Firefox浏览器良好支持W3C标准，是目前对CSS支持最好的浏览器，而ie是出现的比较早，在w3c支持方面做的一直不是很好，所以两种浏览器在很多方面不尽相同。   
下面总结一下这两种浏览器的兼容问题： 

**1：集合类对象问题**  
说明：IE下，可以使用()或[]获取集合类对象。  
Firefox下，只能使用[]获取集合类对象。  
解决方法：统一使用[]获取集合类对象。   
  
**2：HTML对象获取问题**FireFox：document，getElementById("idName")。   
ie：document.idname或者document.getElementById("idName")，   
解决办法：统一使用document.getElementById("idName")。   
  
**3：const问题**说明：Firefox下，可以使用const关键字或var关键字来定义常量。   
IE下，只能使用var关键字来定义常量，    
解决方法：统一使用var关键字来定义常量。   
  
**4：window.event问题**说明：window.event只能在IE下运行，而不能在Firefox下运行，这是因为Firefox的event只能在事件发生的现场使用， Firefox必须从源处加入event作参数传递，IE忽略该参数，用window.event来读取该event。   
解决方法：    
function   Submitted(evt)   {    
evt = evt?evt:(window,event?window.event:null);  
}    
**5：event.x与event.y问题**说明：IE下，even对象有x，y属性，但是没有pageX，pageY属性。   
Firefox下，even对象有pageX，pageY属性，但是没有x，y属性，    
解决方法：使用mX(mX   =   event.x   ?   event.x  :  event.pageX;)来代替IE下的event.x或者Firefox下的event.pageX，    
  
**6：event.srcElement问题**说明：IE下，event对象有srcElement属性，但是没有target属性。   
Firefox下，even对象有target属性，但是没有srcElement属性，    
解决方法：使用obj(obj   =   event.srcElement   ?   event.srcElement   ：   event.target;)来代替IE下的event.srcElement或者Firefox下的event.target， 请同时注意event的兼容性问题。   
  
**7：window.location.href问题**说明：IE或者Firefox2.0.x下，可以使用window.location或window.location.href;   
Firefox1.5.x下，只能使用window.location。  
解决方法：使用window.location来代替window.location.href;  
  
**8：模态和非模态窗口问题**说明：IE下，可以通过showModalDialog和showModelessDialog打开模态和非模态窗口。Firefox下则不能。

解决方法：直接使用window.open(pageURL;name;parameters)方式打开新窗口。   
如果需要将子窗口中的参数传递回父窗口，可以在子窗口中使用window.opener来访问父窗口，    
例如：var   parWin   =   window.opener;   parWin.document.getElementById("Aqing").value   =   "Aqing";  
  
**9：frame问题**以下面的frame为例：   
<frame   src="xxx.html"   id="frameId"   name="frameName"   />   
(1)访问frame对象：   
IE：使用window.frameId或者window.frameName来访问这个frame对象， frameId和frameName可以同名。   
Firefox：只能使用window.frameName来访问这个frame对象：  
另外，在IE和Firefox中都可以使用window.document.getElementById("frameId")来访问这个frame对象。  
  
(2)切换frame内容：   
在 IE和Firefox中都可以使用window.document.getElementById("testFrame").src   =   "xxx.html"或window.frameName.location   =   "xxx.html"来切换frame的内容，如果需要将frame中的参数传回父窗口(注意不是opener，而是parent   frame)，可以在frame中使用parent来访问父窗口。例如：parent.document.form1.filename.value="Aqing"。   
  
**10：body问题**Firefox的body在body标签没有被浏览器完全读入之前就存在；而IE的body则必须在body标签被浏览器完全读入之后才存在。

**11：事件委托方法**IE：document.body.onload   =   inject。   //Function   inject()在这之前已被实现   
Firefox：document.body.onload   =   inject()。    
  
**12：firefox与IE的父元素(parentElement)的区别**IE：obj.parentElement   
firefox：obj.parentNode   
解决方法：因为firefox与IE都支持DOM，因此使用obj.parentNode是不错选择。

**13：cursor：hand   VS   cursor：pointer**firefox不支持hand，但ie支持pointer   
解决方法：统一使用pointer   
  
**14：innerText在IE中能正常工作，但是innerText在FireFox中却不行，需用textContent。**   
解决方法：   
if(navigator.appName.indexOf("Explorer")   >   -1){   
        document.getElementById('element').innerText   =   "my   text";  
} else{   
        document.getElementById('element').textContent   =   "my   text";   
}   
  
**15：FireFox中设置HTML标签的style时，所有位置性和字体尺寸的值必须后跟px。这个ie也是支持的。**  
  
**16：ie，firefox以及其它浏览器对于   table   标签的操作都各不相同， 在ie中不允许对table和tr的innerHTML赋值，使用js增加一个tr时，使用appendChild方法也不管用。**解决方法：   
//向table追加一个空行：   
var   row   =   otable，insertRow(-1);  
var   cell   =   document.createElement("td");   
cell.innerHTML   =   "   ";    
cell.className   =   "XXXX";    
row.appendChild(cell);  
  
**17：padding   问题**padding   5px   4px   3px   1px;   FireFox无法解释简写， 必须改成padding-top：5px;   padding-right：4px;  padding-bottom：3px;   padding-left：1px;   
  
**18：消除ul、ol等列表的缩进时 样式应写成：list-style：’none’; margin：’0px’;padding：’0px’;**   
其中margin属性对IE有效，padding属性对FireFox有效。

**19：CSS透明**IE：filter：progid：DXImageTransform.Microsoft.Alpha(style=0;opacity=60)。    
FF：opacity：0.6。    
  
**20：CSS圆角**IE：不支持圆角。    
FF：-moz-border-radius:4px，或者-moz-border-radius-topleft:4px。-moz-border-radius-topright:4px。-moz-border-radius-bottomleft:4px。-moz-border-radius-bottomright:4px。   
  
