

第5章 鼠标特效

鼠标是电脑必不可少的组成部分，没有鼠标的电脑是无法使用的，当然，对于网页而言，鼠标特效不仅限于类似按键特效式的屏蔽应用，更多的是绚丽缤纷的点缀效果。本章主要讲解网页编写过程中大量应用的各种鼠标特效，在我们制作网页的过程中，更多地应用这些特效，就能让网页更加华丽。

5.1 鼠标单击图片后切换为其他图片

本实例使用 JavaScript 制作一个实例，该实例中的图片，在被鼠标单击以后，能够切换成其他图片。本节主要涉及的 JavaScript 语法是标签。

标签在网页显示的图片用来实现，这里的 picture 为图片在本地计算机上的具体文件信息，或者以 url 存储在网络上的文件位置。

本实例主要代码如下：

```
<script>
    /*更换页面元素背景图片*/
    function show(id1)
    {
        document.getElementById(id1).src = "5102.jpg";
    }
</script>
```

网页效果如图 5.1 所示。



图 5.1 鼠标单击图片后切换为其他图片

5.2 鼠标单击小图无刷新放大图片

本实例使用 JavaScript 制作一个实例，单击该实例中的小图片，将在不刷新网页的情

况下，对该图片进行放大显示。本节主要涉及的 JavaScript 语法如下。

 标签定义的是网页中显示的图片，用 可以控制该 元素在页面中是否可见。经常遇到的 是将图片转换为块级对象。

本实例主要代码如下：

```
<p>
  /*控制元素可见*/
  <a href="#" onclick="document.getElementById('light').style.display=
    'block';document.getElementById('fade').style.display='block'">
    
  </a>
</p>
<div id="light" class="white_content">
  
  /*控制元素不可见*/
  <a href="javascript:void(0)" onclick="document.getElementById('light')
    .style.display='none';document.getElementById('fade').style.display=
    'none'">
    关闭
  </a>
</div>
```

网页效果如图 5.2 所示。



图 5.2 鼠标单击小图无刷新放大图片

5.3 鼠标放到图片上会滑出提示文字

本实例使用 JavaScript 制作一个实例，该实例中的图片当鼠标移入时，会在图片底部出现文字。本节主要涉及的 JavaScript 语法如下。

1. getElementById() 方法

getElementById() 方法的语法如下。getElementById(id)，可以访问 document 中某一设置了 id 的特殊元素。

2. getElementsByTagName()方法

getElementsByTagName() 方法的语法是 getElementsByTagName (tagname) , 通过 tagname (标签名称) 来获得元素 , 一个 document 中当然会有相同的标签 , 所以这个方法也是获取一个数组。

本实例主要代码如下 :

```
<script>
  /*主函数*/
  function go()
  {
    var t, tt;
    var _div = document.getElementById("show"); /*通过 id 获取元素*/
    var obj = _div.getElementsByTagName('h2')[0] /*通过标签名称获取元素*/
    obj.style.bottom = "-50px";
    var change = function() /*变更函数*/
    {
      var obj_h = parseInt(obj.style.bottom);
      if (obj_h < 0)
      {
        obj.style.bottom = (obj_h + Math.floor((0 - obj_h) * 0.1)) + "px"
      }
      else
      {
        clearInterval(t)
      }
    }
    var back = function() /*返回函数*/
    {
      var obj_hh = parseInt(obj.style.bottom);
      if (obj_hh > -50)
      {
        obj.style.bottom = (obj_hh + Math.floor((- 50 - obj_hh) * 0.1))
          + "px"
      }
      else
      {
        clearInterval(tt)
      }
    }
    _div.onmouseover = function() /*鼠标移入函数*/
    {
      clearInterval(tt);
      t = setInterval(change, 10);
    }
    _div.onmouseout = function() /*鼠标移出函数*/
    {
      clearInterval(t);
      tt = setInterval(back, 10)
    }
  }
  window.onload = function() /*页面加载时调用的函数*/
  {
    go();
  }
</script>
```

网页效果如图 5.3 所示。



图 5.3 鼠标放到图片上会滑出提示文字

5.4 鼠标放上图片错位效果

本实例使用 JavaScript 制作一个实例，该实例中的图片当鼠标移入时会发生错位。本节主要涉及的 JavaScript 语法如下。

- onmouseover：鼠标移动到对象上时响应事件。
- onmousedown：鼠标键按下时响应事件。
- onmouseup：鼠标键按下并松开时响应事件。
- onmouseout：鼠标从对象上移出时响应事件。

本实例主要代码如下：

```
/*通过鼠标移入移出和按下松开来让图片错位*/

```

网页效果如图 5.4 所示。



图 5.4 鼠标放上图片错位效果

5.5 鼠标经过时产生浮动层放大图像

本实例使用 JavaScript 制作一个实例，该实例能在鼠标经过时放大图像，产生浮动层放大效果。本节主要涉及的 JavaScript 语法如下。

style.visibility 是控制页面控件是否显示的一个属性，style.visibility = "hidden"表示不显示该控件，style.visibility = "visible"表示显示该控件，当我们看到 style.visibility = ""时，等

同于使用 `style.visibility = "visible"`。

本实例主要代码如下：

```
<script type="text/javascript">
    function picshow()           /*图片显示函数*/
    {
        var a = document.getElementById("m") a.style.visibility =
        'visible'a.style.top = event.clientY a.style.left = event.clientX
    }
    function picmove()           /*图片移动函数*/
    {
        var b = document.getElementById("m") b.style.top = event.clientY
        b.style.left = event.clientX
    }
    function pichide()           /*图片隐藏函数*/
    {
        var c = document.getElementById("m") c.style.visibility = 'hidden'
    }
    /*主函数*/
    function trfunc()
    {
        var d = document.getElementsByTagName("tr") var len = d.length;
        d[1].attachEvent("onmouseover", picshow);
        d[1].attachEvent("onmousemove", picmove);
        d[1].attachEvent("onmouseout", pichide);
    }
</script>
```

网页效果如图 5.5 所示。



图 5.5 鼠标经过放大图像，浮动层放大图像

5.6 鼠标经过图片放大效果

本实例使用 JavaScript 制作一个实例，该实例中的图片在鼠标经过时会放大。本节主要涉及的 JavaScript 语法如下。

❑ `window.event.clientX` 所获取的坐标是相对于窗口左上角而言。

□ window.event.clientY 所获取的坐标是同样也相对于窗口左上角而言。

本实例主要代码如下：

```
<script type="text/javascript">
/*鼠标移入显示大图片*/
function in (t)
{
    var ei = document.getElementById("big_image");
    ei.style.display = "block";
    ei.innerHTML='';
    ei.style.top = document.body.scrollTop + window.event.clientY + 10
    + "px";
    ei.style.left = window.event.clientX + 10 + "px";
}
/*鼠标移出隐藏大图片*/
function out()
{
    var ei = document.getElementById("big_image");
    ei.innerHTML = "";
    ei.style.display = "none";
}
</script>
```

网页效果如图 5.6 所示。



图 5.6 鼠标经过图片放大效果

5.7 鼠标经过图片更换图片

本实例使用 JavaScript 制作一个实例，该实例中的图片在鼠标经过时会自动更换为其他候选图片。本节主要涉及的 JavaScript 语法如下。

 标签在网页显示的图片，用 来实现，这里的 picture 为图片在本地计算机上的具体文件信息，或者以 url 存储在网络上的文件位置。

本实例主要代码如下：

```
<script type="text/javascript">
/*鼠标经过*/
function mouseOver()
```

```

{
    document.b1.src = "5701.jpg"
}
/*鼠标离开*/
function mouseOut()
{
    document.b1.src = "5702.jpg"
}
}
</script>

```

网页效果如图 5.7 所示。



图 5.7 鼠标经过图片更换图片

5.8 鼠标经过图片时，下方出现注释

本实例使用 JavaScript 制作一个实例，该实例中的图片在鼠标经过时，会在图片下方出现对于该图片的注释。本节主要涉及的 JavaScript 语法如下。

`<span>` 标签被用来组合文档中的行内元素，使用 `<span>` 标签，就能更好地通过样式来格式化它们，`<span>` 标签没有固定的格式表现，只有对它应用样式时，它才会产生视觉上的效果。

本实例主要代码如下：

```

/*<span>标签被用来组合文档中的行内元素*/
<span>
    按钮多变幻！
</span>

```

网页效果如图 5.8 所示。



图 5.8 鼠标经过图片时，下方出现注释

5.9 鼠标经过时，图片由灰色变彩色

本实例使用 JavaScript 制作一个实例，该实例中的图片在鼠标经过时会由灰色变为彩色。本节主要涉及的 JavaScript 语法是 `this.className`。

`this.className` 指的是页面中某个元素的样式，本实例中 `this.className = 'frmsearch-active'` 的作用是改变图片的样式为 'frmsearch-active'。

本实例主要代码如下：

```
<script type="text/javascript">
    /*鼠标经过时，图片由灰色变彩色*/
    (function()
    {
        var D = document,
            frm = D.getElementById('frmsearch');
        if (!frm)
        {
            return false;
        }
        /*鼠标移入*/
        frm.onmouseover = function()、
        {
            this.className = 'frmsearch-active';
        };
        /*鼠标移出*/
        frm.onmouseout = function()
        {
            this.className = 'frmsearch-normal';
        };
    })();
</script>
```

网页效果如图 5.9 所示。



图 5.9 鼠标经过时，图片由灰色变彩色

5.10 图片在鼠标的控制下横向滚动

本实例使用 JavaScript 制作一个鼠标控制的横向滚动实例。本节主要涉及的 JavaScript 语法如下。

1. window.setInterval()函数

window.setInterval()的语法是 setInterval(code,millsec)，该函数按照指定的周期来调用函数或计算表达式，单位是毫秒。

2. window.clearInterval()方法

window.clearInterval()将取消由 setInterval()方法设置的定时器。setInterval()方法会不停地调用函数，直到用 clearInterval()方法终止定时或窗口被关闭。

本实例主要代码如下：

```
<script>
    var dir = 1;           /*步长*/
    var speed = 10;        /*循环周期*/
    var MyMar = null;
    /*正常移动*/
    function Marquee() {
        var demo = document.getElementById("demo");
        var demo2 = document.getElementById("demo2");
        if (dir > 0 && (demo2.offsetWidth - demo.scrollLeft) <= 0) {
            demo.scrollLeft = 0;
        }
        if (dir < 0 && (demo.scrollLeft <= 0)) {
            demo.scrollLeft = demo2.offsetWidth;
        }
        demo.scrollLeft += dir;
    }
    /*暂停移动*/
    function onmouseoverMy() {
        window.clearInterval(MyMar);
    }
    /*继续移动*/
    function onmouseoutMy() {
        MyMar = setInterval(Marquee, speed);
    }
    /*换向左移*/
    function r_left() {
        if (dir == -1) dir = 1;
    }
    /*换向右移*/
    function r_right() {
        if (dir == 1) dir = -1;
    }
    /*判断是否 IE 浏览器*/
    function IsIE() {
        var browser = navigator.appName
        if ((browser == "Netscape")) {
            return false;
        } else if (browser == "Microsoft Internet Explorer") {
            return true;
        } else {
            return null;
        }
    }
    var _IsIE = IsIE();
    var _MousePX = 0;
```

```

var _MousePY = 0;
var _DivLeft = 0;
var _DivRight = 0;
var _AllDivWidth = 0;
var _AllDivHeight = 0;
function MoveDiv(e)
{
    var obj = document.getElementById("demo");
    _MousePX = _IsIE ? (document.body.scrollLeft + event.clientX) :
    e.pageX;
    _MousePY = _IsIE ? (document.body.scrollTop + event.clientY) :
    e.pageY;
    /*Opera 浏览器支持 'window.event' 和 'e.pageX' 方法*/
    var obj1 = null;
    if (obj.getBoundingClientRect)
    {
        /*IE 浏览器*/
        obj1 = document.getElementById("demo").getBoundingClientRect();
        _DivLeft = obj1.left;
        _DivRight = obj1.right;
        _AllDivWidth = _DivRight - _DivLeft;
    }
    else if (document.getBoxObjectFor)
    {
        /*FireFox 浏览器*/
        obj1 = document.getBoxObjectFor(obj);
        var borderwidth = (obj.style.borderLeftWidth != null &&
        obj.style.borderLeftWidth != "") ? parseInt(obj.style.borderLeft
        Width) : 0;
        _DivLeft = parseInt(obj1.x) - parseInt(borderwidth);
        _AllDivWidth = Cut_Px(obj.style.width);
        _DivRight = _DivLeft + _AllDivWidth;
    }
    else
    {
        /*其他浏览器 (如 Opera) */
        _DivLeft = obj.offsetLeft;
        _AllDivWidth = Cut_Px(obj.style.width);
        var parent = obj.offsetParent;
        if (parent != obj)
        {
            while (parent)
            {
                _DivLeft += parent.offsetLeft;
                parent = parent.offsetParent;
            }
        }
        _DivRight = _DivLeft + _AllDivWidth;
    }
    var pos1, pos2;
    pos1 = parseInt(_AllDivWidth * 0.4) + _DivLeft;
    pos2 = parseInt(_AllDivWidth * 0.6) + _DivLeft;
    if (_MousePX > _DivLeft && _MousePX < _DivRight)
    {
        /*左移*/
        if (_MousePX > _DivLeft && _MousePX < pos1)
        {
            r_left();
        }
        /*右移*/
    }
}

