**解决IE6、IE7、IE8、Firefox兼容的两种方案**

**浏览器不兼容，你知道源头吗？**

凡事都要知其然，才能知其所以然，前端开发的朋友想毕都会碰到浏览器兼容的问题，今天在网上觅到此文，希望对大家有用。

浏览器的不兼容，大家肯定都是深恶痛绝的，往往我们只是去做修补，却忘了更重要的事情，那就是追溯根源，避免类似的不兼容再次出现。在下不才，归纳几点html编码要素，望能指点各位：

1.文字本身的大小不兼容。同样是font-size:14px的宋体文字，在不同浏览器下占的空间是不一样的，ie下实际占高16px，下留白3px，ff下实际占高17px，上留白1px，下留白3px，opera下就更不一样了。解决方案：给文字设定 line-height 。确保所有文字都有默认的 line-height 值。这点很重要，在高度上我们不能容忍1px 的差异。

2.ff下容器高度限定，即容器定义了height之后，容器边框的外形就确定了，不会被内容撑大，而ie下是会被内容撑大，高度限定失效。所以不要轻易给容器定义height。

3.还讨论内容撑破容器问题，横向上的。如果float 容器未定义宽度，ff下内容会尽可能撑开容器宽度，ie下则会优先考虑内容折行。故，内容可能撑破的浮动容器需要定义width。

小实验：有兴趣大家可以看看这段实验。在不同浏览器下分别测试以下各项代码。

    a.<div style=”border:1px solid red;height:10px”></div>

b.<div style=”border:1px solid red;width:10px”></div>

c.<div style=”border:1px solid red;float:left”></div>

    d.<div style=”border:1px solid red;overflow:hidden”></div>

上面的代码在不同浏览器中是不一样的，实验起源于对小height 值div 的运用，<div style=”height:10px;overflow:hidden”></div>，小height 值要配合overflow:hidden一起使用。实验好玩而已，想说明的是，浏览器对容器的边界解释是大不相同的，容器内容的影响结果各不相同。

4.浮动的清除，ff下不清除浮动是不行的。

纠正大家一个误区，遇到不兼容就说ff烂是不对的，其实更多时候是ie的奇怪表现让我们无所适从。以下列出ie6的种种劣迹。

5.最被痛恨的，double-margin bug。ie6下给浮动容器定义margin-left 或者margin-right 实际效果是数值的2倍。解决方案，给浮动容器定义display:inline。

6.mirrormargin bug，当外层元素内有float元素时，外层元素如定义margin-top:14px，将自动生成margin-bottom:14px。padding也会出现类似问题，都是ie6下的特产，该类bug 出现的情况较为复杂，远不只这一种出现条件，还没系统整理。解决方案：外层元素设定border 或 设定float。

引申：ff 和ie 下对容器的margin-bottom，padding-bottom的解释有时不一致，似乎与之相关。

7.吞吃现象，限于篇幅，我就不展开了。还是ie6，上下两个div，上面的div设置背景，却发现下面没有设置背景的div 也有了背景，这就是吞吃现象。对应上面的背景吞吃现象，还有滚动下边框缺失的现象。解决方案：使用zoom:1。这个zoom好象是专门为解决ie6 bug而生的。

8.注释也能产生bug~~~“多出来的一只猪。”这是前人总结这个bug使用的文案，ie6的这个bug 下，大家会在页面看到猪字出现两遍，重复的内容量因注释的多少而变。解决方案：用“<!–[if !IE]> picRotate start <![endif]–>”方法写注释。

9.<li/>里加 float <div/>，这是一个典型的，棘手的兼容问题，希望引起大家正视 ，给li 不同的属性会有不同的解释效果，ff下的解释稍可理解，ie6下的解释会让你摸不着头脑，由于问题的复杂性，将另起一文专门讨论该问题。在《ul使用心得》一文里有相关成果，却没给出问题解决的过程。

