td></tr>
</tbody>
</table>
</body>
</html>

```

代码键入完毕，存盘为“实例 6.html”。

二、运行程序

双击文件“实例 6.html”。

实例 7 垂直滚动板

实例说明

本例运行效果如图 7-1 所示。这是一个垂直滚动板的例子。

程序运行后,在浏览器中会有一段文字在循环滚动。

本例的知识点有:CSS 中的一些基本的定位属性以及 JavaScript 脚本语言。

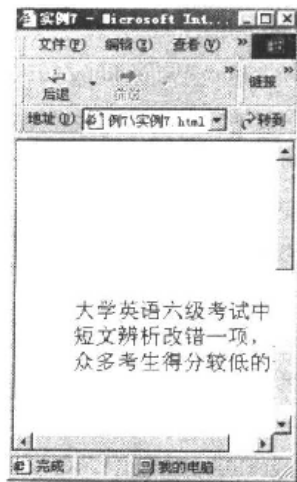


图 7-1 效果图

编程思路

本实例主要应用定位属性对滚动板进行定位,用 JavaScript 脚本语言使滚动板进行滚动。

创作步骤

一、创建脚本代码

```

lstart=200
loop=true
speed=150
pr_step=5
function makeObj(obj,nest)
{nest=(!nest) ? "document."+nest+"":
this.css=(document.layers)?eval(nest+'document'+obj)
:eval(obj+'.style')
this.scrollHeight=document.layers?
this.css.document.height
:eval(obj+'.offsetHeight')
this.up=goUp
this.obj = obj + "Object"

```

```

eval(this.obj + "=this")
return this
}
function goUp(speed) //向上移动
{if(parseInt(this.css.top)>-this.scrollHeight)
{this.css.top=parseInt(this.css.top)-pr_step
setTimeout(this.obj+".up("+speed+")",speed)
}
else if(loop)
{this.css.top=lstart
eval(this.obj+".up("+speed+")")
}
}
function slideInit() //初始化
{oSlide=makeObj('divNews','divCont')
oSlide.css.top=lstart
oSlide.up(speed)
}
onload=slideInit

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 7</title>
<head>
<script>
<!--
键入以上脚本代码。
-->
</script>
</head>
<body>
<div id="divCom" style="position:absolute;
width:200; height:100; top:50; left:50;
clip:rect(0,250,150,0)">
<div id="divNews" style="position:absolute; top:0; left:0">

```

<p>大学英语六级考试中的短文辨析改错一项，是众多考生得分较低的一项。短文辨析改错不仅要求学生要有较高的阅读理解能力，还要求学生有深厚的语法知识，善于分析英语基本句型和句子结构。它主要是考查学生辨析并改正语言知识

在运用时常出现的各种错误的能力。我们在分析历届六级真题的基础上，归纳、总结出以往六级短文改错的命题倾向和常见错误类型，并通过对历年六级短文改错真题的举例分析，培养学生应试技巧。

```
</p>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

代码键入完毕，存盘为“实例 7.html”。

三、运行程序

双击文件“实例 7.html”。

实例 8 多变的超链接

实例说明

本例运行效果如图 8-1 所示。这是一个多变的超链接的例子。

程序运行后，会有几组链接循环显示。

本例的知识点有：CSS 中的边框属性和文本属性，以及 JavaScript 脚本语言知识。



图 8-1 效果图

编程思路

本实例主要使用文本属性和边框属性定义超链接文本的样式，使用 JavaScript 脚本语言控制几组超链接的循环显示。

创作步骤

一、创建脚本代码

```
var speed=2500
var news=new Array() //链接文字数组
news[0]="<a href='index1.htm'>国内新闻</a>"
news[1]="<a href='index2.htm'>国际新闻</a>"
news[2]="<a href='index3.htm'>文化娱乐</a>"
news[3]="<a href='index4.htm'>体育新闻</a>"
i=0
if(document.all)
    tickerobject=document.all.subtickerape.style
else
```

```

    tickerobject=document.tickertape.document
function regenerate()
{ window.location.reload()
}
function regenerate2()
{if(document.layers)
    setTimeout("window.onresize=regenerate",450)
}
function update()
{BgFade(0xff,0xff,0xff, 0x00,0x00,0x00, 10);
if(document.layers)
{document.tickertape.document.subtickertape.document.write('<span
    class="subtickertapefont">'+news[i]+'</span>')
    document.tickertape.document.subtickertape.document.close()
}
else
document.all.subtickertape.innerHTML+=news[i]
if(i<news.length-1)
    i++
else
    i=0
setTimeout("update()",speed)
}
function BgFade(red1,grn1,blu1,red2,grn2,blu2,steps)
{sred=red1;sgm=grn1;sblu=blu1;
ered=red2;egrn=grn2;ebfu=blu2;
inc=steps;
step=0;
RunFader();
}
function RunFader()
{var epct=step/inc;
var spct=1-epct;
if(document.layers)
    tickerobject.bgColor=Math.floor(sred*spct+ered*epct)*256*256+
        Math.floor(sgm*spct+egrn*epct)*256+
        Math.floor(sblu*spct+ebfu*epct);
else
    tickerobject.backgroundColor=Math.floor(sred*spct+ered*epct)*256*256+

```



```
Math.floor(sgm*spct+egm*epct)*256+
```

```
Math.floor(sblu*spct+eblu*epct);
```

```
if(step<inc)
{setTimeout('RunFader()',150);
}
step++;
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<head>
<title>实例 8</title>
<style>
<!--
#tickertape
{position:relative;
layer-background-color:black;
width:400;
height:12;
font family: "宋体";
font size: 9pt;
}
#subtickertape
{background-color:black;
position:absolute;
border: 1px solid black;
width:400;
height:12;
font family: "宋体";
font size: 9pt;
}
.subtickertapefont
{font:bold 9pt "宋体";
text-decoration:none;
color:white;
}
.subtickertapefont a
{color:white;text-decoration:none;
```

```
font family: "宋体"; font size: 9pt;
}
-->
</style>
</head>
<body onload="if (document.all||document.layers) {regenerate2():update()}">
<div id="tickertape">
<div id="subtickertape" class="subtickertapefont">初始化..</div>
</div>
<script language="JavaScript1.2">
键入以上脚本代码。
</script>
</body>
</html>
```

代码键入完毕，存盘为“实例 8.html”。

三、运行程序

双击文件“实例 8.html”。

实例 9 提示文字

实例说明

本例运行效果如图 9-1 所示。这是一个显示提示信息的例子。

程序运行后，将鼠标置于链接文字上会显示出提示信息。

本例的知识点有：CSS 的边框属性和文本属性，以及 JavaScript 脚本语言。

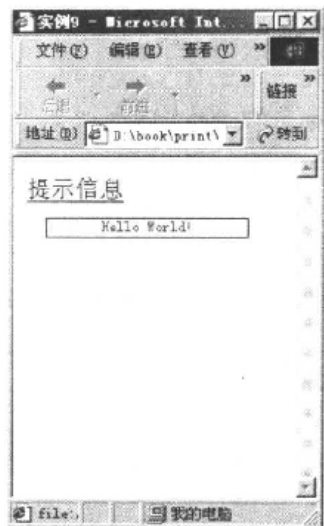


图 9-1 效果图

编程思路

本实例使用 CSS 的基本属性定义提示信息的框架以及文字样式，使用 JavaScript 脚本语言控制提示信息的显示。

创作步骤

一、创建脚本代码

```
var allSupport = document.all!=null
function setupEventObject(e)
{if(e==null)
    return
    window.event=e
    window.event.fromElement=e.target
    window.event.toElement=e.target
    window.event.srcElement=e.target
    window.event.x=e.x
    window.event.y=e.y
    window.event.srcElement.handleEvent(e);
```




```
    }  
    function checkName(src)  
    { while((src!=null)&&(src._tip==null))  
        src = src.parentElement  
        return src  
    }  
    function getElement(elName)  
    { if(allSupport)  
        return document.all[elName]  
        else  
        return document.layers[elName]  
    }  
    function writeContents(el, tip)  
    { if(allSupport)  
        el.innerHTML = tip  
        else  
        { el.document.open()  
            el.document.write("<TABLE WIDTH=200 BORDER=1 bordercolor=black><TR><TD WIDTH=100%  
                BGCOLOR=yellow>")  
            el.document.write(tip)  
            el.document.write("</TD></TR></TABLE>")  
            el.document.close()  
        }  
    }  
    function getOffset(el, which)  
    { var amount=el["offset"+which]  
        if(which=="Top")  
            amount+=el.offsetHeight  
        el=el.offsetParent  
        while(el!=null)  
            { amount+=el["offset"+which]  
                el=el.offsetParent  
            }  
        return amount  
    }  
    function setPosition(el)  
    { src=window.event.srcElement  
        if(allSupport)  
            { el.style.pixelTop=getOffset(src, "Top") +10
```

```
    el.style.pixelLeft=getOffset(src, "Left") +15
  }
  else
  {el.top=src.y + 40
  el.left=src.x +40
  }
}
function setVisibility(el, bDisplay)
{if(bDisplay)
  if(allSupport)
    el.style.visibility="visible"
  else
    el.visibility="show";
else
  if(allSupport)
    el.style.visibility="hidden"
  else
    el.visibility="hidden"
}
function displayContents(tip)
{var el=getElement("tipBox")
writeContents(el, tip)
setPosition(el)
setVisibility(el, true)
}
function doMouseOver(e)
{setupEventObject(e)
var el, tip
if((el=checkName(window.event.srcElement))!=null)
  if(!el._display)
  {displayContents(el._tip)
  el._display = true
  }
}
function doMouseOut(e)
{setupEventObject(e)
el=checkName(window.event.srcElement)
var el, tip
if((el=checkName(window.event.srcElement))!=null)
```

```
    if(el._display)
        if((el.contains==null)||(!el.contains(window.event.toElement)))
            {setVisibility(getElement("tipBox"), false)
              el._display=false
            }
    }
function doLoad()
{if((window.document.captureEvents==null)&&(!allSupport))
    return
  if(window.document.captureEvents!=null)
    window.document.captureEvents(Event.MOUSEOVER | Event.MOUSEOUT)
  window.document.onmouseover=doMouseOver;
  window.document.onmouseout=doMouseOut;
}
window.onload = doLoad
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 9</title>
<head>
<script language=JavaScript>
<!--
键入以上脚本代码。
/-->
</script>
<style type="text/css">
#tipBox
{position:absolute;
width:160px;
z-index:100;
border:1pt black solid;
font family:宋体;
font size:9pt;
background:#FFFF00;
visibility:hidden
}
</style>
</head>
```

```
<body>
<div id="tipBox">
</div>
<a href="#" onMouseOver="this._tip = '<center><font color=#FF0000>Hello World!</font></center>' ">提示
    信息</a>
</body>
</html>
```

代码键入完毕，存盘为“实例 9.html”。

三、运行程序

双击文件“实例 9.html”。

实例 10 有停歇的滚动链接

实例说明

本例运行效果如图 10-1 所示。这是一个有停歇的滚动链接。

程序运行后,以 3 个链接为一组循环显示,每显示一组有一个停歇。

本例的知识点有:CSS 的定位属性,以及 JavaScript 脚本语言知识。



图 10-1 效果图

编程思路

本实例使用定位属性对链接进行定位,使用 JavaScript 脚本语言控制链接的滚动。

创作步骤

一、创建源代码

打开记事本,键入如下代码:

```
<html>
<title>实例 10</title>
<body>
<script language=JavaScript>
var scrollerwidth=115
var scrollerheight=131
var scrollerbgcolor=#FFFFFF
var pausebetweenimages=4000
var slideimages=new Array()
slideimages[0]='<a href="http://www.yahoo.com.cn" target="_blank">中文雅虎</a><br>
```

```

<br><a href="http://www.263.net" target="_blank">首都在线</a><br>
<br><a href="http://fsnet.yes8.com/" target="_blank">网络特区</a><br>
slideimages[1]='<a href="http://www.sina.com.cn" target="_blank">新浪</a><br>
<br><a href="http://www.163.com.cn" target="_blank">网易</a><br>
<br><a href="http://www.edu.com.cn" target="_blank">中国教育网</a><br>
slideimages[2]='<a href="http://www.yahoo.com.cn" target="_blank">中文雅虎</a><br>
<br><a href="http://www.263.net" target="_blank">首都在线</a><br>
<br><a href="http://fsnet.yes8.com/" target="_blank">网络特区</a><br>
slideimages[3]='<a href="http://www.sina.com.cn" target="_blank">新浪</a><br>
<br><a href="http://www.163.com.cn" target="_blank">网易</a><br>
<br><a href="http://www.edu.com.cn" target="_blank">中国教育网</a><br>
if(slideimages.length>1)
    i=2
else
    i=0
function move1(whichlayer)
{tlayer=eval(whichlayer)
if(tlayer.top>0&&tlayer.top<=4)
{tlayer.top=0
setTimeout("move1(tlayer)",pausebetweenimages)
setTimeout("move2(document.main.document.second)",pausebetweenimages)
return
}
if(tlayer.top>=tlayer.document.height*-1)
{tlayer.top=4
setTimeout("move1(tlayer)",100)
}
else
{tlayer.top=scrollerheight
tlayer.document.write(slideimages[i])
tlayer.document.close()
if(i==slideimages.length-1)
    i=0
else
    i++
}
}
function move2(whichlayer)
{tlayer2=eval(whichlayer)

```

```

if(tlayer2.top>0&&tlayer2.top<=4)
{tlayer2.top=0
  setTimeout("move2(tlayer2)",pausebetweenimages)
  setTimeout("move1(document.main.document.first)",pausebetweenimages)
  return
}
if(tlayer2.top>=tlayer2.document.height*-1)
{tlayer2.top=-4
  setTimeout("move2(tlayer2)",100)
}
else
{tlayer2.top=scrollerheight
  tlayer2.document.write(slideimages[i])
  tlayer2.document.close()
  if(i==slideimages.length-1)
    i=0
  else
    i++
}
}
function move3(whichdiv)
{tdiv=eval(whichdiv)
if(tdiv.style.pixelTop>0&&tdiv.style.pixelTop<=4)
{tdiv.style.pixelTop=0
  setTimeout("move3(tdiv)",pausebetweenimages)
  setTimeout("move4(second2)",pausebetweenimages)
  return
}
if(tdiv.style.pixelTop>=tdiv.offsetHeight*-1)
{tdiv.style.pixelTop=-4
  setTimeout("move3(tdiv)",100)
}
else
{tdiv.style.pixelTop=scrollerheight
  tdiv.innerHTML=slideimages[i]
  if(i==slideimages.length-1)
    i=0
  else
    i++
}
}

```

```
    }  
  }  
  function move4(whichdiv)  
  { tdiv2=eval(whichdiv)  
    if(tdiv2.style.pixelTop>0&&tdiv2.style.pixelTop<=4)  
    { tdiv2.style.pixelTop=0  
      setTimeout("move4(tdiv2)",pausebetweenimages)  
      setTimeout("move3(first2)",pausebetweenimages)  
      return  
    }  
    if(tdiv2.style.pixelTop>=tdiv2.offsetHeight*-1)  
    { tdiv2.style.pixelTop=-4  
      setTimeout("move4(second2)",100)  
    }  
    else  
    { tdiv2.style.pixelTop=scrollerheight  
      tdiv2.innerHTML=slideimages[i]  
      if(i==slideimages.length-1)  
        i=0  
      else  
        i++  
    }  
  }  
  }  
  function startscroll()  
  { if(document.all)  
    { move3(first2)  
      second2.style.top=scrollerheight  
    }  
    else if(document.layers)  
    { move1(document.main.document.first)  
      document.main.document.second.top=scrollerheight+5  
      document.main.document.second.visibility='show'  
    }  
  }  
  }  
  window.onload=startscroll  
  if(document.layers)  
    document.write(slideimages[0])  
  if(document.layers)  
    document.write(slideimages[1])
```



```
if(document.layers)
    document.write(slideimages[0])
if(document.all)
{document.writeln('<span id="main2" style="position:relative;
    width:'+scrollerwidth+';height:'+scrollerheight+';overflow:hidden;background-color:'+scrollerbgcolor+'>')
document.writeln('<div style="position:absolute;width:'+scrollerwidth+';height:'+scrollerheight+';
    clip:rect(0'+scrollerwidth+' '+scrollerheight+' 0);left:0;top:0">')
document.writeln('<div id="first2" style="position:absolute;width:'+scrollerwidth+';left:0;top:1;'>')
document.write(slideimages[0])
document.writeln('</div>')
document.writeln('<div id="second2" style="position:absolute;width:'+scrollerwidth+';left:0;top:0">')
document.write(slideimages[1])
document.writeln('</div>')
document.writeln('</div>')
document.writeln('</span>')
}
</script>
</body>
</html>
```

代码键入完毕，存盘为“实例 10.html”。

二、运行程序

双击文件“实例 10.html”。

实例 11 为控件添加样式

实例说明

本例运行效果如图 11-1 所示。这是一个给控件添加样式的例子。

程序运行后，会看到带样式的控件，它可以配合页面的背景色。

本例的知识点有：CSS 的一些基本属性如 color、background-color 等。

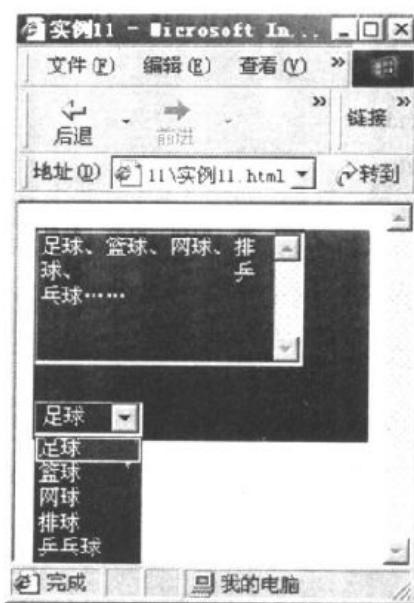


图 11-1 效果图

编程思路

本实例主要运用了一些 CSS 中的颜色属性，使控件能和设计的网页在风格上能够很好地统一起来。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<head>
<title>实例 11</title>
</head>
<body>
<table cellspacing="0" >
<tr>
<td width="36%" bgcolor="black">
<p>
```

```

<textarea rows="5" cols="20" style="color:#FFFFFF;background-color:black">足球、篮球、网球、排球、
乒乓球……
</textarea>
<p>
<p>
</p>
</td>
</tr>
</tr>
<td width="36%" bgcolor="black">
<p><select size="1" style="color: #FFFFFF;background-color:black">
  <option>足球</option>
  <option>篮球</option>
  <option>网球</option>
  <option>排球</option>
  <option>乒乓球</option>
</select>
</p>
</td>
</tr>
</table>
</body>
</html>

```

代码键入完毕，存盘为“实例 11.html”。

二、运行程序

双击文件“实例 11.html”。

实例 12 上下弹跳的文字

实例说明

本例运行效果如图 12-1 所示。这是一个文字可以上下弹跳的例子。

程序运行后，会看到文字上下弹跳。

本例的知识点有：CSS 中的基本定位属性，以及 JavaScript 脚本语言知识。



图 12-1 效果图

编程思路

本实例使用定位属性对文字进行初始定位，使用 JavaScript 脚本语言控制文字的上下弹跳。

创作步骤

一、创建脚本代码

```
done = 0;
step = 4
function anim(yp,yk)
{if(document.layers)
    document.layers["naps"].top=yp;
else
    document.all["naps"].style.top=yp;
if(yp>yk)
    step = -4
if(yp<60)
```

```

        step = 4
        setTimeout('anim('+yp+step+')'+',+yk+)', 10);
    }
    function start()
    {if(done)
        return
        done = 1;
        if(navigator.appName=="Netscape")
        { document.napis.left=innerWidth/2 - 145;
          anim(60,innerHeight - 60)
        }
        else
        {napis.style.left=50;
          anim(60,document.body.offsetHeight - 60)
        }
    }
}

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 12</title>
<head>
<script language="JavaScript">
<!--
键入以上脚本代码。
//-->
</script>
</head>
<body>
<div id="napis"
style="position:absolute;top:100px;width:300px;height:78px;left:50px">
<font size="5">
    <b>Hello!</b>
</font>
</div>
<script language="JavaScript">
<!--
    setTimeout('start()',10);
//-->

```

```
</script>
```

```
</body>
```

```
</html>
```

代码键入完毕，存盘为“实例 12.html”。

三、运行程序

双击文件“实例 12.html”。

实例 13 旋转的文字

实例说明

本例运行效果如图 13-1 所示。这是一个旋转显示文字的例子。

程序运行后，会看到一组文字以某一定点旋转地显示。

本例的知识点有：CSS 的文本属性，以及 JavaScript 脚本语言知识。

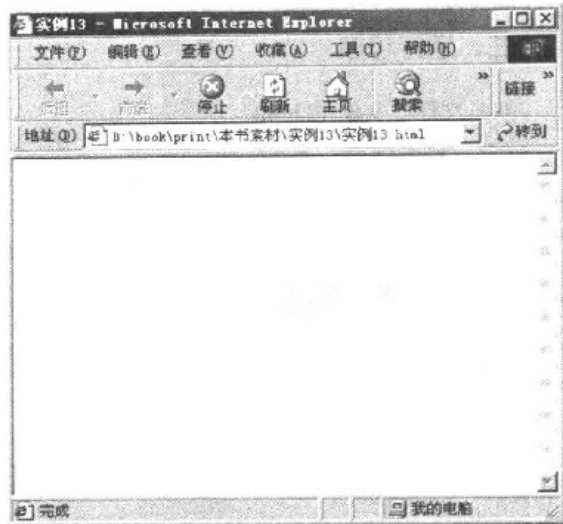


图 13-1 效果图

编程思路

本实例使用文本属性定义文字的样式，使用 JavaScript 脚本语言控制文字的旋转。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 13</title>
<body>
<script language=javascript>
message="旋转的文字"
Balises=""
Taille=40;
Midx=40;
Decal=0.5;
num=message.length;
y=-1000;
```

```
for (x=0;x<num;x++)
{ Balises=Balises + '<DIV Id=L' + x + ' STYLE="width:4;font family: 宋体;
    font-weight:bold;position:absolute;top:100;left:100;z-index:0">' + message.charAt(x) + '</DIV>'
}
document.write (Balises);
Time=window.setInterval("Alors()",10);
Alpha=5;
I_Alpha=0.05;
function Alors()
{ Alpha=Alpha-I_Alpha;
  for(x=0;x<num;x++)
  { Alpha1=Alpha+Decal*x;
    Cosine=Math.cos(Alpha1);
    Ob=document.all("L"+x);
    Ob.style.posLeft=Midx+100*Math.sin(Alpha1)+260;
    Ob.style.zIndex=10*Cosine;
    Ob.style.fontSize=Taille+25*Cosine;
    Ob.style.color="rgb("+ (20+Cosine*80)+150) + "," + (227+Cosine*80+250) + ",0)";
  }
}
</script>
</body>
</html>
```

代码键入完毕、存盘为“实例 13.html”。

二、运行程序

双击文件“实例 13.html”。

实例 14 文字的不同显示

实例说明

本例运行效果如图 14-1 所示。这是一个不同的文本显示效果的例子。

程序运行后，会看到文字从左到右、从右到左等几种不同的显示方式。

本例的知识点有：CSS 中的颜色属性，以及 JavaScript 脚本语言知识。



图 14-1 效果图

编程思路

本实例使用定位属性对文字进行定位，使用 JavaScript 脚本语言控制文字的循环滚动。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 14</title>
<body bgcolor="#fef4d9" onLoad="startNcon()">
<script language="JavaScript">
<!--
neonBaseColor="White";
neonColor="Red";
num=0;
num2=0;
num3=0;
```

```
num4=neonColor;
function startNeon()
{message = neon.innerText;
 neon.innerText = "";
 for(i = 0; i != message.length; i++)
 (neon.innerHTML += "<span id='neond' "
 style='color:"+neonBaseColor+" '>" +message.charAt(i)+"</span>"
 )
 neon2();
}
function neon2()
{if(num != message.length)
 {document.all.neond[num].style.color = neonColor;
 num++;
 setTimeout("neon2()", 100);
 }
 else
 {num = 0;
 num2 = message.length;
 setTimeout("neon4onev()", 2000);
 }
}
function neon4onev()
{document.all.neond[num].style.color = neonBaseColor;
 document.all.neond[num2-1].style.color = neonBaseColor;
 if(Math.floor(message.length / 2) + 1 != num2)
 {num++;
 num2--;
 setTimeout("neon4onev()", 50);
 }
 else
 {setTimeout("neon5()", 50);
 }
}
function neon5()
{if(num3 != message.length && num3 != message.length+1)
 {document.all.neond[num3].style.color = neonColor;
 num3 = num3 + 2;
 setTimeout("neon5()", 100);
```

```

    }
    else
    {setTimeout("neon52()", 50);
    }
}
function neon52()
{if(num3 == message.length)
    {num3++;
    neon52a();
    }
    else
    {num3--;
    neon52a();
    }
}
function neon52a()
{if(num3 != 1)
    {num3 = num3 - 2;
    document.all.neond[num3].style.color = neonColor;
    setTimeout("neon52a()", 100);
    }
    else
    {if(num4 == neonColor)
        {num3 = 0;
        neonColor = neonBaseColor;
        setTimeout("neon5()", 2000);
        }
        else
        {neonColor = num4;
        num3 = 0;
        setTimeout("neon4onev2()", 50);
        }
    }
}
function neon4onev2()
{document.all.neond[num].style.color = neonColor;
document.all.neond[num2 - 1].style.color = neonColor;
if(message.length != num2)
    {num--;

```

```
num2++;
setTimeout("neon4onev2()", 50);
}
else
{num = 0;
num2 = 0;
setTimeout("neon3()", 2000);
}
}
function neon3()
{if(num != message.length)
{document.all.neond[num].style.color = neonBaseColor;
num++;
setTimeout("neon3()", 100);
}
else
{num = 0;
neon2();
}
}
//-->
</script>
<h2><span id="neon">This is very instereting!</span></h2>
</body>
</html>
```

代码键入完毕，存盘为“实例 14.html”。

二、运行程序

双击文件“实例 14.html”。

实例 15 具有打字效果的文字

实例说明

本例运行效果如图 15-1 所示。这是一个文字有打字效果的例子。

程序运行后, 会有一组文字以打字的效果逐个显示。

本例的知识点有: CSS 中的基本的文本属性, 以及 JavaScript 脚本语言知识。



图 15-1 效果图

编程思路

本实例使用文本属性定义显示文字的样式, 使用 JavaScript 脚本语言控制文字的输出。

创作步骤

一、创建源代码

打开记事本, 键入如下代码:

```
<html>
<title>实例 15</title>
<body onload=init()>
<div class=ttl id=ttl0>
</div>
<script language=javascript>
var layers=document.layers,style=document.all,both=layers!style.idme=908601:
if(layers)
{layerRef=document.layers;
styleRef=";
}
if(style)
```

```
{layerRef='document.all';
 styleRef ='.style':
}
function writeOnText(obj,str)
{if(layers)with(document{obj})
 {document.open();
  document.write(str);
  document.close();
 }
 if(style)
  eval(obj+'.innerHTML=str');
}
var dispStr=new Array("具有打字效果的文字");
var overMe=0;
function txtFlyper(str,idx,idObj,spObj,clr1,clr2,delay,plysnd)
{var tmp0=tmp1="",skip=100;
 if(both && idx<=str.length)
 {if(str.charAt(idx)=='<')
  {while(str.charAt(idx)!='>')
   idx++;
   idx++;
 }
 if(str.charAt(idx)=='&'&&str.charAt(idx+1)!='')
 {while (str.charAt(idx)!=';')
  idx++;
  idx++;
 }
 tmp0 = str.slice(0,idx);
 tmp1 = str.charAt(idx++);
 if(overMe==0 && plysnd==1)
 {if(navigator.plugins[0])
  {if(navigator.plugins["LiveAudio"][0].type=="audio/basic" && navigator.javaEnabled())
   {document.embeds[0].stop();
    setTimeout("document.embeds[0].play(false)",100);}
 }
 else if(document.all)
  {ding.Stop();
   setTimeout("ding.Run()",100);
 }
 }
```



```
        overMe=1;
    }
    else overMe=0;
    writeOnText(idObj, "<span class="+spObj+"><font color="+clr1+">" +tmp0+"</font><font color="+clr2
        +">" +tmp1+"</font></span>");
    setTimeout("txtTyper('"+str+"', '"+idx+"', '"+idObj+"', '"+spObj+"', '"+clr1+"',
        '"+clr2+"', '"+delay+"', '"+plysnd+"')",delay);}}
function init() //初始化
{txtTyper(disStr[0], 0, 'td0', 'td1', '#339933', '#99FF33', 300, 0);
}
</script>
</body>
</html>
```

代码键入完毕，存盘为“实例 15.html”。

二、运行程序

双击文件“实例 15.html”。

实例 16 抖动的文字

实例说明

本例运行效果如图 16-1 所示。这是一个文字产生抖动效果的例子。

程序运行后,会看到一组文字产生抖动的效果,一段时间后这种效果会停止,再过一段时间后文字又会重新抖动。

本例的知识点有:CSS 中基本的文本属性和定位属性,以及 JavaScript 脚本语言知识。



图 16-1 效果图

编程思路

本实例使用文本属性定义文字的样式,用定位属性对文字进行定位,使用 JavaScript 脚本语言控制文字的抖动。

创作步骤

一、创建脚本代码

```

var y=80
var x=50
var count_shakes=0
var shake
function initObjects()
{shake_0=document.all.shake0.style
shake_1=document.all.shake1.style
shake_2=document.all.shake2.style
shake_3=document.all.shake3.style
shake()
}
function getrandom()
{var dif=Math.floor(Math.random()*2)
return dif

```



```

    }
    function getRandom2()
    { var dif= Math.floor(Math.random()*5)
      return dif
    }
    function getRandom3()
    { var dif= Math.floor(Math.random()*10)
      return dif
    }
    function intermezzo()
    { count_shakes=0
      var goon=setTimeout('shake()',2000)
    }
    function shake() //抖动函数
    {if(count_shakes<50)
      {shake_0.posTop=y+getRandom();
       shake_0.posLeft=x+getRandom3();
       shake_1.posTop=y+getRandom();
       shake_1.posLeft=x+getRandom2();
       shake_2.posTop=y+getRandom2();
       shake_2.posLeft=x+getRandom();
       shake_3.posTop=y+getRandom();
       shake_3.posLeft=x+getRandom3();
       var timer = setTimeout('shake()',10)
       count_shakes++;
      }
      else
      {shake_0.top=y; shake_0.left=x;
       shake_1.top=y; shake_1.left=x;
       shake_2.top=y; shake_2.left=x;
       shake_3.top=y; shake_3.left=x;
       intermezzo()
       clearTimeout(timer)
      }
    }
  }
}

```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
```

```
<title>实例 16</title>
<head>
<style>
.ticker
{color:#ff0000;
font family:Verdana;
font size:30px;
font-weight:bold;
letter spacing:1px;
position:absolute
}
.main
{position:absolute;
left:10px;
top:80px
}
</style>
<script language=JavaScript>
<!--
键入以上脚本代码。
//-->
</script>
</head>
<body onload=initObjects()>
<div class=ticker id=shake0 style="color: #ffeccc">Hello world!
</div>
<div class=ticker id=shake1 style="color: #ffcccc">Hello world!
</div>
<div class=ticker id=shake2 style="color: #ff9999">Hello world!
</div>
<div class=ticker id=shake3 style="color: #ff7777">Hello world!
</div>
<div class=main id=main></div>
</body>
</html>
```

代码键入完毕，存盘为“实例 16.html”。

三、运行程序

双击文件“实例 16.html”。

实例 17 七彩链接

实例说明

本例运行效果如图 17-1 所示。这是一个改变前景颜色的例子。

程序运行后,将鼠标置于链接文字之上会看到文字前景色不断地在变化。

本例的知识点有:CSS 中基本的颜色属性,以及 JavaScript 脚本语言知识。



图 17-1 效果图

编程思路

本实例使用 JavaScript 脚本语言控制颜色的改变。

创作步骤

一、创建源代码

打开记事本,键入如下代码。

```
<html>
<head>
<title>实例 17</title>
<script language="JavaScript">
<!--
var newcolor=new Array
("FF0000","EE0011","DD0022","CC0033","BB0044","AA0055","990066","880077","770088","660099","
5500AA","4400BB","3300CC","2200DD","1100EE","0000FF","0000FF","0000FF","0011EE","0022DD","
0033CC","0044BB","0055AA","006699","007788","008877","009966","00AA55","00BB44","00CC33","0
0DD22","00EE11","00FF00","00FF00","00FF00","00FF00","11EE00","22DD00","33CC00","44BB00","55
AA00","669900","778800","887700","996600","AA5500","BB4400","CC3300","DD2200","EE1100","FF0
000","FF0000","FF0000")
var delay=30
var timer
```

```
var flag=true
var thislink
var i_color=0
function start(newlink)
{if(document.all)
  {flag=true
  thislink=eval("document.all."+newlink+".style")
  changecolors()
  }
}
function changecolors()
{if(flag)
  {if(i_color>=newcolor.length-1)
  {i_color=0
  }
  thislink.color=newcolor[i_color]
  i_color++
  timer=setTimeout("changecolors()",delay)
  }
else
  {clearTimeout(timer)
  }
}
function stop()
{if(document.all)
  {flag=false
  }
}
-->
</script>
</head>
<body bgcolor=000000 text=888888 link=FFFFFF vlink=FFFFFF alink=FFFFFF>
<a name="link1" onMouseOver="start(this.name)" onMouseOut="stop()" href="#">七彩链接</a>
</body>
</html>
```

代码键入完毕，存盘为“实例 17.html”。

二、运行程序

双击文件“实例 17.html”。

实例 18 总是不动的文本

实例说明

本例运行效果如图 18-1 所示。这是一个文本总出现在屏幕顶端的例子。

程序运行后,在浏览器上会显示出一段文字,拖动滚动条,文字一直在浏览器顶端不会移动。

本例的知识点有:CSS 中的文本属性,以及 JavaScript 脚本语言知识。



图 18-1 效果图

编程思路

本实例使用文本属性定义文字的样式,使用 JavaScript 脚本语言控制文字,使之总是出现在屏幕顶端。

创作步骤

一、创建脚本代码

```
function KB_keepItInIE(theName,theWantTop,theWantLeft)
{theRealTop=parseInt(document.body.scrollTop)
  theTrueTop=theWantTop+theRealTop
  document.all[theName].style.top=theTrueTop
  theRealLeft=parseInt(document.body.scrollLeft)
  theTrueLeft=theWantLeft+theRealLeft
  document.all[theName].style.left=theTrueLeft
}
function KB_keepItInNN(theName,theWantX,theWantY)
{theRealLay=document.layers[theName]
  theBadX=self.pageYOffset
  theBadY=self.pageXOffset
  theRealX=theBadX+theWantX
```

```
theRealY=theBadY+theWantY
theRealLay.moveTo(theRealY,theRealX)
}
IE4=(document.all)?1:0
NN4=(document.layers)?1:0
if(IE4)
    setInterval('KB_keepItInIE("KBStatic",0,0)',1)
if(NN4)
    setInterval('KB_keepItInNN("KBStatic",0,0)',1)
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 18</title>
<head>
<script language=JavaScript>
键入以上脚本代码。
</script>
</head>
<body>
<div id="KBStatic" style="position:absolute; left:3px; top:0px; width:553px; height:70px; z-index:25">
<font size="5">我不会动!</font>
</div>
</body>
</html>
```