```

```

else if (_MousePX < _DivRight && _MousePX > pos2)
{
    r_right();
}
/*停止*/
if (_MousePX > pos1 && _MousePX < pos2)
{
    onmouseoverMy();
    MyMar = null;
}
else if (_MousePX < pos1 || _MousePX > pos2)
{
    if (MyMar == null)
    {
        MyMar = setInterval(Marquee, speed);
    }
}
}
}
/*切割像素*/
function Cut_Px(cswidth)
{
    cswidth = cswidth.toLowerCase();
    if (cswidth.indexOf("px") != -1) {
        cswidth.replace("px", "");
        cswidth = parseInt(cswidth);
    }
    return cswidth;
}
/*移出图层*/
function MoveOutDiv()
{
    if (MyMar == null)
    {
        MyMar = setInterval(Marquee, speed);
    }
}
}
</script>

```

网页效果如图 5.10 所示。



图 5.10 图片在鼠标的控制下横向滚动

5.11 鼠标拖动图片

本实例使用 JavaScript 制作一个实例，该实例中的图片能够用鼠标进行拖动。本节主要涉及的 JavaScript 语法如下。

1. 函数重写

函数重写也就是将原函数覆盖为我们想要执行的函数，我们只要在脚本中编写相同名称的函数，就可以将原函数覆盖，在调用的时候，就已经是我们重写的函数了。

2. window.document.ondragstart()方法

window.document.ondragstart()就是在页面上单击鼠标并开始拖动时触发的事件。

本实例主要代码如下：

```
<SCRIPT language=JavaScript>
    drag = 0
    move = 0
    /*初始化*/
    function init()
    {
        window.document.onmousemove = mouseMove
        window.document.onmousedown = mouseDown
        window.document.onmouseup = mouseUp
        window.document.ondragstart = mouseStop
    }
    /*鼠标按下*/
    function mouseDown()
    {
        if (drag) {
            clickleft = window.event.x - parseInt(dragObj.style.left)
            clicktop = window.event.y - parseInt(dragObj.style.top)
            dragObj.style.zIndex += 1
            move = 1
        }
    }
    /*鼠标停止*/
    function mouseStop()
    {
        window.event.returnValue = false
    }
    /*鼠标移动*/
    function mouseMove()
    {
        if (move) {
            dragObj.style.left = window.event.x - clickleft
            dragObj.style.top = window.event.y - clicktop
        }
    }
    /*鼠标松开*/
    function mouseUp()
    {
        move = 0
    }
</SCRIPT>
```

网页效果如图 5.11 所示。



图 5.11 鼠标拖动图片

5.12 鼠标拖动时图片自由变化大小

本实例使用 JavaScript 制作一个实例，该实例中的图片可以在鼠标的拖动下自动变大、变小。本节主要涉及的 JavaScript 语法如下。

标签定义的是网页中显示的图片，用可以控制该元素在页面中的宽度和长度。

本实例主要代码如下：

```
<SCRIPT LANGUAGE="JavaScript">
  /*重置图片大小函数*/
  function resizeImage(evt, name)
  {
    newX = evt.x newY = evt.y eval("document." + name + ".width=newX")
    eval("document." + name + ".height=newY")
  }
</script>
```

网页效果如图 5.12 所示。

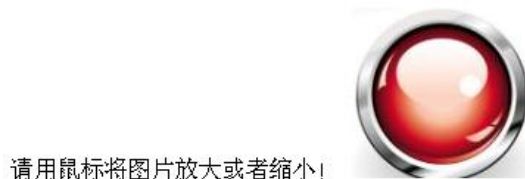


图 5.12 鼠标拖动时图片自由变化大小

5.13 鼠标移出图片抖动

本实例使用 JavaScript 制作一个实例，该实例的图片在鼠标移出时会不停地抖动。本

节主要涉及的 JavaScript 语法如下。

document.getElementById()方法：在 window.onload 事件的使用中，我们常常看到 document.getElementById()方法，该方法常用于获取元素，其最初被定义为 HTML DOM 接口的成员，之后在 2 级 DOM 中移入 XML DOM 接口。document.getElementById 属于 host 对象，是一个方法。

本实例主要代码如下：

```
<script language="JavaScript1.2">
  var rector = 3
  var stopit = 0
  var a = 1
  /*初始化函数*/
  function init(which)
  {
    stopit = 0 shake = which shake.style.left = 0 shake.style.top = 0
  }
  /*图像抖动函数*/
  function rattleimage()
  {
    if ((!document.all && !document.getElementById) || stopit == 1)
      return if (a == 1)
      {
        shake.style.top = parseInt(shake.style.top) + rector
      }
    else if (a == 2)
    {
      shake.style.left = parseInt(shake.style.left) + rector
    }
    else if (a == 3)
    {
      shake.style.top = parseInt(shake.style.top) - rector
    }
    else
    {
      shake.style.left = parseInt(shake.style.left) - rector
    }
    if (a < 4)
      a++
    else
      a = 1
    setTimeout("rattleimage()", 50)
  }
  /*图像停止抖动*/
  function stoprattle(which)
  {
    stopit = 1 which.style.left = 0 which.style.top = 0
  }
</script>
```

网页效果如图 5.13 所示。



图 5.13 鼠标移出图片抖动

5.14 图片透明度随着鼠标移入移出而改变

本实例使用 JavaScript 制作一个实例，该实例中图片透明度会随着鼠标移入移出而改变。本节主要涉及的 JavaScript 语法如下。

window.onload 是网页加载时发生的事件，该事件在页面加载完成后立即发生，可通过将函数赋值给 window.onload 事件，来达到在页面加载完成后立即执行相关函数的目的。

本实例主要代码如下：

```
<script>
/*页面加载函数*/
window.onload = function() {
    var oLi = document.getElementsByTagName("li");
    for (var i = 0; i < oLi.length; i++)
    {
        oLi[i].onmouseover = function()
        {
            this.className = "current"
        };
        oLi[i].onmouseout = function()
        {
            this.className = ""
        }
    }
}
</script>
```

网页效果如图 5.14 所示。



图 5.14 图片透明度随着鼠标移入移出而改变

5.15 鼠标移入缩略图显示出大图

本实例使用 JavaScript 制作一个实例，在该实例中，当鼠标移入缩略图时，能显示该缩略图对应的大图。本节主要涉及的 JavaScript 语法如下。

- document.createElement :document.createElement()是在对象中创建一个对象的方法。
- document.body.appendChild :document.body.appendChild()方法在节点的子节点列表未添加新的子节点。

本实例主要代码如下：

```
<script language="javascript" type="text/javascript">
    var $ = function(thisId)
    {
        return document.getElementById(thisId)
    };
    var $$ = function(othisId, thoseTag)
    {
        return othisId.getElementsByTagName(thoseTag)
    };
    /*增加加载事件*/
    function addLoadEvent(func)
    {
        var oldonload = window.onload;
        if (typeof window.onload != 'function')
        {
            window.onload = func;
        }
        else
        {
            window.onload = function() {
                oldonload();
                func();
            }
        }
    }
    /*准备图片函数*/
    function preparePhoto()
    {
        if (!$)
        {
            return false
        };
        if (!$$)
        {
            return false
        };
        var links = $$($("linkBox"), "a");
        for (i = 0; i < links.length; i++)
        {
            links[i].onclick = function()
            {
                return showPic(this);
            }
        }
    }
</script>
```

```

        links[i].onmousemove = function()
        {
            return showPic(this);
        }
    }
}
/*显示图片*/
function showPic(url)
{
    if (!$("photoHandler"))
    {
        var photoHandler = document.createElement("div");/*创建 DIV 容器*/
        photoHandler.id = "photoHandler";                /*设置 ID*/
        var textBox = document.createElement("p");        /*创建一个段落*/
        var textContent = document.createTextNode("hello,here is come the text");
                                                         /*设置文本*/
        textBox.appendChild(textContent);                /*把文本附加到段落*/
        textBox.id = "textBox";                          /*设置段落的 ID*/
        var imgBox = document.createElement("img"); /*创建一个图片容器*/
        imgBox.id = "imgBox";                            /*设置图片的 ID*/
        photoHandler.appendChild(imgBox);                /*把图片容器附加到 DIV 容器*/
        $("container").appendChild(photoHandler);/*把 DIV 附加到文档中*/
        $("container").insertBefore(textBox, $("linkBox"))
                                                         /*把创建的那个段落插到文档中*/
        /*总的思路就是先创建好树，在 appendChild/insertBefore 文档中*/
    }
    var scoure = url.getAttribute("href");
    var decripPhoto = url.getAttribute("title");
    $("imgBox").setAttribute("src", scoure);
    $("textBox").firstChild.nodeValue = decripPhoto;
    return false;
}
addLoadEvent(preparePhoto);
</script>

```

网页效果如图 5.15 所示。



图 5.15 鼠标移上图片，变换出大图片

5.16 图片在鼠标移入时变清晰

本实例使用 JavaScript 制作一个实例，该实例中的图片在鼠标移入时会变得清晰。本节主要涉及的 JavaScript 语法如下。

- jQuery fadeTo 事件：fadeTo()方法将被选元素的不透明度逐渐地改变为指定的值。
- jQuery hover 事件：hover(over,out)是一个模仿悬停事件，也就是鼠标移动到一个对象上及移出这个对象的方法。这是一个自定义的方法，它为频繁使用的任务提供了一种“保持在其中”的状态。

本实例主要代码如下：

```
<script type="text/javascript">
$(document).ready(function()
{
    $(".rollover").css
    ({
        'opacity': '0'
    });
    /*模仿悬停事件 */
    $('.img_list a').hover(function()
    {
        /*将被选元素的不透明度逐渐地改变为指定的值*/
        $(this).find('.rollover').stop().fadeTo(500, 1);
    },
    function()
    {
        $(this).find('.rollover').stop().fadeTo(500, 0);
    })
});
</script>
```

网页效果如图 5.16 所示。



图 5.16 图片在鼠标移入时变清晰

5.17 图片边框在鼠标经过时变色

本实例使用 JavaScript 制作一个实例，该实例中的图片边框在鼠标经过时会变成灰色。本节主要涉及的 JavaScript 语法是 jQuery hover 事件，该事件的说明 5.16 节已讲过，这里不再复述。

本实例主要代码如下：

```
<style>
  *{margin:0;padding:0;list-style:none;font-size:14px} h1{margin:10px;}
  img{border:1px solid #ccc} .info li { padding:8px;width:140px;
  float:left}
  .info li .img { padding:8px; border:1px solid #CBCBCB;display:
  block;} .info
  li .img:link,.info li .img:visited {border:1px solid #CBCBCB;} .info li
  .img:hover{border:1px solid #CBCBCB;background:#f0f0f0;}
  /*模仿悬停事件*/
</style>
```

网页效果如图 5.17 所示。



图 5.17 图片边框在鼠标经过时变色

5.18 在网页上单击鼠标的地方显示图片

本实例使用 JavaScript 制作一个实例，在网页上单击鼠标的地方显示图片。本节主要涉及的 JavaScript 语法如下。

标签：标签定义的是网页中显示的图片，用可以控制该元素在页面中是否可见。

本实例主要代码如下：

```
<script language="javascript">
  /*显示图片事件*/
  function show()
  {
    pic.style.left = event.x;
    pic.style.top = event.y;
    pic.style.visibility = "visible";
  }
  document.onclick = show;
</script>
```

网页效果如图 5.18 所示。

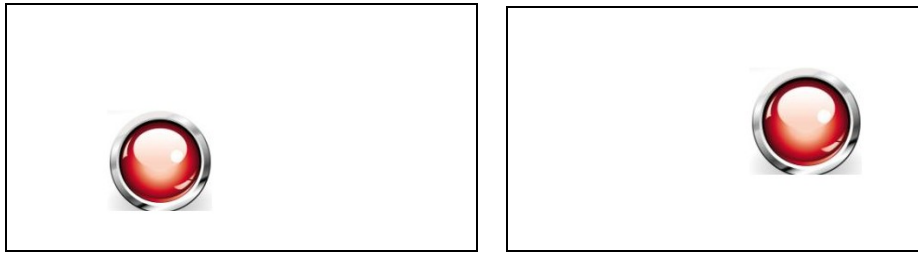


图 5.18 在网页上单击鼠标的地方显示图片

5.19 鼠标滑入时图片变清晰，鼠标划出时图片变模糊

本实例使用 JavaScript 制作一个实例，鼠标滑入时图片变清晰，鼠标划出时图片变模糊。本节主要涉及的 JavaScript 语法如下。

obj.filters.alpha.opacity：该属性用于单独设置某个滤镜。

本实例主要代码如下：

```
<script language="JavaScript">
    /*使图片可见函数*/
    function makevisible(cur, which)
    {
        if (which == 0)
            cur.filters.alpha.opacity = 100      /* obj.filters.alpha.
            opacity 属性用于单独设置某个滤镜*/
        else
            cur.filters.alpha.opacity = 20
        }
        var min = 10;
        var max = 100;
        var number = 1;
        /*鼠标移入事件*/
        function OnMouseOver(obj)
        {
            if (obj.changing) clearInterval(obj.changing);
            obj.changing = setInterval("add(" + obj.id + ")", 10);
        }
        /*鼠标移出事件*/
        function OnMouseOut(obj)
        {
            if (obj.changing) clearInterval(obj.changing);
            obj.changing = setInterval("sub(" + obj.id + ")", 10);
        }
        /*设置滤镜值最大*/
        function add(obj)
        {
            if (obj.filters.Alpha.Opacity > max)
            {
                clearInterval(obj.changing);
            }
        }
    }
</script>
```

```

        obj.changing = false;
        obj.filters.Alpha.Opacity = max;
    }
    else
        obj.filters.Alpha.Opacity += number;
    test1.innerText = "图片一：" + img1.filters.Alpha.Opacity;
    test2.innerText = "图片二：" + img2.filters.Alpha.Opacity;
}
/*设置滤镜值最小*/
function sub(obj)
{
    if (obj.filters.Alpha.Opacity < min)
    {
        clearInterval(obj.changing);
        obj.changing = false;
        obj.filters.Alpha.Opacity = min;
    }
    else
        obj.filters.Alpha.Opacity += -number;
    test1.innerText = "图片一：" + img1.filters.Alpha.Opacity;
    test2.innerText = "图片二：" + img2.filters.Alpha.Opacity;
}
}
</script>

