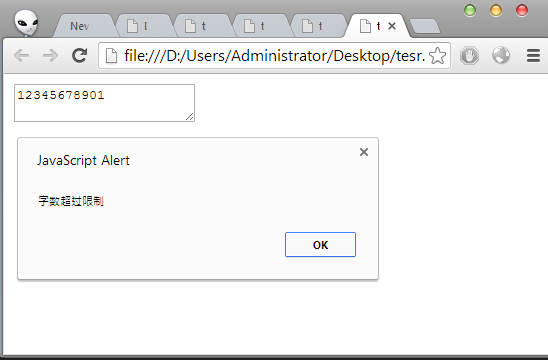
我的js笔记☹: (必须亲自实践, 必须配以插图)

**1.检查表单字数限制:**

|  |  |
| --- | --- |
| 1 | <!doctype html> |
| 2 | <html> |
| 3 | <head> |
| 4 | <metahttp-equiv="content-type"content="text/html; charset=utf-8" /> |
| 5 | </head> |
| 6 | <body> |
| 7 | <scriptlanguage="javascript"type="text/javascript"> |
| 8 | *function*chkmaxsms(*vobj1,vmax*){ |
| 9 | *var* str=vobj1.value; |
| 10 | *var* strlen=str.length; |
| 11 | if(strlen>vmax)   { |
| 12 | alert('字数超过限制'); |
| 13 | eval(vobj1.value=str.substr(0,vmax)); |
| 14 | } |
| 15 | } |
| 16 | </script> |
| 17 | <textareaname="textfield"onkeyup="javascript:chkmaxsms(this,'10')"></textarea> |
| 18 | </body> |
| 19 | </html> |



**2.js的一些表单验证**

**文本框只能输入数字代码(小数点也不能输入)**

<input onkeyup="this.value=this.value.replace(/\D/g,'')" onafterpaste="this.value=this.value.replace(/\D/g,'')">

这里你输入字母小数点自动将你替换为空,你输入不了,注意这里有一个onafterpaste**粘贴事件**,值得学习

**只能输入数字,能输小数点.**

<input onkeyup="if(isNaN(value))execCommand('undo')" onafterpaste="if(isNaN(value))execCommand('undo')">能输小数点

<input onchange="if(/\D/.test(this.value)){alert('只能输入数字');this.value='';}" name="txt1">只能输入数字

注意这里的execCommand方法:<http://www.cnblogs.com/baifan/articles/1034680.html>

[document.execCommand()的用法小记](http://www.cnblogs.com/baifan/articles/1034680.html)