代码键入完毕，存盘为“实例 18.html”。

三、运行程序

双击文件“实例 18.html”。

实例 19 文字的特殊显示

实例说明

本例运行效果如图 19-1 所示。这是一个文字从顶端斜向下弹出的例子。

程序运行后,单击按钮会看到一段文字从屏幕顶端斜向下弹出。

本例的知识点有: CSS 中的基本属性,以及 JavaScript 脚本语言知识。



图 19-1 效果图

编程思路

本实例使用 CSS 的基本属性对文字进行初始定位和定义样式,使用 JavaScript 脚本语言,控制文字的弹出。

创作步骤

一、创建脚本代码

```
function movableObj(startX, startY, endX, endY, delay, speed, refId)
{
    this.sX = startX;this.sY = startY;this.eX = endX;
    this.eY = endY;this.de = delay;this.sp = speed;
    this.ref = refId;
    xL = endX - startX; yL = endY - startY;
    with (Math)
    {
        if(abs(xL) > abs(yL))
        {
            this.xS = (xL > 0)?1:-1; this.yS = (yL > 0)?abs(yL / xL):-abs(yL / xL);
            this.howManySteps = abs(xL / speed);
        }
        else if(abs(yL) > abs(xL))
        {
            this.yS = (yL > 0)?1:-1; this.xS = (xL > 0)?abs(xL / yL):-abs(xL / yL);
        }
    }
}
```

```

        this.howManySteps = abs(yL / speed);
    }
    else
    {this.yS = (yL > 0)?1:-1; this.xS = (xL > 0)?1:-1;
    this.howManySteps = abs(xL / speed);
    }
}
this.startMovement = startMovement;
}
function startMovement()
{ref = document.all(this.ref).style;
doDynamicMovement(this.sX, this.sY, this.eX, this.eY, this.de, this.xS, this.yS, ref, this.sp,
this.howManySteps);
}
function doDynamicMovement(curX, curY, eX, eY, sp, xS, yS, ref, speed, hS) //移动
{if(Math.floor(hS - 1) != 0)
{hS--;curX += xS * speed; curY += yS * speed;
ref.left = Math.round(curX); ref.top = Math.round(curY);
setTimeout("doDynamicMovement(" + curX + ", " + curY + ", " + eX + ", " + eY + ", " + sp + ", " + xS +
" + yS + ", ref, " + speed + ", " + hS + ")", sp);
}
else
{setPos(eX, eY, ref);
}
}
function setPos(x, y, ref)
{ref.left = x; ref.top = y;
}

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<head>
<title>实例 19</title>
<script language="JavaScript">
<!--
键入以上脚本代码。
// -->
</script>

```



```
</head>
<body>
<table>
<tr>
<td><font face="Verdana, Arial"><h3>&nbsp;</h3>
<font size="2">
<form>
<div align="center">
<center><p>
<input type="button" onClick="dynaText.startMovement()" value="按下去看看"> </p>
</center></div>
</form>
<div id="wow" style="position: absolute; left: -100; top: -100; width: 300;
font family: Verdana, Arial; font size: 20pt">
<p>Hello World!</font></p>
</div></td>
</tr>
</table>
<script language="JavaScript">
<!--
dynaText = new movableObj(-100,-100,80,180,10,10,"wow");
// -->
</script>
</body>
</html>
```

代码键入完毕，存盘为“实例 19.html”。

三、运行程序

双击文件“实例 19.html”。

实例 20 文字的动态显示

实例说明

本例运行效果如图 20-1 所示。这是一个文字动态循环显示的例子。

程序运行后，会看到文字从小到大出现，然后从大到小消失，再换另一组文字重复显示。

本例的知识点有：CSS 的基本属性，以及 JavaScript 脚本语言知识。

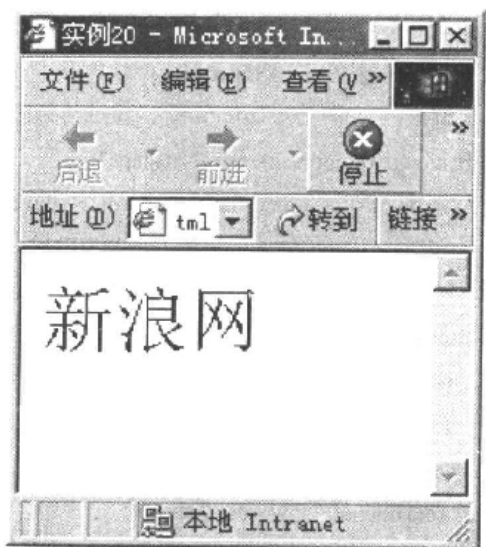


图 20-1 效果图

编程思路

本实例使用 JavaScript 脚本语言，动态改变 font size 属性，并且控制文字循环显示。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 20</title>
<body>
<script language=JavaScript>
<!--
var speed = 20; var delay = 1000; var maxsize = 48;
var x = 0; var y = 0;
var message, size;
var esize = "";
function initArray() //初始化
```

```

(this.length = initArray.arguments.length;
 for (var i = 0; i < this.length; i++)
 {this[i] = initArray.arguments[i];
 }
}
var message2 = new initArray
("新浪网", "网易", "263 信箱", "鲨威体坛"); //信息数组
if(navigator.appName == "Netscape")
' document.write('<layer id="wds"></layer><br>');
if(navigator.appVersion.indexOf("MSIE") != -1)
document.write('<span id="wds"></span><br>');
function upwords()
{message = message2[y];
 if (x < maxsize)
 {x++;
  setTimeout("upwords()",speed);
 }
 else setTimeout("downwords()",delay);
 if(navigator.appName == "Netscape")
 {size = "<font point-size="+x+"pt>";
 document.wds.document.write(size+message+esize);
 document.wds.document.close();
 }
 if (navigator.appVersion.indexOf("MSIE") != -1)
 {wds.innerHTML = message;
  wds.style.fontSize=x+'px'
 }
}
function downwords()
{if (x > 1)
 {x--;
  setTimeout("downwords()",speed);
 }
 else
 {setTimeout("upwords()",delay);
  y++;
  if (y > message2.length - 1) y = 0;
 }
 if(navigator.appName == "Netscape")

```

```
[size = "<font point-size="+x+"pt">";
document.wds.document.write(size+message+size);
document.wds.document.close();
}
if (navigator.appVersion.indexOf("MSIE") != -1)
{wds.innerHTML =message;
wds.style.fontSize=x+'px'
}
}
setTimeout("upwords()",speed);
//-->
</script>
</body>
</html>
```

代码键入完毕，存盘为“实例 20.html”。

二、运行程序

双击文件“实例 20.html”。

实例 21 逐个闪烁的文字

实例说明

本例运行效果如图 21-1 所示。这是一个文字逐个闪烁的例子。

程序运行后, 会看到文字从左到右逐个改变颜色以达到闪烁的效果。

本例的知识点有: CSS 的基本属性, 以及 JavaScript 脚本语言知识。



图 21-1 效果图

编程思路

本实例使用 JavaScript 脚本语言控制文字的 Color 属性动态地变化。

创作步骤

一、创建源代码

打开记事本, 键入如下代码:

```
<html>
<title>实例 21</title>
<body>
<script language="JavaScript">
<!--
text="逐个闪烁的文字";
color1="blue";
color2="red";
```

```
fontSize="6";
speed=100;
i=0;
if(navigator.appName == "Netscape")
{document.write("<layer id=a visibility=show></layer><br><br><br>");
}
else
{document.write("<div id=a></div>");
}
function changeCharColor() //改变颜色函数
{if(navigator.appName == "Netscape")
{document.a.document.write("<font face=arial size=" + fontsize + "><font color=" + color1 +
">");
for (var j = 0; j < text.length; j++)
{if(j == i)
{document.a.document.write("<font face=arial color=" + color2 + ">" + Text.charAt(i) + "</font>");
}
else
{document.a.document.write(text.charAt(j));
}
}
document.a.document.write("</font></font>");
document.a.document.close();
}
if (navigator.appName == "Microsoft Internet Explorer")
{str = "<font face=arial size=" + fontsize + "><font color=" + color1 + ">";
for (var j = 0; j < text.length; j++)
{if(j == i)
{str += "<font face=arial color=" + color2 + ">" + text.charAt(i) + "</font>";
}
else
{str += text.charAt(j);
}
}
str += "</font></font>";
a.innerHTML = str;
}
(i == text.length) ? i=0 : i++;
}
```



```
setInterval("changeCharColor()", speed);  
// -->  
</script>  
</body>  
</html>
```

代码键入完毕，存盘为“实例 21.html”。

二、运行程序

双击文件“实例 21.html”。

实例 22 自动输出的文本

实例说明

本例运行效果如图 22-1 所示。这是一个在文本框中自动输出文本的例子。

程序运行后，在文本框中逐个输出文字。

本例的知识点有：CSS 中的基本属性，以及 JavaScript 脚本语言知识。

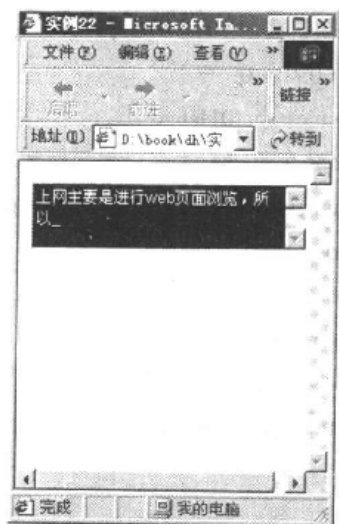


图 22-1 效果图

编程思路

本实例使用 CSS 的基本属性定义文字的样式，使用 JavaScript 脚本语言控制文本的自动输出。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 22</title>
<body bgcolor="#1ef4d9" OnLoad="textticker()">
<script language="JavaScript">
<!--
var max=0;
function textlist()
{max=textlist.arguments.length;
for(i=0;i<max;i++)
    this[i]=textlist.arguments[i];
}
```



```
message=new textlist(
```

```
" 随着 Internet 时代的到来,人们对网络的认识与感知越来越深刻",
```

```
"上网主要是进行 web 页面浏览,所以 web 页面的精彩程度是一个网站能否生存的关键所在","人通过制作个人主页展示自己的才华,而且,由于出现了多种制作网页的软件","但是,光用软件就能制作出你想要的各种页面效果吗?答案是否定的。");
```

```
var x=0;pos=0;
```

```
var l=message[0].length;
```

```
function textticker() //输出函数
```

```
{document.tickform.tickfield.value=message[x].substring(0, pos) + "_";
```

```
if(pos++==l)
```

```
{pos=0;
```

```
setTimeout("textticker()", 400);
```

```
if(++x==max) x=0;
```

```
l=message[x].length;
```

```
}
```

```
else
```

```
setTimeout("textticker()", 200);
```

```
}
```

```
//-->
```

```
</script>
```

```
<form name=tickform>
```

```
<textarea name=tickfield rows=3 cols=38 style="background-color: rgb(0,0,0); color: rgb(255,255,255); cursor: default; font family: Arial; font size: 12px" wrap=virtual>The news will appear here when the page has finished loading.</textarea>
```

```
</form>
```

```
</body>
```

```
</html>
```

代码键入完毕,存盘为“实例 22.html”。

二、运行程序

双击文件“实例 22.html”。

实例 23 弹簧卷轴

实例说明

本例运行效果如图 23-1 所示。这是一个文字呈现卷轴的例子。

程序运行后，会看到一段文字以卷轴式出现和消失。

本例的知识点有：CSS 中的基本属性，以及 JavaScript 脚本语言知识。



图 23-1 效果图

编程思路

本实例使用基本属性对文字进行初始化，使用 JavaScript 脚本语言控制文字的卷轴效果。

创作步骤

一、创建脚本代码

```
var message= new Array()
message[0]="在社会主义改革开放和现代化建设的新时期，在跨越世纪的新征途上，一定要高举邓小平理论的伟大旗帜，用邓小平理论来指导我们整个事业和各项工作。"
message[1]="这是党从历史和现实中得出的不可动摇的结论。"
var scrollerwidth=300
var scrollerleft=10
var scrollerleft=10
var targetlink="_blank"
var font_face="宋体"
var font_color="006600"
var font_size=15
var standstill=3000
var lineheightmax=12
var lineheight=lineheightmax
var linestep=0.2
```

```
var i_message=0
var mes_joined
var mes_split
var contenttext
var pause=20
function initiate()
{if(document.all)
  {document.all.scroller.style.posTop=scroller.top
  document.all.scroller.style.posLeft=scroller.left
  document.all.scroller.style.width=scroller.width
  lineup()
  }
}
function lineup()
{mes_joined=message[i_message]
mes_split=mes_joined.split("")
contenttext="<span style='position:relative;font family:"+font_face+";
color:"+font_color+";font size:"+font_size+"pt;line-height:"+lineheight+";width:"+scrollerwidth+"px'"
contenttext+=mes_split[0]
contenttext+="</span>"
if(lineheight>1.2)
{scroller.innerHTML=contenttext
var timer=setTimeout("lineup()",pause)
lineheight-=linestep
}
else
{mes_joined=message[i_message]
mes_split=mes_joined.split("")
contenttext="<span style='position:relative;font family:"+font_face+";
color:"+font_color+";font size:"+font_size+"pt;line-height:"+lineheight+";width:"+scrollerwidth+"px'"
contenttext+=mes_split[0]
contenttext+="<br><a href="+mes_split[1]+"target="+targetlink+"></a>"
contenttext+="</span>"
scroller.innerHTML=contenttext
clearTimeout(timer)
var timer=setTimeout("squeeze()",standstill)
}
}
function squeeze()
```

```

{mes_joined=message[i_message]
mes_split=mes_joined.split("")
contenttext="<span style=position:relative;font family:"+font_face+";
color:"+font_color+";font size:"+font_size+"pt;line-height:"+lineheight+";width:"+scrollerwidth+"px">
contenttext+=mes_split[0]
contenttext+="</span>"
if(lineheight>0)
{scrollertext.innerHTML=contenttext
var timer=setTimeout("squeeze()",pause)
lineheight-=linestep/2
}
else
{clearTimeout(timer)
scrollertext.innerHTML=""
changemessage()
}
}
function changemessage()
{i_message++
if(i_message>message.length-1)
{i_message=0}
lineheight=lineheightmax
var timer=setTimeout("lineup()",1000)
}

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 23</title>
<body bgcolor="#fef4d9" onload=initiate()>
<script language=JavaScript1.2>
键入以上脚本代码。
</script>
<div id=scrollertext style="position:absolute"></div>
</body></html>

```

代码键入完毕，存盘为“实例 23.html”。

三、运行程序

双击文件“实例 23.html”。

实例 24 文字变幻特效

实例说明

本例运行效果如图 24-1 所示。这是一个动态改变文本及框架样式的例子。

程序运行后, 会看到文本以及框架不停地变幻。

本例的知识点有: CSS 中的基本属性, 以及 JavaScript 脚本语言知识。

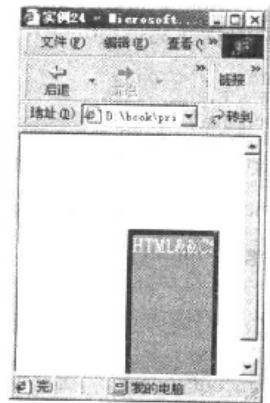


图 24-1 效果图

编程思路

本实例使用基本的文本及边框属性定义初始的样式, 使用 JavaScript 脚本语言动态改变样式。

创作步骤

一、创建脚本代码

```
function morph(e,wait,steps,style.done)
{if(e.morphing)
    return;
    e.morphing=true;
    e.step=0;
    e.done=steps;
    e.wait=wait;
    e.morphDone=done;
    var attribs=_eat_attrib(style);
    if(attribs["top"]!=null)
    {e.deltaTop=Math.round(Number((attribs["top"])-e.style.posTop)/steps);
        e.finalTop=attribs["top"];
    }
    else
    {e.deltaTop=null;
```

```
e.finalTop=e.style.posTop;
}
if(attrs["left"]!=null)
{e.deltaLeft=Math.round(Number((attrs["left"]-e.style.posLeft)/steps));
 e.finalLeft=attrs["left"];
}
else
{e.deltaLeft=null;
 e.finalLeft=e.style.posLeft;
}
if(attrs["height"]!=null)
{e.deltaHeight=Math.round(Number((attrs["height"]-e.style.posHeight)/steps));
 e.finalHeight=attrs["height"];
}
else
{e.deltaHeight=null;
 e.finalHeight=e.style.posHeight;
}
if(attrs["width"]!=null)
{e.deltaWidth=Math.round(Number((attrs["width"]-e.style.posWidth)/steps));
 e.finalWidth=attrs["width"];
}
else
{e.deltaWidth=null;
 e.finalWidth=e.style.posWidth;
}
e.colors=new Array();
for(attr in attrs)
{if(attr=="background")
{e.colors[e.colors.length]=new Array();
 e.colors[e.colors.length-1].name=attr;
 e.colors[e.colors.length-1].safename=attr;
}
if(attr=="color")
{e.colors[e.colors.length]=new Array();
 e.colors[e.colors.length-1].name=attr;
 e.colors[e.colors.length-1].safename=attr;
}
if(attr=="border color")
```

```
{e.colors[e.colors.length]=new Array();
  e.colors[e.colors.length-1].name=attrib;
  e.colors[e.colors.length-1].safename="borderColor";
}
}
for(var i=0;i<e.colors.length;i++)
{var rgb=_eat_rgb(attribs[e.colors[i].name]);
  var prrgb=_eat_rgb(e.style[e.colors[i].safename]);
  e.colors[i].delta=_eat_rgb(attribs[e.colors[i].name]);
  e.colors[i].finish=_eat_rgb(attribs[e.colors[i].name]);
  e.colors[i].delta["red"]=Math.round(Number((rgb["red"]-prrgb["red"])/steps));
  e.colors[i].delta["green"]=Math.round(Number((rgb["green"]-prrgb["green"])/steps));
  e.colors[i].delta["blue"]=Math.round(Number((rgb["blue"]-prrgb["blue"])/steps));
}
setTimeout("_morphing(document.all."+e.id+"")",wait);
}
function _morphing(e)
{e.step++;
  if(e.deltaTop!=null)
    e.style.posTop+=e.deltaTop;
  if(e.deltaLeft!=null)
    e.style.posLeft+=e.deltaLeft;
  if(e.deltaHeight!=null)
    e.style.posHeight+=e.deltaHeight;
  if(e.deltaWidth!=null)
    e.style.posWidth+=e.deltaWidth;
  for(var i=0;i<e.colors.length;i++)
  {var rgb=_eat_rgb(e.style[e.colors[i].safename]);
    e.style[e.colors[i].safename]="rgb"+"(e.colors[i].delta["red"]+rgb["red"])+","+(e.colors[i].delta["green"]
      +rgb["green"])+","+(e.colors[i].delta["blue"]+rgb["blue"]) +)";
  }
  if(e.step==e.done)
  {if(e.deltaTop!=null)
    e.style.posTop=e.finalTop;
  if(e.deltaLeft!=null)
    e.style.posLeft=e.finalLeft;
  if(e.deltaWidth!=null)
    e.style.posWidth=e.finalWidth;
  if(e.deltaHeight!=null)
```

```
e.style.posHeight=e.finalHeight;
for(var i=0;i<e.colors.length;i++)
[e.style[e.colors[i].safename]="rgb("+e.colors[i].finish["red"]+rgb["red"])+","+
(c.colors[i].finish["green"]+rgb["green"])+","+e.colors[i].finish["blue"]+rgb["blue"]
+");
}
e.morphing=false;
eval(e.morphDone);
}
else
{setTimeout("_morphing(document.all."+e.id+"")",e.wait);
}
return;
}
function _eat_attr(str)
{var chunks=new Array();
var all=new Array();
chunks=str.split(";");
for(var i=0;i<chunks.length;i++)
{var tmpA=new Array();
tmpA=chunks[i].split(":");
all[tmpA[0]]=tmpA[1];
}
return all;
}
function _eat_rgb(str)
{var all=new Array();
var a=str.indexOf("(");
var b=str.indexOf(")");
str=str.substring(a+1,b);
var tmpA=str.split(",");
all["red"]=Number(tmpA[0]);
all["green"]=Number(tmpA[1]);
all["blue"]=Number(tmpA[2]);
return all;
}
}
```

二、创建源代码

打开记事本，键入如下代码：


```
<html>
<title>实例 24</title>
<head>
<script>
键入以上脚本代码。
</script>
</head>
<body>
<span id="test" style="border:solid;border width:5;border color:rgb(0,0,0);color:rgb(255,255,255);
position:absolute;top:0;left:0;width:400;height:400;overflow:hidden;background:rgb(0,0,255);">
<a>HTML&&CSS</a>
</span>
<script>
function rander()
{ var top=Math.round(Math.random()*400);
var left=Math.round(Math.random()*600);
var width=Math.round(Math.random()*200);
var height=Math.round(Math.random()*200);
var r=Math.round(Math.random()*255);
var g=Math.round(Math.random()*255);
var b=Math.round(Math.random()*255);
var background="rgb("+r+","+g+","+b+")";
var r=Math.round(Math.random()*255);
var g=Math.round(Math.random()*255);
var b=Math.round(Math.random()*255);
var borderColor="rgb("+r+","+g+","+b+")";
var r=Math.round(Math.random()*255);
var g=Math.round(Math.random()*255);
var b=Math.round(Math.random()*255);
var color="rgb("+r+","+g+","+b+")";
morph(document.all.test,30,20,"top:"+top+"left:"+left+"width:"+width+"height:"+height+"background:"
+background+":border color:"+borderColor+":color:"+color+";","rander()");
}
rander();
</script></body></html>
```

代码键入完毕，存盘为“实例 24.html”。

三、运行程序

双击文件“实例 24.html”。

实例 25 跑马灯式样的提示信息

实例说明

本例运行效果如图 25-1 所示。这是一个跑马灯式的提示信息的例子。

程序运行后，将鼠标置于链接文字之上会出现跑马灯式的提示信息。

本例的知识点有：CSS 中的基本属性，以及 JavaScript 脚本语言知识。



图 25-1 效果图

编程思路

本实例使用基本属性定义文字的样式，使用 JavaScript 脚本语言控制产生跑马灯效果。

创作步骤

一、创建脚本代码

```

if(!document.layers&&!document.all)
    event="test"
function show(current,e,text)
{if(document.all&&document.readyState=="complete")
{document.all.tool.innerHTML+'<marquee style="border:1px solid black">'+text+'</marquee>'
document.all.tool.style.pixelLeft=event.clientX+document.body.scrollLeft+10
document.all.tool.style.pixelTop=event.clientY+document.body.scrollTop+10
document.all.tool.style.visibility="visible"
}
}
else if(document.layers)
{document.tool.document.nstip.document.write('<b>'+text+'</b>')}

```

```
document.tool.document.nstip.document.close()
document.tool.document.nstip.left=0
currentscroll=setInterval("scroll()",100)
document.tool.left=e.pageX+10
document.tool.top=e.pageY+10
document.tool.visibility="show"
    }
}
function hide()
{if(document.all)
    document.all.tool.style.visibility="hidden"
else if(document.layers)
    {clearInterval(currentscroll)
    document.tool.visibility="hidden"
    }
}
function scroll() //滚动函数
{if(document.tool.document.nstip.left>=-document.tool.document.nstip.document.width)
    document.tool.document.nstip.left-=5
else
    document.tool.document.nstip.left=150
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 25</title>
<body>
<div id="tool" style="position:absolute;visibility:hidden;clip:rect(0 150 50 0);
width:150px;background-color:seashell">
<layer name="nstip" width=1000px bgColor="seashell"></layer>
</div>
<script language="JavaScript">
<!--
键入以上脚本代码。
//-->
</script>
<table border="0">
<tr>
```

```
<td><font face="Arial, Helvetica, sans-serif" size="3">  
<a href="http://www.sina.com.cn" onMouseOver="show(this,event,'http://www.sina.com.cn')"  
  onMouseOut="hide()">新浪网</a></font></td>  
<td><font face="Arial, Helvetica, sans-serif" size="3">  
<a href="http://www.263.net" onMouseOver="show(this,event,'http://www.263.net')"  
  onMouseOut="hide()">首都在线</a></font></td>  
</font>  
</td>  
</tr>  
</table>  
</body>  
</html>
```

代码键入完毕，存盘为“实例 25.html”。

三、运行程序

双击文件“实例 25.html”。

实例 26 文字慢慢出现

实例说明

本例运行效果如图 26-1 所示。这是一个文字以不同形式慢慢出现的例子。

程序运行后，四组文字会以四种不同的形式循环显示。

本例的知识点有：CSS 中的基本属性，以及 JavaScript 脚本语言知识。



图 26-1 效果图

编程思路

本实例使用基本属性定义文本的样式，使用 JavaScript 脚本语言控制文字的出现形式。

创作步骤

一、创建脚本代码

```
growWait=90
expandWait=120
scrollWait=100
scrollWidth=40
lineMax=4
lines=new Array(lineMax)
lines[1]=new Line("综合性站点---网易", "http://www.163.net", Expand, 2000)
lines[2]=new Line("新闻发部站点---新浪", "http://www.sina.com.cn", Scroll, 1000)
lines[3]=new Line("搜索站点---搜狐", "http://www.sohu.com", Static, 2000)
lines[4]=new Line("体育论坛---鲨鱼", "http://www.sawci.com", Grow, 2000)
```

```
lineText=""
timerID=null
timerRunning=false
spaces=""
charNo=0
charMax=0
charMiddle=0
lineNo=0
lineWait=0
function Line(text, url, type, wait)
{ this.text=text
  this.url=url
  this.Display=type
  this.wait=wait
}
function StringFill(c, n)
{s=""
  while (--n >= 0)
  { s+=c
  }
  return s
}
function Static()
{ document.formDisplay.buttonFace.value=this.text
  timerID=setTimeout("ShowNextLine()", this.wait)
}
function Grow()
{ lineText=this.text
  lineWait=this.wait
  charMax=lineText.length
  TextGrow()
}
function TextGrow()
{ if (charNo <= charMax)
  { document.formDisplay.buttonFace.value=lineText.substring(0, charNo)
    charNo++
    timerID=setTimeout("TextGrow()", growWait)
  }
  else
```

```
{charNo=0
  timerID=setTimeout("ShowNextLine()", lineWait)
}
}
function Expand()
{lineText=this.text
  charMax=lineText.length
  charMiddle=Math.round(charMax / 2)
  lineWait=this.wait
  TextExpand()
}
function TextExpand()
{if (charNo <= charMiddle)
  {document.formDisplay.buttonFace.value=lineText.substring(charMiddle - charNo, charMiddle + charNo)
  charNo++
  timerID=setTimeout("TextExpand()", expandWait)
}
else
  {charNo=0
  timerID=setTimeout("ShowNextLine()", lineWait)
}
}
function Scroll()
{spaces=StringFill(" ", scrollWidth)
  lineText=spaces+this.text
  charMax=lineText.length
  lineText+=spaces
  lineWait=this.wait
  TextScroll()
}
function TextScroll()
{if (charNo <= charMax)
  {document.formDisplay.buttonFace.value=lineText.substring(charNo, scrollWidth+charNo)
  charNo++
  timerID=setTimeout("TextScroll()", scrollWait)
}
else
  {charNo=0
  timerID=setTimeout("ShowNextLine()", lineWait)
```

```

    }
}
function StartHeadliner()
{StopHeadliner()
  timerID=setTimeout("ShowNextLine()", 1000)
  timerRunning=true
}
function StopHeadliner()
{if (timerRunning)
  {clearTimeout(timerID)
  timerRunning=false
}
}
function ShowNextLine()
{(lineNo < lineMax) ? lineNo++ : lineNo=1
  lines[lineNo].Display()
}
function GotoUrl(url)
{top.location.href=url
}

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 26</title>
<head>
<style>
.stHeadliner
{font-family:lucida console,courier new,monospace;
font-size:11pt;
font-weight:bold;
font-style:italic;
background:blue;
color:white
}
</style>
<script language="JavaScript">
<!--

```

键入以上脚本代码。


```
//-->
</script>
</head>
<body bgcolor="#fef4d9" onLoad="StartHeadliner()">
<form name="formDisplay">
<input class="stHeadLiner" type="button" name="buttonFace"
value="请等待一秒钟!" onClick="GotoUrl(lines[lineNo].url)"></input>
</form>
</body>
</html>
```

代码键入完毕，存盘为“实例 26.html”。

三、运行程序

双击文件“实例 26.html”。

实例 27 消息公告板

实例说明

本例运行效果如图 27-1 所示。这是一个显示消息公告板的例子。

程序运行后，先弹出一个消息公告板，一段时间后该公告板会马上消失。

本例的知识点有：CSS 中的定位属性，以及 JavaScript 脚本语言知识。



图 27-1 效果图

编程思路

本实例使用定位属性定义公告板的位置、宽度和大小，使用 JavaScript 脚本语言控制公告板的出现和消失。

创作步骤

一、创建脚本代码

```
var flyingwin
var popupwidth=200
var popupheight=150
var marginright
var windowcenter
var i_top=200
var i_left=popupwidth-50
var step=40
var timer
var waitingtime=5000
var pause=20
function showWindow()
{flyingwin=window.open(,"flyingwin","toolbar=no,width="+popupwidth+",height="+popupheight+",top
-100,left="+(-popupwidth)+"");
flyingwin.document.open();
```

```

flyingwin.document.write("<html><title>消息公告板</title><body ><p align=center><br><font size='+1'
    color=red><b>Hi!</b></font><br><br>我马上就会消失. </p></body></html>");
flyingwin.document.close();
if(document.all)
{marginright = screen.width+50
}
if(document.layers)
{marginright = screen.width+50
}
windowcenter=Math.floor(marginright/2)-Math.floor(popupwidth/2)
movewindow()
}
function movewindow()
{if(i_left<=windowcenter)
{flyingwin.moveTo(i_left,i_top)
i_left+=step
timer= setTimeout("movewindow()",pause)
}
else
{clearTimeout(timer)
timer= setTimeout("movewindow2()",waitingtime)
}
}
function movewindow2()
{if(i_left<=marginright)
{flyingwin.moveTo(i_left,i_top)
i_left+=step
timer= setTimeout("movewindow2()",pause)
}
else
{clearTimeout(timer)
flyingwin.close()
}
}
}

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 27</title>

```

```
<head>  
<script>  
<!--  
键入以上脚本代码。  
//-->  
</script>  
</head>  
<body onload=showWindow(>  
</body>  
</html>
```

代码键入完毕，存盘为“实例 27.html”。

三、运行程序

双击文件“实例 27.html”。

实例 28 链接窗口说明

实例说明

本例运行效果如图 28-1 所示。这是一个点击链接、弹出窗口说明的例子。

程序运行后,单击任意一个链接会弹出一个窗口,单击窗口中“Jump to!”按钮,会链接到相应的网页。

本例的知识点有:CSS 中的基本属性,以及 JavaScript 脚本语言知识。

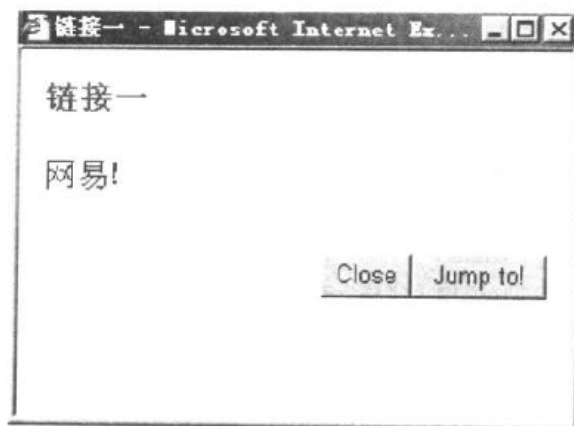


图 28-1 效果图

编程思路

本实例使用基本属性对弹出窗口的样式进行定义,使用 JavaScript 脚本语言控制窗口的弹出和消失。

创作步骤

一、创建脚本代码

```
function PopIt(label, msg, URLE)
{var DQUOTE = "\"
var s1 = "<TITLE>' + label + "</TITLE>" +
"<BODY BGCOLOR=#ffffff><TABLE BORDER=0><TR>"+
"<TD WIDTH=90% HEIGHT=90 VALIGN=TOP ALIGN=LEFT>"+
"<FONT SIZE=4>"
var s2 = "<FONT COLOR=#FF0000><B>"+label+"</B></FONT><P>"
var s3 = "</TD><TD WIDTH=10%> </TD></TR><TR><TD> </TD>"+
"<TD VALIGN=TOP ALIGN=RIGHT>"+
"<FORM><INPUT TYPE='BUTTON' VALUE='Close'" +
"onClick='self.close()'>" +
"<INPUT TYPE='BUTTON' VALUE='Jump to!'"+
"onClick='window.open(\"+DQUOTE+URLE+DQUOTE+\")'" +
"</FORM></TD></TR></TABLE></BODY>"
```

```
popup = window.open("", "popDialog", "height=200,width=300,scrollbars=no")
popup.document.write(s1+s2+msg+s3)
popup.document.close()
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<body>
<script language="JavaScript">
<!--
键入以上脚本代码。
//-->
</script>
<a href="JavaScript:PopIt('链接一','网易!','http://www.163.com')">网易</a>
<br>
<a href="JavaScript:PopIt('链接二','搜狐!','http://www.sohu.com.cn')">搜狐</a>
<br>
<a href="JavaScript:PopIt('链接三','中央电视台!','http://www.cctv.com')">中央电视台</a>
<br>
<a href="JavaScript:PopIt('链接四','新浪!','http://www.sina.com.cn')">新浪</a>
</body>
</html>
```

代码键入完毕，存盘为“实例 28.html”。

三、运行程序

双击文件“实例 28.html”。

第二篇

CSS 导航效果篇

本篇预览

本篇主要应用 CSS 中用于定义边框的一些属性，进行应用程序的开发。本篇将制作一些导航的例子。

通过本篇中各个实例的制作，读者可以对 CSS 中与边框有关的属性有一个大概的了解。导航菜单在网页制作中起着比较重要的作用，导航菜单有利于浏览者的浏览。

实例 29 动态菜单

实例说明

本例运行效果如图 29-1 所示。这是一个显示动态菜单的实例。

程序运行后，将鼠标置于主菜单上，会自动显示出副菜单。

本例的知识点有：text-align、background-color、font-size 等多种属性以及 JavaScript 脚本语言。

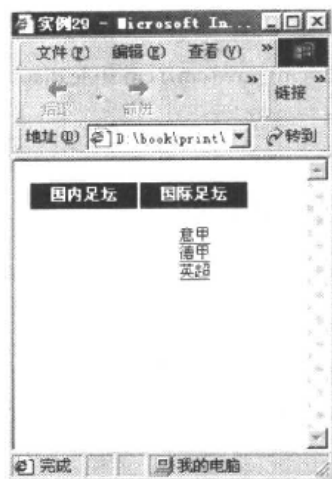


图 29-1 效果图

编程思路

本实例首先定义了两个类：一是 Menu；二是 SubMenu。Menu 定义了显示在菜单项上的样式，而 SubMenu 定义了下拉菜单列的显示样式。

其中，菜单的显示和隐藏是通过响应鼠标事件来控制的，这是通过一段 JavaScript 程序实现的。

创作步骤

一、创建脚本代码

```
var cm=null;
document.onclick = new Function("Show(null)");
function GetPost(el,sProp) {
var Pos = 0;
while (el!=null) {
Pos+=el["offset" + sProp];
el = el.offsetParent;
}
return Pos;
}
function Show(e,m) {
```



```

if (m) {
m.style.display="";
m.style.pixelLeft = GetPos(el,"Left")
m.style.pixelTop = GetPos(el,"Top") + el.offsetHeight
}
if ((m!=cm) && (cm)) cm.style.display='none'
cm=m
}

