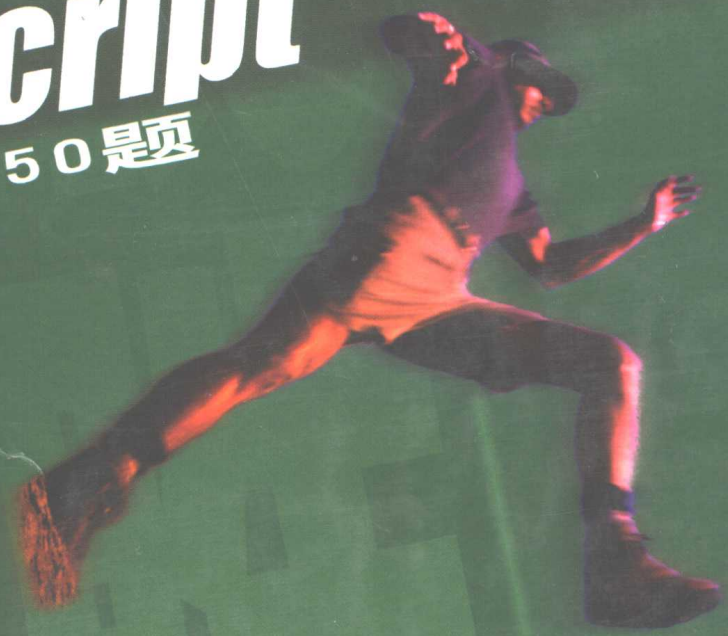
战胜 Javascript

必做练习50题



邓瑞峰 编著

北京大学出版社
http://cbs.pku.edu.cn

战胜 Javascript 必做练习 50 题

邓瑞峰　编著

北京大学出版社

北　京

内 容 简 介

本书以 50 个精选的 Javascript 程序实例作为练习，每个例子均有详细的操作步骤，并附有针对性很强的思考题，有助于读者快速巩固和提高自己的 Javascript 编程水平。

在内容编排上，本书将 50 个练习按照浏览器相关、页面特效、时间日期、图形图像 Cookie，菜单特效、鼠标特效、综合技巧和游戏编程分类，同时又兼顾了从易到难、由浅入深的一般学习顺序。

本书主要面向具有一定 Javascript 和 HTML 语言基础，能够读懂简单的 Javascript 脚本程序，希望进一步提高 Javascript 编程水平的初、中级读者。同时，本书的 50 个练习中，也不乏功能强大，设计精妙的程序实例，对高水平的读者亦有参考价值。希望各个层次的读者都能在本书中有所收获。

本书含 1 张教学光盘。

图 书 名：战胜 Javascript 必做练习 50 题
图书写作者：邓瑞峰
图书责编：段志刚　黄庆生
光盘责编：王　原
本 版 号：ISBN 7-900636-85-4/TP 62
出 版 者：北京大学出版社
地　　址：北京市海淀区中关村北京大学校内　100871
电　　话：出版部 62752015　编辑部 62765013　发行部 62754140
网　　址：http://cbs.pku.edu.cn　　E－mail:xxjs@pup.pku.edu.cn
印 刷 者：河北省滦县印刷厂
发 行 者：北京大学出版社
经 销 者：新华书店
787 毫米×1092 毫米　16 开本　15.25 印张　390 千字
2001 年 11 月第 1 版　2001 年 11 月第 1 次印刷
定　　价：29 元（含光盘 1 张）

前　言

从 Internet 获取信息，还希望自己能够亲手设计网页，向 Internet 发布信息——这就涉及到 Web 页设计与制作，在这种潮流下，网页设计技术已经成为一种时尚，各种用于网页设计的工具更是各领风骚。“网龄”长一点的网友一定有这种感觉：与几年前相比，现在的网页在动态和交互性方面强大得多了。促成这种进步的主要原因，就是脚本编程语言的广泛使用。而其中使用得最为广泛的一种脚本编程语言就是 Javascript。打开任意一个具有动态效果的网页的源文件，几乎都可以从中找到 Javascript 的踪影。这也可以从一个侧面证明 Javascript 的流行和受欢迎的程度。