```

网页效果如图 5.19 所示。

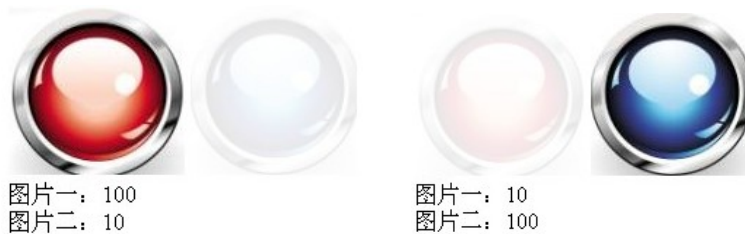


图 5.19 鼠标滑入时图片变清晰，鼠标划出时图片变模糊

5.20 图片响应鼠标特效

本实例使用 JavaScript 制作一个实例，该实例实现图片响应鼠标特效。本节主要涉及的 JavaScript 语法如下。

obj.filters.alpha.opacity：obj.filters.alpha.opacity 属性用于单独设置某个滤镜。

本实例主要代码如下：

```

<script language="JavaScript1.2">
    /*设置图片可见度*/
    function makevisible(cur, which)
    {
        if (which == 0)
            cur.filters.alpha.opacity = 100
        else
            cur.filters.alpha.opacity = 20
    }

```

```
</script>
```

网页效果如图 5.20 所示。



图 5.20 图片响应鼠标特效

5.21 鼠标滑入动画加载大图

本实例使用 JavaScript 制作一个实例，该实例展现的是鼠标滑入动画加载大图的效果。本节主要涉及的 JavaScript 语法如下。

`document.compatMode`：该属性用来判断当前浏览器采用的渲染方式，该属性对我们开发兼容的网页很有帮助。由于 IE 对盒模型的渲染在 Standards Mode 和 Quirks Mode 两种模式下有很大差别，在 Standards Mode 下对于盒模型的解释和其他的标准浏览器是一样的，但在 Quirks Mode 模式下则有很大差别。不声明 Doctype 的时候，IE 默认是 Quirks Mode。只要需要考虑兼容性，我们就要获取当前的文档渲染方式。根据 `document.compatMode` 的值也能得出文档是否加了标准声明。

本实例主要代码如下：

```
<script language="javascript">
    /*获取视窗高度*/
    function getViewportHeight()
    {
        if (window.innerHeight != window.undefined)
            return window.innerHeight;
        if (document.compatMode == 'CSS1Compat')
            return document.documentElement.clientHeight;
        if (document.body)
            return document.body.clientHeight;
        return window.undefined;
    }
    /*获取视窗宽度*/
    function getViewportWidth()
    {
        if (window.innerWidth != window.undefined)
            return window.innerWidth;
        if (document.compatMode == 'CSS1Compat')
            return document.documentElement.clientWidth;
        if (document.body)
            return document.body.clientWidth;
        return window.undefined;
    }
    /*获取滚动条顶端位置*/
```

```

function getScrollTop()
{
    if (self.pageYOffset)
    {
        return self.pageYOffset;
    }
    else if (document.documentElement && document.documentElement.
scrollTop)
    {
        return document.documentElement.scrollTop;
    }
    else if (document.body)
    {
        return document.body.scrollTop;
    }
}
/*获取滚动条左边位置*/
function getScrollLeft() {
    if (self.pageXOffset) {
        return self.pageXOffset;
    } else if (document.documentElement && document.documentElement.s
crollLeft) {
        return document.documentElement.scrollLeft;
    } else if (document.body) {
        return document.body.scrollLeft;
    }
}
var rT = true;           /*允许图像过渡*/
var bT = true;           /*允许图像淡入淡出*/
var tw = 150;            /*提示框宽度*/
/*结束动画*/
var endaction = false;
var ns4 = document.layers;
var ns6 = document.getElementById && !document.all;
var ie4 = document.all;
offsetX = 10;
offsetY = 20;
var toolTipSTYLE = "";
function initToolTips()
{
    tempDiv = document.createElement("div");
    tempDiv.id = "toolTipLayer";
    tempDiv.style.position = "absolute";
    tempDiv.style.display = "none";
    document.body.appendChild(tempDiv);
    if (ns4 || ns6 || ie4)
    {
        /*如果是网景浏览器 4.0*/
        if (ns4)
            toolTipSTYLE = document.toolTipLayer;
        /*如果是网景浏览器 6.0*/
        else if (ns6)
            toolTipSTYLE = document.getElementById("toolTipLayer")
                .style;
        /*如果是网景浏览器 4.0*/
        else if (ie4)
            toolTipSTYLE = document.all.toolTipLayer.style;
        if (ns4)
            document.captureEvents(Event.MOUSEMOVE);
    }
}

```

```

        else
        {
            toolTipSTYLE.visibility = "visible";
            toolTipSTYLE.display = "none";
        }
        document.onmousemove = moveToMouseLoc;
    }
}
/*工具条函数*/
function toolTip(msg, fg, bg)
{
    try
    {
        /*隐藏*/
        if (toolTip.arguments.length < 1)
        {
            if (ns4)
            {
                toolTipSTYLE.visibility = "hidden";
            }
            else
            {
                if (!endaction)
                {
                    toolTipSTYLE.display = "none";
                }
                if (rT) document.all("msg1").filters[1].Apply();
                if (bT) document.all("msg1").filters[2].Apply();
                document.all("msg1").filters[0].opacity = 0;
                if (rT) document.all("msg1").filters[1].Play();
                if (bT) document.all("msg1").filters[2].Play();
                if (rT)
                {
                    if (document.all("msg1").filters[1].status == 1 ||
                        document.all("msg1").filters[1].status == 0)
                    {
                        toolTipSTYLE.display = "none";
                    }
                }
                if (bT)
                {
                    if (document.all("msg1").filters[2].status == 1 ||
                        document.all("msg1").filters[2].status == 0)
                    {
                        toolTipSTYLE.display = "none";
                    }
                }
                if (!rT && !bT) toolTipSTYLE.display = "none";
            }
        }
    }
    else
    {
        if (!fg) fg = "#777777";
        if (!bg) bg = "#eeeeee";
        var content = '<table id="msg1" name="msg1" border="0" ' +
            'cellspacing="0" cellpadding="1" bgcolor="' + fg + '" class=" ' +
            'trans_msg"><td>' + '<table border="1" cellspacing="2" ' +
            'cellpadding="3" bgcolor="' + bg + '"><td><font face="Arial" ' +
            'color="' + fg + '" size="-2">' + msg + '</font></td></table>' +
            '</td></table>';
        if (ns4)

```

```

        {
            toolTipSTYLE.document.write(content);
            toolTipSTYLE.document.close();
            toolTipSTYLE.visibility = "visible";
        }
        if (ns6)
        {
            document.getElementById("toolTipLayer").innerHTML = content;
            toolTipSTYLE.display = 'block'
        }
        if (ie4)
        {
            document.all("toolTipLayer").innerHTML = content;
            toolTipSTYLE.display = 'block'
            /*图像过渡, 淡入处理*/
            var cssopaction = document.all("msg1").filters[0].opacity
            document.all("msg1").filters[0].opacity = 0;
            if (rT) document.all("msg1").filters[1].Apply();
            if (bT) document.all("msg1").filters[2].Apply();
            document.all("msg1").filters[0].opacity = cssopaction;
            if (rT) document.all("msg1").filters[1].Play();
            if (bT) document.all("msg1").filters[2].Play();
        }
    }
}
catch(e) {}
}
/*移动到鼠标位置的函数*/
function moveToMouseLoc(e)
{
    var scrollTop = getScrollTop();
    var scrollLeft = getScrollLeft();
    if (ns4 || ns6)
    {
        x = e.pageX + scrollLeft;
        y = e.pageY - scrollTop;
    }
    else
    {
        x = event.clientX + scrollLeft;
        y = event.clientY;
    }
    if (x - scrollLeft > getViewportWidth() / 2)
    {
        x = x - document.getElementById("toolTipLayer").offsetWidth - 2
        * offsetX;
    }
    if((y + document.getElementById("toolTipLayer").offsetHeight +
    offsetY) > getViewportHeight())
    {
        y = getViewportHeight() - document.getElementById("toolTip
        Layer").offsetHeight - offsetY;
    }
    toolTipSTYLE.left = (x + offsetX) + 'px';
    toolTipSTYLE.top = (y + offsetY + scrollTop) + 'px';
    return true;
}
/*初始化工具条*/
initToolTips();
</script>

```

网页效果如图 5.21 所示。

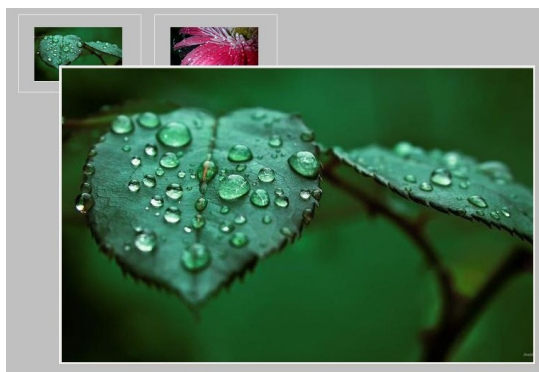


图 5.21 鼠标滑入动画加载大图

5.22 鼠标滑过图片出现边框

本实例使用 JavaScript 制作一个实例，展现鼠标滑过图片出现边框。本节主要涉及的 JavaScript 语法是 `style.borderColor`，用于设置网页中某个元素的边框颜色。

本实例主要代码如下：

```
<script language="JavaScript">
/*显示边框的函数*/
function borderit(which, color)
{
    if (document.all || document.getElementById)
    {
        which.style.borderColor = color /*设置网页中某个元素的边框颜色*/
    }
}
</script>
```

网页效果如图 5.22 所示。



图 5.22 鼠标滑过图片出现边框

5.23 鼠标滑入图片改变透明度

本实例使用 JavaScript 制作一个实例，该实例能通过鼠标滑入图片改变透明度。本节主要涉及的 JavaScript 语法如下。

obj.filters.alpha.opacity：该属性用于单独去设置某个滤镜。

本实例主要代码如下：

```
<script language="JavaScript">
    /*使对象变亮的函数*/
    function high(which2)
    {
        theobject = which2 highlighting = setInterval("highlightit
        (theobject)", 100)
    }
    /*使对象变暗的函数*/
    function low(which2)
    {
        clearInterval(highlighting) which2.filters.alpha.opacity = 30
    }
    /*使对象高亮的函数*/
    function highlightit(cur2)
    {
        if (cur2.filters.alpha.opacity < 200) cur2.filters.alpha.opacity += 20
        else if (window.highlighting) clearInterval(highlighting)
    }
}
</script>
```

网页效果如图 5.23 所示。



图 5.23 鼠标滑入图片改变透明度

5.24 缩略图响应鼠标滑入图片切换

本实例使用 JavaScript 制作一个实例，该实例能通过鼠标滑入缩略图实现大图切换。本节主要涉及的 JavaScript 语法是 Math.ceil(x)函数，返回值为最接近的较大整数。

本实例主要代码如下：

```
<script type="text/javascript">
    function $(id)
    {
        return document.getElementById(id);
    }
    /*增加加载函数*/
    function addLoadEvent(func)
    {
        var oldonload = window.onload;
        if (typeof window.onload != 'function')
        {
            window.onload = func;
        }
        else
    }
```

```

    {
        window.onload = function()
        {
            oldonload();
            func();
        }
    }
}
/*移动事件*/
function moveElement(elementID, final_x, final_y, interval)
{
    if (!document.getElementById)
        return false;
    if (!document.getElementById(elementID))
        return false;
    var elem = document.getElementById(elementID);
    if (elem.movement)
    {
        clearTimeout(elem.movement);
    }
    if (!elem.style.left)
    {
        elem.style.left = "0px";
    }
    if (!elem.style.top)
    {
        elem.style.top = "0px";
    }
    var xpos = parseInt(elem.style.left);
    var ypos = parseInt(elem.style.top);
    if (xpos == final_x && ypos == final_y) {
        return true;
    }
    if (xpos < final_x) {
        var dist = Math.ceil((final_x - xpos) / 10);
        xpos = xpos + dist;
    }
    if (xpos > final_x) {
        var dist = Math.ceil((xpos - final_x) / 10);
        xpos = xpos - dist;
    }
    if (ypos < final_y) {
        var dist = Math.ceil((final_y - ypos) / 10);
        ypos = ypos + dist;
    }
    if (ypos > final_y) {
        var dist = Math.ceil((ypos - final_y) / 10);
        ypos = ypos - dist;
    }
    elem.style.left = xpos + "px";
    elem.style.top = ypos + "px";
    var repeat = "moveElement('" + elementID + "'," + final_x + "," + final_y + "," + interval + ")";
    elem.movement = setTimeout(repeat, interval);
}
/*正常分类*/
function classNormal(iFocusBtnID, iFocusTxID)
{
    var iFocusBtns = $(iFocusBtnID).getElementsByName('li');
    var iFocusTxs = $(iFocusTxID).getElementsByName('li');
    for (var i = 0; i < iFocusBtns.length; i++) {