```

二、创建源代码

```

<html>
<head>
<title>实例 29</title>
<style>
<!--
.menu
{background-color:black;
width:80; height:20;
color: white;
text-align: center;
font-size:9pt;
font-weight:bold;}
.submenu
{color:black;
position:absolute;
top:40;
background-color:lightyellow; width:83;
font-size:9pt;
}
-->
</style>
</head>
<body>
<script>
键入以上脚本代码。
</script>
<table border=0>
<tbody>

```

```
<tr>
<td class=menu id=d1 onmouseover=Show(this,ds1); style="cursor:s-resize">国内足坛</td>
<td class=menu id=d2 onmouseover=Show(this,ds2); style="cursor:s-resize">国际足坛</td>
</tr>
</tbody>
</table>
<div class=submenu id=ds1 style="display: none"><br>
<a href="www">甲 A 联赛</a> <br>
<a href="www">甲 B 联赛</a> <br>
<a href="www">足协杯</a> <br>
</div>
<div class=submenu id=ds2 style="display: none"><br>
<a href="www">意甲</a><br>
<a href="www">德甲</a><br>
<a href="www">英超</a><br>
</div>
</body>
</html>
```

键入完毕，存盘为“实例 29.html”。

三、运行程序

双击文件“实例 29.html”。

实例 30 推拉式菜单

实例说明

本例运行效果如图 30-1 所示。这是一个推拉式菜单的例子。

程序运行后,将鼠标置于浏览器上突出的菜单上,菜单就会自动弹出。

本例的知识点有:CSS 中的边框属性和定位属性,以及 JavaScript 脚本语言。



图 30-1 效果图

编程思路

本实例主要应用边框属性来给菜单绘制边框,使用 JavaScript 脚本语言控制菜单的弹出。

创作步骤

一、创建脚本代码

```
function menu(n)
{if(n=1)
{clearTimeout(out_ID)
if(menu.left>menuW*-1+20+10)
{menu.left-=8
in_ID=setTimeout("menuIn()",1)
}
else if(menu.left<menuW*-1+20)
{menu.left++
in_ID = setTimeout("menuIn()",1)
}
}
else
```

```
[clearTimeout(out_ID)
if(menu.pixelleft>menuW*-1+20+10)
{menu.pixelleft-=4
in_ID=setTimeout("menuIn()",1)
}
else if(menu.pixelleft>menuW*-1+20)
{menu.pixelleft--
in_ID=setTimeout("menuIn()",1)
}
}
function menuOut()
{if(n4)
{clearTimeout(in_ID)
if( menu.left<-10)
{menu.left+=4
out_ID=setTimeout("menuOut()",1)
}
else if(menu.left<0)
{menu.left++
out_ID=setTimeout("menuOut()",1)
}
}
else
{clearTimeout(in_ID)
if(menu.pixelleft<-10)
{menu.pixelleft+=2
out_ID=setTimeout("menuOut()",1)
}
else if(menu.pixelleft<0)
{menu.pixelleft++
out_ID=setTimeout("menuOut()",1)
}
}
}
function fireOver()
{clearTimeout(F_out)
F_over=setTimeout("menuOut()",10)
}
```

```

function fireOut()
{clearTimeout(F_over)
  F_out=setTimeout("menuIn()",10)
}
function init()
{if(n4)
  {menu = document.D1
  menuW = menu.document.width
  menu.left = menuW*-1+20
  document.d1.onmouseover = menuOut
  document.d1.onmouseout = menuIn
  menu.visibility = "visible"
  }
else if(c4)
  {menu=d1.style
  menuW=d1.offsetWidth
  d1.style.pixelleft=menuW*-1+20
  d1.onmouseover=fireOver
  d1.onmouseout=fireOut
  d1.style.visibility="visible"
  }
}
F_over=F_out=in_ID=out_ID=null
n4=(document.layers)?1:0
c4=(document.all)?1:0;

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 30</title>
<head>
<style>
#d1
{border-bottom:white 2px outset;
border-left:white 2px outset;
border-right:white 2px outset;
border-top:white 2px outset;
left:0px;
position:absolute;

```

```
top:70px;
visibility:hidden;
width:200px;
}
</style>
<script language="javascript">
键入以上脚本代码。
</script>
</head>
<body onload="init()">
<div id="d1">
<table border="0">
<tr>
<td align="middle">
<b style="color:red">友情链接</b>
</td>
</tr>
<tr>
<td>
<ul>
<li><a href="http://www.hongen.com">http://www.hongen.com</a></li>
<li><a href="http://www.chinaren.com.cn">http://www.chinaren.com.cn</a></li>
<li><a href="http://www.sina.com.cn">http://www.sina.com.cn</a></li>
</ul>
</td>
</tr>
</table>
</div>
</body>
</html>
```

代码键入完毕，存盘为“实例 30.html”。

三、运行程序

双击文件“实例 30.html”。

实例 31 弹出菜单

实例说明

本例运行效果如图 31-1 所示。这是一个可弹出的菜单。

程序运行后，单击菜单项会弹出菜单。

本例的知识点有：CSS 中基本的文本属性，以及 JavaScript 脚本语言知识。

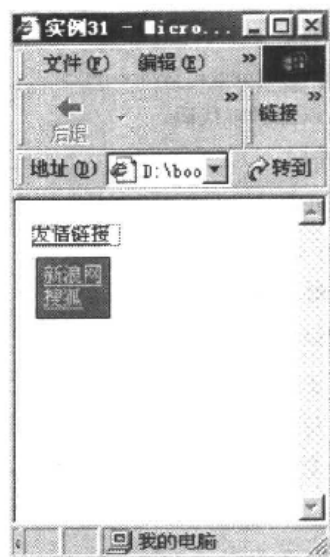


图 31-1 效果图

编程思路

本实例使用 CSS 中的文本属性定义菜单中文字的样式，使用 JavaScript 脚本语言控制菜单的弹出。

创作步骤

一、创建脚本代码

```
var zIndex=100
function dropit2(whichone)
{if(window.themenu&&themenu.id!=whichone.id)
    themenu.style.visibility="hidden"
  themenu=whichone
  if(document.all)
  {themenu.style.left=document.body.scrollLeft+event.clientX-event.offsetX
    themenu.style.top=document.body.scrollTop+event.clientY-event.offsetY+18
    if(themenu.style.visibility=="hidden")
    {themenu.style.visibility="visible"
      themenu.style.zIndex=zIndex++
```

```
    }  
    else  
    {hidemenu()  
    }  
  }  
}  
function dropit(e,whichone)  
{if(window.themenu&&themenu.id!=eval(whichone).id)  
    themenu.visibility="hide"  
    themenu=eval(whichone)  
    if(themenu.visibility=="hide")  
        themenu.visibility="show"  
    else  
        themenu.visibility="hide"  
    themenu.zIndex++  
    themenu.left=e.pageX-e.layerX  
    themenu.top=e.pageY-e.layerY+19  
    return false  
}  
function hidemenu(whichone)  
{if(window.themenu)  
    themenu.style.visibility="hidden"  
}  
function hidemenu2()  
{themenu.visibility="hide"  
}  
if(document.all)  
    document.onclick=hidemenu
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>  
<title>实例 31</title>  
<head>  
<style>  
.iewrap1  
{position:relative;  
height:30px;  
font-family:"宋体";
```



```
font-size:9pt
}
.iewrap2
{position:absolute;
font-family:"宋体";
font-size:9pt
}
#menu1
{visibility:hide;
z-index:100;
font-family:"宋体";
font-size: 9pt
}
</style>
<body>
<script language="JavaScript">
键入以上脚本代码。
</script>
<layer height=35px>
<layer visibility=show>
<span class=iewrap1>
<span class=iewrap2 onClick="dropit2(menu1),event.cancelBubble=true;return false">
<font face=宋体>
<a href=" " onClick="if(document.layers) return dropit(event,'document.menu1')">友情链接
</a>
</font>
</span>
</span>
</layer>
</layer>
<br>
<div id=menu1 style="position:absolute;left:0;top:0;
layer-background-color:#408080;background-color:#408080;
visibility:hidden;border:1px solid black;padding:0px">
<script language="JavaScript">
var menu=new Array()
menu[0]='<a href="http://www.sina.com.cn"><font color="#ffcc00" face="Arial">新浪网</font></a><br>'
menu[1]='<a href="http://www.sohu.com.cn"><font color="#ffcc00" face="Arial">搜狐</font></a><br>'
if (document.all)
```

```
menu1.style.padding="4px"  
for (i=0;i<menu.length;i++)  
document.write(menu[i])  
</script>  
</body>  
</html>
```

代码键入完毕，存盘为“实例 31.html”。

三、运行程序

双击文件“实例 31.html”。

实例 32 自定义右键菜单

实例说明

本例运行效果如图 32-1 所示。这是一个自定义右键菜单的例子。

程序运行后，在浏览器中的任意位置，单击鼠标右键会弹出白定义的右键菜单。

本例的知识点有：CSS 中的文字属性和边框属性，以及 JavaScript 脚本语言知识。

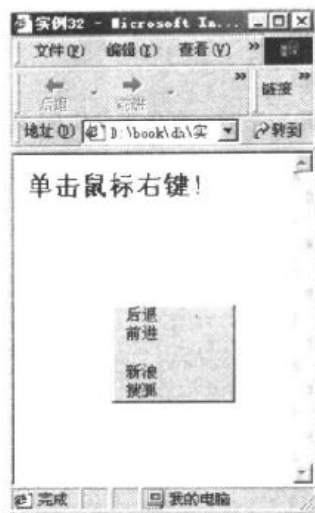


图 32-1 效果图

编程思路

本实例使用文本属性定义菜单中文字的样式，使用边框属性定义菜单的框架，使用 JavaScript 脚本语言控制菜单的显示和隐藏。

创作步骤

一、创建脚本代码

```
var menusk1n="skin";
var display_url=0;
function showmenuie5()
{var rightedge=document.body.clientWidth-event.clientX;
var bottomedge=document.body.clientHeight-event.clientY;
if(rightedge<ie5menu.offsetWidth)
    ie5menu.style.left=document.body.scrollLeft+event.clientX-ie5menu.offsetWidth;
else
    ie5menu.style.left=document.body.scrollLeft+event.clientX;
if(bottomedge<ie5menu.offsetHeight)
    ie5menu.style.top=document.body.scrollTop+event.clientY-ie5menu.offsetHeight;
```

```
else
    ie5menu.style.top=document.body.scrollTop+event.clientY;
ie5menu.style.visibility="visible";
return false;
}
function hidemenuie5()
{ ie5menu.style.visibility="hidden";
}
function highlightie5()
{if(event.srcElement.className=="menuitems")
    {event.srcElement.style.backgroundColor="highlight";
    event.srcElement.style.color="white";
    if(display_url)
        window.status=event.srcElement.url;
    }
}
function lowlightie5()
{if(event.srcElement.className=="menuitems")
    {event.srcElement.style.backgroundColor="";
    event.srcElement.style.color="black";
    window.status="";
    }
}
function jumptoie5()
{if(event.srcElement.className=="menuitems")
    {if(event.srcElement.getAttribute("target")!=null)
        window.open(event.srcElement.url, event.srcElement.getAttribute("target"));
    else
        window.location=event.srcElement.url;
    }
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 32</title>
<head>
<style>
<!--
```

```

.skin
{
  cursor:hand;
  font:menutext;
  position:absolute;
  text-align:left;
  font-family:Arial,Helvetica,sans-serif;
  font-size: 10pt;
  width:100px;
  background-color:menu;
  border:1 solid buttonface;
  visibility:hidden;
  border:2 outset buttonhighlight;
}
.menuitems
{
  padding-left:10px;
  padding-right:10px;
}
-->
</style>
<script language="JavaScript 1.2" >
<!--
键入以上脚本代码。
-->
</script>
</head>
<body>
<h3>单击鼠标右键!</h3>
<br><br>
<div id="ie5menu" class="skin" onMouseover="highlightie5()"
  onMouseout="lowlightie5()" onClick="jumptoeie5();">
  <div class="menuitems" url="javascript:history.back();">后退</div>
  <div class="menuitems" url="javascript:history.forward();">前进</div>
  <div class="menuitems" url="http://www.sina.com.cn">新浪</div>
  <div class="menuitems" url="http://www.sohu.com.cn">搜狐</div>
</div>
<script language="JavaScript 1.2">
if(document.all&&window.print)
{ie5menu.className=menuskii;
  document.oncontextmenu=showmenuie5;

```

```
document.body.onclick=hidemenuie5;  
}  
</script>  
</body>  
</html>
```

代码键入完毕，存盘为“实例 32.html”。

三、运行程序

双击文件“实例 32.html”。

实例 33 循环的导航按钮

实例说明

本例运行效果如图 33-1 所示。这是一个可循环的按钮导航的例子。

程序运行后,在浏览器中有几个导航按钮循环显示。

本例的知识点有:CSS 中的基本属性以及 JavaScript 脚本语言知识。



图 33-1 效果图

编程思路

本实例是用鼠标属性定义鼠标的指针,使用 JavaScript 脚本语言控制导航按钮的循环。

创作步骤

一、创建脚本代码

```
var WhereTo=null;
var TimerID=null;
var now=new Date();
var pos=0;
var StartTime=null;
var PauseTime=1500;
function InitializeArray()
{this.length=InitializeArray.arguments.length
  for(var i=0;i<this.length;i++)
  {this[i+1]=InitializeArray.arguments[i]
  }
}
```

```
}  
function ParseString(data,num)  
{for(var i=0;i<data.length;i++)  
  {if(data.substring(i,i+1)=="|")  
    break;  
  }  
  if(num==0)  
    return(data.substring(0,i));  
  else return(data.substring(i+1,data.length));  
}  
function StartTimer()  
{now=new Date();  
  StartTime=now.getTime();  
  StopTimer();  
  WhereTo=new InitializeArray("新浪主页http://www.sina.com.cn",  
    "搜狐主页http://www.sohu.com.cn",  
    "ChinaRen 主页http://www.chinaren.edu.cn");  
  ShowTimer();  
}  
function StopTimer()  
{TimerID=null;  
  WhereTo=null;  
}  
function ShowTimer()  
{pos=(pos==WhereTo.length)?1:pos+1;  
  document.forms[0].elements[0].value=ParseString(WhereTo[pos],0);  
  TimerID=window.setTimeout('ShowTimer()',PauseTime);  
}  
function DestinationUrl()  
{this.location=ParseString(WhereTo[pos],1);  
  return (false);  
}  
}
```

二、建立程序的基本框架

打开记事本，键入如下代码：

```
<html>  
<title>实例 33</title>  
<head>  
<script language=JavaScript1.2>
```



```
<!--  
键入以上脚本代码。  
</script>  
<body onload=window.StartTimer(>  
<form>  
<input name=Destination onclick=window.DestinationUrl() type=button style="cursor:hand;color:red;" >  
</form>  
</body>  
</html>
```

代码键入完毕，存盘为“实例 33.html”。

三、运行程序

双击文件“实例 33.html”。

实例 34 可拖动的菜单

实例说明

本例运行效果如图 34-1 所示。这是个可以拖动的菜单。

程序运行后，可以用鼠标拖动隐藏的菜单。

本例的知识点有：CSS 的边框属性以及 JavaScript 脚本语言知识。

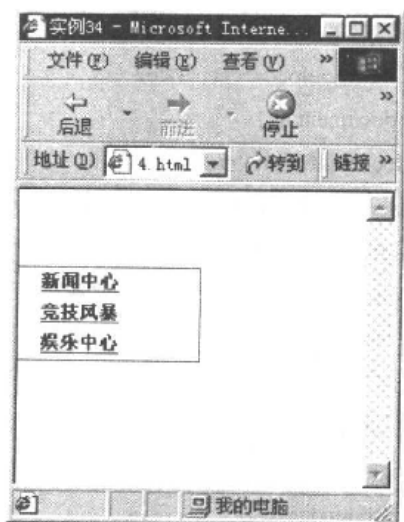


图 34-1 效果图

编程思路

本实例使用边框属性定义菜单的框架，使用 JavaScript 脚本语言控制菜单的拖动。

创作步骤

一、创建脚本代码

```

if(document.all)
document.write('<div id="menu2" style="left:-150">')
var items=new Array()
items[0]="新闻中心"
items[1]="竞技风暴"
items[2]="娱乐中心"
document.write('<center>');
for (i=0;i<=items.length-1;i++)
document.write('<a href=" ../index.htm">'+items[i]+'</a><br>');
document.write('</center>');
function regenerate()
{window.location.reload()}

```

```

    }
    function regenerate2()
    {if(document.layers)
        setTimeout("window.onresize=regenerate",400)
    }
    window.onload=regenerate2
    if(document.all)
    {document.write('</div>')
        themenu=document.all.menu2.style
        rightboundary=0
        leftboundary=-150
    }
    else
    {themenu=document.layers.menu1
        rightboundary=150
        leftboundary=10
    }
    var dragapproved=false
    var x1,x2
    if(document.layers)
        themenu.captureEvents(Event.MOUSEDOWN | Event.MOUSEUP)
    function move(e)
    {if(document.all&&event.button==1&&dragapproved)
        {if(temp1+event.clientX-x1>leftboundary)
            themenu.pixelleft=Math.min(rightboundary,temp1+event.clientX-x1)
        else
            themenu.pixelleft=Math.max(leftboundary,temp1+event.clientX-x1)
        }
    else if(document.layers)
        {if(themenu.left+e.x-x2>leftboundary)
            themenu.left=Math.min(rightboundary+2,themenu.left+e.x-x2)
        else
            themenu.left=Math.max(leftboundary-2,themenu.left+e.x-x2)
        }
    }
    function drags(e)
    {if(document.all)
        {dragapproved=true
            temp1=themenu.pixelleft

```

```

    x1=event.clientX
    document.onmousemove=move
  }
  else
  {thenu.captureEvents(Event.MOUSEMOVE)
    x2=e.x
  }
}
function stopit()
{if(document.all)
  {dragapproved=false
  }
  if(document.layers)
    thenu.releaseEvents(Event.MOUSEMOVE)
}
if (document.all)
{document.all.menu2.onmousedown=drags
  document.all.menu2.onmouseup=new Function("dragapproved=false")
  document.onselectstart=new Function("return false")
}
else if(document.layers)
  {thenu.onmousedown=drags
  thenu.onmousemove=move
  thenu.onmouseup=stopit
  }
}

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 34</title>
<head>
<style>
<!--
#menu1
{background-color:lightyellow;
border-bottom:green 1px solid;
border-left:green 1px solid;
border-right:green 1px solid;
border-top:green 1px solid;

```



```
cursor:hand;  
font:bold 12px/20px Verdana;  
left:-155px;  
position:absolute;  
top:50px;  
width:160px;  
layer-background-color: lightyellow
```

```
}
```

```
#menu2
```

```
{background-color:lightyellow;  
border-bottom:green 1px solid;  
border-left:green 1px solid;  
border-right:green 1px solid;  
border-top:green 1px solid;  
cursor:hand;  
font:bold 12px/20px Verdana;  
left:-155px;  
position:absolute;  
top:50px;  
width:160px;  
layer-background-color: lightyellow
```

```
-->
```

```
</style>
```

```
<script language=JavaScript1.2>
```

键入以上脚本代码。

```
</script>
```

```
</html>
```

代码键入完毕，存盘为“实例 34.html”。

三、运行程序

双击文件“实例 34.html”。

实例 35 图片翻滚导航

实例说明

本例运行效果如图 35-1 所示。这是一个两张图片来回翻滚的例子。

程序运行后，会有两张图片不停地翻滚，单击图片会链接到相应的位置。

本例的知识点有：CSS 中的定位属性，以及 JavaScript 脚本语言知识。

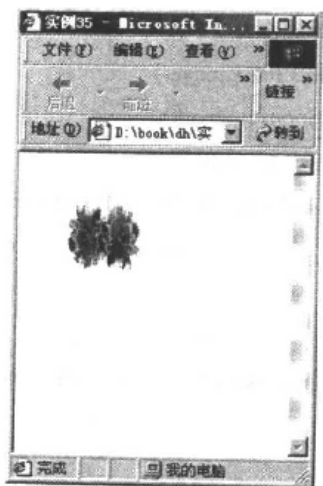


图 35-1 效果图

编程思路

本实例使用 CSS 的定位属性对图片进行初始定位，使用 JavaScript 脚本语言控制图片的翻滚。

创作步骤

一、创建脚本代码

```

var imgwidth=400
var imgheight=60
var imgname=new Array()
imgname[0]="Flow06.gif"
imgname[1]="Flow08.gif"
var imgurl=new Array()
imgurl[0]="#"
imgurl[1]="#"
var imgpreload=new Array()
for (i=0;i<=imgname.length-1;i++)
{imgpreload[i]=new Image()
imgpreload[i].src=imgname[i]
}

```

```

var pause=200
var speed=20
var step=10
var i_loop=0
var i_image=0
function stretchimage()
{if (i_loop<=imgwidth)
  {if (document.all)
    {imgcontainer.innerHTML="<a href="+imgurl[i_image]+" target='_blank'></a>"
    }
    i_loop=i_loop+step
    var timer=setTimeout("stretchimage()",speed)
  }
  else
  {i_loop=imgwidth
    clearTimeout(timer)
    imgcontainer.innerHTML="<a href="+imgurl[i_image]+" target='_blank'></a>"
    var timer=setTimeout("shrinkimage()",pause)
  }
}
function shrinkimage()
{if(i_loop>=0)
  {if(document.all)
    {imgcontainer.innerHTML="<a href="+imgurl[i_image]+" target='_blank'></a>"
    }
    i_loop=i_loop-step
    var timer=setTimeout("shrinkimage()",speed)
  }
  else
  {i_loop=0
    clearTimeout(timer)
    changeimage()
  }
}
function changeimage()
{i_image++

```

```

if(i_image>=imgname.length)
{i_image=0;
}
if(document.layers)
{document.imgcontainer.document.write("<a href='"+imgurl[i_image]+' target='_blank'>
<img src='"+imgname[i_image]+' border='0'></a>")
document.imgcontainer.document.close()
}
stretchimage()
}

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 35</title>
<head>
<script>
<!--
键入以上脚本代码。
//-->
</script>
<style>
.containerstyle
{position:absolute;
left:40px;
top:40px;
}
</style>
</head>
<body onload=stretchimage()>
<span id="imgcontainer" class="containerstyle">
</span>
</body>
</html>

```

代码键入完毕，存盘为“实例 35.html”。

三、运行程序

双击文件“实例 35.html”。

实例 36 可用鼠标驱动菜单

实例说明

本例运行效果如图 36-1 所示。这是一个可以用鼠标驱动隐藏的菜单的例子。

程序运行后，将鼠标置于“>>>”上，会使链接滚动显示出后面的链接。

本例的知识点有：CSS 中的定位属性，以及 JavaScript 脚本语言知识。



图 36-1 效果图

编程思路

本实例使用 CSS 中的定位属性对菜单进行定位，使用 JavaScript 脚本语言控制菜单的显示和隐藏。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 36</title>
<head>
<style>
body{
overflow-x:hidden;
overflow-y:scroll;
}
</style>
</head>
<body>
```

```
<script language="JavaScript 1.2">
var menuwidth=300
var scrollspeed=6
var menucontents='<noabr><a href="http://www.sina.com.cn">新浪主页</a> |
    <a href="http://www.263.net">263 主页</a> | <a href="http://www.sohu.com.cn">搜狐主页</a> |
    <a href="http://www.easthome.net">东方网景</a></noabr>'
var actualwidth=""
var ns_scroll
function fillup()
{if(document.all)
    {test2.innerHTML=menucontents
    actualwidth=test2.offsetWidth
    }
else if(document.layers)
    { ns_scroll=document.ns_scrollmenu.document.ns_scrollmenu2
    ns_scroll.document.write(menucontents)
    ns_scroll.document.close()
    actualwidth=ns_scroll.document.width
    }
}
window.onload=fillup
function moveleft()
{if(document.all&&test2.style.pixelleft>(menuwidth-actualwidth))
    test2.style.pixelleft-=scrollspeed
else if(document.layers&&ns_scroll.left>(menuwidth-actualwidth))
    ns_scroll.left-=scrollspeed
lefttime=setTimeout("moveleft()",50)
}
function moveright()
{if(document.all&&test2.style.pixelleft<0)
    test2.style.pixelleft+=scrollspeed
else if(document.layers&&ns_scroll.left<0)
    ns_scroll.left+=scrollspeed
righttime=setTimeout("moveright()",50)
}
if(document.all||document.layers)
{with(document)
    {write('<table border="0" cellspacing="0" cellpadding="0">')
    write('<td valign="middle"><a href="#" onmouseover="moveleft()')

```

```

        onMouseout="clearTimeout(lefttime)"><<<</a> </td>')
write('<td valign="top">')
if(document.all)
{ write('<span style="position:relative;width:'+menuwidth+';">')
write('<span style="position:absolute;width:'+menuwidth+';clip:rect(0 '+menuwidth+' auto 0)">')
write('<span id="test2" style="position:absolute;left:0;top:0">')
write('</span></span></span>')
}
else if(document.layers)
{ write('<ilayer width='+menuwidth+' name="ns_scrollmenu">')
write('<layer name="ns_scrollmenu2" left=0 top=0></layer></ilayer>')
}
write('</td>')
write('<td valign="middle"> <a href="#" onMouseover="moveright()"
onMouseout="clearTimeout(righttime)">')
write('>>></a>')
write('</td></table>')
}
}
</script>
</body>
</html>

```

代码键入完毕，存盘为“实例 36.html”。

二、运行程序

双击文件“实例 36.html”。

实例 37 可层叠的菜单

实例说明

本例运行效果如图 37-1 所示。这是一个可层叠的菜单的例子。

程序运行后，会出现两个菜单项，单击“友情链接”菜单项，弹出的菜单将盖住下面的菜单项。

本例的知识点有：CSS 中的一些基本的文本、颜色和边框属性，以及 JavaScript 脚本语言知识。

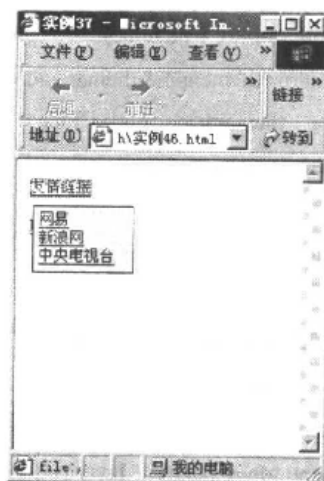


图 37-1 效果图

编程思路

本实例使用 CSS 的基本属性定义菜单的框架，使用 JavaScript 脚本语言控制菜单的显示和隐藏。

创作步骤

一、创建脚本代码

```
var zIndex=100
function dropit2(whichone)
{if(window.themenu&&themenu.id!=whichone.id)
    themenu.style.visibility="hidden"
    themenu=waichone
    if(document.all)
    {themenu.style.left=document.body.scrollLeft+event.clientX-event.offsetX
    themenu.style.top=document.body.scrollTop+event.clientY-event.offsetY+18
    if(themenu.style.visibility=="hidden")
    {themenu.style.visibility="visible"
    themenu.style.zIndex=zIndex++
    }
    else
```

```

        {hidemenu()
        }
    }
}

function dropit(e,whichone)
{if(window.themenu&&themenu.id!=eval(whichone).id)
    themenu.visibility="hide"
    themenu=eval(whichone)
    if (themenu.visibility=="hide")
        themenu.visibility="show"
    else
        themenu.visibility="hide"
    themenu.zIndex++
    themenu.left=e.pageX-e.layerX
    themenu.top=e.pageY-e.layerY+19
    return false
}
function hidemenu(whichone)
{if (window.themenu)
    themenu.style.visibility="hidden"
}
function hidemenu2()
{themenu.visibility="hide"
}
if (document.all)
    document.body.onclick=hidemenu
</script>
<layer height=35px>
<layer visibility=show>
<span class=style1 >
<span class=style2 onClick="dropit2(dropmenu1);event.cancelBubble=true;return false">
<font face=宋体>
<a href="#" onClick="if(document.layers) return dropit(event, 'document.dropmenu1')">友情链接</a>
</font>
</span>
</span>
</layer>
</layer><br>
<layer height=35px>

```

```
<layer visibility=show>
<span class=style1>
<span class=style2 onClick="dropit2(dropmenu2);event.cancelBubble=true;return false">
<font face=宋体>
<a href="#" onClick="if(document.layers) return dropit(event, 'document.dropmenu2')">联络方式</a>
</font>
</span>
</span>
</layer>
</layer><br>
<div id=dropmenu1 style="position:absolute;left:0;top:0;layer-background-color:#CDECF5;
background-color:#CDECF5;width:80;visibility:hidden;border:1px solid black;padding:0px">
<script language="JavaScript1.2">
if (document.all)
dropmenu1.style.padding="4px"
for(i=0;i<menu1.length;i++)
document.write(menu1[i])
</script>
</div>
<script language="JavaScript1.2">
if (document.layers)
{ document.dropmenu1.captureEvents(Event.CLICK)
document.dropmenu1.onclick=hidemenu2
}
</script>
<div id=dropmenu2 style="position:absolute;left:0;top:0;layer-background-color:#CDECF5;
background-color:#CDECF5;width:80;visibility:hidden;border:1px solid black;padding:0px">
<script language="JavaScript1.2">
if (document.all)
dropmenu2.style.padding="4px"
for (i=0;i<menu2.length;i++)
document.write(menu2[i])
</script>
</div>
<script language="JavaScript1.2">
if (document.layers)
{ document.dropmenu2.captureEvents(Event.CLICK)
document.dropmenu2.onclick=hidemenu2
}
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 37</title>
<head>
<script language="JavaScript1.2">
var menu1=new Array()
menu1[0]='<a href=http://www.163.net>网易</a><br>'
menu1[1]='<a href=http://www.sina.com.cn>新浪网</a><br>'
menu1[2]='<a href=http://www.cctv.com>中央电视台</a><br>'
var menu2=new Array()
menu2[0]='<a href=#>论坛</a><br>'
menu2[1]='<a href=#>聊天室</a><br>'
menu2[2]='<a href=#>来信联络</a><br>'
</script>
<style>
<!--
style1
{position:relative;
height:30px;
font-family:"宋体";
font-size:9pt
}
.style2
{position:absolute;
font-family:"宋体";
font-size:9pt
}
#dropmenu1, #dropmenu2
{visibility:hide;
z-index:100;
font-family:"宋体";
font-size:9pt
}
-->
</style>
</head>
<body>
```

```
<script language="JavaScript1.2">
```

键入以上脚本代码。

```
</script>
```

```
</body>
```

```
</html>
```

代码键入完毕，存盘为“实例 37.html”。

三、运行程序

双击文件“实例 37.html”。

实例 38 带边框的菜单

实例说明

本例运行效果如图 38-1 所示。这是一个带边框的菜单的例子。

程序运行后,将鼠标置于菜单项上就会出现边框。

本例的知识点有:CSS 的边框属性,以及 JavaScript 脚本语言知识。

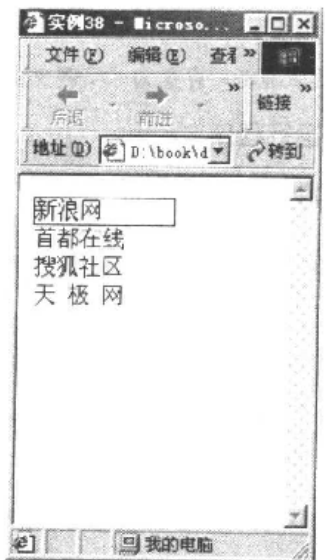


图 38-1 效果图

编程思路

本实例使用边框属性定义菜单边框的样式,使用 JavaScript 脚本语言控制边框的显示和隐藏。

创作步骤

一. 创建脚本代码

```
function borderize(what,color)
{what.style.borderColor=color
:
function borderize_on(e)
{if(document.all)
    source3=event.srcElement
else if(document.getElementById)
    source3=c.target
if(source3.className=="menulines")
{borderize(source3,"black")}
}
```

```

else
{while(source3.tagName!="TABLE")
{source3=document.getElementById? source3.parentNode : source3.parentElement
if(source3.className=="menulines")
borderize(source3,"black")
}
}
}
function borderize_off(e)
{if(document.all)
source4=event.srcElement
else if(document.getElementById)
source4=e.target
if(source4.className=="menulines")
borderize(source4,"white")
else
{while(source4.tagName!="TABLE")
{source4=document.getElementById? source4.parentNode : source4.parentElement
if(source4.className=="menulines")
borderize(source4,"white")
}
}
}

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 38</title>
<head>
<style>
.menulines
{border:1px solid white;
}
.menulines a
{text-decoration:none;
color:black;
}
</style>
<script language="JavaScript1.2">

```

键入以上脚本代码。

```

</script>
</head>
<body>
<table border="0" width="100" cellspacing="0" cellpadding="0"
onmouseover="borderize_on(event)" onmouseout="borderize_off(event)">
<tr><td width="100%" class="menulines">
<font face="Arial" size="2"><a href="http://www.sina.com.cn">新浪网</a></font></td></tr>
<tr><td width="100%" class="menulines">
<font face="Arial" size="2"><a href="http://www.263.net">首都在线</a></font></td></tr>
<tr><td width="100%" class="menulines">
<font face="Arial" size="2"><a href="http://www.sohu.com.cn">搜狐社区</a></font></td></tr>
<tr><td width="100%" class="menulines">
<font face="Arial" size="2"><a href="http://www.yesky.com/">天极网</a></font></td></tr>
</table>
</body>
</html>

```

代码键入完毕，存盘为“实例 38.html”。

三、运行程序

双击文件“实例 38.html”。

实例 39 隐藏的菜单

实例说明

本例运行效果如图 39-1 所示。这是一个隐藏的菜单的例子。

程序运行后,在浏览器任意位置上单击鼠标就会出现菜单。

本例的知识点有:CSS 的文本属性和边框属性,以及 JavaScript 脚本语言知识。



图 39-1 效果图

编程思路

本实例使用文本属性和边框属性定义菜单中文字和边框的样式,使用 JavaScript 脚本语言控制菜单的隐藏和显示。

创作步骤

一、创建源代码

打开记事本,键入如下代码:

```
<html>
<title>实例 39</title>
<body>
<table id="menustyle" border="1" cellspacing="1" bordercolor="#448888" style="display:none">
</tr>
<tr>
<td><a href="a"><font face="Arial">HTML</font></a></td>
</tr>
</tr>
```

```

<td><a href="b"><font face="Arial">Javascript</font></a></td>
</tr>
<tr>
<td><a href="c"><font face="Arial">Frontpage</font></a></td>
</tr>
<tr>
<td><a href="d"><font face="Arial">Photoshop</font></a></td>
</tr>
<tr>
<td><a href="e"><font face="Arial">Dreamweaver</font></a></td>
</tr>
</table>
</center>
</div>
<script language="JavaScript">
document.onclick=popUp
function popUp()
{newX=window.event.x+document.body.scrollLeft
newY=window.event.y+document.body.scrollTop
menu=document.all.menustyle
if(menu.style.display=="")
{menu.style.display="none"
}
else
{menu.style.display=""
}
menu.style.pixelLeft=newX-50
menu.style.pixelTop=newY-50
}
</script>
</body>
</html>
    
