div+css各种浏览器Hack兼容性研究（一）

时间:2012-05-13 19:45 作者:admin 点击: 134 次

ie6,ie7,ie8,ie9 css bug兼容解决记录  
select {  
background-color:red\0; /\* ie 8/9\*/  
background-color:blue\9\0; /\* ie 9\*/  
\*background-color:#dddd00; /\* ie 7\*/  
\_background-color:#CDCDCD; /\* ie 6\*/  
}  
断断续续的在开发过程中收集了好多的bug以及其解决的办法，都在这个文章里面记录下来了！希望以后解决类似问题的时候能够快速解决  
，也希望大家能在留言里面跟进自己发现的ie6 7 8bug和解决办法！  
1：li边距“无故”增加  
任何事情都是有原因的，li边距也不例外。  
先描述一下具体状况：有些时候li边距会突然增 加很多，值也不固定（只在IE6/IE7有这种现象），让人摸不着头脑，仔细“研究”发现是由于其 低级元素ul的padding引 起，padding的上下值对li有影响，左右无影 响。所以只好笨手笨脚地把padding去掉，换成margin。 这是能解决问题，但往往不是我们想要的结果，或许 还会引起其他不必要的怪现象。  
现在终于发现解决这个问题的方法，其实很简单，既然是有ul引 起的，就设置ul的显示形式为\*display:inline-block;即可，前面加\*是只 针对IE6/IE7有效，其他浏览器并不渲染这个属性。  
2：分页数字 字体用“Arial ”加粗不抖动  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<meta http-equiv="Content-Type"content="text/html; charset=gb2312"/>  
<title>无标题文档</title>  
<link href="css/style.css"rel="stylesheet"type="text/css"/>  
<style type="text/css">  
body, ul, h1 {  
font-family:Arial;  
font-size:12px;  
}  
.page {  
text-align:center;  
}  
.page a {  
display:inline-block;  
padding:5px 8px;  
text-decoration:none;  
border:1px solid #09F;  
background-color:#0CF;  
color:#FFF;  
}  
.page a:hover, .page .selected {  
border:1px solid #CCC;  
background-color:#FFF;  
color:#000;  
font-weight:bold;  
}  
</style>  
</head>  
<body>  
<h1>分页样式</h1>  
<div class="page"> <a href="#">1</a> <a href="#"class="selected">2</a> <a href="#">3</a> <a href="#">4</a> <a href="#">5</a>  
<a href="#">6</a> <a href="#">7</a> <a href="#">8</a> <a href="#">9</a> <a href="#">10</a> </div>  
</body>  
</html>  
3：IE6 CSS选择器区分IE6  
IE6不支持子选择器；先针对IE6使用常规申明CSS选择器，然后再用子选择器针对IE7+及其他浏览器。  
/\*IE6 专用 \*/  
.content {color:red;}  
/\* 其他浏览器 \*/  
div>p .content {color:blue;}  
4：IE6最小高度  
IE6 不支持min-height属性，但它却认为height就是最小高度。解决方法：  
使用ie6不支持但其余浏览器支持的属性!important。  
#container{min-height:200px; height:auto !important; height:200px;}  
5：IE6100% 高度  
在IE6下，如果要给元素定义100%高度，必须要明确定义它的父级元素的高度，如果你需要给元素定义满屏的高度，就得  
先给html和body定义height:100%;。  
6：IE6躲猫猫bug  
在IE6和IE7下，躲猫猫bug是一个非常恼人的问题。一个撑破了容器的浮动元素，如果在他之后有不浮动的内容，并且有一  
些定义了:hover的链接，当鼠标移到那些链接上时，在IE6下就会触发躲猫猫。  
解决方法很简单：  
1.在（那个未浮动的）内容之后添加一个<span style="clear: both;"> </span>  
2.触发包含了这些链接的容器的hasLayout，一个简单的方法就是给其定义height:1%;  
7:IE6绝对定位元素的1像素间距bug  
IE6下的这个错误是由于进位处理误差造成（IE7已修复），当绝对定位元素的父元素高或宽为奇数时，bottom和right会  
产生错误。唯一的解决办法就是给父元素定义明确的高宽值，但对于液态布局没有完美的解决方法。  
8: IE下z-index的bug  
在IE浏览器中，定位元素的z-index层级是相对于各自的父级容器，所以会导致z-index出现错误的表现。解决方法是给  
其父级元素定义z-index，有些情况下还需要定义position:relative。  
9: Overflow Bug  
在IE6/7中，overflow无法正确的隐藏有相对定位position:relative;的子元素。解决方法就是给外包容器.wrap加上  
position:relative;。  
10: 横向列表宽度bug  
如果你使用float:left;把<li>横向摆列，并且<li>内包含的<a>（或其他）触发了 hasLayout，在IE6下就会有错误的  
表现。解决方法很简单，只需要给<a>定义同样的float:left;即可。  
11: 列表阶梯bug  
列表阶梯bug通常会在给<li>的子元素<a>使用float:left;时触发，我们本意是要做一个横向的列表(通常 是导航栏)，  
但IE却可能呈现出垂直的或者阶梯状。解决办法就是给<li>定义float:left;而非子元素<a>，或者 给<li>定义  
display:inline;也可以解决。  
12: 垂直列表间隙bug  
当我们使用<li> 包含一个块级子元素时，IE6(IE7也有可能)会错误地给每条列表元素（<li>）之间添加空隙。  
解决方法：把<a>flaot并且清除float来解决这个问题；另外一个办法就是触发<a>的hasLayout（如定 义高宽、  
使用zoom:1;）；也可以给<li> 定义display:inline;来解决此问题；另外还有一个极有趣的方法，给包含的文  
本末尾添加一个空格。  
13: IE6调整窗口大小的 Bug  
当把body居中放置，改变IE浏览器大小的时候，任何在body里面的相对定位元素都会固定不动了。解决办法：  
给body定义position:relative;就行了。  
14: 文本重复Bug  
在IE6中，一些隐藏的元素（如注释、display:none;的元素）被包含在一个浮动元素里，就有可能引发文本重复bug。  
解决办法：给浮动元素添加display:inline;。  
15：链接a的title属性中的文字换行  
啥也不说，先上个图  
  