```

```

        iFocusBtns[i].className = 'normal';
        iFocusTxs[i].className = 'normal';
    }
}
/*当前分类*/
function classCurrent(iFocusBtnID, iFocusTxID, n)
{
    var iFocusBtns = $(iFocusBtnID).getElementsByTagName('li');
    var iFocusTxs = $(iFocusTxID).getElementsByTagName('li');
    iFocusBtns[n].className = 'current';
    iFocusTxs[n].className = 'current';
}
/*变焦事件*/
function iFocusChange()
{
    if (!$('ifocus')) return false;
    $('ifocus').onmouseover = function() {
        atuoke = true
    };
    $('ifocus').onmouseout = function() {
        atuoke = false
    };
    var iFocusBtns = $('ifocus_btn').getElementsByTagName('li');
    var listLength = iFocusBtns.length;
    iFocusBtns[0].onmouseover = function() {
        moveElement('ifocus_piclist', 0, 0, 5);
        classNormal('ifocus_btn', 'ifocus_tx');
        classCurrent('ifocus_btn', 'ifocus_tx', 0);
    }
    if (listLength >= 2) {
        iFocusBtns[1].onmouseover = function() {
            moveElement('ifocus_piclist', 0, -225, 5);
            classNormal('ifocus_btn', 'ifocus_tx');
            classCurrent('ifocus_btn', 'ifocus_tx', 1);
        }
    }
    if (listLength >= 3) {
        iFocusBtns[2].onmouseover = function() {
            moveElement('ifocus_piclist', 0, -450, 5);
            classNormal('ifocus_btn', 'ifocus_tx');
            classCurrent('ifocus_btn', 'ifocus_tx', 2);
        }
    }
    if (listLength >= 4) {
        iFocusBtns[3].onmouseover = function() {
            moveElement('ifocus_piclist', 0, -675, 5);
            classNormal('ifocus_btn', 'ifocus_tx');
            classCurrent('ifocus_btn', 'ifocus_tx', 3);
        }
    }
}
setInterval('autoiFocus()', 3500);
var atuoke = false;
/*自动聚焦*/
function autoiFocus()
{
    if (!$('ifocus')) return false;
    if (atuoke) return false;
    var focusBtnList = $('ifocus_btn').getElementsByTagName('li');
    var listLength = focusBtnList.length;
    for (var i = 0; i < listLength; i++) {

```

```

        if (focusBtnList[i].className == 'current') var currentNum = i;
    }
    if (currentNum == 0 && listLength != 1) {
        moveElement('ifocus_piclist', 0, -225, 5);
        classNormal('ifocus_btn', 'ifocus_tx');
        classCurrent('ifocus_btn', 'ifocus_tx', 1);
    }
    if (currentNum == 1 && listLength != 2) {
        moveElement('ifocus_piclist', 0, -450, 5);
        classNormal('ifocus_btn', 'ifocus_tx');
        classCurrent('ifocus_btn', 'ifocus_tx', 2);
    }
    if (currentNum == 2 && listLength != 3) {
        moveElement('ifocus_piclist', 0, -675, 5);
        classNormal('ifocus_btn', 'ifocus_tx');
        classCurrent('ifocus_btn', 'ifocus_tx', 3);
    }
    if (currentNum == 3) {
        moveElement('ifocus_piclist', 0, 0, 5);
        classNormal('ifocus_btn', 'ifocus_tx');
        classCurrent('ifocus_btn', 'ifocus_tx', 0);
    }
    if (currentNum == 1 && listLength == 2) {
        moveElement('ifocus_piclist', 0, 0, 5);
        classNormal('ifocus_btn', 'ifocus_tx');
        classCurrent('ifocus_btn', 'ifocus_tx', 0);
    }
    if (currentNum == 2 && listLength == 3) {
        moveElement('ifocus_piclist', 0, 0, 5);
        classNormal('ifocus_btn', 'ifocus_tx');
        classCurrent('ifocus_btn', 'ifocus_tx', 0);
    }
}
/*加载变焦事件*/
addLoadEvent(iFocusChange);
</script>

```

网页效果如图 5.24 所示。



图 5.24 缩略图响应鼠标滑入图片切换

5.25 鼠标滑入后悬停，图片交换

本实例使用 JavaScript 制作一个实例，该实例能在鼠标滑入后悬停，并进行图片交换。本节主要涉及的 JavaScript 语法是 document.images，其能获取到当前网页的所有 image。

本实例主要代码如下：

```
<script language="javascript">
    var circleoff;
    var circleon;
    /*判断控件类型*/
    if (document.images)
    {
        circleoff = new Image(120, 120);
        circleoff.src = "5701.jpg";
        circleon = new Image(120, 120);
        circleon.src = "5702.jpg";
    }
    /*鼠标在图片上悬停*/
    function On(name)
    {
        if (document.images)
        {
            document.images['img' + name].src = circleon.src;
        }
    }
    /*鼠标移出图片*/
    function Off(name)
    {
        if (document.images)
        {
            document.images['img' + name].src = circleoff.src;
        }
    }
</script>
```

网页效果如图 5.25 所示。



图 5.25 鼠标滑入后悬停，图片交换

5.26 鼠标滑入小图，跟随鼠标移动展示大图

本实例使用 JavaScript 制作一个实例，鼠标滑入小图，跟随鼠标移动展示大图。本节主要涉及的 JavaScript 语法是 `removeChild(node)` 方法，该方法可从子节点列表中删除某个节点。如删除成功，该方法可返回被删除的节点，如失败，则返回 `null`。

本实例主要代码如下：

```
<script type="text/javascript">
    /*页面加载*/
    window.onload = function()
```

```

{
    var aLi = document.getElementsByTagName("li");
    var oBig = document.getElementById("big");
    var oLoading = oBig.getElementsByTagName("div")[0];
    var i = 0;
    for (i = 0; i < aLi.length; i++)
    {
        aLi[i].index = i;
        /*鼠标滑入小图，跟随鼠标移动展示大图*/
        aLi[i].onmouseover = function()
        {
            var oImg = document.createElement("img");
            /*图片预加载*/
            var img = new Image();
            img.src = oImg.src = aLi[this.index].getElementsByTagName(
                "img")[0].src.replace(".jpg", "_big.jpg");
            oBig.appendChild(oImg);          /*插入大图片*/
            this.className = "active";        /*鼠标移过样式*/
            oBig.style.display = oLoading.style.display = "block";
                                                /*显示 big*/

            /*判断大图是否加载成功*/
            img.complete ? oLoading.style.display = "none": (oImg.onload
            = function()
            {
                oLoading.style.display = "none";
            })
        };
        /*鼠标移动，大图容器跟随鼠标移动*/
        aLi[i].onmousemove = function(event)
        {
            var event = event || window.event;
            var iWidth = document.documentElement.offsetWidth - event.
                clientX;
            oBig.style.top = event.clientY + 20 + "px";/*设置 big 的 top 值*/
            /*设置 big 的 left 值，如果右侧显示区域不够，大图将在鼠标左侧显示*/
            oBig.style.left = (iWidth < oBig.offsetWidth + 10 ?
            event.clientX - oBig.offsetWidth - 10 : event.clientX + 10)
            + "px";
        };
        /*鼠标离开，删除大图并隐藏大图容器*/
        aLi[i].onmouseout = function()
        {
            this.className = "";
            oBig.style.display = "none";
            oBig.removeChild(oBig.lastChild)    /*移除大图片*/
        }
    }
};
</script>

```

网页效果如图 5.26 所示。



图 5.26 鼠标滑入小图，跟随鼠标移动展示大图

5.27 带图片的鼠标悬停提示

本实例使用 JavaScript 制作一个实例，该实例能实现带图片的鼠标悬停提示。本节主要涉及的 JavaScript 语法是 `<img>` 标签，该标签定义的是网页中显示的图片，用 `<img style=" ">` 可以控制该 `<img>` 元素在页面中的样式。

本实例主要代码如下：

```
<script language="javascript">
    /*根据地址显示图片*/
    function showPic(sUrl)
    {
        var x, y;
        x = event.clientX;
        y = event.clientY;
        document.getElementById("Layer1").style.left = x;
        document.getElementById("Layer1").style.top = y;
        document.getElementById("Layer1").innerHTML = "<img src=\"\" + sUrl
        + \"\">";
        document.getElementById("Layer1").style.display = "block";
    }
    /*隐藏图片*/
    function hiddenPic()
    {
        document.getElementById("Layer1").innerHTML = "";
        document.getElementById("Layer1").style.display = "none";
    }
</script>
```

网页效果如图 5.27 所示。



图 5.27 带图片的鼠标悬停提示

5.28 响应鼠标变化的绚丽图片特效

本实例使用 JavaScript 制作一个实例，展示响应鼠标变化的绚丽图片特效。本节主要涉及的 JavaScript 语法如下。

- Math.min：Math.min()返回最小值。
- Math.max：Math.max()返回最小值。

本实例主要代码如下：

```
<script type="text/javascript">
var xm = 0;
var ym = 0;
SP = {cx:0,cy:0,N:0,R:[],I:[],C:[],L:[],Id:0,
/*初始化函数*/
init: function() {
    this.scr = document.getElementById('screen');
    this.pan = document.getElementById('pan');
    this.div = this.pan.getElementsByTagName('div');
    this.scr.onselectstart = function() {
        return false;
    }
    this.scr.ondrag = function() {
        return false;
    }
    for (var i = 0,
    o; o = this.div[i]; i++) {
        if (o.className == 'frame') {
            o.l = document.createElement('div');
            o.l.className = 'legend';
            o.appendChild(o.l);
            o.r = document.createElement('div');
            o.r.className = 'slider';
            o.appendChild(o.r);
            o.r.x = 0;
            o.r.l = o.l;
            o.r.p = 0;
            o.r.s = 2;
            o.r.m = false;
            o.img = o.r.img = o.getElementsByTagName('img')[0];
            o.r.c = Math.random() * 100;
            o.r.o = 0;
            SP.R[SP.N] = o.r;
            SP.I[SP.N] = o.img.src;
            SP.L[SP.N] = o.title;
            o.title = "";
            SP.N++;
            o.r.onmouseover = function() {
                if (!this.m && this.img.complete) {
                    if (SP.O != this && !this.n) {
                        this.x = this.o.offsetWidth;
                        this.l.innerHTML = SP.L[SP.Id];
                        this.img.src = SP.I[SP.Id];
                        this.resize();
                        this.n = true;
                        if (++SP.Id >= SP.N) {
```

```

        SP.Id = 0;
        for (var i = 0,
            o; o = SP.R[i]; i++) o.n = false;
    }
    if (SP.O) {
        SP.O.s = 2;
        SP.C.push(SP.O);
    }
    this.m = true;
    SP.O = this;
    SP.Or = this;
}
}
o.r.resize = function() {
    var i = new Image();
    i.src = this.img.src;
    this.img.style.width = (i.width < this.offsetWidth) ?
    Math.round(this.offsetWidth * 1.25) + 'px': Math.round
    (i.width) + 'px';
    this.img.style.height = (i.height < this.offset
    Height) ? Math.round(this.offsetHeight * 1.25) + 'px':
    Math.round(i.height) + 'px';
    this.w=(this.img.offsetWidth - this.offsetWidth) * .5;
    this.h=(this.img.offsetHeight-this.offsetHeight)* .5;
    this.img.style.visibility = 'visible';
}
}
}
SP.resize();
SP.run();
},
/*重置图片大小的函数*/
resize: function()
{
    var o = SP.scr;
    SP.nw = o.offsetWidth;
    SP.nh = o.offsetHeight;
    SP.iw = SP.pan.offsetWidth;
    SP.ih = SP.pan.offsetHeight;
    for (SP.nx = 0, SP.ny = 0; o != null; o = o.offsetParent) {
        SP.nx += o.offsetLeft;
        SP.ny += o.offsetTop;
    }
    for (var i = 0,
        o; o = SP.R[i]; i++) o.resize();
},
/*运行函数*/
run: function()
{
    SP.cx += (((Math.max( - SP.nw, Math.min(0, (SP.nw * .5 - (xm - SP.nx)
    * 2))) * (SP.iw - SP.nw)) / SP.nw) - SP.cx) * .1;
    SP.cy += (((Math.max( - SP.nh, Math.min(0, (SP.nh * .5 - (ym - SP.ny)
    * 2))) * (SP.ih - SP.nh)) / SP.nh) - SP.cy) * .1;
    SP.pan.style.left = Math.round(SP.cx) + 'px';
    SP.pan.style.top = Math.round(SP.cy) + 'px';
    if (SP.O) {
        SP.O.c += .015;
        SP.O.img.style.left = Math.round( - SP.O.w + SP.O.w * Math.sin
        (SP.O.c * 1.1)) + 'px';
    }
}