首先要说明的是在firefox下支持不好。  
2D-Position 允许通过拖曳移动绝对定位的对象。   
AbsolutePosition 设定元素的 position 属性为“absolute”(绝对)。   
BackColor 设置或获取当前选中区的背景颜色。   
BlockDirLTR 目前尚未支持。   
BlockDirRTL 目前尚未支持。   
Bold 切换当前选中区的粗体显示与否。   
BrowseMode 目前尚未支持。   
Copy 将当前选中区复制到剪贴板。   
CreateBookmark 创建一个书签锚或获取当前选中区或插入点的书签锚的名称。   
CreateLink 在当前选中区上插入超级链接，或显示一个对话框允许用户指定要为当前选中区插入的超级链接的 URL。   
Cut 将当前选中区复制到剪贴板并删除之。   
Delete 删除当前选中区。   
DirLTR 目前尚未支持。   
DirRTL 目前尚未支持。   
EditMode 目前尚未支持。   
FontName 设置或获取当前选中区的字体。   
FontSize 设置或获取当前选中区的字体大小。   
ForeColor 设置或获取当前选中区的前景(文本)颜色。   
FormatBlock 设置当前块格式化标签。   
Indent 增加选中文本的缩进。   
InlineDirLTR 目前尚未支持。   
InlineDirRTL 目前尚未支持。   
InsertButton 用按钮控件覆盖当前选中区。   
InsertFieldset 用方框覆盖当前选中区。   
InsertHorizontalRule 用水平线覆盖当前选中区。   
InsertIFrame 用内嵌框架覆盖当前选中区。   
InsertImage 用图像覆盖当前选中区。   
InsertInputButton 用按钮控件覆盖当前选中区。   
InsertInputCheckbox 用复选框控件覆盖当前选中区。   
InsertInputFileUpload 用文件上载控件覆盖当前选中区。   
InsertInputHidden 插入隐藏控件覆盖当前选中区。   
InsertInputImage 用图像控件覆盖当前选中区。   
InsertInputPassword 用密码控件覆盖当前选中区。   
InsertInputRadio 用单选钮控件覆盖当前选中区。   
InsertInputReset 用重置控件覆盖当前选中区。   
InsertInputSubmit 用提交控件覆盖当前选中区。   
InsertInputText 用文本控件覆盖当前选中区。   
InsertMarquee 用空字幕覆盖当前选中区。   
InsertOrderedList 切换当前选中区是编号列表还是常规格式化块。   
InsertParagraph 用换行覆盖当前选中区。   
InsertSelectDropdown 用下拉框控件覆盖当前选中区。   
InsertSelectListbox 用列表框控件覆盖当前选中区。   
InsertTextArea 用多行文本输入控件覆盖当前选中区。   
InsertUnorderedList 切换当前选中区是项目符号列表还是常规格式化块。   
Italic 切换当前选中区斜体显示与否。   
JustifyCenter 将当前选中区在所在格式化块置中。   
JustifyFull 目前尚未支持。   
JustifyLeft 将当前选中区所在格式化块左对齐。   
JustifyNone 目前尚未支持。   
JustifyRight 将当前选中区所在格式化块右对齐。   
LiveResize 迫使 MSHTML 编辑器在缩放或移动过程中持续更新元素外观，而不是只在移动或缩放完成后更新。   
MultipleSelection 允许当用户按住 Shift 或 Ctrl 键时一次选中多于一个站点可选元素。   
Open 打开。   
Outdent 减少选中区所在格式化块的缩进。   
OverWrite 切换文本状态的插入和覆盖。   
Paste 用剪贴板内容覆盖当前选中区。   
PlayImage 目前尚未支持。   
Print 打开打印对话框以便用户可以打印当前页。   
Redo 重做。   
Refresh 刷新当前文档。   
RemoveFormat 从当前选中区中删除格式化标签。   
RemoveParaFormat 目前尚未支持。   
SaveAs 将当前 Web 页面保存为文件。   
SelectAll 选中整个文档。   
SizeToControl 目前尚未支持。   
SizeToControlHeight 目前尚未支持。   
SizeToControlWidth 目前尚未支持。   
Stop 停止。   
StopImage 目前尚未支持。   
StrikeThrough 目前尚未支持。   
Subscript 目前尚未支持。   
Superscript 目前尚未支持。   
UnBookmark 从当前选中区中删除全部书签。   
Underline 切换当前选中区的下划线显示与否。   
Undo 撤消。   
Unlink 从当前选中区中删除全部超级链接。   
Unselect 清除当前选中区的选中状态。  
  
例1：

**isNaN是测试是否为数值型，限制输入只能为数值如：1981.121，允许最多有一个小数点**

<input type="text" name="text" onkeyup="if(isNaN(value))execCommand('undo')" />

例2：

http://www.cnblogs.com/Images/OutliningIndicators/None.gif<input type=button value=剪切 onclick=document.execCommand('Cut')>  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif<input type=button value=拷贝 onclick=document.execCommand('Copy')>  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif<input type=button value=粘贴 onclick=document.execCommand('Paste')>  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif<input type=button value=撤消 onclick=document.execCommand('Undo')>  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif<input type=button value=重做 onclick=document.execCommand('Redo') id=button2 name=button2>  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif<input type=button value=删除 onclick=document.execCommand('Delete')>  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif<input type=button value=黑体 onclick=document.execCommand('Bold')>  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif<input type=button value=斜体 onclick=document.execCommand('Italic')>  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif<input type=button value=下划线 onclick=document.execCommand('Underline')>  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif<input type=button value=停止 onclick=document.execCommand('stop')>  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif<input type=button value=保存 onclick=document.execCommand('SaveAs')>  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif<input type=button value=另存为 onclick=document.execCommand('Saveas',false,'c:\\test.htm')>  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif<input type=button value=字体 onclick=document.execCommand('FontName',false,fn)>  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif<input type=button value=字体大小 onclick=document.execCommand('FontSize',false,fs)>  
http://www.cnblogs.com/Images/OutliningIndicators/None.gif<input type=button value=刷新 onclick=document.execCommand('refresh',false,0)>