Javascript 是一种嵌入式的脚本语言，它可以嵌入到 HTML 文件中，使网页具有更强的交互性和动态特征；它也可以嵌入到 Asp 文件和 Web 文件中，作为服务器端的脚本语言。

本书的所有示例都在 Internet Explorer 5.5 和 Netscape Navigator 4.61 两种浏览器中测试通过（个别例子仅适用于一种浏览器，在练习中也有明确的说明），读者可以把源代码直接应用于自己的 Web 页中。

如果您在此之前从来没有接触过 Javascript，也没有 C 和 C++的基础知识，那么笔者建议您先找一两本 Javascript 的入门书籍看一看，熟悉了 Javascript 的基础知识（包括语法，程序结构，以及如何使用数据和变量等）之后再来学习本书提供的例子，这样在学习中才不会有太大的困难。

当您一步一步地完成本书的 50 个练习，开始尝试自己独立设计程序之后，您也许会惊奇地发现：原来用 Javascript 编程是一件轻而易举的事情！那么恭喜您，您的 Javascript 编程水平已经上了一个层次，成为一位 Javascript 高手了，这正是本书的目标。

本书主要面向具有一定 Javascript 和 HTML 语言基础，能够读懂简单的 Javascript 脚本程序，希望进一步提高 Javascript 编程水平的初、中级读者。同时，本书的 50 个练习中，也不乏功能强大，设计精妙的程序实例，对高水平的读者亦有参考价值。希望各个层次的读者都能在本书中有所收获。

由于作者水平有限，书中难免会有不足之处，请读者批评指正，也欢迎读者就本书中的一些疑难问题与作者探讨。

最后感谢您成为我们的读者，您的支持是我们前进的最大动力。

本书由邓瑞峰主编，另外，余晓鹏、张伟华、何广、张石勇、战祥森、张松伟、吴绍伟、孙科峰、渠继永、覃文圣、牟南、贾鹏、李衡、钟光辉、钱辰、王晓龙、满晓宇、罗捷、肖健、解灵运等也参加了本书编写工作。

编　者

2001 年 11 月

目　录

练习 1　Title 栏 特 效

练习目的

本练习的目的，是使最小化后的浏览器窗口在任务栏上呈现闪烁效果。请观察下面的两幅图中最小化的 IE 窗口，这是同一个网页相差 1 秒钟抓下来的不同效果。可以看到，左图中任务栏上的星星是暗的，而右图则变亮了。看上去就像星星在一闪一闪，极富动态效果，能够给人留下深刻的印象，尤其是当访问者打开了多个浏览器窗口时，这个窗口就会显得特别引人注目。