```

```

        sP.O.img.style.top = Math.round( - sP.O.h + sP.O.h * Math.sin
(sP.O.c)) + 'px';
        sP.O.l.style.left = Math.round(sP.O.x--) + 'px';
    }
    if (sP.Or) {
        sP.Or.p -= sP.Or.s;
        sP.Or.s *= 1.1;
        if (sP.Or.p < -sP.Or.offsetHeight) {
            sP.Or.p = -sP.Or.offsetHeight;
            sP.Or.s = 2;
            sP.Or.m = false;
            sP.Or = false;
        }
        sP.O.style.top = Math.round(sP.O.p) + 'px';
    }
    for (var i = 0,
c; c = sP.C[i]; i++) {
        if (c != sP.Or) {
            c.p += c.s;
            c.s *= 1.2;
            if (c.p >= 0) {
                c.p = 0;
                c.s = 2;
                c.m = false;
                sP.C.splice(i, 1);
            }
            c.style.top = Math.round(c.p) + 'px';
        } else {
            c.s = 2;
            c.m = false;
            sP.C.splice(i, 1);
        }
    }
    setTimeout(sP.run, 16);
}
}
/*鼠标移入*/
document.onmousemove = function(e)
{
    if (window.event) e = window.event;
    xm = e.clientX;
    ym = e.clientY;
    return false;
}
</script>
<script type="text/javascript">
    sP.init();
    onresize = sP.resize;
</script>

```

网页效果如图 5.28 所示。

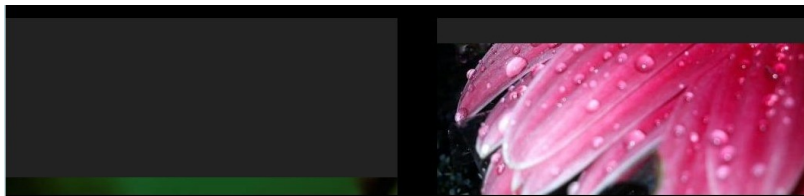


图 5.28 响应鼠标变化的绚丽图片特效

5.29 感应鼠标的图片遮罩动画效果

本实例使用 JavaScript 制作一个实例，其中的图片能感应图标并产生遮罩动画效果。本节主要涉及的 JavaScript 语法是 jQuery 的 `animate( params, [duration], [easing], [callback] )` 函数，是用于创建自定义动画的函数。这个函数的关键在于指定动画形式及结果样式属性对象。

本实例主要代码如下：

```
<script type="text/javascript">
$(document).ready(function()
{
    $('.banner div').css('opacity', 0.4);
    /*模仿悬停事件 */
    $('.banner').hover(function()
    {
        var el = $(this);
        el.find('div').stop().animate(
        {
            width: 200,
            height: 200
        },
        'slow',
        function()
        {
            /*调用渐现事件 */
            el.find('p').fadeIn('fast');
        });
    },
    function()
    {
        var el = $(this);
        el.find('p').stop(true, true).hide();
        /*创建自定义动画的函数*/
        el.find('div').stop().animate(
        {
            width: 60,
            height: 60
        },
        'fast');
    }).click(function()
    {
        window.open($(this).find('a').attr('href'));
    });
});
</script>
```

网页效果如图 5.29 所示。



图 5.29 感应鼠标的图片遮罩动画效果

5.30 鼠标滑入变换内容

本实例使用 JavaScript 制作一个实例，该实例展示鼠标滑入变换内容的效果。本节主要涉及的 JavaScript 语法是 `hide()` 函数，是 jQuery 的隐藏函数。

本实例主要代码如下：

```
<script type="text/javascript">
$(document).ready(function()
{
    $('.banner div').css('opacity', 0.4);
    /*模仿悬停事件 */
    $('.banner').hover(function()
    {
        var el = $(this);
        el.find('div').stop().animate(
        {
            width: 200,
            height: 200
        },
        'slow',
        function()
        {
            /*调用渐现事件 */
            el.find('p').fadeIn('fast');
        });
    },
    function()
    {
        var el = $(this);
        /*隐藏函数 */
        el.find('p').stop(true, true).hide();
        el.find('div').stop().animate(
        {
            width: 60,
            height: 60
        },
        'fast');
    }).click(function()
    {
        window.open($(this).find('a').attr('href'));
    });
});
</script>
```

网页效果如图 5.30 所示。

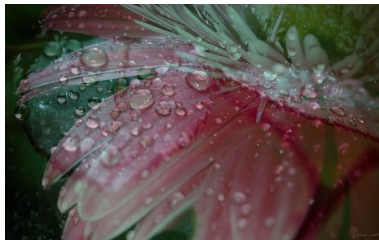


图 5.30 鼠标滑入变换内容

5.31 跟随鼠标旋转的星星

本实例使用 JavaScript 制作一个实例，展现跟随鼠标旋转的星星。本节主要涉及的 JavaScript 语法是 `Math.floor(x)` 函数，用于传回小于或等于指定数字 `x` 的最大整数。

本实例主要代码如下：

```
<script>
var x, y
var step = 5
var flag = 1
var pause
var timersmall
var timerbig
var isbigcircle = 1
var pause = 50
var bigradius
var smallradius = 50
var startwinkel = 1
var imgnumber = 12
var imgsrc = "xingxing.jpg"
var xcenter
var ycenter
var pi = Math.PI
function onresizer() {
    clearTimeout(timerbig) clearTimeout(timersmall) initiate()
}
/*初始化函数 1*/
function initiate()
{
    var initiatetimer = setTimeout("initiate2()", 2000)
}
/*初始化函数 2*/
function initiate2()
{
    if (document.layers)
    {
        xcenter = Math.floor(screen.width / 2) ycenter = Math.floor(
            window.innerHeight / 2)
    }
    if (document.all)
    {
        xcenter = Math.floor(document.body.clientWidth / 2) ycenter =
            Math.floor(document.body.clientHeight / 2)
    }
    bigradius = ycenter bigcircle()
}
function handlerMM(e)
{
    x = (document.layers) ? e.pageX: event.clientX y = (document.layers) ?
        e.pageY: event.clientY flag = 1
}
/*大循环*/
function bigcircle() {
    if (isbigcircle == 1) {
        if (document.all) {
```

```

        for (i = 1; i <= imgnumber; i++) {
            var thisspan = eval("document.all.span" + (i) + ".style")
            thisspan.posLeft = ((xcenter - 20) * Math.cos(startwinkel
            * (pi / 180))) + xcenter thisspan.posTop = ((ycenter - 20)
            * Math.sin(startwinkel * (pi / 180))) + ycenter startwinkel
            = startwinkel + 30
        }
    }
    if (document.layers) {
        for (i = 1; i <= imgnumber; i++) {
            var thisspan = eval("document.span" + i) thisspan.left =
            ((xcenter - 150) * Math.cos(startwinkel * (pi / 180))) +
            (xcenter - 145) thisspan.top = ((ycenter - 20) * Math.sin
            (startwinkel * (pi / 180))) + ycenter startwinkel =
            startwinkel + 30
        }
    }
    step = step + 5 startwinkel = step timerbig = setTimeout
    ("bigcircle()", pause)
} else {
    isbigcircle = 0 clearTimeout(timerbig)
}
}
/*调用小循环*/
function presmallcircle() {
    isbigcircle = 0 clearTimeout(timerbig) smallcircle()
}

function prebigcircle() {
    isbigcircle = 1 clearTimeout(timersmall) bigcircle()
}
/*小循环*/
function smallcircle()
{
    if (isbigcircle == 0)
    {
        if (document.all)
        {
            for (i = 1; i <= imgnumber; i++)
            {
                var thisspan = eval("document.all.span" + (i) + ".style")
                thisspan.posLeft = (smallradius * Math.cos(startwinkel *
                (pi / 180))) + x thisspan.posTop = (smallradius *
                Math.sin(startwinkel * (pi / 180))) + y startwinkel =
                startwinkel + 30
            }
        }
    }
    if (document.layers)
    {
        for (i = 1; i <= imgnumber; i++)
        {
            var thisspan = eval("document.span" + i) thisspan.left =
            (smallradius * Math.cos(startwinkel * (pi / 180))) + x
            thisspan.top = (smallradius * Math.sin(startwinkel * (pi
            / 180))) + y startwinkel = startwinkel + 30
        }
    }
    step = step + 5 startwinkel = step timersmall = setTimeout
    ("smallcircle()", pause)
}
}

```

```

else
{
    isbigcircle = 0 clearTimeout(timersmall)
}
}
window.onload = initiate()          /*调用初始化函数*/
</script>

```

网页效果如图 5.31 所示。



图 5.31 跟随鼠标旋转的星星

5.32 鼠标滑入图片渐变高亮

本实例使用 JavaScript 制作一个实例，该实例中鼠标滑入时，图片会渐渐变为高亮。本节主要涉及的 JavaScript 语法是 `obj.filters.alpha.opacity` 属性，用于单独设置某个滤镜。

本实例主要代码如下：

```

<script language="JavaScript1.2">
    /*高亮显示*/
    function high(image)
    {
        theobject = image highlighting = setInterval("highlightit
        (theobject)", 50)
    }
    /*模糊显示*/
    function low(image)
    {
        clearInterval(highlighting) image.filters.alpha.opacity = 20
    }
    /*改变图像光照效果*/
    function highlightit(cur2)
    {
        if (cur2.filters.alpha.opacity < 100) cur2.filters.alpha.opacity += 5
        else if (window.highlighting) clearInterval(highlighting)
    }
</script>

```

网页效果如图 5.32 所示。



鼠标移入后图片变清晰
鼠标移出后图片变模糊

图 5.32 渐变的图像，鼠标滑入图片高亮

5.33 图片感应鼠标的虚实效果

本实例使用 JavaScript 制作一个实例，实现图片感应鼠标的虚实效果。本节主要涉及的 JavaScript 语法是 `obj.filters.alpha.opacity` 属性，用于单独设置某个滤镜。

本实例主要代码如下：

```
<script language="javascript">
  /*设置可见度函数*/
  function makevisible(cur, which)
  {
    if (which == 0)
      cur.filters.alpha.opacity = 100
    else
      cur.filters.alpha.opacity = 20
  }
</script>
```

网页效果如图 5.33 所示。

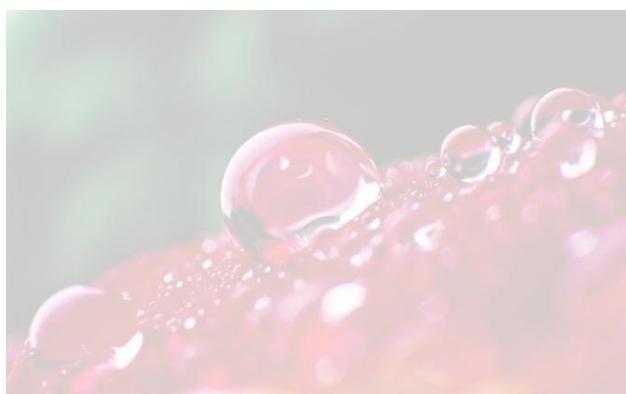


图 5.33 图片感应鼠标的虚实效果

5.34 鼠标单击控制图片左右滚动

本实例使用 JavaScript 制作一个实例，该实例能通过鼠标单击控制图片左右滚动。本节主要涉及的 JavaScript 语法如下。

- setInterval：setInterval()方法，该方法的说明 1.43 节已介绍过，不再复述。
- setTimeout：setTimeout 有两种形式，即 setTimeout(code,interval)和 setTimeout(func,interval, args)，其中 code 是一个字符串,func()是一个函数；interval 表示时间，可以是延迟时间或者交互时间，以毫秒为单位。延迟时间是在载入后延迟指定时间之后去执行一次表达式，仅执行一次；交互时间是从载入后，每隔指定的时间就执行一次表达式。

本实例主要代码如下：

```
<script language="javascript" type="text/javascript">
    /*图片滚动列表*/
    var Speed = 1;           /*速度（单位为毫秒）*/
    var Space = 5;           /*每次移动像素*/
    var PageWidth = 528;     /*翻页宽度*/
    /*整体移位*/
    var fill = 0;
    var MoveLock = false;
    var MoveTimeObj;
    var Comp = 0;
    var AutoPlayObj = null;
    GetObj("List2").innerHTML = GetObj("List1").innerHTML;
    GetObj('ISL_Cont').scrollLeft = fill;
    GetObj("ISL_Cont").onmouseover = function()
    {
        clearInterval(AutoPlayObj);
    }
    GetObj("ISL_Cont").onmouseout = function()
    {
        AutoPlay();
    }
    AutoPlay();
    /*获取对象*/
    function GetObj(objName)
    {
        if (document.getElementById)
        {
            return eval('document.getElementById("' + objName + '")')
        }
        else
        {
            return eval('document.all.' + objName)
        }
    }
    function AutoPlay()      /*自动滚动*/
    {
        clearInterval(AutoPlayObj);
        AutoPlayObj = setInterval('ISL_GoDown();ISL_StopDown();', 3000);
        /*间隔时间*/
    }
    /*上翻开始*/
    function ISL_GoUp()
    {
        if (MoveLock) return;
        clearInterval(AutoPlayObj);
        MoveLock = true;
        MoveTimeObj = setInterval('ISL_ScrUp();', Speed);
    }
    /*上翻停止*/
```

```
function ISL_StopUp()
{
    clearInterval(MoveTimeObj);
    if (GetObj('ISL_Cont').scrollLeft % PageWidth - fill != 0)
    {
        Comp = fill - (GetObj('ISL_Cont').scrollLeft % PageWidth);
        CompScr();
    }
    else
    {
        MoveLock = false;
    }
    AutoPlay();
}
/*上翻动作*/
function ISL_ScrUp()
{
    if (GetObj('ISL_Cont').scrollLeft <= 0)
    {
        GetObj('ISL_Cont').scrollLeft = GetObj('ISL_Cont').scrollLeft +
        GetObj('List1').offsetWidth
    }
    GetObj('ISL_Cont').scrollLeft -= Space;
}
/*下翻*/
function ISL_GoDown()
{
    clearInterval(MoveTimeObj);
    if (MoveLock) return;
    clearInterval(AutoPlayObj);
    MoveLock = true;
    ISL_ScrDown();
    MoveTimeObj = setInterval('ISL_ScrDown()', Speed);
}
/*下翻停止*/
function ISL_StopDown()
{
    clearInterval(MoveTimeObj);
    if (GetObj('ISL_Cont').scrollLeft % PageWidth - fill != 0)
    {
        Comp = PageWidth - GetObj('ISL_Cont').scrollLeft % PageWidth +
        fill;
        CompScr();
    }
    else
    {
        MoveLock = false;
    }
    AutoPlay();
}
/*下翻动作*/
function ISL_ScrDown()
{
    if (GetObj('ISL_Cont').scrollLeft >= GetObj('List1').scrollWidth)
    {
        GetObj('ISL_Cont').scrollLeft = GetObj('ISL_Cont').scrollLeft -
        GetObj('List1').scrollWidth;
    }
    GetObj('ISL_Cont').scrollLeft += Space;
}
```

```

/*比较屏幕*/
function CompScr()
{
    var num;
    if (Comp == 0)
    {
        MoveLock = false;
        return;
    }
    /*上翻*/
    if (Comp < 0)
    {
        if (Comp < -Space)
        {
            Comp += Space;
            num = Space;
        }
        else
        {
            num = -Comp;
            Comp = 0;
        }
        GetObj('ISL_Cont').scrollLeft -= num;
        setTimeout('CompScr()', Speed);
    }
    /*下翻*/
    else
    {
        if (Comp > Space) {
            Comp -= Space;
            num = Space;
        } else {
            num = Comp;
            Comp = 0;
        }
        GetObj('ISL_Cont').scrollLeft += num;
        setTimeout('CompScr()', Speed);
    }
}
//--><![ ]>
</script>