我们都知道，可以给链接a加上title属性，这样鼠标移动上去会显示title属性定义的 文字，常常用来加一些提示语句，  
比如说点击查看详情之类的，代码形如：  
<a href=”#” title=”点击查看详情”>链接xx</a>  
。之前一直没有仔细注意过这个东西。如果鼠标浮动上去要显示更多东西的话，怎么实现呢。第一 反应是jquery的tooltip插件。今天无意中 发现某个学院的网站鼠标移动上去可以显示这么完整的信息，效果看起来似乎还可以，就想看看怎么做的， 找了下，没发现有Javascript脚本，再往链 接的地方一看，终于让我发现了门道了：  
<a href='#' target="\_blank" title="标题: 关于对我校2006年至2009年发展党员工作情况进…  
发布日期: 2010-5-31 16:05:08 类别：院务通知 点击: 139">  
[05-31] 关于对我校2006年至2009年发展党员工作情况进…  
</a>  
注意到了吗。。很简单，只要使用 这样的转义符号,即可实现换行。在一些tooltip要求定制性不高的情况下，这样的显示效果相 当不错，而且是浏览器原生的效果，安逸。

--------------------------------------------------------------------

* 让所有浏览器都支持max-height的CSS样式代码，完整：  
  max-height:1000px;\_height:expression((document.documentElement.clientHeight||document.body.clientHeight)<1000?"1000px":""); overflow:hidden;这里的1000和1000px是你需要的数值，注意3个数值的相同。

让IE6支持最大高度max-height的时候别忘记加上overflow:hidden;

* 让所有浏览器都支持min-height的CSS样式代码，完整：  
  min-height:1000px;\_height:expression((document.documentElement.clientHeight||document.body.clientHeight)>1000?"1000px":""); 这里的1000和1000px是你需要的数值，注意3个数值的相同。
* 让所有浏览器包括IE6即支持最大高度又支持最小高度。  
  .yangshi{Max-Height:620px;Min-Height:40px;\_height:expression(this.scrollHeight > 620 ? "620px" : (this.scrollHeight < 40 ? "40px" : "auto"));}

IE6支持Max-Height和支持Min-Height CSS代码  
\_height:expression(this.scrollHeight > 620 ? "620px" : (this.scrollHeight < 40 ? "40px" : "auto"));

说明：以上代码作用是让对象的最小高度为40px，最大高度为620px的CSS样式属性

**【CSS】最全的CSS浏览器兼容问题**

CSS对浏览器的兼容性有时让人很头疼,或许当你了解当中的技巧跟原理,就会觉得也不是难 事,从网上收集了IE7,6与Fireofx的兼容性处理方法并整理了一下.对于web2.0的过度,请尽量用xhtml格式写代码,而且DOCTYPE 影响 CSS 处理,作为W3C的标准,一定要加 DOCTYPE声名.