```

代码键入完毕，存盘为“实例 39.html”。

二、运行程序

双击文件“实例 39.html”。

实例 40 卷帘菜单

实例说明

本例运行效果如图 40-1 所示。这是一个呈卷帘状的菜单的例子。

程序运行后，单击菜单会看到菜单以卷帘式打开。

本例的知识点有：CSS 中的基本属性，以及 JavaScript 脚本语言知识。

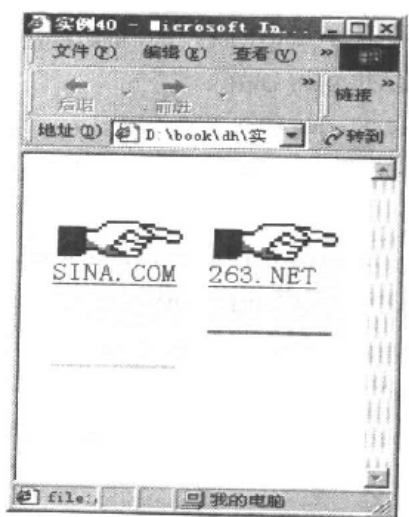


图 40-1 效果图

编程思路

本实例使用 CSS 的基本属性定义菜单的框架和文字的样式，使用 JavaScript 脚本语言控制菜单的下卷和上拉。

创作步骤

一、创建脚本代码

```
ie=document.all?1:0
n=document.layers?1:0
var numberOfMenus=1
var mwidth=90
var pxBetween=10
var fromLeft=20
var fromTop=50
var bgcolor="#CECFCE"
var bgcolorChangeTo="#6380BC"
var imageHeight=41
function makeNewsMenu(obj,nes1)
```

```

{nest=(!nest) ? "":document.'+nest+'
  this.css=(n) ? eval(nest+'document.'+obj);eval('document.all.'+obj+'.style')
  this.scrollHeight=this.css.document.height;eval('document.all.'+obj+'.offsetHeight')
  this.moveTo=b_moveIt;this.bgChange=b_bgChange;
  this.slideUp=b_slideUp; this.slideDown=b_slideDown;
  this.clipTo=b_clipTo;
  this.obj = obj + "Object";
  eval(this.obj + "=this")
}
function b_moveIt(x,y){this.x=x; this.y=y; this.css.left=this.x; this.css.top=this.y}
function b_bgChange(color){if(!ie) this.css.backgroundColor=color; else this.css.bgcolor=color}
function b_clipTo(t,r,b,l)
{if(n)
  {this.css.clip.top=t; this.css.clip.right=r; this.css.clip.bottom=b; this.css.clip.left=l
  }
  else this.css.clip="rect("+t+"," +r+"," +b+"," +l+)";
}
function b_slideUp(ystop,moveby,speed,fn,wh)
{if(!this.slideactive)
  {if(this.y>ystop)
    {this.moveTo(this.x,this.y-5); eval(wh)
    setTimeout(this.obj+'.slideUp("+ystop+", "+moveby+", "+speed+", "+fn+", "+wh+ "',speed)
    }
    else
    {this.slideactive=false; this.moveTo(0,ystop); eval(fn)
    }
  }
}
function b_slideDown(ystop,moveby,speed,fn,wh)
{if(!this.slideactive)
  {if(this.y<ystop)
    {this.moveTo(this.x,this.y+5); eval(wh)
    setTimeout(this.obj+'.slideDown("+ystop+", "+moveby+", "+speed+", "+fn+", "+wh+ "',speed)
    }
    else
    {this.slideactive=false; this.moveTo(0,ystop); eval(fn)
    }
  }
}
}

```

```

function newsMenuInit()
{oTopMenu=new Array()
for(i=0;i<=numberOfMenus;i++)
{oTopMenu[i]=new Array()
oTopMenu[i][0]=new makeNewsMenu('divTopMenu'+i)
oTopMenu[i][1]=new makeNewsMenu('divTopMenuBottom'+i,'divTopMenu'+i)
oTopMenu[i][2]=new makeNewsMenu('divTopMenuText'+i,'divTopMenu'+i)
oTopMenu[i][1].moveIt(0,imageHeight)
oTopMenu[i][0].clipTo(0,mwidth,imageHeight+3,0)
oTopMenu[i][0].moveIt(i*mwidth+fromLeft+(i*pxBetween),fromTop)
oTopMenu[i][0].css.visibility="visible"
}
}
function topMenu(num)
{if(oTopMenu[num][1].y==imageHeight)
oTopMenu[num][1].slideDown(oTopMenu[num][2].scrollHeight+20,10,40,'oTopMenu['+num+']')[0].
clipTo(0,mwidth,oTopMenu['+num+']][1].y+3,0)';oTopMenu['+num+']][0].clipTo(0,mwidth,oTopMc
nu['+num+']][1].y+3,0)')
else if(oTopMenu[num][1].y==oTopMenu[num][2].scrollHeight+20)
oTopMenu[num][1].slideUp(imageHeight,10,40,'oTopMenu['+num+']][0].clipTo(0,mwidth,oTopMen
u['+num+']][1].y+3,0)';oTopMenu['+num+']][0].clipTo(0,mwidth,oTopMenu['+num+']][1].y+3,0)')
}
function menuOver(num){oTopMenu[num][1].bgChange(bgcolorChangeTo)}
function menuOut(num){oTopMenu[num][1].bgChange(bgcolor)}
onload=newsMenuInit;

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 40</title>
<head>
<style>
a
{color:#000080;font-family:ms sans-serif;text-decoration:none
}
a:hover
{cursor:hand;color:#ff2200;font-family:ms sans-serif;text-decoration:none
}
div.clTopMenu

```



```

{z-index:31;visibility:hidden;width:80px;clip:rect(0px 80px 14px 0px);
  position:absolute;height:150px;background-color:#eeeeee;layer-background-color:#EEEEEE
}
div.clTopMenuBottom
{z-index:2;width:80px;clip:rect(0px 80px 3px 0px);
  position:absolute;top:21px;height:3px;background-color:#cecfce;layer-background-color:#CECFCE
}
</style>

```

```
<script>
```

键入以上脚本代码。

```
</script>
```

```
</head>
```

```
<body bgcolor="#FFFFFF" text="#000000">
```

```
<div class=clTopMenu id=divTopMenu0>
```

```
<a onmouseover=menuOver(0) onclick="topMenu(0); if(ie)this.blur(); return false"
```

```
  onmouseout=menuOut(0) href="#">
```

```
</a>
```

```
<div id=divTopMenuText0 style="width:75px;height:100px">
```

```
<a href="http://www.sina.com.cn">SINA.COM</a></div>
```

```
<div class=clTopMenuBottom id=divTopMenuBottom0></div></div>
```

```
<div class=clTopMenu id=divTopMenu1>
```

```
<a onmouseover=menuOver(1) onclick="topMenu(1); if(ie)this.blur(); return false"
```

```
  onmouseout=menuOut(1) href="#">
```

```
</a>
```

```
<div id=divTopMenuText1 style="width:75px;height:100px">
```

```
<a href="http://www.263.net">263.NET</a></div>
```

```
<div class=clTopMenuBottom id=divTopMenuBottom1></div></div>
```

```
</body>
```

```
</html>
```

代码键入完毕，存盘为“实例 40.html”。

三、运行程序

双击文件“实例 40.html”。

实例 41 下拉菜单

实例说明

本例运行效果如图 41-1 所示。这是一个产生下拉菜单的例子。

程序运行后，单击提示文字，菜单会以虚幻的形式慢慢出现。

本例的知识点有：CSS 中的 `revealTrans` 滤镜属性，以及 JavaScript 脚本语言知识。

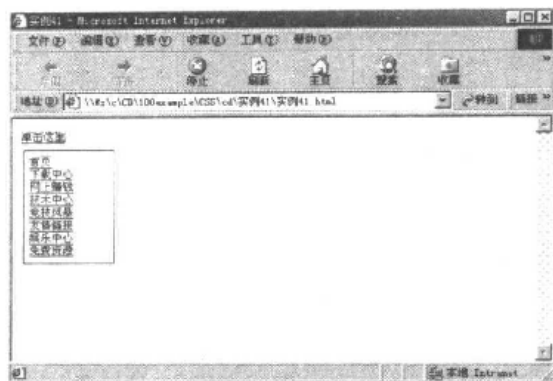


图 41-1 效果图

编程思路

本实例使用 CSS 的基本属性定义下拉菜单的样式，使用 JavaScript 脚本语言控制菜单的显示和隐藏。

创作步骤

一、创建脚本代码

```
var enableeffect=true
var selection=new Array()
selection[0]='<font face="宋体"><a href="a1.htm">首页</a></font><br>'
selection[1]='<font face="宋体"><a href="a2.htm">下载中心</a></font><br>'
selection[2]='<font face="宋体"><a href="a3.htm">网上赚钱</a></font><br>'
selection[3]='<font face="宋体"><a href="a4.htm">技术中心</a></font><br>'
selection[4]='<font face="宋体"><a href="a5.htm">竞技风暴</a></font><br>'
selection[5]='<font face="宋体"><a href="a6.htm">友情链接</a></font><br>'
selection[6]='<font face="宋体"><a href="a7.htm">娱乐中心</a></font><br>'
selection[7]='<font face="宋体"><a href="a8.htm">免费资源</a></font>'
function dropit2()
{if(document.all)
{menu03.style.left=document.body.scrollLeft+event.clientX-event.offsetX
menu03.style.top=document.body.scrollTop+event.clientY-event.offsetY+18
if(menu03.style.visibility=="hidden")
```

```

    {if(enableeffect)
        menu03.filters.revealTrans.apply()
    menu03.style.visibility="visible"
    if(enableeffect)
        menu03.filters.revealTrans.play()
    }
    else
    {hidemenu()
    }
}
}
function dropit(e)
{if(document.menu03.visibility=="hide")
    document.menu03.visibility="show"
else
    document.menu03.visibility="hide"
document.menu03.left=e.pageX-e.layerX
document.menu03.top=e.pageY-e.layerY+19
return false
}
function hidemenu()
{if(enableeffect)
    menu03.filters.revealTrans.stop()
    menu03.style.visibility="hidden"
}
function hidemenu2()
{document.menu03.visibility="hide"
}
if(document.all)
    document.body.onclick=hidemenu

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 41</title>
<head>
<style>
<!--
#style1

```

```

    {position:relative;
      height:30px
    }
#style2
{position:absolute
}
#menu03
{filter:revealTrans(Duration=1.5,Transition=12)
  visibility:hide
}
-->
</style>
</head>
<body>
<span id="style1">
<span id="style2" onClick="dropit2();event.cancelBubble=true;return false">
<font face="宋体"><a href="#">单击这里</a></font>
</span>
</span>
<script language=javascript>
<!--
键入以上脚本代码。
//..>
</script>
<div id="menu03" style="position:absolute;left:10;top:0;
  layer-background-color:seashell;background-color:seashell;width:100;visibility:hidden;border:1px solid
  black;padding:0px">
<script language="JavaScript1.2">
if (document.all)
menu03.style.padding='6px'
for (i=0;i<selection.length;i++)
document.write(selection[i])
</script>
</div>
</body>
</html>

```

代码键入完毕，存盘为“实例41.html”。

三、运行程序

双击文件“实例41.html”。

实例 42 树形菜单

实例说明

本例运行效果如图 42-1 所示。这是一个生成树形菜单的例子。

程序运行后，单击“全部菜单”，菜单会全部展开。

本例的知识点有：CSS 的基本属性、以及 JavaScript 脚本语言知识。

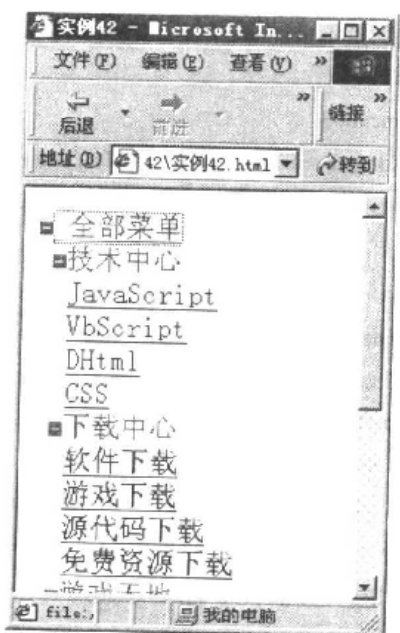


图 42-1 效果图

编程思路

本实例使用 CSS 的基本属性对菜单进行样式定义，使用 JavaScript 脚本语言控制菜单的展开和关闭。

创作步骤

一、创建脚本语言

```
var bV=parseInt(navigator.appVersion);
var NS4=(document.layers)?true:false;
var IE4=((document.all)&&(bV>=4))?true:false;
var ver4=(NS4||IE4)?true:false;
function expand1(){return}
function expandAll(){return}
function nomsg(){self.status="";}
if(ver4)
{document.write("<SCR"+"IPT LANGUAGE='JavaScript' SRC='(treemenu.js)'></SCR"+"IPT>");
```

```

}

```

以上函数请见 `treemenu.js`。

二、创建源代码

打开记事本、键入如下代码：

```

<html>
<title>实例 42</title>
<body>
<script language="JavaScript">
<!--
键入以上脚本代码。
//-->
</script>
<a href="#" onClick="expandAll(false); return false">

<font color="#008000">全部菜单</font></a>
<div id="elOneParent" class=parent style="margin-left: 0.5em">
<a href="#" style="color:#C000FF; text-decoration:none;" onClick="expandIt('elOne'); return false">
技术中心</a>
</div>
<div id="elOneChild" class=child style="margin-left: 1em">
  <a href="1.htm" target="data">JavaScript</a><br>
  <a href="2.htm" target="data">VbScript</a><br>
  <a href="3.htm" target="data">DHtml</a><br>
  <a href="4.htm" target="data">CSS</a>
</div>
<div id="elTwoParent" class=parent style="margin-left: 0.5em">
<a href="#" style="color:#C000FF; text-decoration:none;" onClick="expandIt('elTwo'); return false">
下载中心</a>
</div>
<div id="elTwoChild" class=child style="margin-left: 1em">
  <a href="1.htm" target="data">软件下载</a><br>
  <a href="2.htm" target="data">游戏下载</a><br>
  <a href="3.htm" target="data">源代码下载</a><br>
  <a href="4.htm" target="data">免费资源下载</a>
</div>
<div id="elThreeParent" class=parent style="margin-left: 0.5em">
<a href="#" style="color:#C000FF; text-decoration:none;" onClick="expandIt('elThree'); return false">
游戏天地</a>

```

```

</div>
<div id="elThreeChild" class=child style="margin-left: 1em">
  <a href="1.htm" target="data">星际争霸</a><br>
  <a href="2.htm" target="data">人富翁</a><br>
  <a href="3.htm" target="data">FIFA2000</a><br>
  <a href="4.htm" target="data">帝国时代</a>
</div>
<div id="elFourParent" class=parent style="margin-left: 0.5em">
<a href="#" style="color:#C000FF;text-decoration:none;" onClick="expandIt('elFour'); return false">
竞技风暴</a>
</div>
<div id="elFourChild" class=child style="margin-left: 1em">
  <a href="1.htm" target="data">国内新闻</a><br>
  <a href="2.htm" target="data">国际新闻</a><br>
  <a href="3.htm" target="data">足球世界</a><br>
  <a href="4.htm" target="data">篮球世界</a>
</div>
<script language="JavaScript1.2">
<!--
if(NS4)
{firstEl = "elOneParent";
  firstInd = getIndex(firstEl);
  showAll();
  arrange();
}
//-->
</script>
</body>
</html>

```

代码键入完毕，存盘为“实例 42.html”。

三、运行程序

双击文件“实例 42.html”。

第三篇

CSS 图片效果篇

本篇总览

本篇主要应用 CSS 中的定位技术以及“Height”和“Weight”属性，进行应用程序的开发，本篇将制作一些基于图片的例子。

通过本篇中各个实例的制作，读者可以对 CSS 中的定位技术有一个大概的了解。定位技术是网页制作最重要的技术之一，通过对网页上文字以及其他元素的精确定位，可以使网页看起来更整齐和舒适。

实例 43 抖动的图片

 实例说明

本例运行效果如图 43-1 所示。这是一个图片产生抖动的效果的例子。

程序运行后，鼠标在图片周围来回移动，图片会产生抖动的效果。

本例的知识点有：CSS 中的基本的定位属性，以及一些 JavaScript 脚本语言知识。

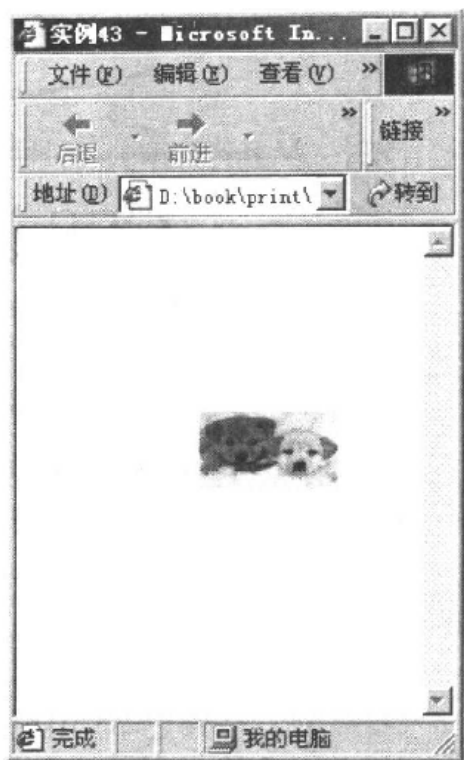


图 43-1 效果图

 编程思路

本实例主要应用 CSS 中的定位属性来控制图片在网页中的位置，并控制图片抖动的大小。

 创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
End Sub  
<html>  
<head>  
<title>实例 43</title>  
<script language=JavaScript>  
function expand()
```

```
{if (small.width<=102)
{x=window.setTimeout('expand()', 10)
small.width=small.width + 5
small.height=small.height + 5
}
else
{
setTimeout('reduce()', 0)
}
}
function reduce()
{if (small.width>80)
{x=window.setTimeout('reduce()', 10)
small.width=small.width - 5
small.height=small.height - 5
}
}
</script>
</head>
<body>
<div id="slot1" onmouseover="expand()" style="height:102px;width:80px">
<dd>

</a>
</dd>
</div>
</body>
</html>
```

键入完毕，存盘为“实例 43.html”。

二、运行程序

双击文件“实例 43.html”。

实例 44 图片播放器

实例说明

本例运行效果如图 44-1 所示。这是一个简单的图片播放器的例子。

程序运行后，单击“下一张”按钮，图片会变换为下一张图片；单击“上一张”按钮，图片会变换为上一张图片。

本例的知识点有：CSS 中的一些基本的文本属性，以及 JavaScript 脚本语言。

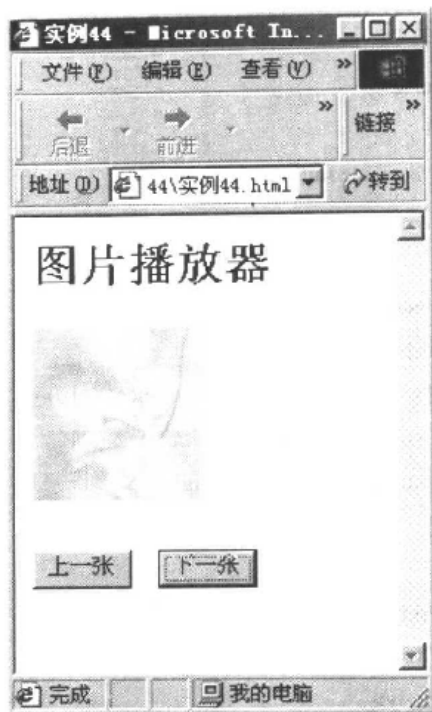


图 44-1 效果图

编程思路

本实例主要应用 CSS 中的文本属性定义文字的样式，使用 JavaScript 脚本语言设计按钮的功能。

创作步骤

一、创建脚本代码

```
preload=new Image();
preload.src="1.jpg";
preload.src="2.jpg";
preload.src="3.jpg";
preload.src="4.jpg";
var count = 0
var pic_num = 4
function imgUp()
```

```
{if (count<pic_num)
{count = count + 1
var fname = count + ".jpg"
document.img.src = fname
}
}
function imgDown()
{if (count > 1)
{count = count - 1
var fname = count + ".jpg"
document.img.src = fname
}
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 44</title>
<head>
<style>
p
{font-family:宋体;
font-weight:bold;
color:blue;
font-style:normal;
font-size:large;
}
</style>
<script language="JavaScript">
<!--
键入以上脚本代码。
//-->
</script>
</head>
<body>
<div>
<p>图片播放器</p>
</div>

```

```
<form name="frm">
<div>
<p>
<input type="button" value="上一张" onClick="imgDown()">
<input type="button" value="下一张" onClick="imgUp()">
</p>
</div>
</form>
</body>
</html>
```

代码键入完毕，存盘为“实例 44.html”。

三、运行程序

双击文件“实例 44.html”。

实例 45 左右移动的图片

实例说明

本例运行效果如图 45-1 所示。这是一个可以左右移动的图片的例子。

程序运行后,会看到图片左右不停地移动。

本例的知识点有:CSS 基本属性以及 JavaScript 脚本语言知识。

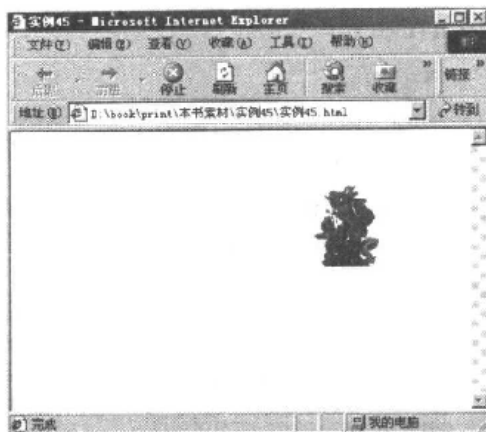


图 45-1 效果图

编程思路

本实例主要应用 CSS 的基本属性对图片进行定位、使用 JavaScript 脚本语言,对图片的左右移动进行控制。

创作步骤

一、创建脚本代码

```

step = 0;
obj = new Image();
function anim(xp,xk,smer)
{obj.style.left = x;
x+=step*smer;
if(x>=(xk+xp)/2)
{if (smer == 1) step--;
else step++;
}
else
{if (smer == -1) step++;
else step--;
}
}

```



```
if (x >= xk)
{
    x = xk;
    smer = -1;
}
if (x <= xp)
{
    x = xp;
    smer = 1;
}
setTimeout('anim('+xp+', '+xk+', '+smer+')', 50);
}

function moveLR(objID,movingarea_width,c)
{
    if (navigator.appName=="Netscape") window_width = window.innerWidth;
    else window_width = document.body.offsetWidth;
    obj = document.images[objID];
    image_width = obj.width;
    x1 = obj.style.left;
    x = Number(x1.substring(0,x1.length-2));
    if (c == 0)
    {
        if (movingarea_width == 0)
        {
            right_margin = window_width - image_width;
            anim(x,right_margin,1);
        }
        else
        {
            right_margin = x + movingarea_width - image_width;
            if (movingarea_width < x + image_width)
                window.alert("No space for moving!");
            else anim(x,right_margin,1);
        }
    }
    else
    {
        if (movingarea_width == 0)
            right_margin = window_width - image_width;
        else
        {
            x = Math.round((window_width-movingarea_width)/2);
            right_margin = Math.round((window_width+movingarea_width)/2)-image_width;
        }
        anim(x,right_margin,1);
    }
}
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 45</title>
<head>
<script language="JavaScript">
<!--
键入以上脚本代码，
//-->
</script>
</head>
<body>

<script language="JavaScript">
<!--
setTimeout("moveLR('picture',300,1)",10);
//-->
</script>
</body>
</html>
```

键入完毕，存盘为“实例 45.html”。

三、运行程序

双击文件“实例 45.html”。

实例 46 移动的广告条

实例说明

本例运行效果如图 46-1 所示。这是一个可移动的广告条的例子。

程序运行后，会看到广告条不停地移动。

本例的知识点有：CSS 中的一些基本的定位属性，以及 JavaScript 脚本语言知识。



图 46-1 效果图

编程思路

本实例利用 CSS 的定位属性对图片进行定位，使用 JavaScript 脚本语言控制广告条的循环移动。

创作步骤

一、创建脚本代码

```
function move()  
{  
if(document.layers)  
{  
document.myid.left += 5;  
if(document.myid.left>300)  
{  
document.myid.left = 0;  
}  
}  
}  
else if(document.all)
```

```
{ myid.style.left=parseInt(myid.style.left)+5;
  if(parseInt(myid.style.left)>300)
  { myid.style.left =0;
  }
}
if((document.layers)||(document.all))
{ setTimeout('move()', 50);
}
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 46</title>
<head>
<script language="javascript">
键入以上脚本代码。
</script>
</head>
<body onload="move()">
<div id="myid" style="position:relative;left:0px;width:100px;height:45px;">
<p align="left">

<br>
</p>
</div>
</body>
</html>
```

键入完毕，存盘为“实例 46.html”。

三、运行程序

双击文件“实例 46.html”。

实例 47 跟随屏幕移动的图像

实例说明

本例运行效果如图 47-1 所示,这是一个可以跟随屏幕移动的图像的例子。

程序运行后,拖动滚动条会看到图片始终位于屏幕的同一位置。

本例的知识点有: CSS 中的定位属性,以及 JavaScript 脚本语言知识。



图 47-1 效果图

编程思路

本实例利用 CSS 中的定位属性对图片进行初始定位,使用 JavaScript 脚本语言控制图片跟随屏幕移动。

创作步骤

一、创建脚本代码

```
self.onError=null;
currentX=currentY=0;
whichIt=null;
lastScrollX=0;
lastScrollY=0;
NS=(document.layers)?1:0;
IE=(document.all)?1:0;
function heartBeat()
{if(IE)
```

```
{diffY=document.body.scrollTop;
  diffX=document.body.scrollLeft;
}
if(NS)
{diffY=self.pageYOffset;
  diffX=self.pageXOffset;
}
if(diffY!=lastScrollY)
{percent=.1*(diffY-lastScrollY);
  if(percent>0)
    percent=Math.ceil(percent);
  else
    percent=Math.floor(percent);
  if(IE)
    document.all.floate.style.pixelTop+=percent;
  if(NS)
    document.floate.top+=percent;
  lastScrollY=lastScrollY+percent;
}
if(diffX!=lastScrollX)
{percent=.1*(diffX-lastScrollX);
  if(percent>0)
    percent=Math.ceil(percent);
  else
    percent=Math.floor(percent);
  if(IE)
    document.all.floate.style.pixelLeft+=percent;
  if(NS)
    document.floate.left+=percent;
  lastScrollX = lastScrollX + percent;
}
}
function checkFocus(x,y)
{stalkerx=document.floate.pageX;
 stalkery=document.floate.pageY;
 stalkerwidth = document.floate.clip.width;
 stalkerheight = document.floate.clip.height;
 if((x>stalkerx&& x<(stalkerx+stalkerwidth))&&(y>stalkery&&y<(stalkery+stalkerheight)))
   return true;
```



```
else
    return false;
}
function grabIt(e)
{ if(IE)
    { whichIt=event.srcElement;
      while(whichIt.id.indexOf("floater")==-1)
        { whichIt=whichIt.parentElement;
          if(whichIt==null)
            return true;
          }
      whichIt.style.pixelLeft=whichIt.offsetLeft;
      whichIt.style.pixelTop=whichIt.offsetTop;
      currentX=(event.clientX+document.body.scrollLeft);
      currentY=(event.clientY+document.body.scrollTop);
    }
  else
    { window.captureEvents(Event.MOUSEMOVE);
      if(checkFocus(e.pageX,e.pageY))
        { whichIt=document.floater;
          StalkerTouchedX=e.pageX-document.floater.pageX;
          StalkerTouchedY=e.pageY-document.floater.pageY;
        }
      }
    return true;
  }
function moveIt(e)
{ if(whichIt==null)
    return false;
  if(IE)
    { newX=(event.clientX+document.body.scrollLeft);
      newY=(event.clientY+document.body.scrollTop);
      distanceX=(newX-currentX);
      distanceY=(newY-currentY);
      currentX=newX;
      currentY=newY;
      whichIt.style.pixelLeft+=distanceX;
      whichIt.style.pixelTop+=distanceY;
      if(whichIt.style.pixelTop<document.body.scrollTop)
```

```

        whichIt.style.pixelTop=document.body.scrollTop;
    if(whichIt.style.pixelLeft<document.body.scrollLeft)
        whichIt.style.pixelLeft=document.body.scrollLeft;
    if(whichIt.style.pixelLeft>document.body.offsetWidth-document.body.scrollLeft-
        whichIt.style.pixelWidth-20)
        whichIt.style.pixelLeft=document.body.offsetWidth-whichIt.style.pixelWidth-20;
    if(whichIt.style.pixelTop>document.body.offsetHeight+document.body.scrollTop-
        whichIt.style.pixelHeight-5)
        whichIt.style.pixelTop=document.body.offsetHeight+document.body.scrollTop-
            whichIt.style.pixelHeight-5;
    event.returnValue=false;
}
else
{whichIt.moveTo(e.pageX-StalkerTouchedX,e.pageY-StalkerTouchedY);
    if(whichIt.left<0+self.pageXOffset)
        whichIt.left=0+self.pageXOffset;
    if(whichIt.top<0+self.pageYOffset)
        whichIt.top=0+self.pageYOffset;
    if((whichIt.left+whichIt.clip.width)>=(window.innerWidth+self.pageXOffset-17))
        whichIt.left=((window.innerWidth+self.pageXOffset)-whichIt.clip.width)-17;
    if((whichIt.top+whichIt.clip.height)>=(window.innerHeight+self.pageYOffset-17))
        whichIt.top=((window.innerHeight+self.pageYOffset)-whichIt.clip.height)-17;
    return false;
}
return false;
}
function dropl()
{whichIt = null;
    if(NS)
        window.releaseEvents(Event.MOUSEMOVE);
    return true;
}
if(NS)
{window.captureEvents(Event.MOUSEUP|Event.MOUSEDOWN);
    window.onmousedown=grabIt;
    window.onmousemove=moveIt;
    window.onmouseup=dropl;
}
if(IE)

```

```
[document.onmousedown=grabIt;
document.onmousemove=moveIt;
document.onmouseup=dropIt;
}
if(NS11IE)
    action=window.setInterval("heartBeat()",1);
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 47</title>
<head>
<style type="text/css">
#floater
{position:absolute;
left:200;
top:50;
width:125;
visibility:visible;
z-index:10;
}
</style>
</head>
<body>
<div id="floater" style="left:100px; top:100px">
<p>

<br>
</p>
</div>
<script language="JavaScript">
键入以上脚本代码。
</script>
</body>
</html>
```

代码键入完毕，存盘为“实例 47.html”。

三、运行程序

双击文件“实例 47.html”。

实例 48 被盖住的图片

实例说明

本例运行效果如图 48-1 所示。这是一个图片被七彩的方块盖住的例子。

程序运行后，将鼠标置于任意七彩的方块上，被盖住的图片会显示出来。

本例的知识点有：CSS 中的定位属性，以及 JavaScript 脚本语言知识。

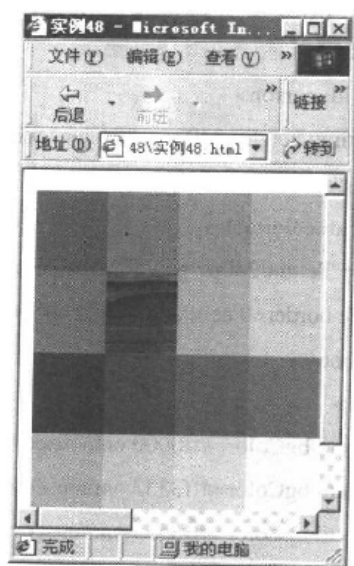


图 48-1 效果图

编程思路

本实例使用 CSS 中的定位属性对方块进行定位，使用 JavaScript 脚本语言控制方块的显示和隐藏。

创作步骤

一、创建脚本代码

```
function positionimage()
{if(document.all)
  {document.all.person.style.posTop=10
  document.all.person.style.posLeft=80
  document.all.person.style.visibility="VISIBLE"
  }
}
function show(thiscolor)
{if(document.all) {document.all.colortables.style.filter="chroma(color="+thiscolor+" )"}
}
}
```


二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 48</title>
<body>
<div id=person>
<p>
</p></div>
<div id=colortables
style="height:240px;position:absolute;top:15px;width:315px">
<table border=0 cellPadding=0 cellSpacing=0 height=240 width=315>
  <tbody>
    <tr>
      <td bgColor=#ff0000 onmouseover="show('FF0000')"></td>
      <td bgColor=#ff3333 onmouseover="show('FF3333')"></td>
      <td bgColor=#ff5555 onmouseover="show('FF5555')"></td>
      <td bgColor=#ff7777 onmouseover="show('FF7777')"></td>
      <td bgColor=#ffa000 onmouseover="show('FFa000')"></td>
      <td bgColor=#ffcccc onmouseover="show('FFcccc')"></td>
    </tr>
    <tr>
      <td bgColor=#ff00ff onmouseover="show('FF00ff')"></td>
      <td bgColor=#ff33ff onmouseover="show('FF33ff')"></td>
      <td bgColor=#ff55ff onmouseover="show('FF55ff')"></td>
      <td bgColor=#ff88ff onmouseover="show('FF88ff')"></td>
      <td bgColor=#ffaaff onmouseover="show('FFaaff')"></td>
      <td bgColor=#ffccff onmouseover="show('FFccff')"></td>
    </tr>
    <tr>
      <td bgColor=#0011ff onmouseover="show('0011FF')"></td>
      <td bgColor=#1133ff onmouseover="show('1133FF')"></td>
      <td bgColor=#2255ff onmouseover="show('2255FF')"></td>
      <td bgColor=#3377ff onmouseover="show('3377FF')"></td>
      <td bgColor=#44aaff onmouseover="show('44aaFF')"></td>
      <td bgColor=#55ddff onmouseover="show('55ddFF')"></td>
    </tr>
    <tr>
      <td bgColor=#22ff22 onmouseover="show('22ff22')"></td>
    </tr>
  </tbody>
</table>

```

```
<td bgColor=#55ff55 onmouseover="show('55ff55')"></td>
<td bgColor=#77ff77 onmouseover="show('77ff77')"></td>
<td bgColor=#aaffaa onmouseover="show('aaffaa')"></td>
<td bgColor=#ccffcc onmouseover="show('ccffcc')"></td>
<td bgColor=#ddfidd onmouseover="show('ddfidd')"></td>
</tr>
</tbody>
</table>
</div>
<div id=redirection
style="left:420px;position:absolute;top:10px;width:180px;">
</div>
<script language=JavaScript>
<!--
键入以上脚本代码。
//-->
</script>
</body>
</html>
```

代码键入完毕，存盘为“实例 48.html”。

三、运行程序

双击文件“实例 48.html”。

实例 49 飞舞的图片链接

实例说明

本例运行效果如图 49-1 所示。这是一个飞舞的图片链接的例子。

程序运行后，会看到三张图片不停地飞舞，鼠标单击图片会链接到相应的位置。

本例的知识点有：CSS 中的定位属性，以及 JavaScript 脚本语言知识。

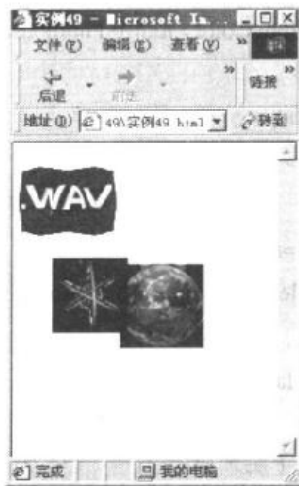


图 49-1 效果图

编程思路

本实例使用 CSS 中的定位属性对图片进行初始定位，使用 JavaScript 脚本语言控制图片的飞舞。

创作步骤

一、创建脚本代码