```

网页效果如图 5.34 所示。



图 5.34 鼠标单击控制图片左右滚动

5.35 全面禁用鼠标右键

本实例使用 JavaScript 制作一个实例，该实例全面禁用网页中的鼠标右键功能。本节主要涉及的 JavaScript 语法如下。

- window.event.ctrlKey：判断是否按下 Ctrl 按键。
- window.event.shiftKey：判断是否按下 Shift 按键。

本实例主要代码如下：

```
<script>
  /*键盘按键函数*/
  function onKeyDown()
  {
    if ((event.keyCode == 116)|| (window.event.ctrlKey)|| (window.
    event.shiftKey)|| (event.keyCode == 122))
    {
      event.keyCode = 0;
      event.returnValue = false;
    }
  }
</script>
```

网页效果如图 5.35 所示。

鼠标右键测试失灵了哦，赶紧买个新的鼠标吧 ^_^

图 5.35 全面禁用鼠标右键

5.36 全面禁用鼠标滚轮

本实例使用 JavaScript 制作一个实例，该实例全面禁用网页中的鼠标滚轮。本节主要涉及的 JavaScript 语法是 document.onmousewheel()函数，该函数是网页中的鼠标滚轮事件。

本实例主要代码如下：

```
<script language="javascript">
  /*重新定义滚轮事件*/
  function document.onmousewheel()
  {
    /*返回 false 表示什么都不操作*/
    return false;
  }
</script>
```

网页效果如图 5.36 所示。

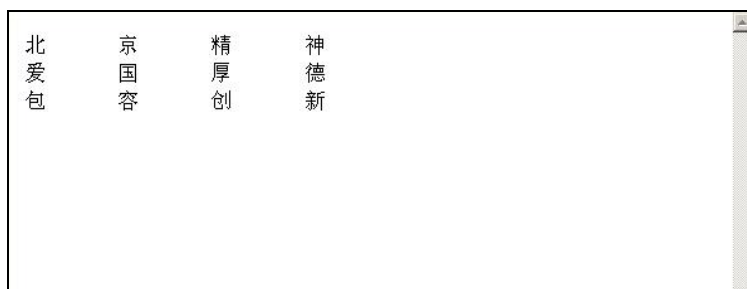


图 5.36 全面禁用鼠标滚轮

5.37 全面禁用鼠标右键复制粘贴

本实例使用 JavaScript 制作一个实例，该实例能全面禁用鼠标右键拷贝和复制。本节主要涉及的 JavaScript 语法如下。

- ❑ document.oncontextmenu：document.oncontextmenu()就是网页中的鼠标右键菜单事件。
- ❑ document.onpaste：document.onpaste()就是网页中的鼠标右键粘贴事件。
- ❑ document.oncopy：document.oncopy()就是网页中的鼠标右键复制事件。
- ❑ document.oncut：document.oncut()就是网页中的鼠标右键剪切事件。
- ❑ document.onselectstart：document.onselectstart()就是网页中的鼠标右键选择事件。

本实例主要代码如下：

```
<script language="javascript" type="text/javascript">
    document.oncontextmenu = function()                /*屏蔽右键菜单*/
    {
        return false;
    }
    document.onpaste = function()                      /*屏蔽粘贴*/
    {
        return false;
    }
    document.oncopy = function()                      /*屏蔽复制*/
    {
        return false;
    }
    document.oncut = function()                      /*屏蔽剪切*/
    {
        return false;
    }
    document.onselectstart = function()                /*屏蔽选择*/
    {
        return false;
    }
</script>
```

网页效果如图 5.37 所示。

该网页内容只能查看，无法选择、复制、粘贴且右键功能已被禁用

图 5.37 全面禁用鼠标右键复制粘贴

5.38 跟踪鼠标显示坐标

本实例使用 JavaScript 制作一个实例，该实例中，在网页上能跟踪鼠标显示坐标。本节主要涉及的 JavaScript 语法是 document.captureEvents(x)函数，用于捕捉指定参数 x 的所有事件。由于能够捕获那些由本地程序自己处理的事件，所以程序员可以随意定义函数来处理事件。

本实例主要代码如下：

```
<script language="JavaScript1.2">
  /*如果浏览器是 Netscape*/
  if (navigator.appName == 'Netscape')
  {
    document.captureEvents(Event.MOUSEMOVE);          /*捕获鼠标移动事件*/
    document.onmousemove = netscapeMouseMove; /*交由 netscapeMouseMove 处理*/
  }
  function netscapeMouseMove(e)
  {
    if (e.screenX != document.test.x.value && e.screenY != document.test.y.value) /*如果鼠标位置有改变*/
    {
      document.test.x.value = e.screenX;          /*刷新横坐标显示*/
      document.test.y.value = e.screenY;          /*刷新纵坐标显示*/
    }
  }
  function ieMouseMove()                               /*IE 鼠标移动事件*/
  {
    if (window.event.x != document.test.x.value && window.event.y != document.test.y.value) /*如果鼠标位置有改变*/
    {
      document.test.x.value = window.event.x;      /*刷新横坐标显示*/
      document.test.y.value = window.event.y;      /*刷新纵坐标显示*/
    }
  }
</script>
```

网页效果如图 5.38 所示。

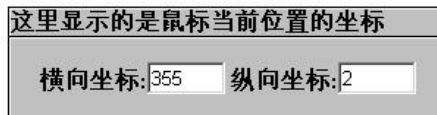


图 5.38 跟踪鼠标显示坐标

5.39 围绕鼠标跳跃的文字

本实例使用 JavaScript 制作一个实例，在网页中出现围绕鼠标跳跃的文字。本节主要涉及的 JavaScript 语法是 document.onmousemove()事件，在鼠标滑过或者鼠标放上去不动时被触发。

本实例主要代码如下：

```
<script language="JavaScript1.2">
var cx = 0;
var cy = 0;
var val = 0;
/*定位*/
function locate()
{
    cx = window.event.x;
    cy = window.event.y;
}
document.onmousemove = locate;
/*跟踪*/
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";
    }
}
/*显示*/
function show(i)
{
    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 < 5) val = setInterval("show(" + (i + 1) + ")", 20);
        }
        fontSize = s;
    }
}
/*开始*/
function start()
{
    for (i = 1; i <= 5; i++)
    {
```

```

        val = setInterval("show(1)", 20);
        setInterval("follow(" + i + ")", 100);
    }
}
var word = new Array(5);
word[1] = "北";
word[2] = "京";
word[3] = "欢";
word[4] = "迎";
word[5] = "你";
for (i = 1; i <= 5; i++) document.write("<div id='word" + i + "'
style='width:20px;height:20px;position:absolute;font-size:1000;
visibility:hidden'><font face='Forte' color='#FF0000'>" + word[i] +
"</font></div>");
start();
</script>

```

网页效果如图 5.39 所示。



图 5.39 围绕鼠标跳跃的文字

5.40 鼠标移入移出某个区域时改变颜色

本实例使用 JavaScript 制作一个实例，该实例中的某个区域在鼠标移入移出时会改变颜色。本节主要涉及的 JavaScript 语法是 `style.color`，就是某个控件的颜色属性。

本实例主要代码如下：

```

/*调用鼠标移入移出事件*/
<input type="submit" value=" 请 把 鼠 标 移 入 移 出 这 个 区 域 " name="btn1"
onMouseOut=this.style.color="green" onMouseOver=this.style.color="red">

```

网页效果如图 5.40 所示。



图 5.40 鼠标移入移出某个区域时改变颜色

5.41 鼠标移入移出时控件特效

本实例使用 JavaScript 制作一个实例，该实例是网页中的控件在鼠标移入移出时会产

生特效。本节主要涉及的 JavaScript 语法是 document.getElementById()方法，已在 5.13 节介绍过，不再复述。

本实例主要代码如下：

```
<script language="javascript">
    var node;
    var initied = false;
    var init = function()                /*初始化方法*/
    {
        node = document.getElementById("chk1"); /*找到页面中的checkbox控件*/
        initied = true;
    }
    var check_over = function()          /*鼠标移入事件绑定的方法*/
    {
        if (!initied) return;
        node.style.borderStyle = "solid";    /*设置边框*/
        node.style.borderColor = "#FFCC00";  /*设置边框颜色*/
        node.style.backgroundColor = "#EEEEEE"; /*设置背景色*/
    }
    var check_out = function()           /*鼠标移出事件绑定的方法*/
    {
        node.style.borderStyle = "none";    /*设置边框*/
        node.style.borderColor = "#FFFFFF";  /*设置边框颜色-白色*/
        node.style.backgroundColor = "#FFFFFF"; /*设置背景色*/
    }
</script>
```

网页效果如图 5.41 所示。

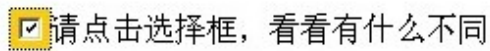


图 5.41 鼠标移入移出时控件特效

5.42 跟随鼠标的汉字

本实例使用 JavaScript 制作一个实例，该实例的网页中会出现跟随鼠标的汉字。本节主要涉及的 JavaScript 语法是 document.captureEvents(x)函数，用于捕捉指定参数 x 的所有事件。由于能够捕获那些由本地程序自己处理的事件，所以程序员可以随意定义函数来处理事件。

本实例主要代码如下：

```
<script language="javascript">
    var x, y;
    var step = 30;
    var flag = 0;
    var message = "粘在鼠标上。";
    /*将文本切割成数组*/
    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)
{
    /*判断浏览器类型并获取鼠标的坐标*/
    x = (document.layers) ? e.pageX: document.body.scrollLeft +
    event.clientX;
    y = (document.layers) ? e.pageY: document.body.scrollTop +
    event.clientY;
    flag = 1;
}
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;          /*文本的 x 坐标距离鼠标 x 坐标的距离*/
        ypos[0] = y;                 /*文本和鼠标的 y 坐标相同*/
        /*设置包装文本的 span 控件的位置*/
        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);
}
/*显示文本的重点*/
for (i = 0; i <= message.length - 1; i++)
{
    document.write("<span id='span" + i + "'class='spanstyle'>");
    document.write(message[i]);
```

```

        document.write("</span>");
    }
    /*针对 navigator 浏览器时的情况*/
    if (document.layers) {
        document.captureEvents(Event.MOUSEMOVE);
    }
    document.onmousemove = handlerMM;          /*将方法绑定到鼠标的移动事件*/
</script>

```

网页效果如图 5.42 所示。



图 5.42 跟随鼠标的汉字

5.43 跟随鼠标的小点点

本实例使用 JavaScript 制作一个实例，该实例中的网页上显示跟随鼠标的小点点。本节主要涉及的 JavaScript 语法是 `clearInterval`，`clearInterval` 事件的作用是清除对 `setInterval()` 函数的调用，它的语法格式是 `clearInterval(intervalid)`，`intervalid` 是调用 `setInterval()` 函数后返回的对象。

本实例主要代码如下：

```

<script language="JavaScript1.2">
    /*高亮显示*/
    function high(image)
    {
        theobject = image highlighting = setInterval("highlightit
        (theobject)", 50)
    }
    /*模糊显示*/
    function low(image)
    {
        clearInterval(highlighting) image.filters.alpha.opacity = 20
    }
    /*改变图像光照效果*/
    function highlightit(cur2)
    {
        if (cur2.filters.alpha.opacity < 100) cur2.filters.alpha.opacity += 5
        else if (window.highlighting) clearInterval(highlighting)
    }
</script>

```

网页效果如图 5.43 所示。

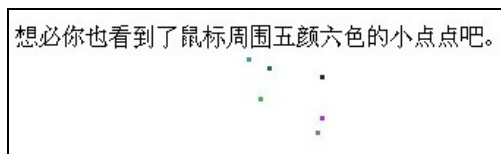


图 5.43 跟随鼠标的小点点

5.44 跟随鼠标漂移的文字

本实例使用 JavaScript 制作一个实例，该实例的网页中出现跟随鼠标漂移的文字。本节主要涉及的 JavaScript 语法如下。

`document.layers`：`document.layers` 是一个代表所有由诸如<div><layer>等定位了的元素数组。通常也是用<div>或<layer>对象的 `id` 属性来引用的，但是这里面不包含除此以外的其他元素。

