浏览器兼容性问题大汇总  
  
  
1. HTML对象获取问题 3  
  
2. const问题 3  
  
3. event.x与event.y问题 3  
  
4. window.location.href问题 3  
  
5. frame问题 3  
  
6. 模态和非模态窗口问题 3  
  
7. firefox与IE的父元素(parentElement)的区别 3  
  
8. document.formName.item(”itemName”) 问题 3  
  
9. 集合类对象问题 3  
  
10. 自定义属性问题 3  
  
11. input.type属性问题 3  
  
12. event.srcElement问题 3  
  
13. body载入问题 3  
  
14. 事件委托方法 3  
  
15. Table操作问题 3  
  
16. 对象宽高赋值问题 3  
  
Ø CSS 3  
  
1. cursor:hand   VS   cursor:pointer 3  
  
2. innerText在IE中能正常工作，但在FireFox中却不行. 3  
  
3. CSS透明 3  
  
4. css中的width和padding 3  
  
5. FF和IE BOX模型解释不一致导致相差2px 3  
  
6. IE5 和IE6的BOX解释不一致 3  
  
7. ul和ol列表缩进问题 3  
  
8. 元素水平居中问题 3  
  
9. Div的垂直居中问题 3  
  
10. margin加倍的问题 3  
  
11. IE与宽度和高度的问题 3  
  
12. 页面的最小宽度 3  
  
13. DIV浮动IE文本产生3象素的bug 3  
  
14. IE捉迷藏的问题 3  
  
15. float的div闭合;清除浮动;自适应高度 3  
  
16. 高度不适应 3  
  
17. IE6下图片下有空隙产生 3  
  
18. 对齐文本与文本输入框 3  
  
19. LI中内容超过长度后以省略号显示 3  
  
20. 为什么web标准中IE无法设置滚动条颜色了 3  
  
21. 为什么无法定义1px左右高度的容器 3  
  
22. 链接(a标签)的边框与背景 3  
  
23. 超链接访问过后hover样式就不出现的问题 3  
  
24. FORM标签 3  
  
25. 属性选择器(这个不能算是兼容,是隐藏css的一个bug) 3  
  
26. 为什么FF下文本无法撑开容器的高度 3  
  
27. 关于空格的解释  
  
28. 条件注释  
29. 强制渲染  
30. js兼容文件  
31. 浏览器识别符  
  
  
Ø JavaScript  
1. 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");  
  
2. 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.  
  
3. event.x与event.y问题  
说明:IE下,event对象有x,y属性,但是没有pageX,pageY属性;  
Firefox下,event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.   
  
4. 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.  
  
5. 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";  
  
6. 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";   
  
7. 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.  
  
8. document.formName.item(”itemName”) 问题  
问题说明：IE下，可以使用document.formName.item(”itemName”) 或document.formName.elements ["elementName"]；Firefox 下，只能使用document.formName.elements["elementName"]。  
解决方法：统一使用document.formName.elements["elementName"]。  
  
9. 集合类对象问题  
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。  
  
10. 自定义属性问题  
问题说明：IE下，可以使用获取常规属性的方法来获取自定义属性，也可以使用getAttribute() 获取自定义属性；Firefox下，只能使用getAttribute() 获取自定义属性。  
解决方法：统一通过getAttribute() 获取自定义属性。  
  
11. input.type属性问题  
问题说明：IE下input.type属性为只读；但是Firefox下input.type属性为读写。  
解决办法：不修改input.type属性。如果必须要修改，可以先隐藏原来的input，然后在同样的位置再插入一个新的input元素。  
  
12. event.srcElement问题  
问题说明：IE下，even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srcObj = event.srcElement ?event.srcElement : event.target;  
如果考虑第8条问题，就改用myEvent代替event即可。  
  
13. body载入问题  
问题说明：Firefox的body对象在body标签没有被浏览器完全读入之前就存在；而IE的body对象则必须在body标签被浏览器完全读入之后才存在。  
[注] 这个问题尚未实际验证，待验证后再来修改。  
[注] 经验证，IE6、Opera9以及FireFox2中不存在上述问题，单纯的JS脚本可以访问在脚本之前已经载入的所有对象和元素，即使这个元素还没有载入完成。  
  
14. 事件委托方法  
问题说明：IE下，使用document.body.onload = inject; 其中function inject()在这之前已被实现；在Firefox下，使用document.body.onload = inject();  
解决方法：统一使用document.body.onload=new Function(’inject()’); 或者document.body.onload = function(){}  
[注意] Function和function的区别。  
  
15. Table操作问题  
问题说明：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);[注] 由于俺很少使用JS直接操作表格，这个问题没有遇见过。建议使用JS框架集来操作table，如JQuery。  
  