10.使用了“float:left;display:inline”的ul的奇怪表现。可以看出这句css是针对ie6下的double margin bug 而加上的display:inline，这也是我的css体系里的重要一环，在《ul使用心得》一文中有相关阐述。而这句css用在ul上会让你痛苦不堪。点到为止，这里不能多说哈。

11.img下的留白，大家看这段代码有啥问题：

<div>  
<img src=”” mce\_src=”" />  
</div>

把div的border打开，你发现图片底部不是紧贴着容器底部的，是img后面的空白字符造成，要消除必须这样写

<div>  
<img src=”" mce\_src=”" /></div>

后面两个标签要紧挨着。ie7下这个bug 依然存在。解决方案：给img设定 display:block。

12.失去line-height。<div style=”line-height:20px”><img />文字</div>，很遗憾，在ie6下单行文字 line-height 效果消失了。。。，原因是<img />这个inline-block元素和inline元素写在一起了。解决方案：让img 和文字都 float起来。

引申：大家知道img 的align 有 text-top，middle，absmiddle啊什么的，你可以尝试去调整img 和文字让他们在ie和ff下能一致，你会发现怎么调都不会让你满意。索性让img 和文字都 float起来，用margin 调整。

13.链接的hover状态。a:hover img{width:300px} 我们想让鼠标hover时，链接里包含的图片宽度变化，可惜在ie6下无效，ie7、ff下有效。

14.非链接的hover状态。div:hover{} 这样的样式ie6是不认的，在ie7、ff下才有效果。

15.block化的a链接，其内套absolute层，absolute层内放置img，ie下，鼠标点击img不会有链接效果，ff、op下正常。

想不出来了，以后想到再加吧。上面的诸多问题如果你掌握了其中奥妙，90%的不兼容问题不需要另起css hack的。

16.无法彻底清除的float。如果让ul下的li具有了float属性，如何clear浮动的li呢？<ul><li class=”c”></li></ul> 或者 <ul><li><div>class=”c”></div></li></ul> 或者 <ul><li></li><div>class=”c”></div></ul> 或者 <ul><li></li></ul><div>class=”c”></div> 或者上述的组合？这个问题，我无法给出解答。下面有个例子与此相关

<!DOCTYPE html PUBLIC -//W3C//DTD XHTML 1.0 Transitional//EN http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>

<style type=text/css>  
.c{clear:both;overflow:hidden;+overflow:visible}  
.bd{border:1px solid red}

ul.ex{list-style:none;}  
ul.ex li{float:left;border:1px solid green;}  
</style>  
<ul class=ex>  
<li>sfsdfsfdf</li>  
<li>sfsdfsfdf</li>  
</ul>  
<div class=c></div>  
<div class=bd style=margin-top:19px>sfsdfsfdf</div>

请在ie下 测试，仅仅将 margin-top:19px 改为margin-top:20px 你发现什么了？要素：doctype必须有，ie6、ie7下margin-top:19px还好好的，margin-top:20px 就出问题了，无法解释。。。大家还可以将 clear 层换不同的位置测试。

解决方案：给ul 属性zoom:1 （给li 加zoom:1 没用）

引申：clear层应该单独使用。也许你为了节省代码把clear属性直接放到下面的一个内容层，这样有问题，不仅仅是ff和op下失去margin效果，ie下某些margin值也会失效  
<div style=”background:red;float:left;”>dd</div>  
<div style=”clear:both;margin-top:18px;background:green”>ff</div>

17.ie下overflow:hidden对其下的绝对层position:absolute或者相对层position:relative无效。解决方案：给overflow:hidden加position:relative或者position:absolute。另，ie6支持overflow-x或者overflow-y的特性，ie7、ff不支持。

18.ie6下严重的bug，float元素如没定义宽度，内部如有div定义了height或zoom:1，这个div就会占满一整行，即使你给了宽度。float元素如果作为布局用或复杂的容器，都要给个宽度的。

19.ie6下的bug，绝对定位的div下包含相对定位的div，如果给内层相对定位的div高度height具体值，内层相对层将具有100%的width值，外层绝对层将被撑大。解决方案给内层相对层float属性。