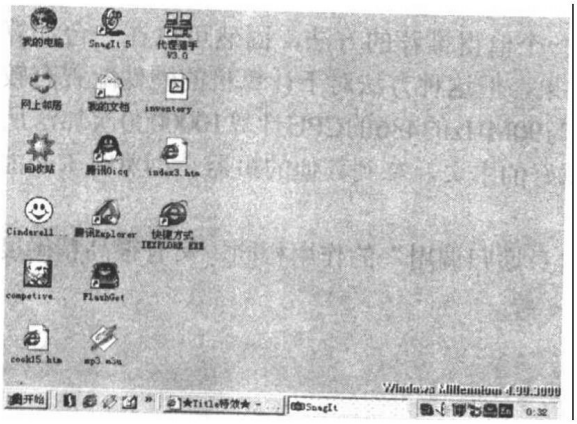


图 1-1　暗时的效果

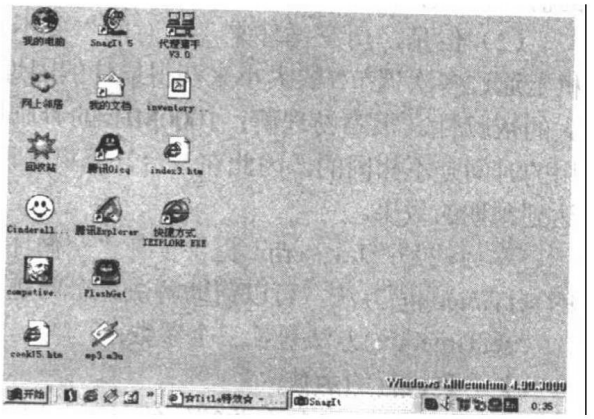


图 1-2　亮时的效果

练习原理

这个例子的实现过程很简单，星星的闪烁效果实际上是由实心的五角星和空心的五角星轮流显示而产生的。即每隔一段时间改变一次document对象的title属性，使“★Title特效★”和“☆Title特效☆”轮流显示。只要设定好合适的时间间隔，就能使肉眼看上去产生星星闪烁的效果。

练习过程

（1）本程序的关键是创建一个star_wink()函数，在该函数中设置一个标志变量flag，取值为0或1，函数每次运行的时候都先检查flag的值，若flag=0，就将document.title设置为“★Title特效★”，并将flag值改为1，表示函数下一次执行时将显示“☆Title特效☆”；若flag=1，则执行与上面相反的动作。

这时，遇到了一个问题：怎样才能使轮流显示两个title的时间间隔合适呢？

熟悉C语言的读者可能马上会想到这样的方法，在star_wink()函数的最后加入如下代码：

```
for(i=0,i<N,i++)
  { } //请注意，花括号里为空
star_wink();
```

其中 N 是一个预定义的常量，它的值通常取得很大，比方取 10000000，for 循环什么事都不做，是一个空循环，它唯一的作用就是“拖时间”，因为 i 要自加 1000000 次才能跳出这个循环，这个过程需要一定时间，然后递归地调用 star_wink()函数自己，改变 document.title 的值，就产生了闪烁效果。

（2）但是，用空循环来“拖时间”并不是一个值得推荐的方法，固然可以在一台计算机上通过多次调节N的大小来获得最佳的闪烁效果，但这种方法对于计算机的硬件配置有较大的依赖性。很容易理解：1000MHz的奔腾III与90MHz的486的CPU计算1000000次加法所用的时间是不相同的，因此在一台计算机上调试好的主页，拿到其他的机器上浏览时未必能达到预期的效果。

幸运的是：Javascript 提供了与“空循环+自身递归调用”的作用相同，又有很高精确度的 setTimeout()方法，可以帮助解决这个问题。

setTimeout()方法包括三个参数：

- 要执行的函数；
- 执行这个函数之前等待的时间，以毫秒为单位；
- 脚本语言的类型（这个参数通常可以省略）。

有了这个工具，上面的问题就迎刃而解了，在 star_wind()函数的末尾加上这样一行：

```
setTimeout("star_wink()",300);
```

就能保证 star_wink()函数每隔 300 毫秒执行一次。

（3）编写star_wink()函数，包含star_wink()函数的源程序如下：

```
<SCRIPT language=JavaScript>
<!—
var flag=0;
function star_wink()
{
switch(flag)
  {
        case 0: document.title='★Title 特效★';
                flag=1;
```

```
break;
        case 1: document.title='☆Title 特效☆';
                flag=0;
break;
        default: false;
       }
     setTimeout("star_wink()",300);
   }
star_wink();
//-->
</SCRIPT>
```

读者在使用时只要把它加入 HTML 文件的<HEAD>和</HEAD>之间，并将“Title 特效”改成网页的实际标题即可。

【练习小结】

本练习介绍了 document 对象的 title 属性和 setTimeout()方法的使用。document 对象是 Javascript 最重要的核心对象之一，包含有 title、form、location、bkColor 等与网页浏览效果密切相关的属性，在后面的很多练习中都要用到这些属性。setTimeout()方法是 Javacript 的内置方法，用来作为一个“定时器”，当某段设定的时间“跑” 完了之后，便执行某个函数。在这个例子中，于 300 毫秒后 （也就是 0.3 秒后）就会执行 star_wink()函数，而 star_wink()函数中又包括了 setTimeout()，再一次执行 star_wink()函数（这实际上也是一种递归调用），从而不断的改变 document.title 的值。Javacript 还有一个功能相似的内置方法 setInterval()，也带有相同的参数，它与 setTimeout()的区别在于：setInterval()是每隔一段时间执行一次某个函数 functionx()，这一过程是连续的，即使上一次的 functionx()尚未执行完毕，本次的 functionx()也要开始，因此使用时不需要递归调用。请读者自己尝试使用 setInterval()代替 setTimeout()来完成本程序。