**只能输入英文字母和数字,不能输入中文**

<input onkeyup="value=value.replace(/[^\w\.\/]/ig,'')">

**只能输入字母和汉字**

<input onkeyup="value=value.replace(/[\d]/g,'') "onbeforepaste="clipboardData.setData('text',clipboardData.getData('text').replace(/[\d]/g,''))" maxlength=10 name="Numbers">

**小数点后只能有最多两位(数字,中文都可输入),不能输入字母和运算符号**

< input onKeyPress="if((event.keyCode<48 || event.keyCode>57) && event.keyCode!=46 || /\.\d\d$/.test(value))event.returnValue=false">

**小数点后只能有最多两位(数字,字母,中文都可输入),可以输入运算符号**

<input onkeyup="this.value=this.value.replace(/^(\-)\*(\d+)\.(\d\d).\*$/,'$1$2.$3')">

**只能输入中文**

|  |  |
| --- | --- |
| <input name="realname" type="text"  maxlength="20" class="input\_style" value="" onkeyup="value=value.replace(/[^\u4E00-\u9FA5]/g,'')" onbeforepaste="clipboardData.setData('text',clipboardData.getData('text').replace(/[^\u4E00-\u9FA5]/g,''))" /> | |
| |  |  | | --- | --- | | 1 | <!doctype html> | | 2 | <html> | | 3 | <head> | | 4 | <metahttp-equiv="content-type"content="text/html; charset=utf-8" /> | | 5 | </head> | | 6 | <body> | | 7 | <scriptlanguage="javascript"type="text/javascript"> | | 8 | *function*chkmaxsms(*vobj1,vmax*){ | | 9 | *var* str=vobj1.value; | | 10 | *var* strlen=str.length; | | 11 | if(strlen>vmax)   { | | 12 | alert('字数超过限制'); | | 13 | eval(vobj1.value=str.substr(0,vmax)); | | 14 | } | | 15 | } | | 16 | </script> | | 17 | <inputonkeyup="this.value=this.value.replace(/\D/g,'')"onafterpaste="this.value=this.value.replace(/\D/g,'')"> | | 18 | <textareaname="textfield"onkeyup="javascript:chkmaxsms(this,'10')"></textarea> | | 19 | <br> | | 20 | <p></p> | | 21 | <inputonkeyup="if(isNaN(value))execCommand('undo')"onafterpaste="if(isNaN(value))execCommand('undo')"> | | 22 | <inputonchange="if(/\D/.test(this.value)){alert('只能输入数字');this.value='';}"name="txt1"> | | 23 | <p></p> | | 24 | <inputtype=buttonvalue=删除onclick=document.execCommand('Delete')><br> | | 25 | <inputtype=buttonvalue=黑体onclick=document.execCommand('selectAll')><br> | | 26 | <inputonkeyup="value=value.replace(/[^\w\.\/]/ig,'')"><br> | | 27 | <inputonKeyPress="if((event.keyCode<48 || event.keyCode>57) && event.keyCode!=46 || /\.\d\d$/.test(value))event.returnValue=false"><br> | | 28 | <inputonkeyup="this.value=this.value.replace(/^(\-)\*(\d+)\.(\d\d).\*$/,'$1$2.$3')"><br> | | 29 | <inputname="realname"type="text"maxlength="20"class="input\_style"value=""onkeyup="value=value.replace(/[^\u4E00-\u9FA5]/g,'')"onbeforepaste="clipboardData.setData('text',clipboardData.getData('text').replace(/[^\u4E00-\u9FA5]/g,''))" /><br> | | 30 | </body> | | 31 | </html> | |  |