CSS技巧

1.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   
  
2. 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 #IamFloat{ float:left; margin:5px;/\*IE下理解为10px\*/ display:inline;/\*IE下再理解为5px\*/}   
  
3.浮动ie产生的双倍距离 #box{ float:left; width:100px; margin:0 0 0 100px; //这种情况之下IE会产生200px的距离 display:inline; //使浮动忽略} 这里细说一下block与inline两个元素：block元素的特点是,总是在新行上开始,高度,宽度,行高,边距都可以控制(块元素);Inline 元素的特点是,和其他元素在同一行上,不可控制(内嵌元素); #box{ display:block; //可以为内嵌元素模拟为块元素 display:inline; //实现同一行排列的效果 diplay:table;   
  
4 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;}   
  
5.页面的最小宽度 min -width是个非常方便的CSS命令，它可以指定元素最小也不能小于某个宽度，这样就能保证排版一直正确。但IE不认得这个，而它实际上把width当 做最小宽度来使。为了让这一命令在IE上也能用，可以把一个<div> 放到 <body> 标签下，然后为div指定一个类, 然后CSS这样设计： #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。   
  
6.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>   
  
7.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。   
  
8.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,为了让高度能自动适应，要在wrapper里面加上overflow:hidden; 当包含float的 box的时候，高度自动适应在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   
  
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. /\* Clear Fix \*/ .clearfix:after { content:"."; display:block; height:0; clear:both; visibility:hidden; } .clearfix { display:inline-block; } /\* Hide from IE Mac \*/ .clearfix {display:block;} /\* End hide from IE Mac \*/ /\* end of clearfix \*/ 或者这样设置：.hackbox{ display:table; //将对象作为块元素级的表格显示}   
  
11.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paddign 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> 解决方法：在P对象上下各加2个空的div对象CSS代码：.1{height:0px;overflow:hidden;}或者为DIV加上 border属性。   
  
12 .IE6下为什么图片下有空隙产生解决这个BUG的方法也有很多,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.   
  
13.如何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>   
  
14.web标准中定义id与class有什么区别吗 一.web标准中是不容许重复ID的,比如 div id="aa" 不容许重复2次,而class 定义的是类,理论上可以无限重复, 这样需要多次引用的定义便可以使用他. 二.属性的优先级问题 ID 的优先级要高于class,看上面的例子三.方便JS等客户端脚本,如果在页面中要对某个对象进行脚本操作,那么可以给他定义一个ID,否则只能利用遍历 页面元素加上指定特定属性来找到它,这是相对浪费时间资源,远远不如一个ID来得简单.   
  
15. LI中内容超过长度后以省略号显示的方法 此方法适用与IE与OP浏览器 <style type="text/css"> <!-- li { width:200px; white-space:nowrap; text-overflow:ellipsis; -o-text-overflow:ellipsis; overflow: hidden; } --> </style>   
  
16.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](http://www.douban.com/link2?url=http%3A//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>   
  
17.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,解决的方法也有很多,例如:overflow:hidden | zoom:0.08 | line-height:1px   
  
18.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" />   
  
19.怎样使一个层垂直居中于浏览器中这里我们使用百分比绝对定位,与外补丁负值的方法,负值的大小为其自身宽度高度除以二 <style type="text/css"> <!-- div { position:absolute; top:50%; lef:50%; margin:-100px 0 0 -100px; width:200px; height:200px; border:1px solid red; } --> </style>   
  
FF与IE   
  
1. Div居中问题 div设置 margin-left, margin-right 为 auto 时已经居中，IE 不行，IE需要设定body居中，首先在父级元素定义text-algin: center;这个的意思就是在父级元素内的内容居中。   
  
2.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。   
  
3.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 决方法是改变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>   
  
4. 游标手指cursor cursor: pointer 可以同时在 IE FF 中显示游标手指状， hand 仅 IE 可以   
  
5.UL的padding与margin ul标签在FF中默认是有padding值的,而在IE中只有margin默认有值,所以先定义 ul{margin:0;padding:0;}就能解决大部分问题   
  
6. 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;}给定义死了,所以后 面就不会为这个头疼了.   
  