16. 对象宽高赋值问题  
问题说明：FireFox中类似obj.style.height = imgObj.height的语句无效。  
  
Ø CSS  
1. cursor:hand   VS   cursor:pointer  
firefox不支持hand，但ie支持pointer  
解决方法:   统一使用pointer  
  
17. innerText在IE中能正常工作，但在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";  
}  
  
18. CSS透明  
IE：filter:progid:DXImageTransform.Microsoft.Alpha(style=0,opacity=60)。  
FF：opacity:0.6。  
opacity 透明，子元素会继承透明属性。解决方式：1、使用 background:rgba(0,0,0,.6) //IE8及以下无效果。 2、使用定位，背景色与子元素处于同级关系。  
  
19. css中的width和padding  
在IE7和FF中width宽度不包括padding，在Ie6中包括padding.  
  
20. FF和IEBOX模型解释不一致导致相差2px  
box.style{width:100;border 1px;}   
ie理解为box.width = 100   
ff理解为box.width = 100 + 1\*2 = 102  //加上边框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， IE不能识别!important这个属性，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;  
  
21. 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}  
  
22. ul和ol列表缩进问题  
消除ul、ol等列表的缩进时，样式应写成：list-style:none;margin:0px;padding:0px;  
经验证，在IE中，设置margin:0px可以去除列表的上下左右缩进、空白以及列表编号或圆点，设置padding对样式没有影响；在 Firefox 中，设置margin:0px仅仅可以去除上下的空白，设置padding:0px后仅仅可以去掉左右缩进，还必须设置list- style:none才能去除列表编号或圆点。也就是说，在IE中仅仅设置margin:0px即可达到最终效果，而在Firefox中必须同时设置margin:0px、 padding:0px以及list-style:none三项才能达到最终效果。  
  
23. 元素水平居中问题  
FF: margin:0 auto;  
  
IE: 父级{ text-align:center; }  
  
24. 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。  
  
25. margin加倍的问题  
设置为float的div在ie下设置的margin会加倍。这是一个ie6都存在的bug。解决方案是在这个div里面加上display:inline;   
  
例如：  
  
<div id=”imfloat”>  
相应的css为  
#imfloat{   
float:left;   
margin:5px;   
display:inline;}  
  
26. IE与宽度和高度的问题  
IE不认得min-这个定义，但实际上它把正常的width和height当作有min的情况来使。这样问题就大了，如果只用宽度和高度，正常的浏览器里这两个值就不会变，如果只用min-width和min-height的话，IE下面根本等于没有设置宽度和高度。  
  
比如要设置背景图片，这个宽度是比较重要的。要解决这个问题，可以这样：  
  
#box{ width: 80px; height: 35px;}html>body #box{ width: auto; height: auto; min-width: 80px; min-height: 35px;}  
  
27. 页面的最小宽度  
如上一个问题，IE不识别min，要实现最小宽度，可用下面的方法：  
  
#container{ min-width: 600px; width:expression(document.body.clientWidth＜ 600? "600px": "auto" );}   
  
第一个min-width是正常的；但第2行的width使用了Javascript，这只有IE才认得，这也会让你的HTML文档不太正规。它实际上通过Javascript的判断来实现最小宽度。  
  
28. DIV浮动IE文本产生3象素的bug  
左边对象浮动，右边采用外补丁的左边距来定位，右边对象内的文本会离左边有3px的间距.   
  
#box{ float:left; width:800px;}   
#left{ float:left; width:50%;}   
#right{ width:50%;}   
\*html #left{ margin-right:-3px; //这句是关键}   
<div id="box">  
<div id="left">＜/div>  
<div id="right">＜/div>  
</div>  
  
29. IE捉迷藏的问题  
当div应用复杂的时候每个栏中又有一些链接，DIV等这个时候容易发生捉迷藏的问题。  
  
有些内容显示不出来，当鼠标选择这个区域是发现内容确实在页面。  
  
解决办法：对#layout使用line-height属性或者给#layout使用固定高和宽。页面结构尽量简单。  
  
30. float的div闭合;清除浮动;自适应高度  
① 例如：＜div id=”floatA”>＜div id=”floatB”>＜div id=”NOTfloatC”>  
  
这里的NOTfloatC并不希望继续平移，而是希望往下排。(其中floatA、floatB的属性已经设置为float:left;)   
  
这段代码在IE中毫无问题，问题出在FF。原因是NOTfloatC并非float标签，必须将float标签闭合。在＜div class=”floatB”>＜div class=”NOTfloatC”>之间加上＜div class=”clear”>这个div一定要注意位置，而且必须与两个具有float属性的div同级，之间不能存在嵌套关系，否则会产生异常。并且将clear这种样式定义为为如下即可：.clear{clear:both;}  
  