这里关于clipboardData对象:

**注意：ie7，与ie8 对网页有个复制的权限，需在“安全”中的“自定义级别”的脚本中设置**

**注意: clipboardData目前为IE专有,chrome浏览器会出现代码报错(会造成其它js无法运行):详见:**

<http://www.w3help.org/zh-cn/causes/BX2001>

可以使用:

判断浏览器类型，若不是 IE 则弹出提示，告诉用户当前浏览器不能访问剪贴板。如：

if (navigator.userAgent.indexOf("MSIE") == -1)

{

alert("您的浏览器不支持此功能,请手工复制文本框中内容");

return false;

}

<http://blog.csdn.net/fox123871/article/details/6454634>有详细介绍

clipboardData 对象提供了对剪贴板的访问。

三个方法   
1.clearData(sDataFormat) 删除剪贴板中指定格式的数据。   
2.getData(sDataFormat) 从剪贴板获取指定格式的数据。   
3.setData(sDataFormat, sData) 给剪贴板赋予指定格式的数据。返回 true 表示操作成功。

|  |  |
| --- | --- |
| 1 | <!doctype html> |
| 2 | <html> |
| 3 | <head> |
| 4 | <metahttp-equiv="content-type"content="text/html; charset=utf-8" /> |
| 5 | <scripttype="text/javascript"> |
| 6 | *function*CopyLinkAddress(){ |
| 7 | *window*.clipboardData.setData("Text", "请复制网址到您的QQ:"+ location.href); |
| 8 | alert("复制成功!"); |
| 9 | *var* tmp = clipboardData.getData("Text",text); |
| 10 | alert(tmp) |
| 11 | } |
| 12 | *var* text ="123"; |
| 13 | if (!*window*.clipboardData.setData('Text', text)) // 赋予 text 格式的数据 |
| 14 | { |
| 15 | alert("复制失败!"); |
| 16 | } |
| 17 | text =*window*.clipboardData.getData('Text'); // 获取 text 格式的数据 |
| 18 | alert(text); |
| 19 | *window*.clipboardData.clearData('Text'); // 清除 text 格式的数据 |
| 20 | text =*window*.clipboardData.getData('Text'); |
| 21 | alert(text); |
| 22 | </script> |
| 23 | </head> |
| 24 | <body> |
| 25 | <inputtype="button"value="复制网址"onclick="CopyLinkAddress()" /> |
| 26 | </body> |
| 27 | </html> |

js验证表单大全:<http://www.jb51.net/article/4976.htm>

**3.关于js的window.location属性详解**

<http://www.5icool.org/a/201105/a564.html>

**window.location 对象所包含的属性**

|  |  |
| --- | --- |
| **属性** | **描述** |
| hash | 从井号 (#) 开始的 URL（锚） |
| host | 主机名和当前 URL 的端口号 |
| hostname | 当前 URL 的主机名 |
| href | 完整的 URL |
| pathname | 当前 URL 的路径部分 |
| port | 当前 URL 的端口号 |
| protocol | 当前 URL 的协议 |
| search | 从问号 (?) 开始的 URL（查询部分） |

window.location.hash

location.hash 设置或获取 href 属性中在井号“#”后面的分段。   
 location.host 设置或获取 location 或 URL 的 hostname 和 port 号码。   
 location.hostname 设置或获取 location 或 URL 的主机名称部分。   
 location.href 设置或获取整个 URL 为字符串。   
 location.pathname 设置或获取对象指定的文件名或路径。   
 location.port 设置或获取与 URL 关联的端口号码。   
 location.protocol 设置或获取 URL 的协议部分。   
 location.search 设置或获取 href 属性中跟在问号后面的部分。

要使用 JS 定位锚点，完全可以使用 window.[hash](javascript:;) 配合元素 ID 完成。比如快速定位到页面的某条评论，则直接使用如下代码即可：

window.location.hash = "#comment-5981";

另外 Twitter、Facebook、Google 等已经开始大量使用 #! 这种形式的 [hash](javascript:;) 方法处理异步交互页面的 URL 可回溯功能。