7. BOX模型解释不一致问题 在FF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， 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:xx px!important; #box{ width:600px; //for ie6.0- w\idth:500px; //for ff+ie6.0} #box{ width:600px!important //for ff width:600px; //for ff+ie6.0 width /\*\*/:500px; //for ie6.0-}   
  
8.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.   
  
9.最狠的手段 - !important; 如果实在没有办法解决一些细节问题,可以用这个方法.FF对于”!important”会自动优先解析,然而IE则会忽略.如下 .tabd1{ background:url(/res/images/up/tab1.gif) no-repeat 0px 0px !important; /\*Style for FF\*/ background:url(/res/images/up/tab1.gif) no-repeat 1px 0px; /\* Style for IE \*/} 值得注意的是，一定要将xxxx !important 这句放置在另一句之上，上面已经提过   
  
10.IE,FF的默认值问题 或许你一直在抱怨为什么要专门为IE和FF写不同的CSS，为什么IE这样让人头疼，然后一边写css，一边咒骂那个可恶的M$ IE.其实对于css的标准支持方面，IE并没有我们想象的那么可恶，关键在于IE和FF的默认值不一样而已，掌握了这个技巧，你会发现写出兼容FF和 IE的css并不是那么困难，或许对于简单的css，你完全可以不用”!important”这个东西了。 我们都知道，浏览器在显示网页的时候，都会根据网页的 css样式表来决定如何显示，但是我们在样式表中未必会将所有的元素都进行了具体的描述，当然也没有必要那么做，所以对于那些没有描述的属性，浏览器将采 用内置默认的方式来进行显示，譬如文字，如果你没有在css中指定颜色，那么浏览器将采用黑色或者系统颜色来显示，div或者其他元素的背景，如果在 css中没有被指定，浏览器则将其设置为白色或者透明，等等其他未定义的样式均如此。所以有很多东西出现FF和IE显示不一样的根本原因在于它们的默认显 示不一样，而这个默认样式该如何显示我知道在w3中有没有对应的标准来进行规定，因此对于这点也就别去怪罪IE了。   
  
11.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; }   
  
12.FireFox下如何使连续长字段自动换行 众所周知IE中直接使用 word-wrap:break-word 就可以了, FF中我们使用JS插入&#10;的方法来解决 <style type="text/css"> <!-- div { width:300px; word-wrap:break-word; border:1px solid red; } --> </style> <div id="ff">aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa</div> <scrīpt type="text/javascrīpt"> /\* <![CDATA[ \*/ function toBreakWord(el, intLen){ var ōbj=document.getElementById(el); var strContent=obj.innerHTML; var strTemp=""; while(strContent.length>intLen){ strTemp+=strContent.substr(0,intLen)+"&#10;"; strContent=strContent.substr(intLen,strContent.length); } strTemp+="&#10;"+strContent; obj.innerHTML=strTemp; } if(document.getElementById && !document.all) toBreakWord("ff", 37); /\* ]]> \*/ </scrīpt>   
  
13.为什么IE6下容器的宽度和FF解释不同呢 <?xml version="1.0" encoding="gb2312"?> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "[http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd](http://www.douban.com/link2?url=http%3A//www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd)"> <meta http-equiv="Content-Type" content="text/html; charset=gb2312" /> <style type="text/css"> <!-- div { cursor:pointer; width:200px; height:200px; border:10px solid red } --> </style> <div ōnclick="alert(this.offsetWidth)">让FireFox与IE兼容</div> 问题的差别在于容器的整体宽度有没有将边框（border）的宽度算在其内,这里IE6解释为200PX ,而FF则解释为220PX,那究竟是怎么导致的问题呢？大家把容器顶部的xml去掉就会发现原来问题出在这,顶部的申明触发了IE的qurks mode,关于qurks mode、 standards mode的相关知识,请参考:http: //[www.microsoft.com/china/msdn/library/webservices/asp.net/](http://www.douban.com/link2?url=http%3A//www.microsoft.com/china/msdn/library/webservices/asp.net/) ASPNETusStan.mspx?mfr=true   
  
IE6,IE7,FF IE7.0 出来了，对CSS的支持又有新问题。浏览器多了，网页兼容性更差了，疲于奔命的还是我们 ，为解决IE7.0的兼容问题，找来了下面这篇文章：现在我大部分都是用!important来hack，对于ie6和firefox测试可以正常显示， 但是ie7对!important可以正确解释，会导致页面没按要求显示！下面是三个浏览器的兼容性收集.   
  