```
var brOK=false;
var mie=false;
var aver=parseInt(navigator.appVersion.substring(0,1));
var aname=navigator.appName;
function checkbrOK()
{if(aname.indexOf("Internet Explorer")!=-1)
  {if(aver>=4) brOK=navigator.javaEnabled();
   mie=true;
  }
  if(aname.indexOf("Netscape")!=-1)
  {if(aver>=4) brOK=navigator.javaEnabled();
  }
}
```

```
var vmin=2;
var vmax=5;
var vr=2;
var timer1;
function Chip(chipname,width,height)
{this.named=chipname;
  this.vx=vmin+vmax*Math.random();
  this.vy=vmin+vmax*Math.random();
  this.w=width;
  this.h=height;
  this.xx=0;
  this.yy=0;
  this.timer1=null;
}
function movechip(chipname)
{if(brOK)
  {eval("chip="+chipname);
   if(!mie)
   {pageX=window.pageXOffset;
    pageW=window.innerWidth-20;
    pageY=window.pageYOffset;
    pageH=window.innerHeight-20;
    }
   else
   {pageX=window.document.body.scrollLeft;
    pageW=window.document.body.offsetWidth-20;
    pageY=window.document.body.scrollTop;
    pageH=window.document.body.offsetHeight-20;
    }
   chip.xx=chip.xx+chip.vx;
   chip.yy=chip.yy+chip.vy;
   chip.vx+=vr*(Math.random()-0.5);
   chip.vy+=vr*(Math.random()-0.5);
   if(chip.vx>(vmax+vmin)) chip.vx=(vmax+vmin)*2-chip.vx;
   if(chip.vx<(-vmax-vmin)) chip.vx=(-vmax-vmin)*2-chip.vx;
   if(chip.vy>(vmax+vmin)) chip.vy=(vmax+vmin)*2-chip.vy;
   if(chip.vy<(-vmax-vmin)) chip.vy=(-vmax-vmin)*2-chip.vy;
   if(chip.xx<=pageX)
   {chip.xx=pageX;
```

```
        chip.vx=vmin+vmax*Math.random();
    }
    if(chip.xx>=pageX+pageW-chip.w)
    {chip.xx=pageX+pageW-chip.w;
    chip.vx=-vmin-vmax*Math.random();
    }
    if(chip.yy<=pageY)
    {chip.yy=pageY;
    chip.vy=vmin+vmax*Math.random();
    }
    if(chip.yy>=pageY+pageH-chip.h)
    {chip.yy=pageY+pageH-chip.h;
    chip.vy=-vmin-vmax*Math.random();
    }
    if(document.layers)
    {eval('document.'+chip.named+'.top='+chip.yy);
    eval('document.'+chip.named+'.left='+chip.xx);
    }
    else if (document.all)
    {eval('document.all.'+chip.named+'.style.pixelLeft='+chip.xx);
    eval('document.all.'+chip.named+'.style.pixelTop='+chip.yy);
    }
    else if (document.getElementById)
    {document.getElementById(chip.named).style.left=chip.xx;
    document.getElementById(chip.named).style.top =chip.yy;
    }
    chip.timer1=setTimeout("movechip("+chip.named+")",100);
}
}
function stopme(chipname)
{if(brOK)
{eval("chip="+chipname);
if(chip.timer1!=null)
{clearTimeout(chip.timer1)}
}
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 49</title>
<head>
<script language="JavaScript">
键入以上脚本代码.
</script>
</head>
<body onLoad="pagestart();" onUnload="if(brOK) {stopme('chip1'); stopme('chip2'); stopme('chip3');}" >
<script>
var chip1;
var chip2;
var chip3;
function pagestart()
{checkbrOK();
 chip1=new Chip("chip1",60,80);
 chip2=new Chip("chip2",60,80);
 chip3=new Chip("chip3",60,80);
 if(brOK)
 {movechip("chip1");
 movechip("chip2");
 movechip("chip3");
 }
}
</script>
<div id="chip1" style="position:absolute; width:47; height:68;">
<a href="http://www.163.com"></a>
</div>
<div id="chip2" style="position:absolute; width:47; height:68;">
<a href="http://www.sina.com.cn"></a>
</div>
<div id="chip3" style="position:absolute; width:47; height:68;">
<a href="http://www.sohu.com"></a>
</div>
</body>
</html>
```

代码键入完毕，存盘为“实例 49.html”。

三、运行程序

双击文件“实例 49.html”。

实例 50 静态导航

实例说明

本例运行效果如图 50-1 所示。这是一个静态导航图片的例子。

程序运行后，三张图片固定在屏幕上，鼠标单击任意一张图片会链接到相应的位置。

本例的知识点有：CSS 中的定位属性，以及 JavaScript 脚本语言知识。



图 50-1 效果图

编程思路

本实例使用 CSS 中的定位属性对图片进行初始定位，使用 JavaScript 脚本语言对导航进行控制。

创作步骤

一、创建脚本代码

```
function setVariables()  
{if(navigator.appName=="Netscape")  
{v=".top=";  
dS="document.";  
sD="";  
y="window.pageYOffset";  
}  
else  
{v=".pixelTop=";  
dS="";  
sD=".style";  
}
```

```

        y="document.body.scrollTop";
    }
}
function checkLocation()
{object="object1";
  yy=eval(y);
  eval(dS+object+sD+v+yy);
  setTimeout("checkLocation()",10);
}

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 50</title>
<head>
<script language=javascript>
<!--
键入以上脚本代码。
//-->
</script>
</head>
<body bgcolor="#fef4d9" OnLoad="setVariables();checkLocation()">
<div id="object1" style="position:absolute;visibility:show;left:0px; top:0px;z-index:2">
<table width=128 border=0 cellspacing=20 cellpadding=0>
<tr>
<td height="97">
<a href="#"></a><br>
<a href="#"></a><br>
<a href="#"></a>
</td>
</tr>
</table>
</div>
</body>
</html>

```

代码键入完毕，存盘为“实例 50.html”。

三、运行程序

双击文件“实例 50.html”。

实例 51 图片扭曲变形

实例说明

本例运行效果如图 51-1 所示。这是一个图片扭曲变形的例子。

程序运行后，会看到图片的大小不停地变化。

本例的知识点有：CSS 中的基本属性，以及 JavaScript 脚本语言知识。



图 51-1 效果图

编程思路

本实例使用 JavaScript 脚本语言动态改变图片的 Width、Height 属性，使图片产生扭曲变形的效果。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 51</title>
<body bgcolor="#fcf4d9" onLoad="fade()">

<script language="JavaScript">
var b=1;
var e=true;
function fade()
{if(document.all);
  if(e==true)
```

```
        {b++;  
        }  
    if(b==100)  
    {b--;  
    c=false  
    }  
    if(b==10)  
    {b++;  
    c=true;  
    }  
    if(c==false)  
    {b--;  
    }  
    u.width=150+b;  
    u.height=125-b;  
    setTimeout("fade()",50);  
    }  
</script>  
</body>  
</html>
```

代码键入完毕，存盘为“实例 51.html”。

二、运行程序

双击文件“实例 51.html”。

实例 52 抖动的窗口超链接

实例说明

本例运行效果如图 52-1 所示。这是一个能使窗口抖动的超链接的例子。

程序运行后,将鼠标置于图片之上会产生窗口抖动的效果。

本例的知识点有:CSS 中的定位属性,以及 JavaScript 脚本语言。



图 52-1 效果图

编程思路

本实例主要使用 CSS 中的定位属性对图片的起始位置进行设置,使用 JavaScript 脚本语言产生抖动的效果。

创作步骤

一、创建脚本代码

```
function boom(n) //抖动函数
{if (window.top.moveBy)
  {for (i = 10; i > 0; i--)
    {for (j = n; j > 0; j--)
      { window.top.moveBy(0,i);
        window.top.moveBy(i,0);
        window.top.moveBy(0,-i);
        window.top.moveBy(-i,0);
      }
    }
  }
```

```
    )  
  }  
}  
function callJS(jsStr)  
{return eval(jsStr)  
}  
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>  
<title>实例 52</title>  
<head>  
<script language=JavaScript 1.2>  
<!--  
键入以上脚本代码。  
//-->  
</script>  
</head>  
<body>  
<div>  
<p style="line-height:22px">  
<a href="http://www.sina.com.cn" onmouseover="callJS('boom(1)');">  
  
</a>  
</p>  
</div>  
</body>  
</html>
```

代码键入完毕，存盘为“实例 52.html”。

三、运行程序

双击文件“实例 52.html”。

第四篇

CSS 页面效果篇

本篇总览

本篇主要应用 CSS 中的大部分属性，进行应用程序的开发，本篇将制作一些有关页面效果的例子。

通过本篇中各个实例的制作，读者可以对 CSS 中大部分属性有一个大致了解。页面效果是网页上出现最多，也是地位最重要的效果，通过对页面效果的制作，可以使网页看起来更漂亮。

实例 53 为控件添加样式

实例说明

本例运行效果如图 53-1 所示。这是一个给控件添加样式的例子。

程序运行后，在屏幕上会出现添加样式的控件。

本例的知识点有：CSS 的一些基本属性如 color、background-color 等。

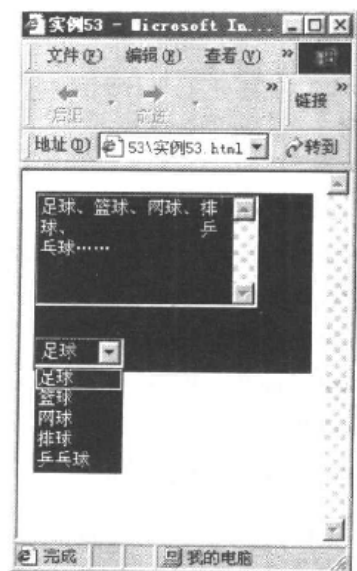


图 53-1 效果图

编程思路

本实例主要运用了一些 CSS 中的颜色属性，使控件能和设计的网页在风格上很好地统一起来。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<head>
<title>实例 53</title>
</head>
<body>
<table cellspacing="0" >
<tr>
<td width="36%" bgcolor="black">
<p>
<textarea rows="5" cols="20" style="color:#FFFFFF;background-color:black">足球、篮球、网球、排球、
```

乒乓球……

```
</textarea>
</p>
<p>
</p>
</td>
</tr>
</tr>
<td width="36%" bgcolor="black">
<p><select size="1" style="color: #FFFFFF;background-color:black">
  <option>足球</option>
  <option>篮球</option>
  <option>网球</option>
  <option>排球</option>
  <option>乒乓球</option>
</select>
</p>
</td>
</tr>
</table>
</body>
</html>
```

键入完毕，存盘为“实例 53.html”。

二、运行程序

双击文件“实例 53.html”。

实例 54 背景时钟

实例说明

本例运行效果如图 54-1 所示。这是一个显示背景时钟的例子。

程序运行后，会在页面上出现两个不同样式的时钟相互重叠。

本例的知识点有：CSS 中的一些基本的文本属性，以及 JavaScript 脚本语言知识。



图 54-1 效果图

编程思路

本实例使用 CSS 中的文本属性对时钟的数字进行样式定义，使用 JavaScript 脚本语言的函数对时钟进行控制。

创作步骤

一、创建脚本代码

```
function clockon() //时钟函数
{
    this.time = new Date()
    var hours=this.time.getHours()
    var minutes=this.time.getMinutes()
    var seconds=this.time.getSeconds()
    if (eval(hours) < 10) {hours="0"+hours;}
    if (eval(minutes) < 10) {minutes="0"+minutes;}
    if (seconds < 10) {seconds="0"+seconds;}
    this.time = hours+"."+minutes+"."+seconds
    if(document.all)

```



```

{ bgclocknoshade.innerHTML=thisime
  bgclockshade.innerHTML=thisime
}
if(document.layers)
{ document.bgclockshade.document.write
  ('<div id="bgclockshade"
  style="position:absolute;visibility:visible;
  font-family:Verdana;color:FFAAAAA;
  font-size:120px;top:10px;left:152px">'+thisime+'</div>')
  document.bgclocknoshade.document.write
  ('<div id="bgclocknoshade"
  style="position:absolute;visibility:visible;
  font-family:Verdana;color:DDDDDD;
  font-size:120px;top:10px;left:150px">'+thisime+'</div>')
  document.close()
}
var timer=setTimeout("clockon()",200)
}

```

二、创建源代码

打开记事本,键入如下代码:

```

<html>
<title>实例 54</title>
<head>
<script language=JavaScript>
<!--
键入以上脚本代码。
//-->
</script>
</head>
<body onLoad="clockon()">
<div id="bgclockshade"
  style="position:absolute;visibility:visible;
  font-family:Arial;color:FF8888,
  font-size:120px;top:102px;left:152px">
</div>
<div id="bgclocknoshade"
  style="position:absolute;visibility:visible;
  font-family:Arial;color:DDDDDD;

```

```
font-size:120px;top:100px;left:150px">  
</div>  
<div id="mainbody" style="position:absolute;visibility:visible">  
</div>  
</body>  
</html>
```

代码键入完毕，存盘为“实例 54.html”。

三、运行程序

双击文件“实例 54.html”。

实例 55 滚动的背景

实例说明

本例运行效果如图 55-1 所示。这是一个背景文字可以滚动的例子。

程序运行后, 会看到背景文字不停地滚动。

本例的知识点有: CSS 中的文本属性, 以及 JavaScript 脚本语言知识。



图 55-1 效果图

编程思路

本实例使用 CSS 中的文本属性, 定义背景文字的风格, 使用 JavaScript 脚本语言控制背景文字的滚动。

创作步骤

一、创建脚本代码

```
var y=100
var bgticker
function initObjects()
{if(document.all)
{bgticker=document.all.bgtick.style
bgticker.posLeft=0
bgticker.posTop=400
mainbody=document.all.mainbod.style
mainbody.posLeft=0
mainbody.posTop=0
}
```

```
if(document.layers)
{
  bgticker=document.bgticker
  bgticker.left=0
  bgticker.top=300
  mainbody=document.mainbody
  mainbody.left=0
  mainbody.top=0
}
scroll()
}
function scroll() //背景滚动函数
{
  if(y>200)
  {
    y--
    if(document.all)
    {
      bgticker.posTop=y
      mainbody.posTop=0
    }
    if(document.layers)
    {
      bgticker.top=y
      mainbody.top=0
    }
  }
  else
  {
    y=300
  }
  var timer = setTimeout('scroll()',20)
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<head>
<title>实例 55</title>
<script language="JavaScript1.1">
<!--
键入以上脚本代码。
//-->
</script>
<style>
```

```
.ticker  
{position:absolute;  
font-family:Arial;  
color:DDDDDD;  
font-size:60px;  
letter-spacing:1;  
}
```

```
</style>
```

```
</head>
```

```
<body onLoad="initObjects()">
```

```
<div id="bgtick" class="ticker">
```

西部地区物力资源优势的开发首先取决于人力资源的状况。这就需要我们所开发的内容和领域有新的转变，由过去以开发物力资源为主转向既开发物，又开发人，尤其应注重人力资源的开发。

```
</div>
```

```
<div id="mainbod" style="position:absolute; visibility:visible">
```

```
</div>
```

```
</body>
```

```
</html>
```

代码键入完毕，存盘为“实例 55.html”。

三、运行程序

双击文件“实例 55.html”。

实例 56 倒计时

实例说明

本例运行效果如图 56-1 所示。这是一个倒计时的例子。

程序运行后，会显示出倒计时条，当时间到达时会显示“时间到了！！”。源代码中的 `setcountdown(2001,4,5)` 语句可以改变倒计时的时间。

本例的知识点有：CSS 中基本的文本属性、以及 JavaScript 脚本语言知识。

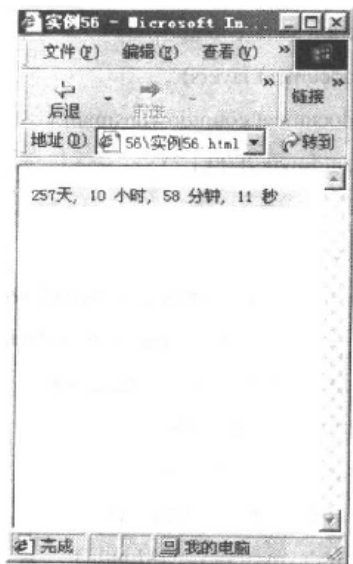


图 56-1 效果图

编程思路

本实例使用 CSS 中的文本属性，定义计时文字的样式，使用 JavaScript 脚本语言控制倒计时时钟的运行。

创作步骤

一、创建脚本代码

```
function setcountdown(theyear,themoth,theday)
{
    yr=theyear;
    mo=themoth;
    da=theday
}
setcountdown(2002,1,1) //6时限
var occasion="时间到了！！"
var message_on_occasion="时间到了！！"
var countdownwidth='200px'
var countdownheight='20px'
```

```

var countdownbgcolor='lightblue'
var opentags='<font face="宋体" style="font-size:9pt">'
var closetags='</font>'
var montharray=new Array("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec")
function start_countdown()
{if(document.layers)
    document.countdownnsmain.visibility="show"
    countdown(2000,1,1)
}
if(document.all)
document.write('<span id="countdownie"
    style="width:'+countdownwidth+'; background-color:'+countdownbgcolor+'"></span>')
window.onload=start_countdown
function countdown()
{ var today=new Date()
  var todayy=today.getYear()
  var todaym=today.getMonth()
  var todayd=today.getDate()
  var todayh=today.getHours()
  var todaymin=today.getMinutes()
  var todaysec=today.getSeconds()
  var todaystring=montharray[todaym]+" "+todayd+" "+todayy+" "+todayh+" "+todaymin+" "+todaysec
  futurestring=montharray[mo-1]+" "+da+" "+yr
  dd=Date.parse(futurestring)-Date.parse(todaystring)
  dday=Math.floor(dd/(60*60*1000*24)*1)
  dhour=Math.floor(((dd%(60*60*1000*24))/(60*60*1000)*1)
  dmin=Math.floor((((dd%(60*60*1000*24))%(60*60*1000))/(60*1000)*1)
  dsec=Math.floor((((dd%(60*60*1000*24))%(60*60*1000))%(60*1000))/1000*1)
  if(dday<=0&&dhour<=0&&dmin<=0&&dsec<=1)
  {if(document.layers)
    [ document.countdownnsmain.document.countdownnssub.document.write
      (opentags+message_on_occasion+closetags)
      document.countdownnsmain.document.countdownnssub.document.close()
    ]
  else if(document.all)
      countdownie.innerHTML=opentags+message_on_occasion+closetags
  return
  }
  else
  }
}

```

```

if(document.layers)
{
document.countdownnsmain.document.countdownnssub.document.write(opentags+dday+ " 天 ,
"+dhour+" 小时,"+dmin+" 分钟, "+dsec+" 秒")
document.countdownnsmain.document.countdownnssub.document.close()
}
else if(document.all)
countdownie.innerHTML=opentags+dday+"天,"+dhour+" 小时,"+dmin+" 分钟,"+dsec+" 秒 "
}
setTimeout("countdown()",1000)
}

```

二、建立程序的基本框架

打开记事本，键入如下代码：

```

<html>
<title>实例 56</title>
<body bgcolor="#FFFFFF">
<script language="JavaScript1.2">
键入以上脚本代码。
</script>
<layer id="countdownnsmain" width={countdownwidth};
height={countdownheight}; bgColor={countdownbgcolor}; visibility=hide>
<layer id="countdownnssub" width={countdownwidth}; height={countdownheight}; left=0 top=0>
</layer>
</layer>
</body>
</html>

```

代码键入完毕，存盘为“实例 56.html”。

三、运行程序

双击文件“实例 56.html”。

实例 57 十字准星

实例说明

本例运行效果如图 57-1 所示。这是一个十字准星的例子。

程序运行后,任意移动鼠标会看到一个十字准星跟随鼠标移动。

本例的知识点有: CSS 中的基本属性,以及 JavaScript 脚本语言。



图 57-1 效果图

编程思路

本实例使用 CSS 中的基本属性,定义十字准星的样式。使用 JavaScript 脚本语言控制准星的移动。

创作步骤

一、创建脚本代码

```
if(document.all&&!window.print)
{leftright.style.width=document.body.clientWidth-2
topdown.style.height=document.body.clientHeight-2
}
else if(document.layers)
{document.leftright.clip.width=window.innerWidth
document.leftright.clip.height=1
document.topdown.clip.width=1
document.topdown.clip.height=window.innerHeight
}
```

```
function followmouse1()
{left:right.style.pixelTop=document.body.scrollTop+event.clientY+1
  topdown.style.pixelTop=document.body.scrollTop
  if(event.clientX<document.body.clientWidth-2)
    topdown.style.pixelLeft=document.body.scrollLeft+event.clientX+1
  else
    topdown.style.pixelLeft=document.body.clientWidth-2
}
function followmouse2(e)
{document.left:right.top=e.y+1
  document.topdown.top=pageYOffset
  document.topdown.left=e.x+1
}
if(document.all)
  document.onmousemove=followmouse1
else if(document.layers)
  {window.captureEvents(Event.MOUSEMOVE)
    window.onmousemove=followmouse2
  }
function regenerate1()
{window.location.reload()
}
function regenerate2()
{setTimeout("window.onresize=regenerate",400)
}
if((document.all&&!window.print)||document.layers)
  window.onload=regenerate2
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 57</title>
<head>
<style>
<!--
#left:right,#topdown
{position:absolute;
left:0;
top:0;
```



```
width:1px;
height:1px;
layer-background-color:black;
background-color:black;
z-index:100;
font-size:1px;
}
-->
</style>
<script language="JavaScript">
<!--
键入以上脚本代码。
//-->
</script>
</head>
<body>
<div id="leftright" style="width:expression(document.body.clientWidth-2)">
</div>
<div id="topdown" style="height:expression(document.body.clientHeight-2)">
</div>
</body>
</html>
```

代码键入完毕，存盘为“实例 57.html”。

三、运行程序

双击文件“实例 57.html”。

实例 58 模拟 Flash 效果

实例说明

本例运行效果如图 58-1 所示。这是一个模拟 Flash 效果的例子。

程序运行后，将鼠标置于提示文字之上，会出现模拟 Flash 的效果。

本例的知识点有：CSS 中基本的属性，以及 JavaScript 脚本语言知识。



图 58-1 效果图

编程思路

本实例使用 CSS 中的文本属性，定义提示文字的样式，使用 JavaScript 脚本语言控制模拟 Flash 效果。

创作步骤

一、创建脚本代码

```
var x,y
var timer
var i_fontsize=0
var step=0
var thisx,thisy
function handlerMM(e)
{x=(document.layers)?e.pageX:event.clientX
 y=(document.layers)?e.pageY:event.clientY
}
function ringup()
{if(document.all)
```

```

    {thisx = x
      thisy = y
      ringup2()
    }
  }
function ringup2()
{if (i_fontsize<=1530)
  {document.all.ring.style.fontSize=i_fontsize
  document.all.ring.style.color=
  "rgb(255,"+Math.floor(i_fontsize/6)+",""+Math.floor(i_fontsize/6)+")"}
  document.all.ring.style.posLeft=thisx-(Math.floor(i_fontsize/3))
  document.all.ring.style.posTop=thisy-(Math.floor(i_fontsize/1.4))
  step+=4
  i_fontsize+=step
  timer=setTimeout("ringup(2)",50)
}
else
{clearTimeout(timer)
  i_fontsize=0
  step=0
  document.all.ring.style.posTop=-10000
}
}
document.onmousemove = handlerMM;

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 58</title>
<head>
<style>
.ringstyle
{position:absolute;
visibility:visible;
top:-50px;
font-size:5pt;
font-family:Verdana;
color:FF0000
}

```

```
.explain
{position:absolute;
top:80px;
left:40px;
width:150px;
color:000000;
text-align:center;
font-size:10pt;
font-family:Times;
}
A
{font-family: Times;
color:#000000;
text-decoration:underline
}
A:visited
{color:#000000;
}
</style>
<script>
<!--
键入以上脚本代码。
-->
</script>
</head>
<body>
<div id="ring" class="ringstyle">o</div>
<div id="redirection" class="explain">
<p>
<a onMouseOver="ringup()">
请把鼠标放到这里来试试.....
</a>
</div>
</body>
</html>
```

代码键入完毕，存盘为“实例 58.html”。

三、运行程序

双击文件“实例 58.html”。

实例 59 陨石背景

 实例说明

本例运行效果如图 59-1 所示。这是一个从背景中产生陨石的例子。

程序运行后，从背景中会随机产生陨石。

本例的知识点有：CSS 中的基本属性，以及 JavaScript 脚本语言知识。

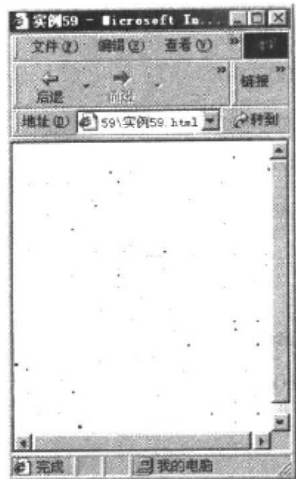


图 59-1 效果图

 编程思路

本实例使用 CSS 中的基本属性，定义陨石的样式，使用 JavaScript 脚本语言控制陨石的产生。

 创作步骤

一、创建脚本代码

```
var starnum=175;
var isNS=(document.layers);
var _all=(isNS)?":all;";
var _style=(isNS)?":style;";
var xoffset,yoffset,w_x,w_y,tmpx,tmpy,scr1x,scr1y;
function getstarpos(obj)
{obj.deltay=Math.floor(Math.random()*12);
obj.deltax=Math.floor(Math.random()*12);
obj.xdir=(Math.floor(Math.random()*2)==1)?'+':'-';
obj.ydir=(Math.floor(Math.random()*2)==1)?'+':'-';
obj.counter=1;
if(isNS)
{obj.clip.width=1;
```

```

obj.clip.height=1;
obj.moveTo(xoffset+pageXOffset,yoffset+pageYOffset);
}
else
{obj.width=1;
obj.height=1;
obj.pixelTop=yoffset+document.body.scrollTop;
obj.pixelLeft=xoffset+document.body.scrollLeft;
}
}
function movestar(starN)
{tmpx=starN.deltax*starN.counter+starN.counter;
tmpy=starN.deltay*starN.counter+starN.counter;
if(isNS)
{starN.clip.width=starN.counter/3;
starN.clip.height=starN.counter/3;
scrIx=pageXOffset;
scrly=pageYOffset;
if((starN.left+tmpx>=w_x+scrIx)||(starN.top+tmpy>=w_y+scrly)||(starN.left-tmpx <= scrIx)
|| (starN.top-tmpy<=scrly))
{getstartpos(starN);
}
else
{eval('starN.moveBy('+starN.xdir+tmpx+', '+starN.ydir+tmpy+')');
}
}
else
{starN.width=starN.counter/3;
starN.height=starN.counter/3;
scrIx=document.body.scrollLeft;
scrly=document.body.scrollTop;
if((starN.pixelLeft+tmpx>=w_x+scrIx)||(starN.pixelTop+tmpy>=w_y+scrly)||(starN.pixelLeft-tmpx
<=scrIx)||(starN.pixelTop-tmpy<=scrly))
{getstartpos(starN);
}
else
{eval('starN.pixelTop'+starN.ydir+'=tmpy');
eval('starN.pixelLeft'+starN.xdir+'=tmpx');
}
}
}

```



```

    }
    starN.counter++;
    }
    function animate()
    {for(i=1;i<=starnum;i++)
      {movestartar(eval('star'+i));
      }
    setTimeout('animate()', 100);
    }
    function findwindowparams()
    {w_x=(isNS)?window.innerWidth:document.body.clientWidth;
      w_y=(isNS)?window.innerHeight:document.body.clientHeight;
      xoffset=w_x/2;
      yoffset=w_y/2;
      for(i=1;i<=starnum;i++)
      {getstartpos(eval('star'+i));
      }
    }
    function resizeNS()
    {setTimeout('document.location.reload()', 400);
    }
    (isNS)?window.onresize=resizeNS:window.onresize=findwindowparams;
    window.onload=new Function("findwindowparams(); animate();");

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 59</title>
<head>
<style type="text/css">
.star
{position:absolute;
  layer-background-color:black;
  visibility:visible;
  top:-50px;
  width:50px;
  height:50px;
  font-size:1px;
  background-color:black;

```

```
)  
</style>  
<script language="JavaScript">  
<!--  
键入以上脚本代码。  
//-->  
</script>  
</body>  
</html>
```

代码键入完毕，存盘为“实例 59.html”。

三、运行程序

双击文件“实例 59.html”。

实例 60 输入框的消失

实例说明

本例运行效果如图 60-1 所示。这是一个使输入框消失的例子。

程序运行后，在输入框重输入文字，然后单击按钮，输入框的框架会自动消失。

本例的知识点有：CSS 中的基本定位属性，以及 JavaScript 脚本语言知识。



图 60-1 效果图

编程思路

本实例使用 CSS 中的基本定位属性对文字进行定位，使用 JavaScript 脚本语言控制输入框的消失。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 60</title>
<body>
<script>
if(document.layers)
{ document.writeln ('<STYLE>');
  document.writeln (' .changeable { position: relative; }');
  document.writeln ('<VSTYLE>');
}
function setElement (id, html)
{ if(document.layers)
  { var l=document[id];
```

```
if(!l.overLayer)
{l.overLayer=new Layer(l.clip.width);
l.overLayer.left=l.pageX;
l.overLayer.top=l.pageY;
l.visibility='hide';
l.overLayer.visibility='show';
}
l.overLayer.document.open();
l.overLayer.document.write(html);
l.overLayer.document.close();
}
else if(document.all)
{document.all[id].innerHTML=html;
}
}
function changeNow()
{var user = document.aForm.user.value
document.aForm.aField.value = "Welcome "+user+"";
setElement ('someParagraph', document.aForm.aField.value);
}
var nw = "";
document.write('<FORM NAME="aForm">');
document.write('<INPUT TYPE="hidden" NAME="aField" VALUE="">');
document.write('<SPAN ID="someParagraph" CLASS="changeable">');
document.write('<input type=text name=user size=15> <INPUT TYPE="button" VALUE="ok"
ONCLICK="changeNow();>');
document.write('</SPAN></form>');
</script>
</body>
</html>
```

代码键入完毕，存盘为“实例 60.html”。

二、运行程序

双击文件“实例 60.html”。

实例 61 下雨背景

实例说明

本例运行效果如图 61-1 所示。这是一个产生下雨背景的例子。

程序运行后，会随机出现雨点效果的背景。

本例的知识点有：CSS 中的基本属性，以及 JavaScript 脚本语言知识。



图 61-1 效果图

编程思路

本实例使用 CSS 基本属性控制雨点的颜色和位置，使用 JavaScript 脚本语言控制雨点的产生。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 61</title>
<body>
<script language="JavaScript">
<!--
var no = 50;
var speed=1;
var ns4up=(document.layers)?1:0;
var ie4up=(document.all)?1:0;
var s,x,y,sn,cs;
```

```
var a,r,cx,cy;
var i,doc_width=800,doc_height=600;
if(ns4up)
{doc_width = self.innerWidth;
 doc_height = self.innerHeight;
}
else
{if(ie4up)
{doc_width = document.body.clientWidth;
 doc_height = document.body.clientHeight;
}
}
x=new Array();
y=new Array();
r=new Array();
cx=new Array();
cy=new Array();
s=8;
for(i=0;i<no;+i)
{initRain();
if(ns4up)
{if(i==0)
{document.write("<layer name='dot"+ i +"\" left='1\" ");
 document.write("top='1\" visibility='show\"><font color='red\">");
 document.write("</font></layer>");
}
else
{document.write("<layer name='dot"+ i +"\" left='1\" ");
 document.write("top='1\" visibility='show\"><font color='red\">");
 document.write("</font></layer>");
}
}
}
else
{if(ie4up)
{if(i==0)
{document.write("<div id='dot"+ i +"\" style='POSITION: ");
 document.write("absolute; Z-INDEX: "+ i +"; VISIBILITY: ");
 document.write("visible; TOP: 15px; LEFT: 15px;\"><font color='red\">");
 document.write("</font></div>");
}
}
}
```



```
document.layers["dot"+i].left=x[i];
}
setTimeout("raindropNS()", speed);
}
function raindropIE()
{for(i=0;i<no;+i)
{updateRain();
if((x[i]<=1)||(x[i]>=(doc_width-20))||(y[i]>=(doc_height-20)))
{makeRain();
doc_width=document.body.clientWidth;
doc_height=document.body.clientHeight;
}
document.all["dot"+i].style.pixelTop=y[i];
document.all["dot"+i].style.pixelLeft=x[i];
}
setTimeout("raindropIE()", speed);
}
if(ns4up)
{raindropNS();
}
else
if(ie4up)
{raindropIE();
}
-->
</script>
</body>
</html>
```

代码键入完毕，存盘为“实例 61.html”。

二、运行程序

双击文件“实例 61.html”。

实例 62 水韵效果

实例说明

本例运行效果如图 62-1 所示。这是一个产生水韵效果的例子。

程序运行后,在页面上随机出现水韵效果的光圈。

本例的知识点有:CSS 中的基本属性,以及 JavaScript 脚本语言知识。



图 62-1 效果图

编程思路

本实例使用 CSS 中的定位属性对图片进行定位。使用 JavaScript 脚本语言控制产生水韵效果。

创作步骤

一、创建脚本代码

```

I=new Image();
I.src="0109ring.gif";
if (document.all){
document.write('<div id="C" style="position:absolute;top:0px;left:0px">');
document.write('<div style="position:relative">');
document.write('');
document.write('</div></div>');
}
S=null,fadeStep=4,fade=80,currentStep=0,step=2,R,Y=0,RX=0,Yarea=0,Xarea=0;
function Expand(){
if (document.all){
pic.width=currentStep*2;
pic.height=currentStep*2;
pic.style.top=-currentStep;
pic.style.left=-currentStep;
pic.style.filter='alpha(opacity='+fade+')';

```

```

currentStep+=step;
if (currentStep > 20) fade=fadeStep;
if (fade < -50)
{
currentStep=0;
fade=80;
Yarea=window.document.body.clientHeight-80;
Xarea=window.document.body.clientWidth-80;
RY=Math.round(50+Math.random()*Yarea);
RX=Math.round(50+Math.random()*Xarea);
C.style.top=RY+document.body.scrollTop;
C.style.left=RX+document.body.scrollLeft;
}
S=setTimeout('Expand()',40);
}
}
if (document.all) window.onload=Expand;
if (document.layers)
{
alert(" Your Browser is not capable of displaying this effect.\nPick another.");
opener.gO():window.close();
}

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 62</title>
<body bgcolor=#000000 onload="if (document.all)SmokeRing()">
<script language="JavaScript">
<!--
键入以上脚本代码。
// -->
</script>
<div style="position:relative"></div>
</body></html>

```

代码键入完毕，存盘为“实例 62.html”。

三、运行程序

双击文件“实例 62.html”。

实例 63 星空漫步

实例说明

本例运行效果如图 63-1 所示。这是一个模拟太空漫步的例子。

程序运行后,宇航员图片在屏幕中不停地移动,以模拟太空漫步。

本例的知识点有:CSS 中的定位属性,以及 JavaScript 脚本语言知识。



图 63-1 效果图

编程思路

本实例使用 CSS 中的定位属性对图片进行定位,使用 JavaScript 脚本语言控制图片的移动。

创作步骤

一、创建源代码

打开记事本,键入如下代码:

```
<html>
<title>实例 63</title>
<body bgcolor="black" onLoad="space()">
<script language="JavaScript">
<!--
astropic="01105.jpg";
var xval=new Array();
var yval=new Array();
stars=100;
speed=200;
xp=20+100*Math.random();
yp=20+100*Math.random();
angle=3.14 * Math.random();
xinc=3*Math.sin(angle);
```