②作为外部 wrapper 的 div 不要定死高度,为了让高度能自适应，要在wrapper里面加上overflow:hidden; 当包含float的box的时候，高度自适应在IE下无效，这时候应该触发IE的layout私有属性(万恶的IE啊！)用zoom:1;可以做到，这样就达到了兼容。  
例如某一个wrapper如下定义：  
  
.colwrapper{overflow:hidden; zoom:1; margin:5px auto;}  
  
③对于排版,我们用得最多的css描述可能就是float:left.有的时候我们需要在n栏的float div后面做一个统一的背景,譬如:   
  
<div id=”page”>  
  
<div id=”left”>＜/div>  
<div id=”center”>＜/div>  
<div id=”right”>＜/div>  
  
</div>  
  
比如我们要将page的背景设置成蓝色,以达到所有三栏的背景颜色是蓝色的目的,但是我们会发现随着left center right的向下拉长,而page居然保存高度不变,问题来了,原因在于page不是float属性,而我们的page由于要居中,不能设置成float,所以我们应该这样解决：  
  
<div id=”page”>  
  
<div id=”bg” style=”float:left;width:100%”>  
  
<div id=”left”>＜/div>  
<div id=”center”>＜/div>  
<div id=”right”>＜/div>  
  
</div>  
  
</div>  
  
再嵌入一个float left而宽度是100%的DIV解决之。  
  
或者另一种方法：用选择器（：after）在page之后插入一个空标签,并清除浮动  
  
.page:after {  content: ""; display: table; clear: both; }  
  
④万能float 闭合(非常重要!)   
  
关于 clear float 的原理可参见 [How To Clear Floats Without Structural Markup],将以下代码加入Global CSS 中,给需要闭合的div加上class="clearfix" 即可,屡试不爽。  
  
  
.clearfix:after { content:"."; display:block; height:0; clear:both; visibility:hidden; }   
.clearfix { display:inline-block; }   
  
.clearfix {display:block;}   
  
  
  
或者这样设置：.hackbox{ display:table; //将对象作为块元素级的表格显示}  
  
31. 高度不适应  
高度不适应是当内层对象的高度发生变化时外层高度不能自动进行调节，特别是当内层对象使用margin 或padding时。  
  
例：  
  
#box {background-color:#eee; }   
#box p {margin-top: 20px;margin-bottom: 20px; text-align:center; }   
<div id="box">  
<p>p对象中的内容＜/p>  
</div>  
  
解决技巧：在P对象上下各加2个空的div对象CSS代码{height:0px;overflow:hidden;}或者为DIV加上border属性。  
  
32. IE6下图片下有空隙产生  
解决这个BUG的技巧有很多,可以是改变html的排版,或者设置img为display:block或者设置vertical-align属性为vertical-align:top/bottom/middle/text-bottom 都可以解决.  
  
33. 对齐文本与文本输入框  
加上vertical-align:middle;   
  
<style type="text/css">  
<!--  
input {   
width:200px;   
height:30px;   
border:1px solid red;   
vertical-align:middle;   
}   
-->  
</style>  
  
经验证，在IE下任一版本都不适用，而ff、opera、safari、chrome均OK！  
  
34. LI中内容超过长度后以省略号显示  
此技巧适用与IE、Opera、safari、chrom浏览器，FF暂不支持。  
  
<style type="text/css">  
<!--  
li {   
width:200px;   
white-space:nowrap;   
text-overflow:ellipsis;   
-o-text-overflow:ellipsis;   
overflow: hidden;   
}  
  
-->  
</style>  
  
35. 为什么web标准中IE无法设置滚动条颜色了  
解决办法是将body换成html   
  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />  
<style type="text/css">  
<!--   
html {   
scrollbar-face-color:#f6f6f6;   
scrollbar-highlight-color:#fff;   
scrollbar-shadow-color:#eeeeee;   
scrollbar-3dlight-color:#eeeeee;   
scrollbar-arrow-color:#000;   
scrollbar-track-color:#fff;   
scrollbar-darkshadow-color:#fff;   
}   
-->  
＜/style>  
  
36. 为什么无法定义1px左右高度的容器  
IE6下这个问题是因为默认的行高造成的,解决的技巧也有很多：  
  
例如:overflow:hidden　 zoom:0.08 　 line-height:1px  
  
16.怎么样才能让层显示在FLASH之上呢  
  
解决的办法是给FLASH设置透明  
  
<param name="wmode" value="transparent" />  
  
37. 链接(a标签)的边框与背景  
a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。  
  
38. 超链接访问过后hover样式就不出现的问题  
被点击访问过的超链接样式不在具有hover和active了,很多人应该都遇到过这个问题,解决技巧是改变CSS属性的排列顺序: L-V-H-A   
  
Code:   
  