**window.location.search**

如果有这样一个 URL 地址：

http://www.google.com.hk/search?hl=zh-CN&source=hp&biw=1400&bih=935&q=%E8%8A%92%E6%9E%9C%E5%B0%8F%E7%AB%99&aq=f&aqi=&aql=&oq=

如何利用 JS 脚本捕获页面 GET 方式请求的参数？其实直接使用 window.location.search 获得，然后通过 split 方法结合循环遍历自由组织数据格式。

另外，如果根据用户的搜索条件刷新页面，只需直接设置 window.location.search 即可。

**4.js 页面刷新location.reload和location.replace的区别小结**

<http://www.jb51.net/article/21573.htm>

js刷新页面方法大全: http://www.jb51.net/article/14397.htm

在实际应用的时候，重新刷新页面的时候，我们通常使用： location.reload() 或者是 history.go(0) 来做。下面有一些相关的内容，大家看完了就会有更多的收获。

首先介绍两个方法的语法：   
**reload 方法，该方法强迫浏览器刷新当前页面。**   
**语法： location.reload([bForceGet])**   
参数： bForceGet，可选参数，默认为 false，从客户端缓存里取当前页。true, 则以 GET 方式，从服务端取最新的页面, 相当于客户端点击 F5("刷新")   
  
**replace 方法**，该方法通过指定URL替换当前缓存在历史里（客户端）的项目，因此当使用replace方法之后，你不能通过“前进”和“后退”来访问已经被替换的URL。   
**语法： location.replace(URL)**   
在实际应用的时候，重新刷新页面的时候，我们通常使用： location.reload() 或者是 history.go(0) 来做。因为这种做法就像是客户端点F5刷新页面，所以页面的method="post"的时候，会出现“网页过期”的提示。那是因为Session的安全保护机制。可以想到：当调用 location.reload() 方法的时候， aspx页面此时在服务端内存里已经存在，因此必定是 IsPostback 的。如果有这种应用：我们需要重新加载该页面，也就是说我们期望页面能够在服务端重新被创建，我们期望是 Not IsPostback 的。这里，location.replace() 就可以完成此任务。被replace的页面每次都在服务端重新生成。你可以这么写： location.replace(location.href)   
  
下面是相关的应用与知识点：   
window.location.href   
window.top.location.replace("http://www.jb51.net")   
top.location.href("http://www.jb51.net")   
window.navigate ("http://www.jb51.net")   
Html：   
<meta http-equiv="Refresh" content="0; url=http://baidu.com/" >   
---------------------------------------   
URL即:统一资源定位符 (Uniform Resource Locator, URL)   
完整的URL由这几个部分构成：   
scheme://host:port/path?query#fragment   
scheme:通信协议   
常用的http,ftp,maito等   
host:主机   
服务器(计算机)域名系统 (DNS) 主机名或 IP 地址。   
port:端口号   
整数，可选，省略时使用方案的默认端口，如http的默认端口为80。   
path:路径   
由零或多个"/"符号隔开的字符串，一般用来表示主机上的一个目录或文件地址。   
query:查询   
可选，用于给动态网页（如使用CGI、ISAPI、PHP/JSP/ASP/ASP.NET等技术制作的网页）传递参数，可有多个参数，用"&"符号隔开，每个参数的名和值用"="符号隔开。   
fragment:信息片断   
字符串，用于指定网络资源中的片断。例如一个网页中有多个名词解释，可使用fragment直接定位到某一名词解释。(也称为锚点.)   
对于这样一个URL   
http://www.jb51.net:80/fisker/post/0703/window.location.html?ver=1.0&id=6#imhere   
  