【思考题】

1. 改用 setInterval()方法完成本程序。
2. 调整 setTimeout()方法的时间间隔，看看闪烁速度有什么变化。
3. 仿照本练习，设计一个最小化后能在任务栏上显示时间的脚本程序。

练习2 模拟街霸

练习目的

相信读者对“街霸”格斗游戏一定不陌生，但你有没有想过在你的主页上能模拟“街霸”的动画效果呢？Javascript能够帮助你做到这一点。请看图2-1和图2-2：

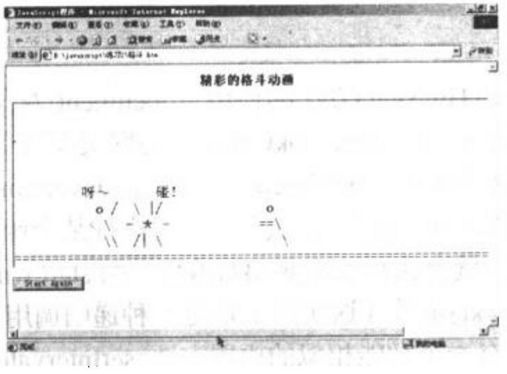


图2-1 动画I

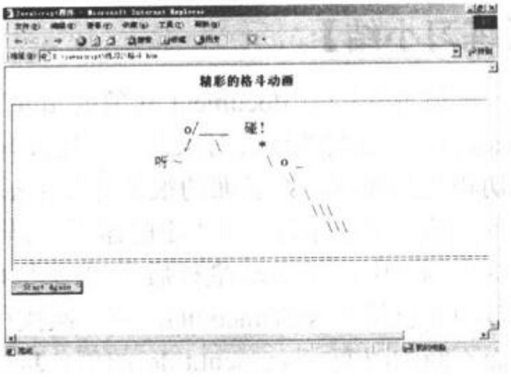


图2-2 动画II

与GIF动画和FLASH动画不同，本练习中的动画每一帧都是用ASCII字符模拟出来的图形，因此不会显著地增加文件大小。这也是ASCII动画的最大优势。

练习原理

本程序实现的原理是在一个文本框中按帧显示预先设定好的画面，即每隔一段时间更新一次文本框中的内容，这里又要用到setTimeout()方法。

练习过程

（1）用Windows自带的“记事本”程序中用ASCII字符绘制好每一帧的图画。

（2）建一个html文件，在页面上建一个表单，命名为animation，在该表单上放置一个文本框命名为animation，以及一个值为“Start Again”按钮。

（3）建立一个tlist对象，定义如下：

```
function tlist()
{
 max=tlist.arguments.length;
 for (i=0; i<max; i++)
 this[i]=tlist.arguments[i];
}
```

max是预先定义的全局变量，表示动画的帧数。

Javascript中定义对象是通过构造函数的方式来完成的。使用在定义tlist()函数的时候我们并没有指定函数的参数，这也是Javascript语言的一个特点，即使用函数时可以不完全按照函数的定义来使用参数。每次调用函数时，Javascript会自动生成arguments数组，可以用它来访问被调用函数的参数（亦即函数所定义的对象的属性）。在本练习中，没有在tlist()对象的定义中指明参数，而是在后面的代码中动态地生成一个tlist实例，并指明具体的参数（属性）。

（4）为了使用对象，需要创建一个对象实例tl:

```
tl=new tlist(property1，property2,……propertyn)
```

new语句的作用是实例化一个对象。property1到propertyn是给对象实例添加的n个属性，在本练习中，这n个属性就代表动画的n个帧，每个帧在属性中用一个字符串表示，比如tl的第一个属性表示为：

```
""+a+
""+a+
""+a+
""+a+
""+a+
"                    o                    o "+a+
"                   <|-    Round 1        -|> "+a+
"                    >>                  <<  "+a+
"=============================================================="
="+a,
```