本实例主要代码如下：

```
<script language="javascript">
function Div_Layer(divleft, divtop, divfnx, divfny, mydiv, divbilder,
divloop, divto, divcnt, divstep) {
    if ((document.layers) || (document.all))
    {
        with(Math)
        {
            yynextx = eval(divfnx)
        }
        with(Math)
        {
            yynexty = eval(divfny)
        }
        divcnt = (divloop && divcnt >= divstep * divbilder) ? 0 : divcnt
        + divstep;
        /*网景浏览器下设置横坐标和纵坐标*/
        if (document.layers)
        {
            eval(mydiv + ".top=" + (yynexty + divtop)) eval(mydiv +
            ".left=" + (yynextx + divleft))
        }
        if (document.all)
        {
            eval("mydiv=mydiv.replace(/.layers/gi, '.all')");
            eval(mydiv + ".style.pixelTop=" + (yynexty + divtop));
            /*设置 div 的横坐标*/
            eval(mydiv + ".style.pixelLeft=" + (yynextx + divleft));
            /*设置 div 的纵坐标*/
        }
        argStr = 'Div_Layer(' + divleft + ',' + divtop + ',' + divfnx
        + ',' + divfny + ',' + mydiv + ',' + divbilder + ',' + divloop
        + ',' + divto + ',' + divcnt + ',' + divstep + ')';
        if (divcnt <= divstep * divbilder)
        {
            eval(mydiv + ".divto=setTimeout(argStr,divto)");
            /*设置定时器-实现星星的闪烁效果*/
        }
    }
}
function YY_Mousetrace(evnt) /*设置鼠标移动的事件*/
{
    /*网景浏览器的情况*/
```

```

if (yyms4)
{
    if (evnt.pageX)
    {
        yy_ml = evnt.pageX;
        yy_mt = evnt.pageY;
    }
}
else
{
    yy_ml = (event.clientX + document.body.scrollLeft);/*横坐标位置*/
    yy_mt = (event.clientY + document.body.scrollTop);/*纵坐标位置*/
}
/*转换为数值型变量*/
if (yy_tracescript)
    eval(yy_tracescript)
}
</script>

```

网页效果如图 5.44 所示。

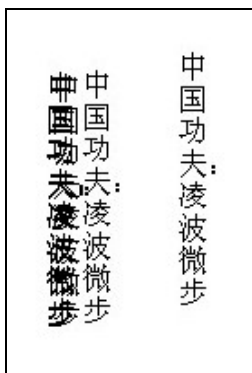


图 5.44 跟随鼠标漂移的文字

5.45 鼠标燃放烟花

本实例使用 JavaScript 制作一个实例，在网页上用鼠标控制燃放烟花。本节主要涉及的 JavaScript 语法是 `addEventListener`，`addEventListener` 事件监听会自动传递一个事件对象参数给处理函数，可是不能传递其他参数。

本实例主要代码如下：

```

<script type="text/javascript">
    var fgm =
    {
        /*事件监听*/
        on: function(element, type, handler) {
            return element.addEventListener ? element.addEventListener(type,
                handler, false) : element.attachEvent("on" + type, handler)
        },
        /*取消监听*/
    }

```

```

un: function(element, type, handler) {
    return element.removeEventListener ? element.removeEventListener(type, handler, false) : element.detachEvent("on" + type, handler)
},
/*绑定*/
bind: function(object, handler) {
    return function() {
        return handler.apply(object, arguments)
    }
},
randomRange: function(lower, upper)          /*随机范围*/
{
    return Math.floor(Math.random() * (upper - lower + 1) + lower)
},
getRanColor: function()                      /*获取颜色*/
{
    var str = this.randomRange(0, 0xFFFFFF).toString(16);
    while (str.length < 6) str = "0" + str;
    return "#" + str
}
};
function FireWorks()                          /*初始化对象*/
{
    this.type = 0;
    this.timer = null;
    this.fnManual = fgm.bind(this, this.manual)
}
FireWorks.prototype =                         /*初始化函数原型*/
{
    initialize: function()                    /*初始化函数*/
    {
        clearTimeout(this.timer);
        fgm.un(document, "click", this.fnManual);
        switch (this.type)
        {
            case 1:
                fgm.on(document, "click", this.fnManual);
                break;
            case 2:
                this.auto();
                break;
        }
    };
},
/*手动*/
manual: function(event)
{
    event = event || window.event;
    this.__create__({
        x: event.clientX,
        y: event.clientY
    });
},
/*自动*/
auto: function()
{
    var that = this;
    that.timer = setTimeout(function() {

```

```

        that.__create__({
            x: fgm.randomRange(50, document.documentElement.clientWidth - 50),
            y: fgm.randomRange(50, document.documentElement.clientHeight - 150)
        }) that.auto();
    },
    fgm.randomRange(900, 1100))
},
/*创建*/
__create__: function(param) {
    var that = this;
    var oEntity = null;
    var oChip = null;
    var aChip = [];
    var timer = null;
    var oFrag = document.createDocumentFragment();
    oEntity = document.createElement("div");
    with(oEntity.style) {
        position = "absolute";
        top = document.documentElement.clientHeight + "px";
        left = param.x + "px";
        width = "4px";
        height = "30px";
        borderRadius = "4px";
        background = fgm.getRanColor();
    };
    document.body.appendChild(oEntity);
    oEntity.timer = setInterval(function() {
        oEntity.style.top = oEntity.offsetTop - 20 + "px";
        if (oEntity.offsetTop <= param.y) {
            clearInterval(oEntity.timer);
            document.body.removeChild(oEntity); (function()
            {
                /*在 50~100 之间随机生成碎片, 由于 IE 浏览器处理效率低, 随机范围缩小至 20~30 ,
                自动放烟花时, 随机范围缩小至 20-3*/
                var len = (/msie/i.test(navigator.userAgent) ||
                    that.type == 2) ? fgm.randomRange(20, 30) : fgm.random
                    Range(50, 100) for (i = 0; i < len; i++) {
                    oChip = document.createElement("div");
                    with(oChip.style) {
                        position = "absolute";
                        top = param.y + "px";
                        left = param.x + "px";
                        width = "4px";
                        height = "4px";
                        overflow = "hidden";
                        borderRadius = "4px";
                        background = fgm.getRanColor();
                    };
                    oChip.speedX = fgm.randomRange( - 20, 20);
                    oChip.speedY = fgm.randomRange( - 20, 20);
                    oFrag.appendChild(oChip);
                    aChip[i] = oChip
                };
                document.body.appendChild(oFrag);
                timer = setInterval(function() {
                    for (i = 0; i < aChip.length; i++) {
                        var obj = aChip[i];
                        with(obj.style) {
                            top = obj.offsetTop + obj.speedY + "px";

```

```

        left = obj.offsetLeft + obj.speedX + "px";
    };
    obj.speedY++; (obj.offsetTop < 0 || obj.offsetLeft < 0 || obj.offsetTop > document.documentElement.clientHeight || obj.offsetLeft > document.documentElement.clientWidth) && (document.body.removeChild(obj), aChip.splice(i, 1));
    }; ! aChip[0] && clearInterval(timer);
    },
    30)
    })()
    }
    },
    30)
    }
};
/*重写 load()函数*/
fgm.on(window, "load", function()
{
    var oTips = document.getElementById("tips");
    var aBtn = oTips.getElementsByTagName("a");
    var oFireWorks = new FireWorks();
    fgm.on(oTips, "click",
    function(event) {
        var oEvent = event || window.event;
        var oTarget = oEvent.target || oEvent.srcElement;
        var i = 0;
        if (oTarget.tagName.toUpperCase() == "A")
        {
            for (i = 0; i < aBtn.length; i++) aBtn[i].className = "";
            switch (oTarget.id) {
                case "manual":
                    oFireWorks.type = 1;
                    break;
                case "auto":
                    oFireWorks.type = 2;
                    break;
                case "stop":
                    oFireWorks.type = 0;
                    break;
            }
            oFireWorks.initialize();
            oTarget.className = "active";
            oEvent.stopPropagation ? oEvent.stopPropagation() : oEvent.cancelBubble = true
        }
    });
});
/*重写 contextmenu()函数*/
fgm.on(document, "contextmenu", function(event)
{
    var oEvent = event || window.event;
    oEvent.preventDefault ? oEvent.preventDefault() : oEvent.returnValue = false
});
</script>

```

网页效果如图 5.45 所示。



图 5.45 鼠标燃放烟花

5.46 带时钟的鼠标

本实例使用 JavaScript 制作一个实例，该实例网页里的鼠标旁边会带有时钟。本节主要涉及的 JavaScript 语法如下。

window.pageYOffset：Netscape 属性，指滚动条顶部到网页顶部的距离。

本实例主要代码如下：

```
<script language="JavaScript1.2">
/*初始化变量*/
dCol = '000000';
fCol = '000000';
sCol = '000000';
mCol = '000000';
hCol = '000000';
ClockHeight = 40;
ClockWidth = 40;
ClockFromMouseY = 0;
ClockFromMouseX = 100;
d = new Array("SUNDAY", "MONDAY", "TUESDAY", "WEDNESDAY", "THURSDAY",
"FRIDAY", "SATURDAY");
m = new Array("JANUARY", "FEBRUARY", "MARCH", "APRIL", "MAY", "JUNE",
"JULY", "AUGUST", "SEPTEMBER", "OCTOBER", "NOVEMBER", "DECEMBER");
date = new Date();
day = date.getDate();
year = date.getYear();
if (year < 2000) year = year + 1900;
TodaysDate = " " + d[date.getDay()] + " " + day + " " + m[date.getMonth()]
+ " " + year;
D = TodaysDate.split(' ');
H = '...';
H = H.split(' ');
M = '....';
M = M.split(' ');
S = '.....';
S = S.split(' ');
Face = '1 2 3 4 5 6 7 8 9 10 11 12';
font = 'Arial';
size = 1;
speed = 0.6;
```

```

ns = (document.layers);
ie = (document.all);
Face = Face.split(' ');
n = Face.length;
a = size * 10;
ymouse = 0;
xmouse = 0;
scrll = 0;
props = "<font face=" + font + " size=" + size + " color=" + fCol + "><B>";
props2 = "<font face=" + font + " size=" + size + " color=" + dCol + "><B>";
Split = 360 / n;
Dspllt = 360 / D.length;
HandHeight = ClockHeight / 4.5 HandWidth = ClockWidth / 4.5 HandY = -7;
HandX = -2.5;
scrll = 0;
step = 0.06;
currStep = 0;
/*定义数组*/
y = new Array();
x = new Array();
Y = new Array();
X = new Array();
for (i = 0; i < n; i++)
{
    y[i] = 0;
    x[i] = 0;
    Y[i] = 0;
    X[i] = 0
}
/*定义数组*/
Dy = new Array();
Dx = new Array();
DY = new Array();
DX = new Array();
/*初始化数组*/
for (i = 0; i < D.length; i++)
{
    Dy[i] = 0;
    Dx[i] = 0;
    DY[i] = 0;
    DX[i] = 0
}
/*网景浏览器*/
if (ns)
{
    for (i = 0; i < D.length; i++) document.write('<layer name="nsDate'
+ i + '" top=0 left=0 height=' + a + ' width=' + a + '><center>' +
props2 + D[i] + '</font></center></layer>');
    for (i = 0; i < n; i++) document.write('<layer name="nsFace' + i +
'" top=0 left=0 height=' + a + ' width=' + a + '><center>' + props
+ Face[i] + '</font></center></layer>');
    for (i = 0; i < S.length; i++) document.write('<layer name=nsSeconds'
+ i + ' top=0 left=0 width=15 height=15><font face=Arial size=3 color='
+ sCol + '><center><b>' + S[i] + '</b></center></font></layer>');
    for (i = 0; i < M.length; i++) document.write('<layer name=nsMinutes'
+ i + ' top=0 left=0 width=15 height=15><font face=Arial size=3 color='
+ mCol + '><center><b>' + M[i] + '</b></center></font></layer>');
    for (i = 0; i < H.length; i++) document.write('<layer name=nsHours'
+ i + ' top=0 left=0 width=15 height=15><font face=Arial size=3 color='
+ hCol + '><center><b>' + H[i] + '</b></center></font></layer>');
}