我们可以用javascript获得其中的各个部分   
1, window.location.href   
整个URl字符串(在浏览器中就是完整的地址栏)   
本例返回值: http://www.jb51.net:80/fisker/post/0703/window.location.html?ver=1.0&id=6#imhere   
2,window.location.protocol   
URL 的协议部分   
本例返回值:http:   
3,window.location.host   
URL 的主机部分   
本例返回值:www.jb51.net   
4,window.location.port   
URL 的端口部分   
如果采用默认的80端口(update:即使添加了:80)，那么返回值并不是默认的80而是空字符   
本例返回值:""   
5,window.location.pathname   
URL 的路径部分(就是文件地址)   
本例返回值:/fisker/post/0703/window.location.html   
6,window.location.search   
查询(参数)部分   
除了给动态语言赋值以外，我们同样可以给静态页面,并使用javascript来获得相信应的参数值   
本例返回值:?ver=1.0&id=6   
7,window.location.hash   
锚点   
本例返回值:#imhere

**5.如何获取select下拉框的值(option没有及有value属性)**

1.下拉框的option没有value属性时

<select id="param1">   
 <option>学号</option>   
 <option>姓名</option>   
 <option>年龄</option>   
</seclect>   
<script>   
 window.onload = funciton(){   
 var param = document.getElementById("param1");   
 param.onchange = getParam(){   
 var text = param.value;   
 }   
}   
</script>

2.下拉框的option有value属性时

<select id="param1">   
 <option>学号</option>   
 <option>姓名</option>   
 <option>年龄</option>   
</seclect>   
<script>   
 window.onload = funciton(){   
 var param = document.getElementById("param1");   
 param.onchange = getParam(){   
 var selectIndex = param.selectIndex; //获得被选项的引索号   
 var text = param.options[selectIndex].text;   
</script>

**js实现完美拖拽效果可拖动层与回放拖动规迹并显示拖动距离参数**

<http://www.jb51.net/article/32822.htm>

<http://www.jb51.net/article/25203.htm>

**javascript阻止scroll事件多次执行的思路及实现**

<http://www.jb51.net/article/42972.htm>

**如何弹出QQ临时对话框实现不添加好友在线交谈效果**

<http://www.jb51.net/web/115867.html>

手风琴效果

<http://www.jb51.net/article/31107.htm>

<http://tympanus.net/Tutorials/ElegantAccordion/>

**下面是一个产生一定范围整数的随机函数**

|  |
| --- |
|  |
| 3 | *function*getRandom(*min,max*) |
| 4 | { |
| 5 | return((*Math*.floor(*Math*.random()\*(max-min)))+min); |
| 6 | } |

**下面是一个限制值范围的函数:**

***function*** clamp**(**val**,** min**,** max**){**//将val的值限制在min 和 max之间

***return*** Math**.**max**(**min**,** Math**.**min**(**max**,** val**));**

**}**

**关于js在webkit内核可以运行,但是在IE中不能被运行,一般都是由于以下的错误导致:**

1.网页错误详细信息

用户代理: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; InfoPath.2; .NET CLR 2.0.50727; .NET4.0C; .NET4.0E)

时间戳: Mon, 11 Nov 2013 00:53:41 UTC

消息: 'console' 未定义

行: 27

字符: 2

代码: 0

URI: http://localhost:8077/work/work/Z/themis/themis/js/script.js

2. 网页错误详细信息

用户代理: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; InfoPath.2; .NET CLR 2.0.50727; .NET4.0C; .NET4.0E)

时间戳: Wed, 13 Nov 2013 05:41:20 UTC

消息: 缺少标识符、字符串或数字

行: 60

字符: 37

代码: 0

URI: <http://localhost:8077/work/work/Z/demo-2/demo/js/themis.js>

错误代码如下:

**$(this).animate({"height":"0px",},getRandom(500,1000),'easeInOutExpo',function(){}**

**原因就是数组或对象最后的元素后面的逗号所导致,这在IE8(以下的浏览器(IE7,6)中会导致报错,代码无法运行.**

**js中cookie的使用详细分析**

**详细出处参考：**[**http://www.jb51.net/article/14566.htm**](http://www.jb51.net/article/14566.htm)

**jquery表单验证使用插件formValidator**

**详细出处参考：http://www.jb51.net/article/31835.htm**

[**http://www.yhuan.com/formvalidator/userguide.html**](http://www.yhuan.com/formvalidator/userguide.html)

**http://www.yhuan.com/formvalidator/demo.html**