它是一个包含了若干个回车换行符的字符串，显示出来就是动画的第一帧。因为动画中画面最高的帧有9行，而文本框是从上到下显示字符串的。因此该帧的前5行是空行，这是为了保证整个动画在同一条水平线上演示而加上去的。变量a是预先定义的回车换行符，如果不在每行末尾加上回车换行符的话，文本框就会把所有这些字符当成一行，无法显示出指定的图画效果。

tl其余的属性请看后面的源代码。

（5）定义一个fight()函数，它的作用是每隔一段时间依次在文本框中显示tl的各个属性，直到整个动画播放完毕。

```
function fight()
{
 document.animation.cheerleader.value = " " + a +
```

```
        tl[i];
        i++;
        if (i != max)
            setTimeout("fight()", 400);
        else
            i = 0;
    }
```

变量i是预先定义的计数器，初值为0。每次用“document.animation.cheerleader.value = "" + a + tl[i]”语句将一个空行和tl的第i个属性设置为表单animation中文本框cheerleader的值，然后将i加1。若i等于动画的最大帧数，将i值置0，动画播放结束；否则使用setTimeout()方法将fight()函数在400毫秒后再执行一次。

余下部分代码〖**参见光盘**〗（代码段较长，请读者注意看注释）。

【练习小结】

本程序的原理并不复杂，但体现了面向对象编程的思想，介绍了 Javascript 中构造和使用对象和对象实例的方法，并介绍了如何向对象实例中动态地添加属性和用 arguments 数组访问对象的属性。另外，读者应注意 Javascript 的控制字符的使用。

【思考题】

1. 向对象中添加属性有哪两种方法？
2. Javascript 中反斜杆“\”有什么特殊作用？
3. 编写一个脚本程序，让一个小人在文本框内跳舞。

练习 3　检测浏览器信息并跳转到相应页面

练习目的

有过网页制作经验的读者或许都遇到过这样的情况，设计好的网页在 Internet Exploer 里浏览时效果非常好，但在 Netscape 里浏览时却远不是那么回事，有时候甚至可以用“惨不忍睹”来形容。下面两幅图就是同一个页面在两种浏览器里打开的效果（如图 3-1 和图 3-2 所示），是不是相差很大？

图 3-1　某页面在 IE 中的效果（局部）

图 3-2　同一页面在 Netscape 中的效果

之所以出现这种情况，是因为两种浏览器之间的不兼容性造成的，IE 和 Netscape 虽然都支持同一个 HTML 4.0 标准，但它们又各自对这个标准作了扩展，支持更多的新功能。由于这些扩展功能尚未形成统一的标准，如果网页上使用了一种浏览器的扩展功能，那么它在另一种浏览器中就有可能无法正常浏览。

解决这一问题较好的方法是为同一个网页内容设计适合于 IE 和适合于 Netscape 的两个页面，然后编写 Javascript 脚本程序，检测用户浏览器的类型，自动跳转到针对该浏览器制作的页面。

通过这个练习，读者将认识 navigator 对象，以及 window 对象的 location 属性。并利用它们来实现浏览器类型判断和页面跳转。

练习原理

本练习主要使用了 navigator 对象的 appName 属性，window 对象的 location 属性，和 string

对象的 indexOf()方法。

navigator 对象是一个 Javascript 内置对象，包含了有关加载文档的浏览器的信息。navigator 对象包括如下属性：

- appCodeName：浏览器的代码名（例如，Mozilla）；
- appName：浏览器的名字；
- appVersion：浏览器的版本号；
- userAgent：由客户机送到服务器的用户与代理头标文本。

window 对象的 location 属性本身也是一个对象，它是指窗口所显示文档的完整（绝对）URL，用户可以改变 window.location，表示用另一个文档取代当前文档。

string 对象的 indexOf()方法返回括号内的字符串 str1 在调用此方法的字符串 str 中的起始位置，如果在 str 中找不到 str1 则返回-1。