```

```

}
/*IE 浏览器*/
if (ie)
{
    document.write('<div id="Od" style="position:absolute;top:0px;
left:0px"><div style="position:relative">');
    for (i = 0; i < D.length; i++) document.write('<div id="ieDate"
style="position:absolute;top:0px;left:0;height:' + a + ';width:' +
a + ';text-align:center">' + props2 + D[i] + '</B></font></div>');
    document.write('</div></div>');
    document.write('<div id="Of" style="position:absolute;top:0px;
left:0px"><div style="position:relative">');
    for (i = 0; i < n; i++) document.write('<div id="ieFace" style=
"position:absolute;top:0px;left:0;height:' + a + ';width:' + a + ';
text-align:center">' + props + Face[i] + '</B></font></div>');
    document.write('</div></div>');
    document.write('<div id="Oh" style="position:absolute;top:0px;
left:0px"><div style="position:relative">');
    for (i = 0; i < H.length; i++) document.write('<div id="ieHours"
style="position:absolute;width:16px;height:16px;font-family:Arial;
font-size:16px;color:' + hCol + ';text-align:center;font-weight:
bold">' + H[i] + '</div>');
    document.write('</div></div>');
    document.write('<div id="Om" style="position:absolute;top:0px;
left:0px"><div style="position:relative">');
    for (i = 0; i < M.length; i++) document.write('<div id="ieMinutes"
style="position:absolute;width:16px;height:16px;font-family:Arial;
font-size:16px;color:' + mCol + ';text-align:center;font-weight:
bold">' + M[i] + '</div>');
    document.write('</div></div>'); document.write('<div id="Os"
style="position:absolute;top:0px;left:0px"><div style="position:
relative">');
    for (i = 0; i < S.length; i++) document.write('<div id="ieSeconds"
style="position:absolute;width:16px;height:16px;font-family:Arial;
font-size:16px;color:' + sCol + ';text-align:center;font-weight:
bold">' + S[i] + '</div>');
    document.write('</div></div>');
} (ns) ? window.captureEvents(Event.MOUSEMOVE) : 0;
function Mouse(evt) {
    ymouse = (ns) ? evt.pageY + ClockFromMouseY - (window.pageYOffset) :
    event.y + ClockFromMouseY;
    xmouse = (ns) ? evt.pageX + ClockFromMouseX : event.x + ClockFromMouseX;
} (ns) ? window.onMouseMove = Mouse : document.onmousemove = Mouse;
/*计时分发*/
function ClockAndAssign()
{
    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;
    if (ie) {
        Od.style.top = window.document.body.scrollTop;
        Of.style.top = window.document.body.scrollTop;
        Oh.style.top = window.document.body.scrollTop;
        Om.style.top = window.document.body.scrollTop;
        Os.style.top = window.document.body.scrollTop;
    }
}

```

```

    for (i = 0; i < n; i++) {
        var F = (ns) ? document.layers['nsFace' + i] : ieFace[i].style;
        F.top = y[i] + ClockHeight * Math.sin( - 1.0471 + i * Split * Math.PI
        / 180) + scrll;
        F.left = x[i] + ClockWidth * Math.cos( - 1.0471 + i * Split * Math.PI
        / 180);
    }
    for (i = 0; i < H.length; i++) {
        var HL = (ns) ? document.layers['nsHours' + i] : ieHours[i].style;
        HL.top = y[i] + HandY + (i * HandHeight) * Math.sin(hrs) + scrll;
        HL.left = x[i] + HandX + (i * HandWidth) * Math.cos(hrs);
    }
    for (i = 0; i < M.length; i++) {
        var ML = (ns) ? document.layers['nsMinutes' + i] : ieMinutes[i].
        style;
        ML.top = y[i] + HandY + (i * HandHeight) * Math.sin(min) + scrll;
        ML.left = x[i] + HandX + (i * HandWidth) * Math.cos(min);
    }
    for (i = 0; i < S.length; i++) {
        var SL = (ns) ? document.layers['nsSeconds' + i] : ieSeconds
        [i].style;
        SL.top = y[i] + HandY + (i * HandHeight) * Math.sin(sec) + scrll;
        SL.left = x[i] + HandX + (i * HandWidth) * Math.cos(sec);
    }
    for (i = 0; i < D.length; i++) {
        var DL = (ns) ? document.layers['nsDate' + i] : ieDate[i].style;
        DL.top = Dy[i] + ClockHeight * 1.5 * Math.sin(currStep + i * Dsplit
        * Math.PI / 180) + scrll;
        DL.left = Dx[i] + ClockWidth * 1.5 * Math.cos(currStep + i * Dsplit
        * Math.PI / 180);
    }
    currStep -= step;
}
/*延迟*/
function Delay()
{
    scrll = (ns) ? window.pageYOffset : 0;
    Dy[0] = Math.round(DY[0] += ((ymouse) - DY[0]) * speed);
    Dx[0] = Math.round(DX[0] += ((xmouse) - DX[0]) * speed);
    for (i = 1; i < D.length; i++) {
        Dy[i] = Math.round(DY[i] += (Dy[i - 1] - DY[i]) * speed);
        Dx[i] = Math.round(DX[i] += (Dx[i - 1] - DX[i]) * speed);
    }
    y[0] = Math.round(Y[0] += ((ymouse) - Y[0]) * speed);
    x[0] = Math.round(X[0] += ((xmouse) - X[0]) * speed);
    for (i = 1; i < n; i++) {
        y[i] = Math.round(Y[i] += (y[i - 1] - Y[i]) * speed);
        x[i] = Math.round(X[i] += (x[i - 1] - X[i]) * speed);
    }
    ClockAndAssign();
    setTimeout('Delay()', 20);
}
/*页面加载*/
if (ns || ie)
    window.onload = Delay;
</script>

```

网页效果如图 5.46 所示。

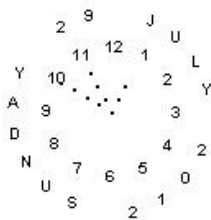


图 5.46 带时钟的鼠标

5.47 吐泡泡的鼠标

本实例使用 JavaScript 制作一个实例，展示网页上吐泡泡的鼠标。本节主要涉及的 JavaScript 语法如下。

Math.random()函数：该函数返回值是一个大于等于 0 且小于 1 的随机数，如 0.0105901374530933 或 0.872525005541986。

本实例主要代码如下：

```
<script language="JavaScript1.2">
  Image0 = new Image();
  Image0.src = "wbb.gif";
  Amount = 20;
  Ymouse = -50;
  Xmouse = -50;
  Ypos = new Array();
  Xpos = new Array();
  Speed = new Array();
  rate = new Array();
  grow = new Array();
  Step = new Array();
  Cstep = new Array();
  Size = new Array();
  /*鼠标事件*/
  function Mouse(evnt)
  {
    Ymouse = event.y - 20;
    Xmouse = event.x;
  }
  /*鼠标事件*/
  document.onmousemove = Mouse;
  for (i = 0; i < Amount; i++)
  {
    Ypos[i] = Ymouse;
    Xpos[i] = Xmouse;
    Speed[i] = Math.random() * 4 + 1;
    Cstep[i] = 0;
    Step[i] = Math.random() * 0.1 + 0.05;
    grow[i] = 8;
    rate[i] = Math.random() * 0.5 + 0.1;
  }
}
```

```

{
    for (i = 0; i < Amount; i++)
    {
        document.write('');
    }
}
/*主函数*/
function MouseBubbles()
{
    /*定义冒泡范围*/
    var hscrll = document.body.scrollTop;
    var wscrll = document.body.scrollLeft;
    for (i = 0; i < Amount; i++)
    {
        sy = Speed[i] * Math.sin(270 * Math.PI / 180);
        sx = Speed[i] * Math.cos(Cstep[i] * 4);
        Ypos[i] += sy;
        Xpos[i] += sx;
        if (Ypos[i] < -40)
        {
            Ypos[i] = Ymouse;
            Xpos[i] = Xmouse;
            Speed[i] = Math.random() * 6 + 4;
            grow[i] = 8;
            Size[i] = Math.random() * 15 + 5;
        }
        {
            si[i].style.pixelLeft = Xpos[i] + wscrll;
            si[i].style.pixelTop = Ypos[i] + hscrll;
            si[i].style.width = grow[i];
            si[i].style.height = grow[i];
        }
        grow[i] += rate[i];
        Cstep[i] += Step[i];
        if (grow[i] > 24) grow[i] = 25;
    }
    setTimeout('MouseBubbles()', 10);
}
MouseBubbles();          /*执行冒泡函数*/
</script>

```

网页效果如图 5.47 所示。



图 5.47 吐泡泡的鼠标

5.48 会变化的鼠标指针

本实例使用 JavaScript 制作一个实例，该实例中的鼠标指针是会变化的。本节主要涉及的 JavaScript 语法是 `document.body.style.cursor`，用于给网页定义鼠标样式。

本实例主要代码如下：

```
<SCRIPT LANGUAGE="JavaScript">
  var x, y, xold, yold, xdiff, ydiff;
  var dir = Array();
  dir[0] = "n-resize";dir[1] = "ne-resize";dir[2] = "e-resize";dir[3] =
  "se-resize";dir[4] = "s-resize";dir[5] = "sw-resize";dir[6] = "w-resize";
  dir[7] = "nw-resize";document.onmousemove = FindXY;
  function display(direction)                                /*显示函数*/
  {
    document.body.style.cursor = dir[direction];           /*定义鼠标样式*/
  }
  /*查询横纵坐标*/
  function FindXY(loc)
  {
    x = (document.layers) ? loc.pageX: event.clientX;
    y = (document.layers) ? loc.pageY: event.clientY;
    xdiff = x - xold;
    ydiff = y - yold
    if ((xdiff < 2) && (ydiff < -2)) display(0);
    if ((xdiff < 2) && (ydiff > 2)) display(4);
    if ((xdiff > 2) && (ydiff < 2)) display(2);
    if ((xdiff < -2) && (ydiff < 2)) display(6);
    if ((xdiff > 2) && (ydiff > 2)) display(3);
    if ((xdiff > 2) && (ydiff < -2)) display(1);
    if ((xdiff < -2) && (ydiff > 2)) display(5);
    if ((xdiff < -2) && (ydiff < -2)) display(7);
    xold = x;
    yold = y;
  }
</script>
```

5.49 跟踪鼠标的雪花

本实例使用 JavaScript 制作一个实例，在鼠标周围有雪花跟随。本节主要涉及的 JavaScript 语法是 `document.layers` 数组，已在 5.44 节介绍过，这里不再复述。

本实例主要代码如下：

```
<script language="javascript">
  /*定义和初始化变量*/
  var images = '551.jpg';
  var amount = 7;
  var cnter = 70;
```

```

var step;
var currStep = 0;
var Xpos = 0;
var Ypos = 0;
/*document.all 是 IE 浏览器所具有的对象集合，一般用 if (document.all)来判断是
否是 IE 浏览器，这个集合代表 document 对象下所有元素*/
if (document.all)
{
    document.write('<div id="ieDiv" style="position:absolute;top:0px;
left:0px">')
    document.write('<div id="c" style="position:relative">');
    for (n = 0; n < amount; n++)
        document.write('')
    document.write('</div>') document.write('</div>') function Msie
MouseFollow()
    {
        Xpos = document.body.scrollLeft + event.x - 5;
        Ypos = document.body.scrollTop + event.y - 5;
    }
    document.onmousemove = MsieMouseFollow;
}
/* document.layers 是网景 Netscape 浏览器所具有的对象集合 这个集合代表 document
对象下所有的 layer*/
else if (document.layers)
{
    window.captureEvents(Event.MOUSEMOVE);
    for (ns = 0; ns < amount; ns++)
        document.write("<layer name='n" + ns + "' left='0' top='0'><img
src='" + images + "'></layer>");
    function NsMouseFollow(evt)
    {
        Xpos = evt.pageX - 5;
        Ypos = evt.pageY - 5;
    }
    window.onMouseMove = NsMouseFollow;
}
/*旋转函数*/
function Swirl()
{
    if (currStep < 0.0550) step = 0.001;
    if (document.all)
    {
        for (i = 0; i < ieDiv.all.c.all.length; i++) {
            ieDiv.all.c.all[i].style.top = Ypos + cnter * Math.cos
            ((currStep + i * 4.5) / 5) * Math.sin((currStep) * 150);
            ieDiv.all.c.all[i].style.left = Xpos + cnter * Math.sin
            ((currStep + i * 4.5) / 5) * Math.sin((currStep) * 150);
        }
    }
    else if (document.layers)
    {
        for (i = 0; i < ns; i++) {
            var temp = "n" + i document.layers[temp].top = Ypos + cnter
            * Math.cos((currStep + i * 4.5) / 5) * Math.sin((currStep) *
            150);
            document.layers[temp].left = Xpos + cnter * Math.sin((currStep
            + i * 4.5) / 5) * Math.sin((currStep) * 150);
        }
    }
}

```

```

    }
  }
  currStep += step;
  setTimeout("Swirl()", 10);
  if (currStep > 0.0540)
  {
    step += 0.002;
    if (document.layers)
    {
      for (i = 0; i < ns; i++)
      {
        var temp = "n" + i;
        document.layers[temp].top = Ypos + cnter *
          Math.cos((currStep + i * 4.5) / 5);
        document.layers[temp].left = Xpos + cnter *
          Math.sin((currStep + i * 4.5) / 5);
      }
    }
    else if (document.all)
    {
      for (i = 0; i < ieDiv.all.c.all.length; i++)
      {
        ieDiv.all.c.all[i].style.top = Ypos + cnter *
          Math.cos((currStep + i * 4.5) / 5);
        ieDiv.all.c.all[i].style.left = Xpos + cnter *
          Math.sin((currStep + i * 4.5) / 5);
      }
    }
  }
  if (step > 0.5)
  {
    step = 0.5;
    cnter -= 8;
  }
  if (document.layers) _y = -window.innerWidth;
  else if (document.all) _y = -document.body.clientWidth;
  if (cnter <= _y)
  {
    currStep = 0;
    step = 0.001;
    cnter = 70;
  }
}
/*调用旋转函数*/
Swirl();
</script>

```

网页效果如图 5.48 所示。

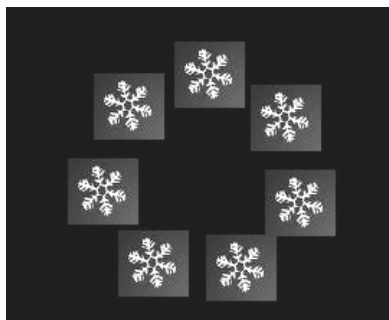


图 5.48 跟踪鼠标的雪花