```
yinc=3;
ie=(document.all)? 1 : 0;
ns=(document.layers)? 1 : 0;
if(ie)
{width=document.body.clientWidth;
height=document.body.clientHeight;
for(var i=0;i<=stars;i++)
{document.write("<div id=\"div\" + i+ \"\" style=\"color:white;");
document.write("position:absolute;left:100;\"><div>");
}
document.write("<div id=\"diva\" style=\"position:absolute;\">");
document.write("</div>");
for(i=0;i<=stars;i++)
{xval[i]=width*Math.random();
yval[i]=height*Math.random();
document.all["div" + i].style.left=xval[i];
document.all["div" + i].style.top=yval[i];
}
}
if(ns)
{width=self.innerWidth;
height=self.innerHeight;
for(var i=0;i<=stars;i++)
{document.write("<layer id=\"div\" + i+ \"\" style=\"color:white;");
document.write("position:absolute;left:100;\"></layer>");
}
document.write("<layer id=\"diva\" style=\"position:absolute;\">");
document.write("</layer>");
for(i=0;i<=stars;i++)
{xval[i]=width*Math.random();
yval[i]=height*Math.random();
document.layers["div" + i].left=xval[i];
document.layers["div" + i].top=yval[i];
}
}
function starf()
{for(i=0;i<=stars;i=i+4)
{xval[i]+=5;
if((i%20)==0) xval[i]+=10;
```

```
if(xval[i]>=width) xval[i]=width*Math.random();
if(ie) document.all["div" + i].style.left=xval[i];
if(ns) document.layers["div" + i].left=xval[i];
}
return true;
}
function goangle()
{angle=3.14*Math.random();
xinc=3*Math.sin(angle);
return true;
}
function space()
{if(ie)
{document.all["diva"].style.left = xp;
document.all["diva"].style.top = yp;
}
if(ns)
{document.layers["diva"].left = xp;
document.layers["diva"].top = yp;
}
xp += xinc;
yp += yinc;
star();
if((xp>=(width-50))||(xp<=10))
{goangle();
xinc = ( 1) * xinc;
}
if((yp>=(height-50))||(yp<=10))
{goangle();
yinc=(-1)*yinc;
}
setTimeout('space()', speed);
}
//-->
</script></body></html>
```

代码键入完毕，存盘为“实例 63.html”。

二、运行程序

双击文件“实例 63.html”。

实例 64 页面打开方式(一)

实例说明

本例运行效果如图 64-1 所示。这是一个屏幕左右展开的例子。

程序运行后,屏幕将以左右展开的方式打开。

本例的知识点有: CSS 中的颜色属性,以及 JavaScript 脚本语言。

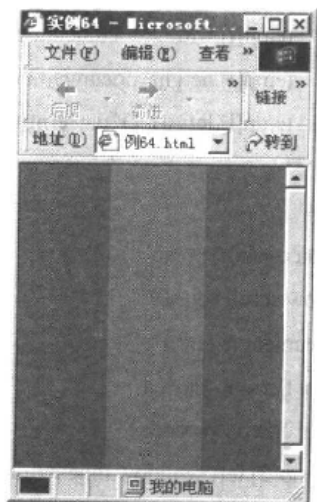


图 64-1 效果图

编程思路

本实例使用 CSS 中的颜色属性,定义初始颜色,使用 JavaScript 脚本语言控制屏幕的左右展开。

创作步骤

一、创建脚本代码

```

var speed=20
var temp=new Array()
var temp2=new Array()
if (document.layers)
{for (i=1;i<=2;i++)
{temp[i]=eval("document.i"+i+".clip")
temp2[i]=eval("document.i"+i)
temp[i].width=window.innerWidth/2
temp[i].height=window.innerHeight
temp2[i].left=(i-1)*temp[i].width
}
}

```

```

else if(document.all)
    { var clipright=document.body.clientWidth/2,clipleft=0
      for(i=1;i<=2;i++)
      { temp[i]=eval("document.all.i"+"i+".style")
        temp[i].width=document.body.clientWidth/2
        temp[i].height=document.body.offsetHeight
        temp[i].left=(i-1)*parseInt(temp[i].width)
        }
      }
function openit()
{ window.scrollTo(0,0)
  if(document.layers)
  { temp[1].right-=speed
    temp[2].left+=speed
    if(temp[2].left>window.innerWidth/2)
      clearInterval(stopit)
    }
  else if(document.all)
    { clipright-=speed
      temp[1].clip="rect(0 "+clipright+" auto 0)"
      clipleft+=speed
      temp[2].clip="rect(0 auto auto "+clipleft+")"
      if(clipright<=0)
        clearInterval(stopit)
      }
  }
function go()
{ stopit=setInterval("openit()",100)
  }
go()

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<head>
<title>实例 64</title>
<style>
<!--
.style

```

```
{position:absolute;
left:0;
top:0;
layer-background-color:red;
background-color:red;
border:0.1px solid red
|
-->
</style>
</head>
<body bgcolor="#009999">
<div id="i1" class="style">
</div>
<div id="i2" class="style1">
</div>
<script language="JavaScript1.2">
键入以上脚本代码。
</script>
</body>
</html>
```

代码键入完毕，存盘为“实例 64.html”。

三、运行程序

双击文件“实例 64.html”。

实例 65 页面打开方式(二)

实例说明

本例运行效果如图 65-1 所示。这是一个屏幕缩放的例子。

程序运行后, 屏幕将以从大到小的缩放方式的展开。

本例的知识点有: CSS 中的颜色属性, 以及 JavaScript 脚本语言。

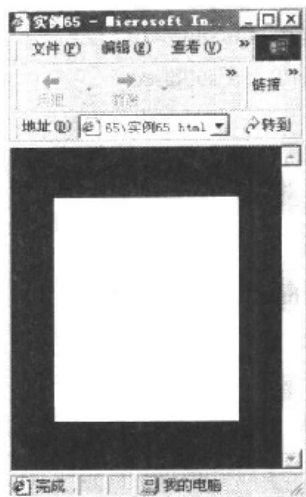


图 65-1 效果图

编程思路

本实例使用 CSS 中的颜色属性, 定义初始颜色, 使用 JavaScript 脚本语言控制屏幕的缩放展开。

创作步骤

一、创建脚本代码

```
var speed=5
if(document.layers)
{var reference=window.innerWidth/window.innerHeight
var temp=eval("document.all.clip")
temp.left=temp.top=0
temp.right=window.innerWidth
temp.bottom=window.innerHeight
}
else if(document.all)
{var reference=document.body.clientWidth/document.body.clientHeight
var rightlip,leftlip,toplip,bottomlip
var temp=document.all[1].style
```

```

    topclip=leftclip=0
    rightclip=temp.width=document.body.clientWidth
    bottomclip=temp.height=document.body.clientHeight
  }
function doit()
{ window.scrollTo(0,0)
  if(document.layers)
  { if(temp.left>window.innerWidth/2)
    clearInterval(stopit)
    temp.left+=reference*speed
    temp.top+=speed
    temp.right-=reference*speed
    temp.bottom-=speed
  }
  else if(document.all)
  { if(leftclip>document.body.clientWidth/2)
    clearInterval(stopit)
    temp.clip="rect "+topclip+" "+rightclip+" "+bottomclip+" "+leftclip+" "
    leftclip+=reference*speed
    topclip+=speed
    rightclip-=reference*speed
    bottomclip-=speed
  }
}
stopit=setInterval("doit()",100)

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<head>
<title>实例 65</title>
<style>
<!--
.style
{ position:absolute;
  left:0;
  top:0;
  layer-background-color:white;
  background-color:white;

```

```
border:0.1px solid white
}
-->
</style>
</head>
<body bgcolor="#000000">
<div id="i1" class="style">
</div>
<script language="JavaScript1.2">
键入以上脚本代码。
</script>
</body>
</html>
```

代码键入完毕，存盘为“实例 65.html”。

三、运行程序

双击文件“实例 65.html”。

实例 66 页面打开方式(三)

实例说明

本例运行效果如图 66-1 所示。这是一个百叶窗式的打开方式的例子。

程序运行后,屏幕将以百叶窗式的展开方式打开。

本例的知识点有: CSS 中的颜色属性,以及 JavaScript 脚本语言。

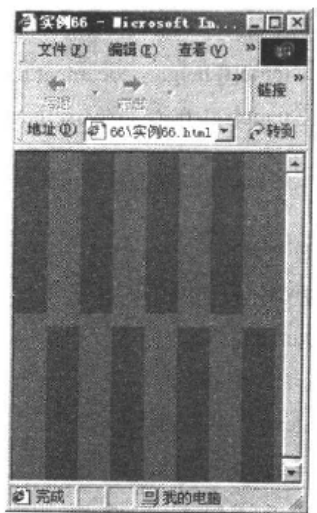


图 66-1 效果图

编程思路

本实例使用 CSS 中的颜色属性,定义初始颜色,使用 JavaScript 脚本语言控制屏幕以百叶窗式展开。

创作步骤

一、创建脚本代码

```

var speed=20
var temp=new Array()
var temp2=new Array()
if(document.layers)
{for(i=1;i<=8;i++)
{temp[i]=eval("document.i"+"i"+".clip")
temp2[i]=eval("document.i"+"i")
temp[i].width=window.innerWidth/8-0.3
temp[i].height=window.innerHeight
temp2[i].left=(i-1)*temp[i].width
}
}

```



```
else if(document.all)
    { var clipbottom=document.body.offsetHeight,cliptop=0
      for(i=1;i<=8;i++)
        { temp[i]=eval("document.all.i"+"."+style")
          temp[i].width=document.body.clientWidth/8
          temp[i].height=document.body.offsetHeight
          temp[i].left=(i-1)*parseInt(temp[i].width)
        }
      }
function openit()
{ window.scrollTo(0,0)
  if(document.layers)
    { for(i=1;i<=8;i=i+2)
      temp[i].bottom-=speed
      for(i=2;i<=8;i=i+2)
        temp[i].top+=speed
      if(temp[2].top>window.innerHeight)
        clearInterval(stopit)
    }
  else if(document.all)
    { clipbottom-=speed
      for(i=1;i<=8;i=i+2)
        { temp[i].clip="rect(0 auto+"+clipbottom+" 0)"
        }
      cliptop+=speed
      for(i=2;i<=8;i=i+2)
        { temp[i].clip="rect("+cliptop+" auto auto)"
        }
      if(clipbottom<=0)
        clearInterval(stopit)
    }
}
function go()
{ stopit=setInterval("openit()",100)
  |
  go()
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 66</title>
<head>
<style>
<!--
.style
{position:absolute;
left:0;
top:0;
layer-background-color:green;
background-color:green;
border:0.1px solid green
}
-->
</style>
<body bgcolor="#009999">
<div id="i1" class="style">
</div>
<div id="i2" class=" style ">
</div>
<div id="i3" class=" style">
</div>
<div id="i4" class="style">
</div>
<div id="i5" class="style">
</div>
<div id="i6" class="style">
</div>
<div id="i7" class="style">
</div>
<div id="i8" class="intro">
</div>
<script language="JavaScript1.2">
键入以上脚本代码。
</script></body></head></html>
代码键入完毕，存盘为“实例 66.html”。
```

三、运行程序

双击文件“实例 66.html”。

实例 67 刷新页面

实例说明

本例运行效果如图 67-1 所示。这是一个刷新页面的例子。

程序运行后，在输入框中输入文字，然后单击按钮，刷新文本框中的文字，单击“R”、“G”、“B”可以改变文本框中文字的颜色。

本例的知识点有：CSS 的基本属性，以及 JavaScript 脚本语言知识。



图 67-1 效果图

编程思路

本实例使用 CSS 的基本属性，定义页面和控件的样式，使用 JavaScript 脚本语言控制输入框的输入字数。

创作步骤

一、创建脚本代码

```
function Init() //初始化
{
    document.all.cmb1.innerHTML=traintext.value;
}
function Judge() //判断函数
{
    var string=traintext.value;
    if(string.length>15)
    {
        alert("Please input 15 characters or less.");
        traintext.value="";
    }
    else
    {
        document.all.cmb1.innerText=traintext.value;
    }
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 67</title>
<head>
<style type="text/css">
textarea
{font:"宋体";
}
input
{font:18px "宋体";
background:yellow;
color:purple;
height:30px;
}
body
{background:silver;
}
span
{font-family:"wide latin";
font-size:36px;
}
.mystyle1
{position:absolute;
top:30px;
left:10px;
font-family:Arial;
margin-left:-2px;
font-size:60px;
text-align:center;
color:black;
}
.mystyle1
{position:absolute;
top:30px;
left:10px;
font-family:Arial;
margin-left:-8px;
font-size:60px;
```



```
text-align:center;
color:white;
}
.textbox
{position:absolute;
top:142px;
left:10px;
font-family:Georgin;
font size:26px;
text-align:center;
color:purple;
}
</style>
<script language="JavaScript">
<!--
键入以上脚本代码,
//-->
</script>
</head>
<body onload="Init()">
<textarea name=cmb1 rows=1 cols=15 style="background-color: rgb(0,0,0); color: rgb(255,255,255); cursor:
default; font-family: Arial; font-size: 30px"></textarea>
<div class="textbox">
<dd>
<input type="text" name="traintext" value="Change me!" size="15">
<input type="button" name="btn1" value="Change it!" onclick="Judge()">
<p>
<span style="color:red;cursor:hand;font-size:30px" onclick="cmb1.style.color='#ff0000'">R</span>
<span style="color:lime;cursor:hand;font-size:30px" onclick="cmb1.style.color='#00ff00'">G</span>
<span style="color:blue;cursor:hand;font-size:30px" onclick="cmb1.style.color='#0000ff'">B</span>
</p>
</dd>
</div>
<div id="cmd1 class="mystyle1"></div>
</body></html>
```

代码键入完毕,存盘为“实例 67.html”。

三、运行程序

双击文件“实例 67.html”。

实例 68 任意改变页面样式

实例说明

本例运行效果如图 68-1 所示。这是一个可以动态改变页面样式的例子。

程序运行后,单击表格中的样式文字会改变文本框中时钟文字的样式。

本例的知识点有: CSS 的基本属性,以及 JavaScript 脚本语言知识。



图 68-1 效果图

编程思路

本实例使用 CSS 中的基本属性,定义页面的样式,使用 JavaScript 脚本语言控制时钟的显示。

创作步骤

一、创建脚本代码

```
function run() //时钟函数
{
    var hour,minute,second;
    var temp_h,temp_m,temp_s;
    var today;
    today=new Date();
    temp_h=today.getHours();
    temp_m=today.getMinutes();
    temp_s=today.getSeconds();
    if(temp_h<10)
        hour="0"+temp_h+":";
    else
        hour=temp_h+":";
    if(temp_m<10)
```

```

minute="0"+temp_m+":";
else
    minute=temp_m+":";
if(temp_s<10)
    second="0"+temp_s+" ";
else
    second=temp_s+" ";
string=hour+minute+second;
Clock.innerHTML=string;
window.setInterval("run();",100);
}
window.onload=run;

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 68</title>
<head>
<script language="JavaScript">
<!--
键入以上脚本代码。
//-->
</script>
</head>
<body>
<textarea name=Clock rows=1 cols=15 style="background-color:
    rgb(0,0,0);color:rgb(255,255,255);cursor:default;"></textarea>
<table border="1">
<tr>
<td style="color='blue'" >前景色</td>
<td style="color='blue'" >背景色</td>
<td style="color='blue'" >字体</td>
<td style="color='blue'" >字体大小</td>
<td style="color='blue'" >字体粗细</td>
</tr>
<tr>
<td><span onmouseover="this.style.cursor='hand'" onmouseout="this.style.cursor='default'"
    onclick="Clock.style.color='red'">红色</span>
</td>

```

```
<td><span onmouseover="this.style.cursor='hand'" onmouseout="this.style.cursor='default'"
      onclick="Clock.style.background='black'">黑色</span>
</td>
<td><span onmouseover="this.style.cursor='hand'" onmouseout="this.style.cursor='default'"
      onclick="Clock.style.fontFamily='宋体'">宋体</span>
</td>
<td><span onmouseover="this.style.cursor='hand'" onmouseout="this.style.cursor='default'"
      onclick="Clock.style.fontSize='12px'">小字体</span>
</td>
<td><span onmouseover="this.style.cursor='hand'" onmouseout="this.style.cursor='default'"
      onclick="Clock.style.fontWeight='normal'">普通</span>
</td>
</tr>
<tr>
<td><span onmouseover="this.style.cursor='hand'" onmouseout="this.style.cursor='default'"
      onclick="Clock.style.color='yellow'">黄色</span>
</td>
<td><span onmouseover="this.style.cursor='hand'" onmouseout="this.style.cursor='default'"
      onclick="Clock.style.background='white'">白色</span>
</td>
<td><span onmouseover="this.style.cursor='hand'" onmouseout="this.style.cursor='default'"
      onclick="Clock.style.fontFamily='verdana'">verdana</span>
</td>
<td><span onmouseover="this.style.cursor='hand'" onmouseout="this.style.cursor='default'"
      onclick="Clock.style.fontSize='48px'">大字体</span>
</td>
<td><span onmouseover="this.style.cursor='hand'" onmouseout="this.style.cursor='default'"
      onclick="Clock.style.fontWeight='bold'">粗体</span>
</td>
</tr>
</table>
</body>
</html>
```

代码键入完毕，存盘为“实例 68.html”。

三、运行程序

双击文件“实例 68.html”。

实例 69 跟随鼠标的文字

实例说明

本例运行效果如图 69-1 所示,这是一个产生跟随鼠标文字的例子。

程序运行后,任意移动鼠标,会有跟随的文字出现。

本例的知识点有:CSS 中的基本文本属性,以及 JavaScript 脚本语言知识。



图 69-1 效果图

编程思路

本实例运用 CSS 中的基本文本属性如: font-size, font-weight 等,定义跟随文字的样式,使用 JavaScript 脚本语言来控制文字跟随鼠标。

创作步骤

一、创建脚本代码

```
var x,y
var step=20
var flag=0
var message="Hello World!"
message=message.split("")
var xpos=new Array()
for(i=0;i<=message.length-1;i++)
{xpos[i]=-50
}
var ypos=new Array()
for(i=0;i<=message.length-1;i++)
{ypos[i]=-50
```

```

}
function handlerMM(e)
{x = (document.layers) ? e.pageX : document.body.scrollLeft+event.clientX
 y = (document.layers) ? e.pageY : document.body.scrollTop+event.clientY
 flag=!
}
function makesnake()
{if (flag==1 && document.all)
 {for (i=message.length-1; i>=1; i--)
  {xpos[i]=xpos[i-1]+step
   ypos[i]=ypos[i-1]}
  }
 xpos[0]=x+step
 ypos[0]=y
 for (i=0; i<message.length-1; i++)
 {var thisspan = eval("span"+(i)+".style")
  thisspan.posLeft=xpos[i]
  thisspan.posTop=ypos[i]
 }
}
else if (flag==1 && document.layers)
 {for (i=message.length-1; i>=1; i--)
  {xpos[i]=xpos[i-1]+step
   ypos[i]=ypos[i-1]}
  }
 xpos[0]=x+step
 ypos[0]=y
 for (i=0; i<message.length-1; i++)
 {var thisspan = eval("document.span"+i)
  thisspan.left=xpos[i]
  thisspan.top=ypos[i]
 }
}
var timer=setTimeout("makesnake()",30)
}

```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
```

```

<title>实例 69</title>
<head>
<style>
.spanstyle
{position: absolute;
visibility: visible;
top: -50px;
font-size: 10pt;
font-family: Verdana;
font-weight: bold;
color: black;
}
</style>
<script>
键入以上脚本代码。
</script>
</head>
<body onLoad= 'makesnake()' style="width: 100%;overflow-x: hidden;overflow-y: scroll">
<script language=javascript>
<!--
for(i=0;i<=message.length-1;i++)
(document.write("<span id='span"+i+"' class='spanstyle'>")
document.write(message[i])
document.write("</span>"))
}
if(document.layers)
(document.captureEvents(Event.MOUSEMOVE);
)
document.onmousemove = handlerMM;
/-->
</script>
</body>
</html>

```

键入完毕，存盘为“实例 69.html”。

三、运行程序

双击文件“实例 69.html”。

实例 70 动态鼠标指针

实例说明

本例运行效果如图 70-1 所示,这是一个动态鼠标指针的例子。

程序运行后,在鼠标周围产生几个彩色小球,这些小球会随着鼠标的移动而移动。

本例的知识点有:CSS 的基本颜色属性和定位属性,以及 JavaScript 脚本语言知识。



图 70-1 效果图

编程思路

本实例使用 CSS 基本属性,定义小球的样式,使用 JavaScript 脚本语言控制小球的移动。

创作步骤

一、创建脚本语言

```
function YY_Layerfx(yyleft,yytop,yyfnx,yyfny,yydiv,yybilder,yyloop,yyto,yyent,yystep)
{if ((document.layers)||(document.all))
{with (Math)
{yynextx= eval(yyfnx)
}
with (Math)
{yynexty= eval(yyfny)
}
yyent=(yyloop && yyent>=yystep*yybilder)?0:yyent+yystep;
if(document.layers)
{eval(yydiv+".top="+yyent*yystep)
eval(yydiv+".left="+yynextx+yyleft)}
```



```

    }
    if(document.all)
    (eval("yydiv=yydiv.replace(/.layers/gi, '.all')");
    eval(yydiv+".style.pixelTop="+(+yynexty+yytop));
    eval(yydiv+".style.pixelLeft="+(+yynextx+yyleft));
    }
    argStr='YY_Layerfx('+yyleft+', '+yytop+', '+yyfnx+', '+yyfny+', '+yydiv+', '+yybilder+', '+yyloop+', '+y
        yto+', '+yycnt+', '+yystep+')';
    if(yycnt<=yystep*yybilder)
    {eval(yydiv+".yyto=setTimeout(argStr,yyto)");
    }
    }
}
function YY_Mousetrace(evt)
{if(yyns4)
    {if(evt.pageX)
        {yy_ml=evt.pageX;
        yy_mt=evt.pageY;
        }
    }
else
    {yy_ml=(event.clientX + document.body.scrollLeft);
    yy_mt=(event.clientY + document.body.scrollTop);
    }
if(yy_tracescript)
    eval(yy_tracescript)
}

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 70</title>
<head>
<script language="JavaScript">
键入以上脚本代码。
</script>
</head>
<body>
<div id="yyd0"

```

```
style="background-color:#19636c;border-bottom:#000000 1px;
border-left:#000000 1px;border-right:#000000 1px;
border-top:#000000 1px;clip:rect(0px 3px 3px 0px);height:3px;
left:10px;position:absolute;top:50px;
width:3px;z-index:1;layer-background-color:#19636c">
</div>
<div id="yyd1"
style="background-color:#708574;border-bottom:#000000 1px;
border-left:#000000 1px;border-right:#000000 1px;
border-top:#000000 1px;clip:rect(0px 3px 3px 0px);height:3px;
left:20px;position:absolute;top:50px;
width:3px;z-index:1;layer-background-color:#708574">
</div>
<div id="yyd2"
style="background-color:#379bbf;border-bottom:#000000 1px;
border-left:#000000 1px;border-right:#000000 1px;
border-top:#000000 1px;clip:rect(0px 3px 3px 0px);height:3px;
left:30px;position:absolute;top:50px;
width:3px;z-index:1;layer-background-color:#379bbf">
</div>
<div id="yyd3"
style="background-color:#25184c;border-bottom:#000000 1px;
border-left:#000000 1px;border-right:#000000 1px;
border-top:#000000 1px;clip:rect(0px 3px 3px 0px);height:3px;
left:40px;position:absolute;top:50px;
width:3px;z-index:1;layer-background-color:#25184c">
</div>
<div id="yyd4"
style="background-color:#31bd3c;border-bottom:#000000 1px;
border-left:#000000 1px;border-right:#000000 1px;
border-top:#000000 1px;clip:rect(0px 3px 3px 0px);height:3px;
left:50px;position:absolute;top:50px;
width:3px;z-index:1;layer-background-color:#31bd3c">
</div>
<div id="yyd5"
style="background-color:#c11efd;border-bottom:#000000 1px;
border-left:#000000 1px;border-right:#000000 1px;
border-top:#000000 1px;clip:rect(0px 3px 3px 0px);height:3px;
left:60px;position:absolute;top:50px;
```

```

width:3px;z-index:1;layer-background-color:#c11efd">
</div>
<script>
var yyns4=window.Event?true:false;
var yy_mt=0;
var yy_ml=0;
document.onmousemove=YY_Mousetrace;
yy_tracescript = "";
if (yyns4)
{document.captureEvents(Event.MOUSEMOVE);
  YY_Mousetrace(",document.YY_Mousetrace1")
}
YY_Layerfx(0,0,'yy_ml+cos((15*sin(yycnt/39.83007847812662))+0)*150*(sin(10+yycnt/20)+0.2)*cos(yycnt/20)',yy_mt+sin((15*sin(yycnt/34.224861639800686))+0)*150*(sin(10+yycnt/20)+0.2)*cos(yycnt/20)',document.layers['yyd0'],2000,true,80,0,1);
YY_Layerfx(0,0,'yy_ml+cos((15*sin(yycnt/27.66510707209673))+30)*150*(sin(10+yycnt/20)+0.2)*cos(yycnt/20)',yy_mt+sin((15*sin(yycnt/9.240632767417667))+30)*150*(sin(10+yycnt/20)+0.2)*cos(yycnt/20)',document.layers['yyd1'],2000,true,80,0,1);
YY_Layerfx(0,0,'yy_ml+cos((15*sin(yycnt/16.45318944579641))+60)*150*(sin(10+yycnt/20)+0.2)*cos(yycnt/20)',yy_mt+sin((15*sin(yycnt/16.0564452288292))+60)*150*(sin(10+yycnt/20)+0.2)*cos(yycnt/20)',document.layers['yyd2'],2000,true,80,0,1);
YY_Layerfx(0,0,'yy_ml+cos((15*sin(yycnt/6.95348954836835))+90)*150*(sin(10+yycnt/20)+0.2)*cos(yycnt/20)',yy_mt+sin((15*sin(yycnt/44.13697049887155))+90)*150*(sin(10+yycnt/20)+0.2)*cos(yycnt/20)',document.layers['yyd3'],2000,true,80,0,1);
YY_Layerfx(0,0,'yy_ml+cos((15*sin(yycnt/33.90077294583733))+120)*150*(sin(10+yycnt/20)+0.2)*cos(yycnt/20)',yy_mt+sin((15*sin(yycnt/2.2378828869411587))+120)*150*(sin(10+yycnt/20)+0.2)*cos(yycnt/20)',document.layers['yyd4'],2000,true,80,0,1);
YY_Layerfx(0,0,'yy_ml+cos((15*sin(yycnt/37.858312521039835))+150)*150*(sin(10+yycnt/20)+0.2)*cos(yycnt/20)',yy_mt+sin((15*sin(yycnt/18.08383979599098))+150)*150*(sin(10+yycnt/20)+0.2)*cos(yycnt/20)',document.layers['yyd5'],2000,true,80,0,1);
</script>
</body>
</html>

```

代码键入完毕，存盘为“实例 70.html”。

三、运行程序

双击文件“实例 70.html”。

实例 71 流星攻击鼠标

实例说明

本例运行效果如图 71-1 所示。这是一个流星尾随鼠标的例子。

程序运行后,有一串流星追随鼠标直到打到鼠标为止。

本例的知识点有: CSS 中的定位属性,以及 JavaScript 脚本语言知识。



图 71-1 效果图

编程思路

本实例使用 CSS 中的定位属性对流星进行位置控制,使用 JavaScript 脚本语言控制流星的产生和消失。

创作步骤

一、创建源代码

打开记事本,键入如下代码:

```
<html>
<title>实例 71</title>
<head>
<script language="JavaScript">
<!--
var speed=20,divTop=0,divLeft=0,angle=0,ypos=0,xpos=0,amount=13;
if(document.layers)
{for(i=0;i<amount;i++)
{document.write('<layer name=nsa'+i+' top=0 left=0 width='+i/3+' height='+i/3+'
bgcolor=#ffffff></layer>');
}
window.captureEvents(Event.MOUSEMOVE);
```



```
function nsMouse(evt)
{
  xpos=evt.pageX;
  ypos=evt.pageY;
}
window.onMouseMove = nsMouse;
}
else if(document.all)
{
  document.write('<div style="position:absolute;top:0px;left:0px">');
  document.write('<div style="position:relative">');
  for(i=0;i<amount;i++)
  {
    document.write('<div id=k style="position:absolute;top:0px;left:0px;width:'+i/3+';height:'+i/3+';
      background:#ffff;font-size:'+i/3+'"></div>')
  }
  document.write('</div></div>');
  function iMouse()
  {
    xpos=event.x;
    ypos=document.body.scrollTop+event.y
  }
  document.onmousemove = iMouse;
}
function getAngle()
{
  Y=ypos-divTop;X=xpos-divLeft;angle=Math.round(Math.atan2(Y,X)*180/Math.PI);
  if(angle<0)
    angle+=360;
}
function bomb()
{
  if(document.layers)
  {
    Yscroll=window.pageYOffset;
    ybase=Math.round(Math.random()*window.innerHeight);
    xbase=Math.round(Math.random()*window.innerWidth);
    yybase=window.innerHeight-10;xxbase=window.innerWidth-10;
  }
  else if(document.all)
  {
    Yscroll=document.body.scrollTop;
    ybase=Math.round(Math.random()*window.document.body.offsetHeight);
    xbase=Math.round(Math.random()*window.document.body.offsetWidth);
    xxbase=window.document.body.offsetWidth-10;yybase=window.document.body.offsetHeight-10;
  }
  if((xbase>10&&xxbase<xbase)&&(ybase>10&&yybase<ybase))

```

```

    bomb();
else
{if((divLeft>xpos-speed/1.5)&&(divLeft<xpos+speed/1.5)&&(divTop>=ypos-speed/1.5)&&(divTop<=
    ypos+speed/1.5))
    {divTop=ybase+Yscroll;divLeft=xbase
    }
y=Math.round(speed*Math.sin(angle*Math.PI/180));
x=Math.round(speed*Math.cos(angle*Math.PI/180));
divTop+=y;divLeft+=x;
getAngle();
T=setTimeout('bomb()',20);
}
}
function Split()
{Cls=new Array('fff000','fffff','ffa500')
var ntscp=document.layers,msie=document.all;
if(document.layers)
{for(i=0;i<amount;i++)
    {if(i<amount-1)
        {ntscp['nsa'+i].top=ntscp['nsa'+(i+1)].top;
        ntscp['nsa'+i].left=ntscp['nsa'+(i+1)].left
        }
    else
        {ntscp['nsa'+i].top=divTop;
        ntscp['nsa'+i].left=divLeft
        }
    }
}
}
else if(document.all)
    {for(i=0;i<amount;i++)
        {if(i<amount-1)
            {msie.k[i].style.top=msie.k[i+1].style.top;
            msie.k[i].style.left=msie.k[i+1].style.left
            }
            else
                {msie.k[i].style.top=divTop;
                msie.k[i].style.left=divLeft
                }
            }
        }
}

```

```
    }  
    for(i=0;i<amount-1;i++)  
    { for(n=0;n<Clrs.length;n++)  
      {var x=Math.round(Math.random()*n)  
      }  
      if(document.layers)  
      {ntscp['nsa'+i].bgColor=Clrs[x]  
      }  
      else if(document.all)  
      {msie.k[i].style.background=Clrs[x]  
      }  
    }  
    setTimeout("Split()",20);  
  }  
  function StArT()  
  {bomb();  
  .Split()  
  }  
  window.onload=StArT;  
  //-->  
</script>  
</head>  
<body bgcolor=black>  
</body>  
</html>
```

代码键入完毕，存盘为“实例 71.html”。

二、运行程序

双击文件“实例 71.html”。

实例 72 鼠标痕迹

实例说明

本例运行效果如图 72-1 所示。这是一个产生鼠标痕迹的例子。

程序运行后，任意移动鼠标，在鼠标移动过程中会出现模糊的鼠标痕迹。

本例的知识点有：CSS 中的定位属性，以及 JavaScript 脚本语言知识。



图 72-1 效果图

编程思路

本实例使用 CSS 中的定位属性对鼠标痕迹进行定位，使用 JavaScript 脚本语言跟随鼠标确定痕迹。

创作步骤

一、创建源代码

打开记事本，输入如下代码。

```
<html>
<title>实例 72</title>
<head>
<style type="text/css">
body
{overflow:scroll;
overflow-x:hidden;
}
</style>
```



```

<script language="JavaScript 1.2">
<!--
var trailLength=8;
var path="cursor1.gif";
var isIE=false,isNav=false,range="all",style="style",i,d=0;
var topPix="pixelTop",leftPix="pixelLeft",images,storage;
if (document.layers)
{isNav=true,range="layers",style="",topPix="top",leftPix="left";
}
else if(document.all)
    {isIE = true;
    }
function initTrail()
{images=new Array();
for(i=0;i<parseInt(trailLength);i++)
    {images[i]=new Image();
    images[i].src=path;
    }
storage=new Array();
for(i=0;i<images.length*3;i++)
    storage[i] = 0;
for(i=0;i<images.length;i++)
    {(isIE)?document.write('<div id="obj" + i + "' style="position: absolute; z-Index: 100; height: 0;
    width: 0"></div>') : document.write('<layer name="obj" + i + "'
    width="0" height="0" z-index="100"></layer>');
    }
trail();
}
function trail()
{for(i=0;i<images.length;i++)
    {eval("document." + range + "obj" + i + style + topPix + "=" + storage[d]),
    eval("document." + range + "obj" + i + style + leftPix + "=" + storage[d+1]);
    d=d+2;
    }
for(i=storage.length;i>=2;i--)
    storage[i] = storage[i-2];
d=0;
clearTimeout(timer);
var timer=setTimeout("trail()", 10);
    }
    
```

```
    }  
    function processEvent(e)  
    {  
        if (isIE)  
        {  
            storage[0] = window.event.y+document.body.scrollTop+10;  
            storage[1] = window.event.x+document.body.scrollLeft+10;  
        }  
        else  
        {  
            storage[0] = e.pageY+12;  
            storage[1] = e.pageX+12;  
        }  
    }  
    }  
    if (isNav)  
    {  
        document.captureEvents(Event.MOUSEMOVE);  
    }  
    if (isIE || isNav)  
    {  
        initTrail();  
        document.onmousemove = processEvent;  
    }  
    // End -->  
    </script>  
    </head>  
    </html>
```