<style type="text/css">  
<!--  
a:link {}   
a:visited {}   
a:hover {}   
a:active {}   
-->  
</style>  
  
39. FORM标签  
这个标签在IE中,将会自动margin一些边距,而在FF中margin则是0,因此,如果想显示一致,所以最好在css中指定margin和 padding,针对上面两个问题,我的css中一般首先都使用这样的样式ul,form{margin:0;padding:0;}。  
  
40. 属性选择器(这个不能算是兼容,是隐藏css的一个bug)  
p[id]{}div[id]{}   
  
这个对于IE6.0和IE6.0以下的版本都隐藏,FF和OPera作用.属性选择器和子选择器还是有区别的,子选择器的范围从形式来说缩小了,属性选择器的范围比较大,如p[id]中,所有p标签中有id的都是同样式的.  
  
41. 为什么FF下文本无法撑开容器的高度  
标准浏览器中固定高度值的容器是不会象IE6里那样被撑开的,那我又想固定高度,又想能被撑开需要怎样设置呢？办法就是去掉height设置min-height:200px; 这里为了照顾不认识min-height的IE6 可以这样定义:  
  
{   
height:auto!important;   
height:200px;   
min-height:200px;   
}   
  
  
43. IE和FireFox 对空格的尺寸解释不同，FireFox为4px,IE为8px; FireFox对div与div之间的空格是忽略的，但是IE是处理的。因此在两个相邻div之间不要有空格跟回车，否则可能造成不同浏览间之间格式不正确，比如著名的3px偏差（多个img标签连着，然后定义float: left;结果在firefox里面正常，而IE里面显示的每个img都相隔了3px。我把标签之间的空格都删除都没有作用。解决方法是在img外面套li，并且对li定义margin: 0; 避免方式：在必要的时候不要无视 list 标签）而且原因难以查明。  
  
44. 条件注释  
  
<link rel="stylesheet" type="text/css" href="css.css" />  
  
<!--[if IE 7]>  
<link rel="stylesheet" type="text/css" href="ie7.css" />  
<![endif]-->  
  
<!--[if lte IE 6]>  
<link rel="stylesheet" type="text/css" href="ie.css" />  
<![endif]-->  
  
lte -- 小于等于  
lt  -- 小于  
gte --  大于等于  
gt  --  大于  
！ --  不等于  
  
  
45.强制渲染  
<meta http-equiv=X-UA-Compatible content=IE=EmulateIE7>    //这句话的意思是强制使用IE7模式来解析网页代码！  
  
<meta http-equiv=“X-UA-Compatible” content=“IE=8″>  
  
<meta http-equiv=“X-UA-Compatible” content=“chrome=1″ />    //Google Chrome Frame也可以让IE用上Chrome的引擎  
  
<meta http-equiv=“X-UA-Compatible” content=“IE=EmulateIE7″><!– IE7 mode –> 或者 <meta http-equiv=“X-UA-Compatible” content=“IE=7″><!– IE7 mode –>       //强制IE8使用IE7模式来解析  
  
<meta http-equiv=“X-UA-Compatible” content=“IE=6″><!– IE6 mode –>   <meta http-equiv=“X-UA-Compatible” content=“IE=5″><!– IE5 mode –>   //强制IE8使用IE6或IE5模式来解析  
  
<meta http-equiv=“X-UA-Compatible” content=“IE=5; IE=8″ />   //一个特定版本的IE支持所要求的兼容性模式多于一种  
  
46.js兼容文件  
  
使IE5,IE6兼容到IE7模式（推荐）  
  
<!–[if lt IE 7]>  
<script src=”http://ie7-js.googlecode.com/svn/version/2.0(beta)/IE7.js” type=”text/javascript”></script>  
<![endif]–>  
使IE5,IE6,IE7兼容到IE8模式  
  
<!–[if lt IE 8]>  
<script src=”http://ie7-js.googlecode.com/svn/version/2.0(beta)/IE8.js” type=”text/javascript”></script>  
<![endif]–>  
使IE5,IE6,IE7,IE8兼容到IE9模式  
  
<!–[if lt IE 9]>  
<script src=”http://ie7-js.googlecode.com/svn/version/2.1(beta4)/IE9.js”></script>  
<![endif]–>  
  
  
47. 浏览器识别符  
  
p{ \_color:red; }           IE6 专用  
\*html p{ color:#red; }  IE6 专用  
p{ +color:red; }           IE6,7 专用  
p{ \*color:red; }           IE6,7 专用  
\*html p{ color:red; }    IE6,7 专用  
p{\*+color: red;}          IE7 专用  
Body> p{ color: red; }  屏蔽 IE6  
p{ color:red\9; }          IE8     
  
Firefox: -moz-  
Safari: -webkit-  
Opera: -o-  
IE: -ms-