练习过程

（1）假设某个 html 文件的名字是 sample.htm，它使用了一些只有 IE 才支持的功能，希望它在 Netscape 浏览器中也能正常显示。于是把针对 IE 制作的 sample.htm 改成 sampleie.htm，另外针对 Netscape 制作一个页面，去掉那些只有 IE 才支持的功能，或者用 Netscape 可以识别的扩展来重新设置这些功能，然后把它命名为 samplens.htm。同时改写 sample.htm 为空页面，将在这个页面上实现检测浏览器和跳转 url 的功能。

（2）检测浏览器的类型，由于并不知道 navigator.appName 等属性返回值的具体情况，所以先做一个测试，向 sample.htm 的 BODY 段中添加如下内容：

```
<script language = "JAVASCRIPT">
document.write("浏览器代码名称：" + navigator.appCodeName + "<BR>");
document.write("浏览器名称：" + navigator.appName + "<BR>");
document.write("浏览器版本号：" + navigator.appVersion + "<BR>");
document.write("用户代理：" + navigator.userAgent + "<BR>");
</script>
```

然后保存，分别在 IE 和 Netscape 浏览器中打开它，看看 navigator 对象的这些属性的返回值到底是什么（如图 3-3 和图 3-4 所示）：

图 3-3　IE 浏览器的信息

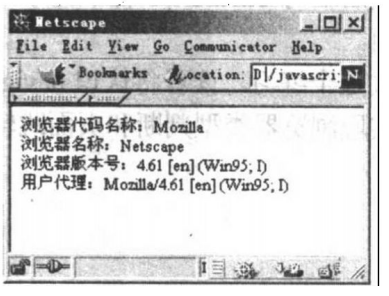


图 3-4　Netscape 浏览器的信息

可见两种浏览器的代码名称都是 Mozilla，只有浏览器名称（appName 属性）可以用来识别不同的浏览器。于是可以对 appName 属性的返回值（一个字符串）进行判断，如果字符串中含有“Microsoft”就表示这是 IE 浏览器，如果含有“Netscape”就表示这是 Netscape 浏览器。

（3）解决了判断浏览器类型的问题，接下来就可以编写跳转的代码了。

```
var ie = navigator.appName.indexOf("Microsoft");
var ns = navigator.appName.indexOf("Netscape");
var loc;
if (ie!=-1)
   {
   document.write("您使用的是 Internet Explore 浏览器，<BR>");
   document.write("5 秒钟之后将自动跳转到适合该浏览器的页面...");
   loc = "sampleie.htm";
   }
else if (ns!=-1)
   {
   document.write("您使用的是 Netscape Navigator 浏览器，<BR>");
   document.write("5 秒钟之后将自动跳转到适合该浏览器的页面...");
   loc = "samplens.htm";
   }
setTimeout("start()",5000);

function start()
{
window.location = loc;
}
```

用 ie 和 ns 两个变量分别确定 navigator 对象的 appName 属性是否包含“Microsoft”和“Netscape”字符串。根据它们的返回值就可以判断出浏览器的类型，并设定好将要跳转的页面的 url，然后用 setTimeout()方法等待 5 秒钟后执行 start()函数，实现跳转。