**21：CSS双线凹凸边框**IE：border:2px   outset;  
FF： -moz-border-top-colors:#d4d0c8   white;-moz-border-left-colors:#d4d0c8   white;-moz-border-right-colors:#404040   #808080;-moz-border-bottom-colors:#404040   #808080;   
  
**22：对select的options集合操作**枚举元素除了[]外，SelectName，options，item()也是可以的，   另外SelectName.options.length，   SelectName.options.add/remove都可以在两种浏览器上使用。注意在add后赋值元素，否则会失败。   
  
**23：XMLHTTP的区别**  
//mf   
if   (window，XMLHttpRequest)   //mf   
    {   
    xmlhttp=new   XMLHttpRequest()   
    xmlhttp，onreadystatechange=xmlhttpChange   
    xmlhttp，open("GET"，url，true)   
    xmlhttp，send(null)   
    }   
//ie   
else   if   (window，ActiveXObject)     //   code   for   IE   
    {   
    xmlhttp=new   ActiveXObject("Microsoft，XMLHTTP")   
        if   (xmlhttp)   
        {   
        xmlhttp，onreadystatechange=xmlhttpChange   
        xmlhttp，open("GET"，url，true)   
        xmlhttp，send()   
        }   
    }   
}   
  
**24：css中的width和padding**在IE7和FF中width宽度不包括padding，在Ie6中包括padding，   
  
**25：css hack**根据不同浏览器对CSS样式的支持程度，解析结果和识别CSS的优先级不同，设计师们就可以根据这些不同浏览器 的特点来书写不同的CSS样式代码。   
IE6能识别下划线\_和星号\*，IE7能识别星号\*，不能识别下划线\_，   
而firefox两个都不能识别，如此，就可 以针对IE6，IE7和FF通过对这些特殊符号的使用写不同的代码了。   
<style>   
div{   
background：green。 /\* for FireFox \*/   
\*background：red。 /\* for IE7 \*/   
\_background：blue。 /\* for IE6 \*/   
}   
</style>   
  
该样式显示的效果是：在FireFox中背景色为green；在IE7中背景色为red；在IE6中背景色为blue。   
  
此外，!important声明也可以很好地提升指定样式规则的应用优先权。   
在IE6和FF中用!important声明可以提高优先级别，但在IE6中 的!important声明会被之后的同名属性定义替换。   
所以，通过\*和!important声明两者的搭配也可以很好地解决IE6，IE7和FF三者之 间的兼容性问题。   
  
区别FF，IE7，IE6：background：red。 \*background：green !important。 \*background：blue。   
  
注：IE都能识别\*。FF不能识别\*；IE6能识别\*，但不能识别 !important；   
IE7能识别\*，也能识别!important。FF不能识别\*，但能识别!important。

针对IE7/firefox 在css的前面加 [xmlns]， 如下面的left属性，如果我想要只针对IE7/firefox起作用，写法如下：   
[xmlns] #left {   
float：left。   
border：4px solid #999。   
padding：5px。   
width：200px。   
height：200px。   
}   
只针对IE6起作用，可以在css前面加\* html，如：   
\* html #left {   
clear：both。   
}   
  
只针对IE7起作用，在css里前面加\*+html，如：   
\*+html #left {   
clear：both。   
}   
  
  
书写的顺序都是FireFox的写在前面，IE7的写在中间，IE6的写在最后面。   
  
**26：使用IE专用的条件注释**  
<!--其他浏览器 -->   
<link rel="stylesheet" type="text/css" href="css，css" />   
  
<!--[if IE 7]>   
<!-- 适合于IE7 -->   
<link rel="stylesheet" type="text/css" href="ie7，css" />   
<![endif]-->   
  
<!--[if lte IE 6]>   
<!-- 适合于IE6及一下 -->   
<link rel="stylesheet" type="text/css" href="ie，css" />   
<![endif]-->   
  
**27：div 的垂直居中问题：**

vertical-align：middle。 将行距增加到和整个DIV一样高 line-height：200px。 然后插入文字，就垂直居中了。缺点是要控制内容不要换行   
  
**28：cursor： pointer;**

可以同时在 IE FF 中显示游标手指状， hand 仅 IE 可以 。  
  
**29：**FF： 链接加边框和背景色，需设置 display:block， 同时设置 float:left 保证不换行。参照 menubar， 给 a 和 menubar 设置高度是为了避免底边显示错位， 若不设 height， 可以在 menubar 中插入一个空格。   
  
**30：**在mozilla firefox和IE中的BOX模型解释不一致导致相差2px解决方法：div{margin：30px!important。margin：28px。}   
注意这两个margin的顺序一定不能写反，据阿捷的说法!important这个属性IE不能识别，但别的浏览器可以识别。所以在IE下其实解释成这样：div{maring：30px。margin：28px}   
重复定义的话按照最后一个来执行，所以不可以只写margin：XXpx!important。   
  
**31：IE5 和IE6的BOX解释不一致**IE5下div{width：300px。margin：0 10px 0 10px。}   
div 的宽度会被解释为300px-10px(右填充)-10px(左填充)最终div的宽度为280px，而在IE6和其他浏览器上宽度则是以 300px+10px(右填充)+10px(左填充)=320px来计算的。这时我们可以做如下修改 div{width:300px!important;width /\*\*/:340px;margin:0 10px 0 10px;} ;