代码输入完毕，存盘为“实例 72.html”。

二、运行程序

双击文件“实例 72.html”。

实例 73 飞出来的字母

实例说明

本例运行效果如图 73-1 所示,这是一个飞出七彩字母的例子。

程序运行后,在任意位置单击鼠标会飞出七彩的字母。

本例的知识点有:CSS 中的基本文本属性,以及 JavaScript 脚本语言知识。



图 73-1 效果图

编程思路

本实例使用 CSS 中的基本文本属性,定义飞出来的字母的样式,使用 JavaScript 脚本语言控制字母的出现。

创作步骤

一、创建脚本代码

```
var ver=navigator.appVersion;
var dom=document.getElementById ? 1 : 0;
var ie5=(ver.indexOf("MSIE 5") > -1 && dom) ? 1 : 0;
var n=(document.layers);
var ie=(document.all);
var sparksAflyin=0;
var totalnumber=0;
var sparksOn=1;
function initMouseEvents()
```

```

[document.onmousedown = mouseDown;
  if(n) document.captureEvents(Event.MOUSEDOWN | Event.MOUSEMOVE);
]
function mouseDown(e) //鼠标按下处理函数
{if(sparksOn)
  {var mousex=(n) ? e.pageX : event.x+document.body.scrollLeft;
  var mousey=(n) ? e.pageY : event.y+document.body.scrollTop;
  if(!sparksAflyin)
  {for(var k=0;k<=4;k++) eval('show("sDiv'+k+'");');
  sparksAflyin=1;
  totalnumber=0;
  for(i=0;i<=4;i++)
    eval('moveTo('+i+',0,'+mousex+','+mousey+')');
  }
}
}
function moveTo(i,j, mousex, mousey)
{if(j<eval('anim_'+i+'_x.length'))
  {var tempx=eval('anim_'+i+'_x[j]+mousex');
  var tempy=eval('anim_'+i+'_y[j]+mousey');
  if(!ie)
  {if((tempy+30)>(document.body.offsetHeight+document.body.scrollTop))
    tempy = document.body.offsetHeight+document.body.scrollTop-30;
  if((tempx+30)>(document.body.offsetWidth+document.body.scrollLeft))
    tempx = document.body.offsetWidth+document.body.scrollLeft-30;
  eval('document.all.sDiv'+i+'.style.left = tempx;');
  eval('document.all.sDiv'+i+'.style.top = tempy;');
  }
  if(n)
  {eval('document.layers.sDiv'+i+'.left = tempx;');
  eval('document.layers.sDiv'+i+'.top = tempy;');
  }
  j++;
  setTimeout("moveTo('+i+', '"+j+', '"+mousex+', '"+mousey+')",50)
}
else
{eval('hide("sDiv'+i+'");');
totalnumber++;
}
}

```

```
if(totalnumber==5)
{ sparksAflyin=0;
totalnumber=0;
}
}
function show(divName) //显示函数
{if(document.all) eval('document.all.'+divName+'.style.visibility = "visible"');
else if(document.layers)
eval('document.layers["'+divName+'"].visibility = "visible"');
}
function hide(divName)
{if(document.all) eval('document.all.'+divName+'.style.visibility = "hidden"');
else if(document.layers)
eval('document.layers["'+divName+'"].visibility = "hide"');
}
anim_0_x=new Array(20,20,10,0,0,0,0,0,0,0,0);
anim_0_y=new Array(-20,-40,-60,-80,60,-40,20,0,20,40,60,80);
anim_1_x=new Array(20,20,17,36,60,78,90,92,93,98,108,120,133,152,181);
anim_1_y=new Array(-20,-20,-33,-38,-38,-27,-2,25,51,84,113,141,162,212,253);
anim_2_x=new Array(20,20,2,3,4,5,6,7,8,9,10,12,13,15,18);
anim_2_y=new Array(-20,-20,-33,-38,-38,-27,-2,25,51,84,113,141,162,212,253);
anim_3_x=new Array(-20,-20,-2,-1,7,10,18,35,60,102,94,94,93,97,108,111,117,127);
anim_3_y=new Array(-20,-25,-64,-89,-104,-150,-173,-197,-213,-199,-151,-101,-66,-17,27,87,140,189);
anim_4_x=new
Array(-20,-20,-10,-39,-30,-69,-64,-138,-154,-200,-181,-209,-191,-207,-203,-213,-202,-221,-211);
anim_4_y=new Array(-20,-20,-28,-51,-79,-100,-135,-154,-193,-183,-149,-134,-89,-60,8,51,107,157,201);
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 73</title>
<head>
<script language="JavaScript1.2">
键入以上脚本代码。
</script>
</head>
<body bgcolor="#f4d9" OnLoad="initMouseEvents()">
<div id="sparks">
<div id="sDiv0" style="position:absolute; visibility: hidden;">
```

```
<font face="arial black" color="red">a</font></div>
<div id="sDiv1" style="position:absolute; visibility: hidden;">
<font face="arial black" color="yellow">b</font></div>
<div id="sDiv2" style="position:absolute; visibility: hidden;">
<font face="arial black" color="blue">c</font></div>
<div id="sDiv3" style="position:absolute; visibility: hidden;">
<font face="arial black" color="red">d</font></div>
<div id="sDiv4" style="position:absolute; visibility: hidden;">
<font face="arial black" color="orange">e</font></div>
</div>
</body>
</html>
```

代码键入完毕，存盘为“实例 73.html”。

三、运行程序

双击文件“实例 73.html”。

实例 74 跟随鼠标跳动的文字

实例说明

本例运行效果如图 74-1 所示。这是一个跟随鼠标的文字的例子。

程序运行后,任意移动鼠标会有跳动的文字跟随鼠标。

本例的知识点有: CSS 的基本属性,以及 JavaScript 脚本语言知识。



图 74-1 效果图

编程思路

本实例使用 CSS 的基本属性,定义文字的样式,使用 JavaScript 脚本语言控制文字跟随鼠标并进行跳动。

创作步骤

一、创建脚本代码

```
var cx=0;
var cy=0;
var val=0;
function locate()
{cx=window.event.x;
cy=window.event.y;
}
document.onmousemove=locate;
function follow(i)
{var x;
```

```

if(i<4) x=cx-50+i*10;
else x=cx-25+i*10;
var y=cy-20+Math.floor(Math.random()*40);
w=eval("word"+i);
with(w.style)
{left=x.toString()+"px";
top=y.toString()+"px";
}
|
function show(i) //显示函数
{var w=eval("word"+i);
with(w.style)
{visibility="visible";
s=parseInt(fontSize);
if(s>=200) s=100;
else if(s>90&& s<=100)
{s-=85;
clearInterval(val);
if(i<7) val=setInterval("show("+i+1+")",20);
}
fontSize=s;
}
}
function start()
{for(i=1;i<=7;i++)
{val=setInterval("show(1)",20);
setInterval("follow("+i+")",100);
}
}
}

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 74</title>
<head>
<script language=javascript>
<!--
键入以上脚本代码。
//-->

```



```
</script>
</head>
<body>
<script language=javascript>
var word=new Array(7);
word[1]="w";
word[2]="c";
word[3]="1";
word[4]="c";
word[5]="o";
word[6]="m";
word[7]="e";
for(i=1;i<=7;i++)
document.write("<div id= word"+i+"
    style='width:20px;height:20px;position:absolute;font-size:1000;visibility:hidden'>
<font face='Forte' color='#cc0000'>"+word[i]+"</font></div>");
start();
</script>
</body>
</html>
```

代码键入完毕，存盘为“实例 74.html”。

三、运行程序

双击文件“实例 74.html”。

实例 75 跟随鼠标的叠影文字

实例说明

本例运行效果如图 75-1 所示。这是一个跟随鼠标的叠影文字的例子。

程序运行后，任意移动鼠标会有叠影文字跟随。

本例的知识点有：CSS 的定位属性，以及 JavaScript 脚本语言知识。



图 75-1 效果图

编程思路

本实例使用 CSS 的定位属性对文字进行定位，使用 JavaScript 脚本语言产生叠影效果。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 75</title>
<body>
<script language="JavaScript">
<!
message='HTML&&CSS';
Font='Verdana';
Color='223399';
Size=3;
```

```
var amount=5,ypos=-50,xpos=0,Ay=0,Ax=0,By=0,Bx=0,Cy=0,Cx=0,Dy=0,Dx=0,Ey=0,Ex=0;
if(document.all)
{ document.write("<div id='outer' style='position:absolute;top:0px;left:0px'>");
  document.write("<div style='position:relauve'>");
  for(i=0;i<amount;i++)
  { document.write("<div id='text"+i+'
    style='position:absolute;top:0px;left:0px;width:400px;height:20px'>
    <font face='+FonT+' s ize='+SizE+'
    color='+Color+'>'+message+'</font></div>")
  }
  document.write("</div>");
  document.write("</div>");
  function mouse()
  { ypos=event.y+20;
    xpos=event.x+20;
  }
  window.document.onmousemove=mouse;
}
function makefollow()
{if(document.all)
{ outer.style.pixelTop=document.body.scrollTop;
  text[0].style.pixelTop=ay;
  text[0].style.pixelLeft=ax;
  text[1].style.pixelTop=by;
  text[1].style.pixelLeft=bx;
  text[2].style.pixelTop=cy;
  text[2].style.pixelLeft=cx;
  text[3].style.pixelTop=Dy;
  text[3].style.pixelLeft=Dx;
  text[4].style.pixelTop=Ey;
  text[4].style.pixelLeft=Ex;
  }
}
function move()
{ ey=Ey+=(ypos-Ey)*0.2;
  ex=Ex+=(xpos-Ex)*0.2;
  dy=Dy+=(ey-Dy)*0.3;
  dx=Dx+=(ex-Dx)*0.3;
  cy=Cy+=(dy-Cy)*0.4;
```

```
cx=Cx+(dx-Cx)*0.4;
by=By+(cy-By)*0.5;
bx=Bx+(cx-Bx)*0.5;
ay=Ay+(by-Ay)*0.6;
ax=Ax+(bx-Ax)*0.6;
makefollow();
setTimeout('move()', 10);
}
window.onload=move;
//-->
</script>
</body>
</html>
```

代码键入完毕，存盘为“实例 75.html”。

二、运行程序

双击文件“实例 75.html”。

实例 76 跟随鼠标的时钟

实例说明

本例运行效果如图 76-1 所示。这是一个走动的时钟的例子。

程序运行后,会有一个时钟跟随鼠标移动。

本例的知识点有:CSS 中的基本文本属性,以及 JavaScript 脚本语言知识。

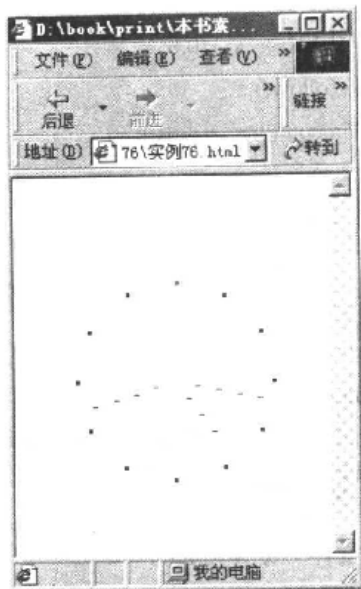


图 76-1 效果图

编程思路

本实例使用 CSS 中的基本属性,定义时钟的样式,使用 JavaScript 脚本语言控制时钟走动以及跟随鼠标移动。

创作步骤

一、创建源代码

打开记事本,键入如下代码:

```
<html>
<title>实例 76</title>
<body>
<script language="JavaScript">
<!--
colors=new Array('330099','00ff00','ff00ff');
sCol=black;
mCol=blue;
```

```

hCol=red';
H='-----';H=H.split("");H=H.reverse();
M='-----';M=M.split("");M=M.reverse();
S='-----';S=S.split("");S=S.reverse();
dots=12;
var Ypos=0,Xpos=0,Ybase=0,Xbase=0;
var ay=0,ax=0,Ay=0,Ax=0,by=0,bx=0,By=0,Bx=0,cy=0,cx=0,Cy=0,Cx=0,dy=0,dx=0,Dy=0,Dx=0;
count=0;
count_a=0;
move=1;
ns=(document.layers)?1:0;
viz=(document.layers)?'hide':'hidden';
document.write('<div id="W" style="position:absolute;top:0px;left:0px"><div style="position:relative">');
for(i=0;i<dots;i++)
{ document.write('<div id="face" style="position:absolute;top:0px;left:0px;width:3px;height:3px;
font-size:3px;background:#ffffff"></div>');
}
document.write('</div></div>');
document.write('<div id="X" style="position:absolute;top:0px;left:0px"><div style="position:relative">');
for(i=0;i<S.length;i++)
{ document.write('<div id="x" style="position:absolute;width:36px;height:36px;font-family:Verdana;
font-size:12px;color:'+sCol+';text-align:center;padding-top:10px">'+S[i]+'</div>');
}
document.write('</div></div>');
document.write('<div id="Y" style="position:absolute;top:0px;left:0px"><div style="position:relative">');
for(i=0;i<M.length;i++)
{ document.write('<div id="y" style="position:absolute;width:36px;height:36px;font-family:Verdana;
font-size:12px;color:'+mCol+';text-align:center;padding-top:10px">'+M[i]+'</div>');
}
document.write('</div></div>');
document.write('<div id="Z" style="position:absolute;top:0px;left:0px"><div style="position:relative">');
for(i=0;i<H.length;i++)
{ document.write('<div id="z" style="position:absolute;width:36px;height:36px;font-family:Verdana;
font-size:12px;color:'+hCol+';text-align:center;padding-top:10px">'+H[i]+'</div>');
}
document.write('</div></div>');
function Mouse()
{ Ypos=event.y+100;
Xpos=event.x+100;

```

```

    }
    document.onmousemove=Mouse;
    function clock()
    {time=new Date();
    secs=time.getSeconds();
    sec=-1.57+Math.PI*secs/30;
    mins=time.getMinutes();
    min=-1.57+Math.PI*mins/30;
    hr=time.getHours();
    hrs=-1.575+Math.PI*hr/6+Math.PI*parseInt(time.getMinutes())/360;
    Ybase=15;
    Xbase=15;
    var scrll=document.body.scrollTop;
    W.style.pixelTop=scrll;
    X.style.pixelTop=scrll;
    Y.style.pixelTop=scrll;
    Z.style.pixelTop=scrll;
    x[0].style.visibility=viz;
    y[0].style.visibility=viz;
    z[0].style.visibility=viz;
    for(i=0;i<S.length;i++)
    {x[i].style.pixelTop=ay-12+(i*Ybase)*Math.sin(sec);
    x[i].style.pixelLeft=ax-12+(i*Xbase)*Math.cos(sec);
    }
    for(i=0;i<M.length;i++)
    {y[i].style.pixelTop=by-12+(i*Ybase)*Math.sin(min);
    y[i].style.pixelLeft=bx-12+(i*Xbase)*Math.cos(min);
    }
    for(i=0;i<H.length;i++)
    {z[i].style.pixelTop=cy-12+(i*Ybase)*Math.sin(hrs);
    z[i].style.pixelLeft=cx-12+(i*Xbase)*Math.cos(hrs);
    }
    for(i=0;i<dots;++i)
    {face[i].style.pixelTop=dy+6+(70*Math.sin(-0.49+dots+i/1.9));
    face[i].style.pixelLeft=dx+4+(70*Math.cos(-0.49+dots+i/1.9));
    }
    }
    function MouseFollow()
    {ay=Math.round(Ay+((Ypos)-Ay)*4/15);

```

```
ax=Math.round(Ax+==(Xpos)-Ax)*4/15);
by=Math.round(By+==(ay-By)*4/15);
bx=Math.round(Bx+==(ax-Bx)*4/15);
cy=Math.round(Cy+==(by-Cy)*4/15);
cx=Math.round(Cx+==(bx-Cx)*4/15);
dy=Math.round(Dy+==(cy-Dy)*4/15);
dx=Math.round(Dx+==(cx-Dx)*4/15);
clock();
setTimeout('MouseFollow()',10);
}
function colorstep()
{count+=move;
if(count>=dots) {count=0;count_a+=move}
if(count_a==colors.length) count_a=0;
face[count].style.background=colors[count_a];
setTimeout('colorstep()',100)
}
function StartAll()
{MouseFollow();
colorstep();
}
if(document.all) window.onload = StartAll;
!-->
</script>
</body>
</html>
```

代码键入完毕，存盘为“实例 76.html”。

二、运行程序

双击文件“实例 76.html”。

实例 77 围绕鼠标旋转的小球

实例说明

本例运行效果如图 77-1 所示。这是一个旋转的小球的例子。

程序运行后，会有一串旋转的小球跟随鼠标移动。

本例的知识点有：CSS 的基本属性、以及 JavaScript 脚本语言知识。



图 77-1 效果图

编程思路

本实例使用 CSS 的基本属性，定义小球的样式，使用 JavaScript 脚本语言控制小球的旋转。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 77</title>
<body>
<script language=JavaScript>
if(document.all)
{with(document)
 {write('<div id="starsDiv" style="position:absolute;top:0px;left:0px">')
  write('<div style="position:relative;width:1px;height:1px;
      background:#000000;font-size:1px;visibility:visible;"></div>')
  write('<div style="position:relative;width:1px;height:1px;
```

```
background:#aaaaaa;font-size:1px;visibility:visible;"></div>')
write('<div style="position:relative;width:1px;height:1px;
background:#bbbbbb;font-size:1px;visibility:visible;"></div>')
write('<div style="position:relative;width:1px;height:1px;
background:#cccccc;font-size:1px;visibility:visible;"></div>')
write('<div style="position:relative;width:1px;height:1px;
background:#ddddd;d;font-size:1px;visibility:visible;"></div>')
write('<div style="position:relative;width:1px;height:1px;
background:#eeeeee;font-size:1px;visibility:visible;"></div>')
write('<div style="position:relative;width:1px;height:1px;
background:#ffffff;font-size:1px;visibility:visible;"></div>')
write('<div style="position:relative;width:2px;height:2px;
background:#222222;font-size:2px;visibility:visible;"></div>')
write('<div style="position:relative;width:2px;height:2px;
background:#444444;font-size:2px;visibility:visible;"></div>')
write('<div style="position:relative;width:2px;height:2px;
background:#666666;font-size:2px;visibility:visible;"></div>')
write('<div style="position:relative;width:2px;height:2px;
background:#888888;font-size:2px;visibility:visible;"></div>')
write('<div style="position:relative;width:2px;height:2px;
background:blue;font-size:2px;visibility:visible;"></div>')
write('<div style="position:relative;width:2px;height:2px;
background:green;font-size:2px;visibility:visible;"></div>')
write('<div style="position:relative;width:3px;height:3px;
background:red;font-size:3px;visibility:visible;"></div>')
write('</div>')
}
}
if(document.layers)
{window.captureEvents(Event.MOUSEMOVE);
}
var yBase=200;
var xBase=200;
var step=1;
var currStep=0;
var Xpos=1;
var Ypos=1;
if(document.all)
{function MoveHandler()
```

```

        {Xpos=document.body.scrollLeft+event.x;
        Ypos=document.body.scrollTop+event.y;
        }
        document.onmousemove=MoveHandler;
    }
    else if(document.layers)
        {function xMoveHandler(evt)
        {Xpos=evt.pageX;
        Ypos=evt.pageY;
        }
        window.onMouseMove=xMoveHandler;
    }
    function animateLogo()
    {if(document.all)
    {yBase=window.document.body.offsetHeight/6;
    xBase=window.document.body.offsetWidth/6;
    }
    else if(document.layers)
    {yBase=window.innerHeight/8;
    xBase=window.innerWidth/8;
    }
    if(document.all)
    {for(i=0;i<starsDiv.all.length;i++)
    {starsDiv.all[i].style.top=Ypos+yBase*Math.sin((currStep+i*4)/12)*Math.cos(400+currStep/200);
    starsDiv.all[i].style.left=Xpos+xBase*Math.sin((currStep + i*3)/10)*Math.sin(currStep/200);
    }
    }
    else if(document.layers)
    {for(j=0;j<14;j++)
    {var templayer="a"+j
    document.layers[templayer].top=Ypos+yBase*Math.sin((currStep+j*4)/12)*Math.cos(400+currStep/200);
    document.layers[templayer].left=Xpos+xBase*Math.sin((currStep+j*3)/10)*Math.sin(currStep/200);
    }
    }
    currStep+=step;
    setTimeout("animateLogo()", 5);
    }
    }
    
```

```
animateLogo());  
</script>  
</body>  
</html>
```

代码键入完毕，存盘为“实例 77.html”。

二、运行程序

双击文件“实例 77.html”。

实例 78 图文混排

实例说明

本例运行效果如图 78-1 所示。这是一个图文混排的例子。

程序运行后,在浏览器上会出现几幅图片与文字混排的效果。

本例的知识点有:CSS 中的定位技术,以及表格的应用。



图 78-1 效果图

编程思路

本实例将表格的应用与 CSS 的定位技术结合起来,实现图文混排。

创作步骤

一、创建源代码

打开记事本,键入如下代码:

```
<html>
<title>实例 78</title>
<head>
<style type="text/css">
<!--
.text
{font-family:"宋体";
font-size:10pt;
}
.image1
{margin-top:10px;
margin-right:10px;
}
.image2
{margin-left:20px;
}
```

```

-->
</style>
</head>
<body bgcolor="#FFFFFF">
<table width="70%" border="0" height="70%">
  <tr>
    <td width="23%">
      <table width="180" border="0" height="350">
        <tr valign="bottom">
          <td height="231">
            <p><span class="text">为实现社会主义现代化而不懈奋斗的精神，包括解放思想、实事求是的精神；
            紧跟时代、勇于创新的精神；知难而进、一往无前的精神；艰苦奋斗、务求实效的精神；淡泊名利、
            无私奉献的精神。</span></p>
          </td>
        </tr>
        <tr>
          <td></td>
        </tr>
      </table>
    </td>
    <td width="50%">
      <table width="180" border="0" height="350">
        <tr valign="bottom">
          <td height="140"></td>
        </tr>
        <tr valign="top">
          <td height="100" class="text">
            <p>它们构成一个有机的整体，其中，解放思想、实事求是核心精神。这是因为，解放思想、实事
            求是实现现代化的基本要求，也是其他几种精神的统领。因此，大力宣传和弘扬为实现社会主义现
            代化而不懈奋斗的精神，首先是要大力宣传和弘扬解放思想、实事求是的精神。</p>
          </td>
        </tr>
      </table>
    </td>
    <td width="27%">
      <table width="180" border="0" height="350">
        <tr align="left" valign="top">
          <td height="61">&nbsp;&nbsp;&nbsp;</td>

```

```
</tr>
<tr align="left" valign="bottom">
  <td></td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```

代码键入完毕，存盘为“实例 78.html”。

二、运行程序

双击文件“实例 78.html”。

实例 79 项目列表

实例说明

本例运行效果如图 79-1 所示。这是一个建立项目列表的例子。

程序运行后,在浏览器上会出现不同的列表形式。

本例的知识点有: CSS 的分级属性。

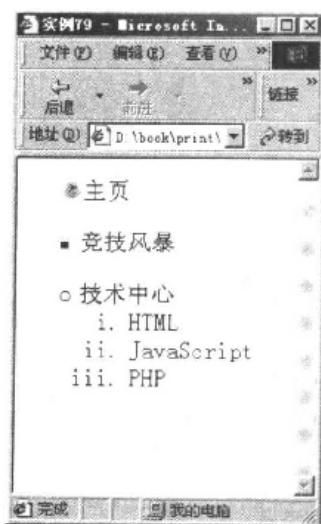


图 79-1 效果图

编程思路

本实例主要使用分级属性制定项目列表。

创作步骤

一、创建源代码

打开记事本,键入如下代码

```
<title>实例 79</title>
<head>
<style>
<!--
ul.1
{list-style:square outside url(dot0001.gif)
}
ul.2
{list style:squarc outside url()
}
ul.3
```



```
{list-style:circle outside
}
ol
{list-style:lower-roman
}
-->
</style>
</head>
<body>
<ul class=1>
  <li>主页</li>
</ul>
<ul class=2>
  <li>竞技风暴</li>
</ul>
<ul class=3>
  <li>技术中心</li>
  <ol>
    <li>HTML</li>
    <li>JavaScript</li>
    <li>PHP</li>
  </ol>
</ul>
</body>
</html>
```

代码键入完毕，存盘为“实例 79.html”。

二、运行程序

双击文件“实例 79.html”。

实例 80 自动重排列

实例说明

本例运行效果如图 80-1 所示。这是一个自动重排列的例子。

程序运行后，将鼠标置于图片之上，图片会自动变换为一张大图片，并且周围的文字会自动重排列。

本例的知识点有：CSS 中的基本属性。

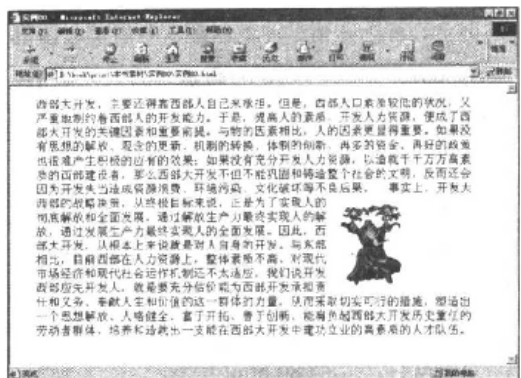


图 80-1 效果图

编程思路

本实例是用鼠标事件的响应改变图片，使文字进行自动重排列。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 80</title>
<body topmargin="20" leftmargin="40">
<p>西部大开发，主要还得靠西部人自己来承担。但是，西部人口素质较低的状况，又严重地制约着西部人的开发能力。于是，提高人的素质，开发人力资源，便成了西部大开发的关键因素和重要前提。与物的因素相比，人的因素更显得重要。如果没有思想的解放、观念的更新、机制的转换、体制的创新、再多的资金、再好的政策也很难产生积极的应有的效果；如果没有充分开发人力资源，以造就千千万万高素质的西部建设者，那么西部大开发不但不能巩固和铸造整个社会的文明，反而还会因为开发失当造成资源浪费、环境污染、文化破坏等不良后果。
<a onmouseover="img2.src='Sj127.jpg';this.style.cursor='hand'"
onmouseout="img2.src='Player3.gif';this.style.cursor='default'">

</a>
```

事实上，开发大西部的战略决策，从终极目标来说，正是为了实现人的彻底解放和全面发展。通过解

放生产力最终实现人的解放，通过发展生产力最终实现人的全面发展。因此，西部大开发，从根本上来说就是对自身的开发。与东部相比，目前西部在人力资源上，整体素质不高，对现代市场竞争和现代社会运作机制还不太适应。我们说开发西部应先开发人，就是要充分估价能为西部开发承担责任和义务，奉献人生和价值的这一群体的力量，从而采取切实可行的措施，塑造出一个思想解放、人格健全、富于开拓、善于创新、能肩负起西部大开发历史重任的劳动者群体，培养和造就出一支能在西部大开发中建功立业的高素质的人才队伍。

```
</p>
```

```
</body>
```

```
</html>
```

代码键入完毕，存盘为“实例 80.html”。

二、运行程序

双击文件“实例 80.html”。

实例 81 图片的显示和隐藏

实例说明

本例运行效果如图 81-1 所示，这是一个控制图片显示和隐藏的例子。

程序运行后，单击“图片一”会显示出图片一，而图片二会自动隐藏；单击“图片二”会显示出图片二，而图片一会自动隐藏。

本例的知识点有：CSS 的 `visibility` 属性。



图 81-1 效果图

编程思路

本实例使用 CSS 中的 `visibility` 属性控制图片的显示和隐藏，使用按钮改变图片的 `visibility` 属性。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 81</title>
<head>
<style>
#picture1
{position:absolute;
top:100;
}
#picture2
{position:absolute;
top:100;
visibility:hidden;
```

```
}
</style>
</head>
<body>
<div id="picture1">

</div>
<div id="picture2">

</div>
<form name="myform">
<p>
<input type="button" value="图片一"
onclick="picture1.style.visibility='visible';picture2.style.visibility='hidden'">
<input type="button" value="图片二"
onclick="picture2.style.visibility='visible';picture1.style.visibility='hidden'">
</p>
</form>
</body>
</html>
```

代码键入完毕，存盘为“实例 81.html”。

二、运行程序

双击文件“实例 81.html”。

实例 82 文字环绕图片

实例说明

本例运行效果如图 82-1 所示。这是一个文字环绕图片的例子。

程序运行后,在浏览器上出现两端不同效果的文字,第一段文字环绕图片排列,第二段文字不环绕图片排列。

本例的知识点有:CSS 的文字环绕属性,以及清除环绕属性。

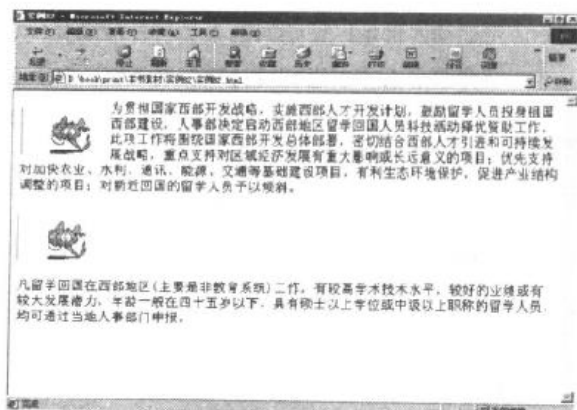


图 82-1 效果图

编程思路

本实例使用 CSS 中的文字环绕属性使第一段文字环绕图片,使用清除环绕属性使第二段文字不环绕图片。

创作步骤

一、创建源代码

打开记事本,键入如下代码:

```
<html>
<title>实例 82</title>
<head>
<style>
<!--
img.1
{float:left;
}
img.2
{clear:left;
}
-->
</style>
```



```
</head>
```

```
<body>
```

```

```

```
<p>为贯彻国家西部开发战略，实施西部人才开发计划，鼓励留学人员投身祖国西部建设，人事部决定启动西部地区留学回国人员科技活动择优资助工作。此项工作将围绕国家西部开发总体部署，密切结合西部人才引进和可持续发展战略，重点支持对区域竞技发展有重大影响或长远意义的项目；优先支持对加快农业、水利、通讯、能源、交通等基础建设项目，有利生态环境保护，促进产业结构调整的项目；对新近回国的留学人员予以倾斜。
```

```
</p>
```

```

```

```
<p>凡留学回国在西部地区(主要是非教育系统)工作，有较高学术技术水平，较好的业绩或有较大发展潜力，年龄一般在四十五岁以下，具有硕士以上学位或中级以上职称的留学人员，均可通过当地人事部门申报。
```

```
</p>
```

```
</body>
```

```
</html>
```

代码键入完毕，存盘为“实例 82.html”。

二、运行程序

双击文件“实例 82.html”。

实例 83 文本列

实例说明

本例运行效果如图 83-1 所示。这是一个生成文本列的例子。

程序运行后，在浏览器上出现两列文字，这种效果常被用于新闻列表。

本例的知识点有：CSS 的 float、width 和 border 属性。

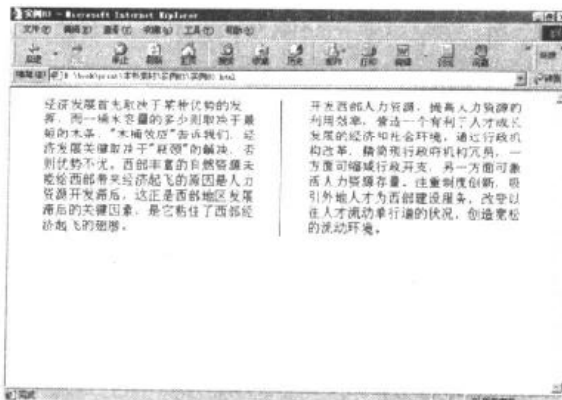


图 83-1 效果图

编程思路

本实例使用 CSS 中的 float、width 以及 border 属性对文本进行定位。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 83</title>
<head>
<style type="text/css">
div.left
{text-align:justify;
float:left;
margin-left:0.5cm;
padding-right:0.3in;
width:45%;
border-right:solid thin;
}
div.right
{text-align:justify;
```



```

float:right;
width:45%;
margin-left:1cm;
padding-right:0.25in;
}
</style>
</head>
<body bgcolor="#FFFFFF">
<div class="left">
<p> 竞技发展首先取决于某种优势的发挥,而一桶水容量的多少则取决于最短的木条。“木桶效应”告诉我们,竞技发展关键取决于“瓶颈”的解决,否则优势不优,西部丰富的自然资源未能给西部带来竞技起飞的原因是人力资源开发滞后,这正是西部地区发展滞后的关键因素,是它粘住了西部竞技起飞的翅膀。 </p>
</div>
<div class="right">
<p>开发西部人力资源,提高人力资源的利用效率,营造一个有利于人才成长发展的竞技和社会环境。通过行政机构改革,精简现行政府机构冗员,一方面可缩减行政开支,另一方面可激活人力资源存量,注重制度创新,吸引外地人才为西部建设服务,改变以往人才流动单行道的状况,创造宽松的流动环境。 </p>
</div>
</body>
</html>
    
```

代码键入完毕,存盘为“实例 83.html”。

二、运行程序

双击文件“实例 83.html”。

实例 84 背景色渐变

实例说明

本例运行效果如图 84-1 所示。这是一个使屏幕背景色渐变的例子。

程序运行后，背景色会逐渐变化。

本例的知识点有：CSS 的颜色属性、以及 JavaScript 脚本语言知识。

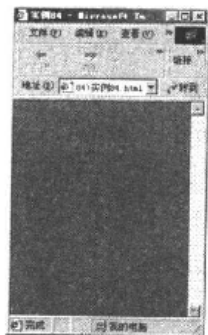


图 84-1 效果图

编程思路

本实例使用 JavaScript 脚本语言动态改变背景颜色。

创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 84</title>
<head>
<script language="JavaScript">
<!--
r=0;
g=0;
b=0;
flag=0;
t=new Array;
o=new Array;
d=new Array;
function hex(a,c)
f[a]=Math.floor(c/16)
o[a]=c%16
```

```
switch(t[a])
{case 10:t[a]='A';break;
 case 11:t[a]='B';break;
 case 12:t[a]='C';break;
 case 13:t[a]='D';break;
 case 14:t[a]='E';break;
 case 15:t[a]='F';break;
 default:break;
}
switch(o[a])
{case 10:o[a]='A';break;
 case 11:o[a]='B';break;
 case 12:o[a]='C';break;
 case 13:o[a]='D';break;
 case 14:o[a]='E';break;
 case 15:o[a]='F';break;
 default:break;
}
}
function ran(a,c)
{if((Math.random())>2/3||c==0)&&c<255)
 {c++;d[a]-2;
 }
 else
 {if((Math.random())<=1/2||c==255)&&c>0)
 {c--;d[a]=1;
 }
 else d[a]=0;
 }
 return c
}
function do_it(a,c)
{if((d[a]==2&&c<255)||c==0)
 {c++;d[a]=2
 }
 else if(d[a]==1&&c>0)||c==255)
 {c--;d[a]=1;
 }
}
if(a==3)
```

```

    {if(d[1]==0&&d[2]==0&&d[3]==0)
        flag=1
    }
    return c
}
function Init()
{if(flag==0)
    {r=ran(1, r);
    g=ran(2, g);
    b=ran(3, b);
    hex(1,r)
    hex(2,g)
    hex(3,b)
    document.bgColor="#"+t[1]+o[1]+t[2]+o[2]+t[3]+o[3]
    flag=50
    }
    else
    {r=do_it(1, r)
    g=do_it(2,g)
    b=do_it(3,b)
    hex(1,r)
    hex(2,g)
    hex(3,b)
    document.bgColor="#"+t[1]+o[1]+t[2]+o[2]+t[3]+o[3]
    flag--
    }
    if(document.all)
        setTimeout('Init()',50)
}
//-->
</script>
</head>
<body onload="Init()">
</body></html>

```

代码键入完毕，存盘为“实例 84.html”。

二、运行程序

双击文件“实例 84.html”。

实例 85 打开任意大小的窗口

实例说明

本例运行效果如图 85-1 所示。这是一个可以打开任意大小的窗口的例子。

程序运行后,在输入框中输入窗口的大小,单击按钮会生成定制的窗口。

本例的知识点有:CSS 中的 Width、Height 属性,以及 JavaScript 脚本语言知识。

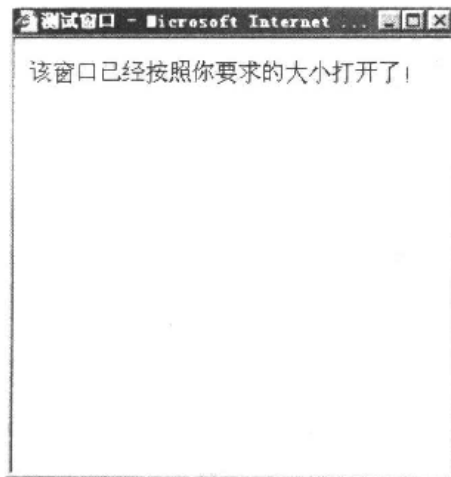


图 85-1 效果图

编程思路

本实例使用 JavaScript 脚本语言动态控制打开窗口的 Width、Height 的属性值。

创作步骤

一、创建脚本代码