20.ie6下的bug，<head></head>内有<base target=”\_blank”/>的情况下，position:relative层下的float层内文字无法选中。这个bug迫使我修公用样式库。

21.终于来了个ff的缺点。width:100%这个东西在ie里用很方便，会向上逐层搜索width值，忽视浮动层的影响，ff下搜索至浮动层结束，如此，只能给中间的所有浮动层加width:100%才行，累啊。opera这点倒学乖了跟了ie。

**解决IE6、IE7、Firefox兼容的两种方案**

第一种方案：CSS Hack

具体写法很容易：

#someNode  
{  
    position: fixed;  
   #position: fixed;  
   \_position: fixed;  
}  
第一排给Firefox以及其他浏览器看   
第二排给IE7（可能以后的IE8、IE9也是如此，谁知道呢）看   
第三排给IE6以及更老的版本看   
最好的应用就是可以让IE6也“支持”position:fixed，而且，配合这个原理，可以做到不引入JavaScript代码（仅用IE6的expression），我这里有一个现成的页面，CSS如下写：

#ff-r  
{  
position: fixed;  
\_position: absolute;  
right:     15px;  
top:       15px;  
\_top:   expression(eval(document.compatMode &&  
            document.compatMode=='CSS1Compat') ?  
            documentElement.scrollTop+15 :  
            document.body.scrollTop +  
            (document.body.clientHeight  
            -this.clientHeight));  
}

第二种方案：

为不同的浏览器建立不同的.css方案，在用户浏览时，根据浏览器类型及版本号不同，分别加载不同的css文件。

当然，这里又产生两种方案，用JS判断或服务器端方法。

**附：各款浏览器对选择器的支持情况**

在正常的 selector { property:value; } 的基础上常会根据具体情况为相同元素使用Hack以达到浏览器间统一。

IE系列:

selector { +property:value; } 在属性名前加上加号"+",这个Hack只有IE系列可以识别.   
selector { \*property:value; } 在属性名前加上星号"\*",这个Hack只有IE系列可以识别.   
selector { \_property:value; } 在属性名前加上下划线"\_",这个Hack只有IE系列 (除IE7外) 识别.   
\* html selector{ property:value; } 在选择器上运用继承法 \* html selector, 这个Hack只有IE系列 (除IE7外) 可以识别.   
html/\*\*/ >body selector { property:value; } 在选择器上运用继承法 html/\*\*/ >body selector ,这个Hack只有IE系列 (除IE7外) 可以识别.   
selector { property/\*\*/:value; } 在属性名和冒号":"之间加入注释,屏蔽IE6用.   
selector/\*\*/ { property/\*\*/:value; } 在选择器和花括号"{"之间以及在属性名和冒号":"之间加入注释,屏蔽IE5和IE6用 (不屏蔽IE5.5) .   
select/\*\*/ { property:value; } 在选择器和花括号"{"之间加入注释,屏蔽IE5用.   
\*+html selector { property:value !important; } 在选择器上运用继承法 \*+html selector 再加上 !important, 这个Hack只有IE7可以识别.

Firefox:

\*:lang(lang) selector { property:value !important; } 用伪类lang(语言)再加上!important进行定义的话,目前只有Firefox可以识别.

Safari:

selector:empty { property:value !important; } 用伪类empty再加上!important进行定义的话,目前只有Safari可以识别.

Opera:

@media all and (min-width: 0px){ selector { property:value; } } 利用特殊继承法进行定义的话,目前只有Opera可以识别.

对Hack的运用,最普遍的是CSS盒模型Hack,清除浮动Hack.

CSS盒模型在IE5.X上是有严重解析错误的.这个Hack针对IE5.X:

selctor { width:IE5.X宽度; voice-family :"\"}\""; voice-family:inherit; width:正确宽度; }

清除浮动Hack

selector:after { content:"."; display:block; height:0; clear:both; visibility:hidden; }

浏览器兼容一览图：