第一种，是CSS HACK的方法 height:20px; /\*For Firefox\*/ \*height:25px; /\*For IE7 & IE6\*/ \_height:20px; /\*For IE6\*/ 注意顺序。 这样也属于CSS HACK，不过没有上面这样简洁。 #example { color: #333; } /\* Moz \*/ \* html #example { color: #666; } /\* IE6 \*/ \*+html #example { color: #999; } /\* IE7 \*/   
  
<!--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]-->   
  
第三种，css filter的办法，以下为经典从国外网站翻译过来的。. 新建一个css样式如下： #item { width: 200px; height: 200px; background: red; } 新建一个div,并使用前面定义的css的样式： <div id="item">some text here</div> 在body表现这里加入lang属性,中文为zh： <body lang="en"> 现在对div元素再定义一个样式： \*:lang(en) #item{ background:green !important; } 这样做是为了用!important覆盖原来的css样式,由于:lang选择器ie7.0并不支持,所以对这句话不会有任何作用,于是也达到了 ie6.0下同样的效果,但是很不幸地的是,safari同样不支持此属性,所以需要加入以下css样式： #item:empty { background: green !important } :empty选择器为css3的规范,尽管safari并不支持此规范,但是还是会选择此元素,不管是否此元素存在,现在绿色会现在在除ie各版本以外的 浏览器上。 对IE6和FF的兼容可以考虑以前的!important 个人比较喜欢用

**10个 DIV+CSS 需要注意的问题**  
1. 检查HTML元素是否有拼写错误、是否忘记结束标记即使是老手也经常会弄错div的嵌套关系。可以用dreamweaver的验证功能检查一下有无错误。   
2. 检查CSS是否正确   
检查一下有无拼写错误、是否忘记结尾的 } 等。可以利用CleanCSS来检查 CSS的拼写错误。CleanCSS本是为CSS减肥的工具，但也能检查出拼写错误。   
3. 确定错误发生的位置   
假如错误影响了整体布局，则可以逐个删除div块，直到删除某个div块后显示恢复正常，即可确定错误发生的位置。   
4. 利用border属性确定出错元素的布局特性   
使用float属性布局一不小心就会出错。这时为元素添加border属性确定元素边界，错误原因即水落石出。   
5. float元素的父元素不能指定clear属性   
MacIE下假如对float的元素的父元素使用clear属性，四周的float元素布局就会混乱。这是MacIE的闻名的bug，倘若不知道就会走弯路。   
6. float元素务必指定width属性   
很多浏览器在显示未指定width的float元素时会有bug。所以不管float元素的内容如何，一定要为其指定width属性。另外指定元素时尽量使用em而不是px做单位。   
7. float元素不能指定margin和padding等属性   
IE在显示指定了margin和padding的float元素时有bug。因此不要对float元素指定margin和padding属性（可以在float元素内部嵌套一个div来设置margin和padding）。也可以使用hack方法为IE指定非凡的值。   
8. float元素的宽度之和要小于100%   
假如float元素的宽度之和正好是100%，某些古老的浏览器将不能正常显示。因此请保证宽度之和小于99%。   
9. 是否重设了默认的样式？   
某些属性如margin、padding等，不同浏览器会有不同的解释。因此最好在开发前首先将全体的margin、padding设置为0、列表样式设置为none等。   
10. 是否忘记了写DTD？   
假如无论怎样调整不同浏览器显示结果还是不一样，那么可以检查一下页面开头是不是忘了写下面这行DTD：   
< !DOCTYPE HTML PUBLIC \"-//W3C//DTD HTML 4.01 Transitional//EN\" \"[http://www.w3.org/TR/html4/loose.dtd/](http://www.douban.com/link2?url=http%3A//www.w3.org/TR/html4/loose.dtd/)">