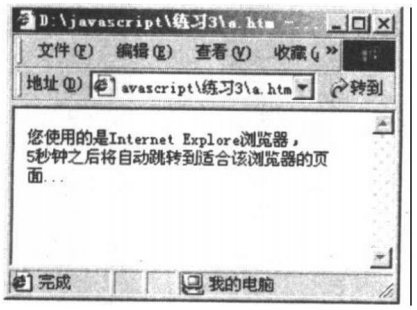


图 3-5　检测到 IE 浏览器

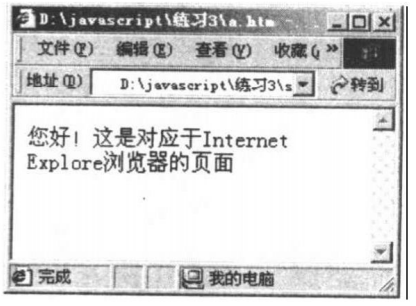


图 3-6　跳转到对应 IE 的页面

（4）保存 sample.htm，在两种浏览器中观看效果（如图 3-5~3-8 所示）：

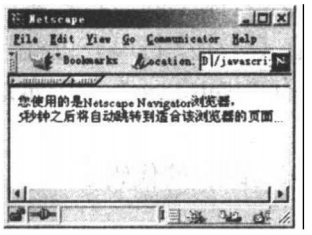


图 3-7　检测到 Netscape 浏览器

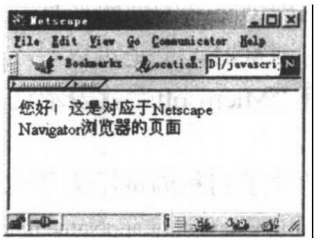


图 3-8　跳转到对应 Netscape 的页面

【练习小结】

在这个练习里，读者接触了 navigator 对象，它代表用户使用的浏览器。通常在编写跨浏览器平台的网页时，需要先判断浏览器的相关信息，然后决定如何编写代码，此时就可以使用 navigator 对象。

除了浏览器的类型之外，版本号对于页面能否正确显示也有关系，同属于 Netscape 公司的产品，低版本的浏览器就不支持许多高版本浏览器中的功能，比如说 2.0 以下的浏览器就不支持框架。所以有些使用了框架的网页其源代码中往往有“本网页使用了框架，但是您的浏览器不支持框架”的字样，以便给使用低版本浏览器的用户明确的提示。

全面考虑不同类型和不同版本的浏览器对网页的支持，利用 Javascript 实现判断和跳转，能够使你的网页变得友好。

【思考题】

1．document 对象也有一个属性名叫 location，请查阅有关的工具书，看看它与 window 对象的 location 属性有何区别。

2．根据 navigator 对象的 appVersion 属性，获取浏览器的版本号，然后判断它是否支持当前页面的某些功能。

练习4 表 单 检 验

练习目的

虽然表单处理通常都需要服务器端的支持，但还是有一些工作可以在客户端完成，比较典型的就是表单检验的工作。表单检验是指确定用户提交的表单数据是否合法。例如：是否填写了所有的必填字段，某些字段是否符合要求（例如电子邮件地址是否含有“@”字符）。如果表单验证也交给服务器端程序来做，势必增加额外的网络开销和服务器开销，因为每次表单提交都要与服务器进行交互。如果把这个工作放到客户端来做，由于相应的脚本已经下载到客户端浏览器，可以只让合法的表单提交到服务器，因此可以大大减少各种开销。

练习原理

如果用户填写的数据非法，表单将不提交，这一思想的实现依赖与 IE 浏览器处理事件的“事件浮升”机制：当事件发生时，它首先定向到发生事件的最底层对象，然后依次上浮，由各级对象定义的事件处理语句或事件处理函数依次处理；如果在同一对象中触发了两个以上的事件，那么它们的优先级由事件发生的先后来决定。本练习中将用到同属于 form 对象的 onsubmit()和 submit 事件，但 onsubmit()的发生早于 submit()，故先由 onsubmit()的处理函数 reg()来处理，从而实现表单检验。在本练习中，还将用到字符串对象（string）的多个通用操作方法，以及表单（Form）中的单行文本框对象（text），口令框对象（password），多行文本框对象（textarea）等多个对象。另外，弹出消息框是 Javascript 中经常用到的一种提示用户的方法，本练习也将介绍它的使用。

练习过程

（1）在页面上放置如图 4-1 所示的各个表单元素，可以使用 Dreamweaver 或 FrontPage 等主页编辑器，也可以直接输入 HTML 代码，请注意填写密码的文本框应使用口令框〖**参见光盘**〗。

表单的action属性指定了服务器处理表单提交数据的程序，一般来说是CGI、ASP或PHP程序，但本练习中练习的是客户端的Javascript编程，不能使用上述服务器程序，但可以指定 action 属性为 mailto:mimi@263.net，即将表单发送到指定的邮箱里，这是客户端 Javascript