```
function Create()
{
    var h=document.winfrm.big.value
    var w=document.winfrm.wid.value
    testWindow=window.open("about:blank","testwin",{toolbar=no,location=no,directories=no,
        menubar=no,scrollbars=no,resizable=no,width=' + w + ',height=' + h + "}");
    testWindow.document.writeln("<head><title>测试窗口</title></head>");
    testWindow.document.writeln("<body bgcolor=#ffffcc>");
    testWindow.document.writeln("该窗口已经按照你要求的大小打开了! </body>");
    testWindow.document.writeln("");
}
```

二、创建源代码

打开记事本,键入如下代码:

```
<html>
<head>
<script language="JavaScript">
<!--
键入以上脚本代码。
// -->
</script>
</head>
<body>
<form name="winfrm">
窗口宽度:
<input type="text" name="wid" maxlength="3">
<br>
窗口高度:
<input type="text" name="hig" maxlength="3">
<br>
<input type="button" name="s" value="打开窗口" onClick="Create()" class="yk9">
</form>
</body>
</html>
```

代码键入完毕，存盘为“实例 85.html”。

三、运行程序

双击文件“实例 85.html”。

第五篇

CSS 滤镜效果篇

本篇总览

本篇主要应用 CSS 中的滤镜属性，进行应用程序的开发。本篇将制作一些有关滤镜效果的例子。

通过本篇中各个实例的制作，读者可以对 CSS 中的滤镜属性有一个大概的了解。滤镜效果是网页效果中不可缺少的一项，通过对滤镜效果的制作，可以使网页看起来更精彩。

实例 86 花的世界

 实例说明

本例运行效果如图 86-1 所示。这是一个显示出不同的滤镜效果的例子。

程序运行后，在浏览器上显示出四种不同滤镜效果的花的图片。

本例的知识点有：CSS 的滤镜属性。

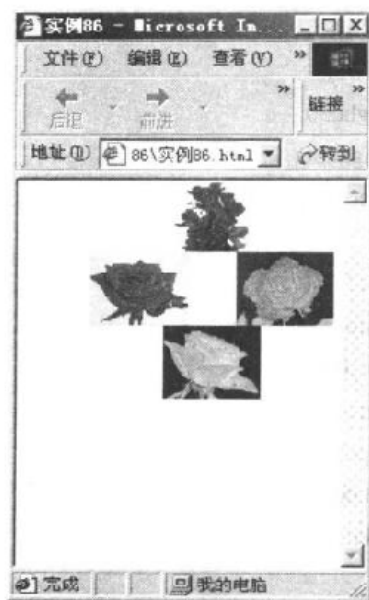


图 86-1 效果图

 编程思路

本实例使用 CSS 中的滤镜属性对图片进行处理。

 创作步骤

一、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 86</title>
<head>
<style>
.init
{position:absolute;
top:0;
left:100;
height:50;
```



```

weight:50;
}
style1
{position:absolute;
top:50;
left:50;
height:50;
width:50;
filter:gray;
}
.style2
{position:absolute;
top:100;
left:100;
height:50;
width:50;
filter:invert;
}
.style3
{position:absolute;
top:50;
left:150;
height:50;
width:50;
filter:xray;
}
</style>
<body>




</body>
</html>

```

代码键入完毕，存盘为“实例 86.html”。

二、运行程序

双击文件“实例 86.html”。

实例 87 动态的滤镜效果

实例说明

本例运行效果如图 87-1 所示。这是一个动态滤镜效果的实例。

程序运行后,在浏览器上会出现有水波波动效果的文字。

本例的知识点有:CSS 中的 Wave 属性和 JavaScript 脚本语言知识。



图 87-1 效果图

编程思路

本实例使用 CSS 中的 Wave 属性和 JavaScript 脚本语言。

Wave 属性可以使 HTML 元素在垂直方向上产生波纹状的变形,它具有 5 个参数:

Add 值用来设置是否在已经使用了滤镜的元素对象上显示原来的对象。取 0 表示不显示,取非 0 值表示要显示原来的对象;也可以使用布尔值,取 false 表示不显示,取 true 表示显示。

Freq 值用来设置波动的个数,也就是波纹的频率,用它来指定一个对象上要产生多少个完整的波纹。

Lightstrength 值用来设置波浪效果的光照强度,其取值从 0 到 100,取 0 值表示最弱,取 100 值表示最强。

Phase 值用于设置波浪的起始相角,从 0 到 100 的百分数值。

JavaScript 脚本语言主要用来使对象产生动态的效果。

创作步骤

一、创建脚本代码

```
function DoWave()
{if(wave.filters.Wave.Phase>99)
    wave.filters.Wave.Phase = 5
if(wave.filters.Wave.Phase<100)
    setInterval("wave.filters.Wave.Phase=wave.filters.Wave.Phase+2",50)}
```

|

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 87</title>
<head>
<style>
.wave
{background: green;
filter: Wave(Add=add,Freq=50,LightStrength=50,Phase=15,Strength=20);
width: auto;
position: absolute;
top: 10px
}
p
{font-weight: bold;
font-size: 40pt;
color: white
}
</style>
<script language=javascript>
<!--
键入以上脚本代码。
-->
</script>
</head>
<body onload=DoWave()>
<div class=wave id=wave
style="background: green; filter: Wave(Add=add,Freq=1,LightStrength=50,
Phase=15,Strength=20); width: auto; position: absolute; top: 10px">
<p>动态的滤镜效果</p>
</div>
</body>
</html>
```

代码键入完毕，存盘为“实例 87.html”。

三、运行程序

双击文件“实例 87.html”。

实例 88 图片的淡入淡出

实例说明

本例运行效果如图 88-1 所示。这是一个图片淡入淡出的例子。

程序运行后，单击图片，图片会产生淡入淡出的效果，变换为另一张图片。

本实例主要应用的技术有：CSS 中的 BlendTrans 滤镜属性和 JavaScript 脚本语言。

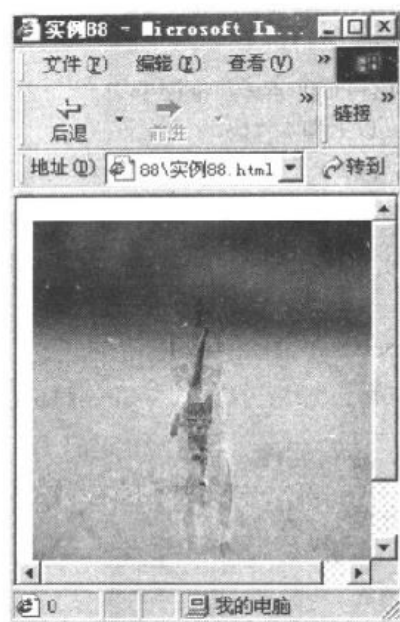


图 88-1 效果图

编程思路

本实例主要使用 CSS 中的 BlendTrans 滤镜属性。

BlendTrans 属性是用来处理网页或是 HTML 元素对象之间的淡入和淡出效果的。它具有唯一的参数 duration，用它指定淡入和淡出的时间，时间的取值是秒。

创作步骤

一、创建脚本代码

```
function BlendTrans()
{
window.defaultStatus=picture.filters.blendTrans.status;
picture.filters.blendTrans.Apply();
picture.src = "0392.jpg";
picture.filters.blendTrans.Play();
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 88</title>
<head>
<script language="JavaScript">
键入以上脚本代码.
</script>
</head>

</html>
```

代码键入完毕，存盘为“实例 88.html”。

三、运行程序

双击文件“实例 88.html”。

实例 89 多变的转换效果

实例说明

本例运行效果如图 89-1 所示。这是一个使用不同形式进行图片转换的例子。

程序运行后,单击图片会出现随机溶解的效果变换为另一张图片。

本实例主要应用的技术有: CSS 中的 RevealTrans 滤镜属性和 JavaScript 脚本语言。



图 89-1 效果图

编程思路

本实例主要使用 CSS 中的 RevealTrans 滤镜属性和一些 JavaScript 脚本语言知识。

RevealTrans 属性和 BlendTrans 属性不同,BlendTrans 属性值提供一种淡入淡出的效果,而 RevealTrans 属性还提供了更多的转换效果。RevealTrans 属性除了提供 duration 参数外,还提供了—个转换类型的参数 transition,通过它可以设置 24 种转换类型。

具体类型如表 1-1 所示:

表 1-1 转换类型

转换滤镜的转换形式	所对应的代码	转换滤镜的转换形式	所对应的代码
矩形区从大到小	0	随机溶解	12
矩形区从小到大	1	垂直向内裂开	13
圆形区从大到小	2	垂直向外裂开	14
圆形区从小到大	3	水平向内裂开	15
由下向上推开	4	水平向外裂开	16
由上向下推开	5	向左下剥开	17
由左向右推开	6	向左上剥开	18
由右向左推开	7	向右下剥开	19
垂直型百叶窗	8	向右上剥开	20
水平型百叶窗	9	随机水平细纹	21
水平棋盘	10	随机垂直细纹	22
垂直棋盘	11	随机选取一种形式	23



创作步骤

一、创建脚本代码

```
function RevealTrans()  
{ picture.filters.filters.item(0).Apply();  
  picture.src = "0432.jpg";  
  picture.filters.filters.item(0).Play();  
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>  
<title>实例 89</title>  
<head>  
<script language="JavaScript">  
键入以上脚本代码。  
</script>  
</head>  
<body>  
  
</body>  
</html>
```

代码键入完毕，存盘为“实例 89.html”。

三、运行程序

双击文件“实例 89.html”。

实例 90 倒计时过程

实例说明

本例运行效果如图 90-1 所示。这是一个以不同转换效果显示的倒计时过程。

程序运行后,在浏览器上单击方框会显示出倒计时过程,并且每次计数都以不同的转换效果显示。

本例的知识点有:CSS 中的 RevealTrans 滤镜属性和 JavaScript 脚本语言。

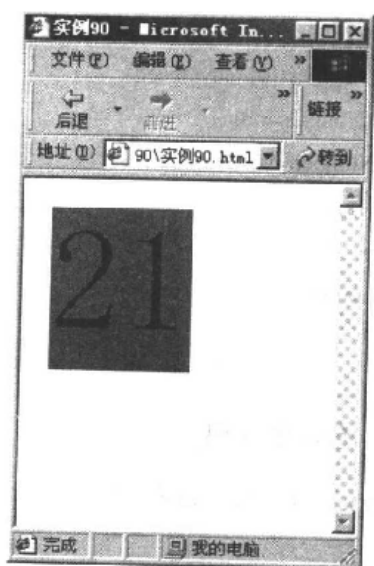


图 90-1 效果图

编程思路

本实例使用 CSS 中的 RevealTrans 滤镜属性进行转换效果的控制,使用 JavaScript 脚本语言控制数字的倒数

创作步骤

一、创建脚本代码

```
var MaxNumber=23;
var Number=MaxNumber;
var speed=2;
var Runing=0;
var colors=new Array(10);
function Again()
{Runing=0;
  Count();
}
function Count()
```



```

if(Runing==0)
{
    Runing=1;
    div1.filters[0].transition=Number;
    div1.filters[0].apply();
    div1.innerText=Number;
    div1.filters[0].play(speed);
    if(Number==0)
    {
        Number=MaxNumber;
        return;
    }
    Number--;
}
}

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 90</title>
<head>
<script language="JavaScript">
<!--
键入以上脚本代码。
-->
</script>
</head>
<body>
<div id="div1"
style="position:absolute;top:20;left:20;width:100;height:100;font-size:100;text-align:center;background-col
or:red;filter:revealTrans(Transition=1,Duration=0.1)" onclick="Count()" onfilterchange="Again()">
</div>
</body>
</html>

```

代码键入完毕，存盘为“实例 90.html”。

三、运行程序

双击文件“实例 90.html”。

实例 91 动态光源效果

实例说明

本例运行效果如图 91-1 所示。这是一个模拟光源效果的例子。

程序运行后，在图片中不停地单击鼠标，会出现最多 4 条颜色、强度、位置都不相同的光源，它们将随着鼠标的移动而移动。

本例的知识点有：CSS 中的滤镜属性的 Light 属性以及 JavaScript 脚本语言。



图 91-1 效果图

编程思路

本实例主要使用的是 Light 滤镜属性，另外，本例中还运用了一些 JavaScript 脚本语言的知识。

Light 属性可以模拟光源的投影效果，它甚至可以控制 10 个不同的光源照亮在一个元素上，以调节亮度、颜色和光源位置。

Light 滤镜属性还需要脚本语言以使其显示出来，一旦为对象定义了 Light 属性，就可以调用它的方法来设置或改变属性。

Light 滤镜可用的方法有如下 7 种：

- (1) AddAmbient：加入包围的光源；
- (2) AddCone：加入锥形光源；
- (3) AddPoint：加入点光源；
- (4) ChangColor：改变光的颜色；
- (5) ChangStrength：改变光源的强度；
- (6) Clear：清除所有的光源；
- (7) MoveLight：移动光源。



一、创建脚本代码

```
var light_num=0;
window.onload=set_lights;
document.onclick=lights;
light.onmousemove=move;
function set_lights()
{light.filters[0].clear();
light.filters[0].addcone(0,0,5,100,100,255,255,0,60,15);
if(light_num>0)
{light.filters[0].addcone(0,170,5,100,100,255,0,0,60,15);
if(light_num>1)
{light.filters[0].addcone(220,180,5,100,100,0,255,255,60,15);
if(light_num>2)
{light.filters[0].addcone(255,0,5,100,100,255,0,255,60,15);
}
}
}
}
function lights()
{light_num=(light_num+=1)%4;
set_lights();
}
function move()
{x=(window.event.x-80);
y=(window.event.y-80);
light.filters[0].movelight(0,x,y,5,1);
if(light_num>0)
{light.filters[0].movelight(1,x,y,5,1);
if(light_num>1)
{light.filters[0].movelight(2,x,y,5,1);
if(light_num>2)
{light.filters[0].movelight(3,x,y,5,1);
}
}
}
}
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 91</title>
<body>
<table>
<tr>
  <td colspan="2" >
  </td>
</tr>
<tr>
  <td id="light" style="filter:light(enabled=1);width: 200px">
  
  </td>
  <td style="vertical-align: top">
  </td>
</tr>
</table>
<script language="Javascript">
<!--
键入以上脚本代码。
-->
</script>
</body>
</html>
```

代码键入完毕，存盘为“实例 91.html”。

三、运行程序

双击文件“实例 91.html”。

实例 92 火焰字

实例说明

本例运行效果如图 92-1 所示。这是一个可以产生火焰淡化效果的例子。

程序运行后，单击浏览器中的文字，文字会出现火焰效果。

本例的知识点有：CSS 中的 Glow 滤镜属性和 JavaScript 脚本语言。



图 92-1 效果图

编程思路

本实例主要使用的是 CSS 中的 Glow 滤镜属性。

Glow 滤镜属性可以让 HTML 元素对象的轮廓产生一种比较柔和的边框，也可以说是一个光晕，并且还可以在像火一样的淡化效果。可以通过定义数值来指定这种效果的颜色和强度。

该属性有如下两个参数：

- (1) Color 值用来指定晕圈效果的颜色，可以使用 #RRGGBB 格式的颜色值。
- (2) Strength 值用来指定晕圈的强度范围，取值的范围从 1 到 255，数字越大晕的效果就越强，数字越小则晕的效果越不明显。

创作步骤

一、创建脚本代码

```
function my_glow()
{if(myid.filters.glow.enabled==1)
{myid.filters.glow.enabled=0;
}
```

```
if(myid.filters.blendTrans.status==0)
{
myid.filters.blendTrans.apply();
myid.filters.glow.enabled=1;
myid.filters.blendTrans.play();
}
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 92</title>
<head>
<script languag=JavaScript>
<'--
键入以上脚本代码。
-->
</script>
</head>
<body bgcolor="black" text="pink">
<div id="myid" onclick="my_glow()" style="font-size:54; height:200;
filter:glow(color=#fff00,strength=20,enabled=0)
blendTrans(duration=2)">
<p>单击这里</p>
</div>
</body>
</html>
```

代码键入完毕，存盘为“实例 92.html”。

三、运行程序

双击文件“实例 92.html”。

实例 93 雷达演示

实例说明

本例运行效果如图 93-1 所示, 这是一个雷达演示的例子。

程序运行后, 在浏览器上显示出一个雷达系统, 目标将一直向中心移动。

本例的知识点有: CSS 中的 `light` 属性及其方法, 另外还有一些 JavaScript 脚本语言知识。

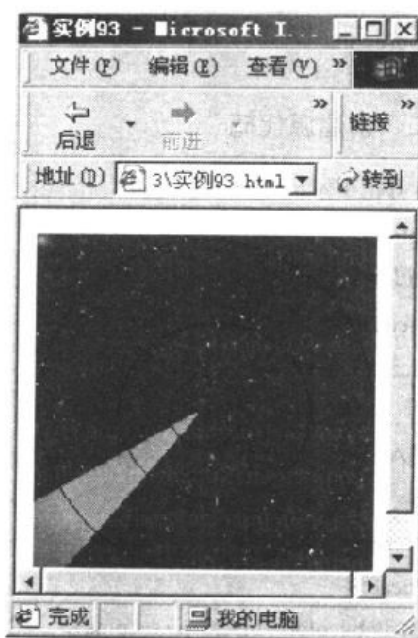


图 93-1 效果图

编程思路

本实例使用了 `light` 属性及其方法, 另外还使用了 JavaScript 脚本语言控制雷达的动作。

创作步骤

一、创建脚本代码

```
var light_x=20;  
var light_y=20;  
var light_z=40;  
var xinc=10;  
var yinc=10;  
var r=100;  
var deg=0;  
var deg1;  
var rad;
```

```

var pointangle;
var planelight_x=3;
var planelight_y=3;
var planelight_z=3;
var conversion=(2*3.14159)/360;
function movefilt()
{light_x=r+r*Math.cos(deg*conversion);
light_y=r+r*Math.sin(deg*conversion);
deg+=10;
if(deg==360)
    deg=0;
    body1.filters[0].movelight(0,light_x,light_y,light_z,1);
pointangle=Math.atan((planelight_y-r)/(planelight_x-r))/conversion;
if((planelight_x<r)&&(planelight_y<r))
    pointangle+=180;
if((planelight_x>r)&&(planelight_y>r))
    pointangle+-360;
if((planelight_x<r)&&(planelight_y>r))
    pointangle+=180;
if((deg-10<=pointangle)&&(pointangle<=deg))
    body1.filters[0].movelight(2,planelight_x,planelight_y,planelight_z,1);
mytimeout=setTimeout('movefilt()',100);
}
function moveplanes()
{planelight_x++;
planelight_y++;
if(planelight_y>100) planelight_y=0;
if(planelight_x>100) planelight_x=0;
timeout2=setTimeout('moveplanes()',500);
}
function go()
{body1.filters.light.addcone(100,100,0,light_x,light_y,0,225,0,150,10);
body1.filters.light.addambient(0,225,0,20)
body1.filters.light.addpoint(planelight_x,planelight_y,3,0,225,0,100);
var x=0;
movefilt();
moveplanes();
}

```


二、创建源代码

```

<html>
<title>实例 93</title>
<head>
<style>
div
{filter: blur(animate,direction=135,strength=10);
width: 800px
}
</style>
<script language=JavaScript>
键入以上脚本代码。
</script>
</head>
<body onload=go(>

</body>
</html>

```

键入完毕，存盘为“实例 93.html”。

三、运行程序

双击文件“实例 93.html”。

实例 94 隐藏的文字

实例说明

本例运行效果如图 94-1 所示。这是一个隐藏文字的例子。

程序运行后，当鼠标置于图片范围之内时，隐藏的文字就会显示出来，鼠标移出图片范围文字就会隐藏。

本例的知识点有：CSS 中的滤镜属性的 Alpha 属性和 Visibility 属性，以及一些 JavaScript 脚本语言知识。



图 94-1 效果图

编程思路

本例主要使用 CSS 中滤镜效果的“Alpha”属性。“Alpha”属性可以把一个目标元素与背景相混合，设计者可以指定数值来控制混合的程度。通过指定坐标，可以指定点、线、面的透明度，它们的参数含义分别如下：

“Opacity”代表透明度水准。默认的范围是从 0 到 100，0 代表完全透明，100 代表完全不透明。

“FinishOpacity”是一个可选参数，范围也是从 0 到 100。如果想要设置渐变的透明效果，就可以使用它们来指定结束时的透明度。

“Style”参数指定了透明区域的形状特征。其中 0 代表统一形状，1 代表线形，2 代表放射状，3 代表长方形。

“StartX”和“StartY”代表渐变透明效果开始的 X 和 Y 坐标。

“FinishX”和“FinishY”代表渐变透明效果结束的 X 和 Y 坐标。

隐藏属性的改变通过 JavaScript 函数控制实现。

创作步骤

一、创建脚本代码

```
function display()
{
  if (picture.style.visibility=="hidden")
  { picture.style.visibility="";
    words.style.visibility="";
  }
  else
  { picture.style.visibility="hidden";
    words.style.visibility="hidden";
  }
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 94</title>
<head>
<style>
.pos
{position:absolute;
  top=15;
  left=10;
}
</style>
<script>
键入如上脚本代码。
</script>
</head>
<body>
<div id="picture" class="pos" style="visibility:hidden; height=25; width=200; background-color:blue;
  filter:alpha(opacity=60)">
</div>
<div id="words" class="pos" style="visibility:hidden; left:25; color:yellow; font-size:14pt,
  font-weight:bold; filter:alpha(opacity=60)">
<p>
```

Hello world!

```
</p>
```

```
</div>
```

```

```

```
</body>
```

```
</html>
```

代码键入完毕，存盘为“实例 94.html”。

三、运行程序

双击文件“实例 94.html”。

实例 95 若隐若现的文字

实例说明

本例运行效果如图 95-1 所示，这是一个显示若隐若现的文字的例子。

程序运行后，几组文字不停地循环显示，并产生若隐若现的效果。

本例的知识点有：CSS 中的 Glow 滤镜属性，以及 JavaScript 脚本语言知识。



图 95-1 效果图

编程思路

本实例使用 CSS 中的 Glow 属性对显示的文字添加滤镜效果，使用 JavaScript 脚本语言控制文字的循环显示。

创作步骤

一、创建脚本代码

```

var thissize=20
var textfont="隶书"
var textcolor= new Array()
textcolor[0]="000000"
textcolor[1]="000000"
textcolor[2]="000000"
textcolor[3]="111111"
textcolor[4]="222222"
textcolor[5]="333333"
textcolor[6]="444444"
textcolor[7]="555555"
textcolor[8]="666666"
textcolor[9]="777777"
    
```

```
textcolor[10]="888888"
textcolor[11]="999999"
textcolor[12]="aaaaaa"
textcolor[13]="bbbbbb"
textcolor[14]="cccccc"
textcolor[15]="ddddd"
textcolor[16]="eeeee"
textcolor[17]="fffff"
textcolor[18]="fffff"
var message = new Array()
message[0]="message_1"
message[1]="message_2"
message[2]="message_3"
message[3]="message_4"
i_message=0
var i_strength=0
var i_message=0
var timer
function glowtext()
{if(document.all)
  {if (i_strength <=17)
    {glowdiv.innerText=message[i_message]
    document.all.glowdiv.style.filter="glow(color="+textcolor[i_strength]+", strength=4)"
    i_strength++
    timer=setTimeout("glowtext()",100)
    }
    else
    {clearTimeout(timer)
    setTimeout("deglowtext()",500)
    }
  }
}
function deglowtext()
{if(document.all)
  {if (i_strength >=0)
    {glowdiv.innerText=message[i_message]
    document.all.glowdiv.style.filter="glow(color="+textcolor[i_strength]+", strength=4)"
    i_strength--
    timer=setTimeout("deglowtext()",100)
    }
  }
}
```

```

    }
    else
    {clearTimeout(timer)
    i_message++
    if(i_message>=message.length)
    {i_message=0
    }
    l_strength=0
    intermezzo()
    }
}
}
function intermezzo()
{glowdiv.innerText=""
setTimeout("glowtext()",500)
}

```

二、创建源代码

打开记事本，键入如下代码：

```

<html>
<title>实例 95</title>
<head>
</head>
<script language=JavaScript>
<!--
键入以上脚本代码。
//-->
</script>
<body bgcolor="#000000" onLoad="glowtext()">
<div id="glowdiv"
    style="position:absolute;visibility:visible;width:300px;text-align:center;
    top:100px;left:0px;font-family:隶书;font-size:30pt;color:000000">
</div>
</body>
</html>

```

代码键入完毕，存盘为“实例 95.html”。

三、运行程序

双击文件“实例 95.html”。

实例 96 光影效果的图片

实例说明

本例运行效果如图 96-1 所示，这是一个产生光影效果的例子。

程序运行后，图片将产生光影效果。

本例的知识点有：CSS 中的 Light 滤镜属性，以及 JavaScript 脚本语言知识。



图 96-1 效果图

编程思路

本实例主要应用 CSS 中的 Light 属性的方法，以及 JavaScript 脚本语言实现光影效果。

创作步骤

一、创建脚本代码

```
if (document.all&&window.spotlight)
{
    var x=new Array()
    var direction=new Array()
    var y=new Array()
    if(spotlight.length==null)
    {
        spotlight[0]=document.all.spotlight
        x[0]=0
        direction[0]="right"
        y[0]=spotlight[0].height
        spotlight[0].filters.light.addPoint(100,50,100,255,255,255,90)
    }
}
```



```

    }
    else
    for(i=0;i<spotlight.length;i++)
    {x[i]=0
    direction[i]="right"
    y[i]=spotlight[i].height
    spotlight[i].filters.light.addPoint(100,50,100,255,255,255,90)
    }
}

function slidelight(cur)
[spotlight[cur].filters.light.MoveLight(0,x[cur],y[cur],200,-1)
if(x[cur]<spotlight[cur].width+200&&direction[cur]=="right")
    x[cur]+=10
else if(x[cur]>spotlight[cur].width+200)
    {direction[cur]="left"
    x[cur]-=10
    }
else if(x[cur]>-200&&x[cur]<-185)
    {direction[cur]="right"
    x[cur]+=10
    }
else
    {x[cur]-=10
    direction[cur]="left"
    }
}

if (document.all&&window.spotlight)
{if(spotlight.length==null)
    setInterval("slidelight(0)",spotlight[0].speed)
else
    for (t=0;t<spotlight.length;t++)
    {var temp='setInterval("slidelight('+t+')",'+spotlight[t].speed+'
    eval(temp)
    }
}

```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
```

```
<title>实例 96</title>
<head>
<style>
<!--
#spotlight
{filter:light
}
-->
</style>
</head>
<body>

</body>
<script language="JavaScript1.2">
键入以上脚本代码。
</script>
</html>
```

代码键入完毕，存盘为“实例 96.html”。

三、运行程序

双击文件“实例 96.html”。

实例 97 文字渐变

实例说明

本例运行效果如图 97-1 所示。这是一个几组文字渐变的例子。

程序运行后，将有几组文字循环显示，并产生渐变的效果。

本例的知识点有：CSS 的 `revealTrans` 滤镜属性和文本属性，以及 JavaScript 脚本语言知识。



图 97-1 效果图

编程思路

本实例使用文本属性定义文字的风格，用 CSS 中的 `revealTrans` 属性控制文字渐变的效果，用 JavaScript 脚本语言控制文字的渐变。

创作步骤

一、创建脚本代码

```
messages = new Array()  
messages[0] = "德甲"  
messages[1] = "英超"  
messages[2] = "意甲"  
messages[3] = "西甲"  
messages[4] = "法甲"  
color = new Array()  
color[0] = "0080ff"
```

```
color[1] = "0080ff"
color[2] = "123456"
color[3] = "789abc"
color[4] = "654321"
size = new Array()
size[0] = "20pt"
size[1] = "20pt"
size[2] = "40pt"
size[3] = "40pt"
size[4] = "30pt"
var i_messages = 0
var timer
function dotransition()
{if(document.all)
  {content.filters[0].apply()
  content.innerHTML = "<span
    style='color:"+color[i_messages]+";font-size:"+size[i_messages]+";
    font-family:"+Arial black+";filter: revealTrans(Transition=12, Duration=3)'>
    "+messages[i_messages]+"</span>"
  content.filters[0].play()
  if(i_messages>=messages.length-1)
  {i_messages = 0
  }
  else
  {i_messages++
  }
  }
if(document.layers)
{document.content.document.write("<span
  style='color:"+color[i_messages]+">"+messages[i_messages]+"</span>")
if(i_messages>=messages.length-1)
{i_messages = 0
}
else
{i_messages++
}
}
timer=setTimeout("dotransition()",5000)
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 97</title>
<head>
<script>
键入以上脚本代码。
</script>
</head>
<body onload=dotransition()>
<div id=content style="filter:revealTrans(Transition=12, Duration=2);
height:150px;left:50px;position:absolute;top:50px;width:559px">
</div>
</body>
</html>
```

代码键入完毕，存盘为“实例 97.html”。

三、运行程序

双击文件“实例 97.html”。

实例 98 图片的循环播放

实例说明

本例运行效果如图 98-1 所示，这是一个几个图片循环播放的例子。

程序运行后，几张图片将循环显示。

本例的知识点有：CSS 中的 Alpha 滤镜属性，以及 JavaScript 脚本语言知识。



图 98-1 效果图

编程思路

本实例使用 CSS 中的 Alpha 属性对图片的透明度进行定义，使用 JavaScript 脚本语言知识控制图片的循环播放。

创作步骤

一、创建脚本代码

```
sandra0=new Image();
sandra0.src="Boylaug.gif";
sandra1=new Image();
sandra1.src="Fidog.gif";
sandra2=new Image();
sandra2.src="Hlpglobe.gif";
var i_stmngth=1
```

```
var i_image=0
var imageurl=new Array()
imageurl[0]="Boylaug.gif"
imageurl[1]="Fidog.gif"
imageurl[2]="Hjpglobe.gif"
function showimage()
{if(document.all)
  {if(i_strngth<=110)
    {testimage.innerHTML="";
    i_strngth=i_strngth+10
    var timer=setTimeout("showimage()",100)
  }
  else
    {clearTimeout(timer)
    var timer=setTimeout("hideimage()",1000)
  }
}
if(document.layers)
{clearTimeout(timer)
document.testimage.document.write("")
document.close()
i_image++
if(i_image>=imageurl.length)
  i_image=0
var timer=setTimeout("showimage()",2000)
}
}
function hideimage()
{if(i_strngth>=-10)
  {testimage.innerHTML="";
  i_strngth=i_strngth-10
  var timer=setTimeout("hideimage()",100)
}
else
{clearTimeout(timer)
i_image++
if(i_image>=imageurl.length)
```

```
    i_image=0  
    i_strngth=i  
    var timer=setTimeout("showimage()",500)  
  }  
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>  
<title>实例 98</title>  
<head>  
<script language=JavaScript>  
<!--//  
键入以上脚本代码。  
//-->  
</script>  
</head>  
<body onLoad="showimage()">  
<div id="testimage" style="position:absolute;visibility:visible;top:80px;left:80px">  
</div>  
</body>  
</html>
```

代码键入完毕，存盘为“实例 98.html”。

三、运行程序

双击文件“实例 98.html”。

实例 99 虚幻的图片

实例说明

本例运行效果如图 99-1 所示。这是一个虚幻显示图片的例子。

程序运行后，图片将产生虚幻的效果。

本例的知识点有：CSS 的 Alpha 滤镜属性，以及 JavaScript 脚本语言知识。



图 99-1 效果图

编程思路

本实例使用 CSS 中的 Alpha 属性产生透明的效果，使用 JavaScript 脚本语言控制小球的滚动。

创作步骤

一、创建脚本代码

```
var x,y
var marginbottom
var marginleft=0
var margintop=0
var marginright
var cliptop
var clipbottom
var clipleft
var clipright
var clippoints
var ballheight=150
var ballwidth=150
```

```
var imageheight=525
var imagewidth=457
var tempo=25
var stepx=12
var stepy=6
var timer
function setValues()
{if(document.all)
  {marginbottom=imageheight-ballheight
  marginright=imagewidth-ballwidth
  document.all.ball.style.posLeft=randommaker(400)
  document.all.ball.style.posTop=0
  document.all.textContent.style.posLeft=0
  document.all.textContent.style.posTop=0
  document.all.ball.style.filter="alpha(opacity=0,finishopacity=100,style=2,startX=0px,startY=0px,finishX
    =100px,finishY=100px)"
  moveball()
  }
}
function randommaker(range)
{rand=Math.floor(range*Math.random())
return rand
}
function moveball()
{checkposition()
if(document.all)
  {document.all.ball.style.posLeft+=stepx
  document.all.ball.style.posTop+=stepy
  cliptop=document.all.ball.style.posTop
  clipbottom=cliptop+ballheight
  clipleft=document.all.ball.style.posLeft
  clipright=clipleft+ballwidth
  clippoints="rect("+cliptop+" "+clipright+" "+clipbottom+" "+clipleft+")"
  document.all.textContent.style.clip=clippoints
  timer=setTimeout("moveball()",tempo)
  }
}
function checkposition()
{if(document.all)
```

```
(if(document.all.ball.style.posLeft>=marginright)
{stepx=stepx*-1
document.all.ball.style.posLeft-=10
}
if(document.all.ball.style.posLeft<=marginleft)
{stepx=stepx*-1
document.all.ball.style.posLeft+=10
}
if(document.all.ball.style.posTop>=marginbottom)
{stepy=stepy*-1
document.all.ball.style.posTop-=10
}
if(document.all.ball.style.posTop<=margintop)
{stepy=stepy*-1
document.all.ball.style.posTop+=10
}
}
}
```

二、创建源代码

打开记事本，键入如下代码：

```
<html>
<title>实例 99</title>
<body bgcolor="#fef4d9" onload=setValues(>
<script language=JavaScript>
<!--
键入以上脚本代码。
// >
</script>
<span id=textcontent style="left:-5000px;position:absolute;top:-5000px">

</span>
<span id=ball
style="background-color:white;height:150px;position:absolute;top:-50px;width:150px">
</span></body></html>
```

代码键入完毕，存盘为“实例 99.html”。

三、运行程序

双击文件“实例 99.html”。

实例 100 多变的注释

实例说明

本例运行效果如图 100-1 所示。这是一个为图片添加注释的例子。

程序运行后,在浏览器上将显示出添加了注释的图片。

本例的知识点有: CSS 中的滤镜属性。



图 100-1 效果图

编程思路

本实例使用 CSS 中的滤镜属性为注释添加样式,作为图片的点缀。

创作步骤

一、创建源代码

打开记事本,键入如下代码:

```
<html>
<title>实例 100</title>
<head>
<style>
<!--
div.1
{position: absolute;
top: 200;
```

```

left:10;
weith:80;
height:15;
background-color:white;
filter:alpha(opacity=40);
}
div.2
{position:absolute;
top:20;
left:150;
weith:190;
height:10;
filter:glow(color=red,strength=2);
}
-->
</style>
</head>
<body>
<div class="1">
<p style="font-family:隶书;font-size:20pt">多变的注释</p>
</div>
<div class="2">
<p style="font-size:10pt">http://www.sina.com.cn</p>
</div>

</body>
</html>

```

代码键入完毕、存盘为“实例 100.html”。

二、运行程序

双击文件“实例 100.html”。