div+css 浏览器兼容问题解决方法

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*/}

OK

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 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:e-xpression(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}

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:　　　　　　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　　　　　　　　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; }　　 p 对象中的内容　　　　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"> <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>

font-size：0；

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" />

删除框采用此做法

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">aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa</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)+" "; strContent=strContent.substr(intLen,strContent.length); } strTemp+=" "+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"> <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/ 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： some text here 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 个人比较喜欢用第一种，简洁，兼容性比较好。

分割线

div+css 网站布局设计常见错误大全

div+css 是网站设计标准（或称“WEB 标准”）中常用的术语之一，通常为了说明与 HTML 网页设计语言中的表格（table）定位方式的区别，因为 XHTML 网站设计标准中，不再使用表格定位技术，而是采用 css+div 的方式实现各种定位。应用应用 DIV+CSS 编码时很容易犯一些错误。本文列举了一些常见的错误：

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

如果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 下如果对 float 的元素的父元素使用 clear 属性，周围的 float 元素布局就会混乱。这是 MacIE 的著名的 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 指定特别的值。

8. float 元素的宽度之和要小于 100%

如果 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?

如果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。

DIV CSS 完美兼容 IE6/IE7/FF 的通用方法
关于 CSS 对各个浏览器兼容已经是老生常谈的问题了, 网络上的教程遍地都是.以下内容没有太多新颖, 纯属个人总结, 希望能对初学者有一定的帮助.

一、CSS HACK

以下两种方法几乎能解决现今所有 HACK.

1, !important

随着 IE7 对!important 的支持, !important 方法现在只针对 IE6 的 HACK.(注意写法,记得该声明位置需要提前.)

以下为引用的内容：

```
<style>
#wrapper
{
width: 100px!important; /* IE7+FF */
width: 80px; /* IE6 */
}
</style>
```

2, IE6/IE77 对 FireFox

以下为引用的内容：

*+html 与 *html 是 IE 特有的标签, firefox 暂不支持.而*+html 又为 IE7 特有标签.

```
<style>
#wrapper
{
#wrapper { width: 120px; } /* FireFox */
*html #wrapper { width: 80px;} /* ie6 fixed */
*+html #wrapper { width: 60px;} /* ie7 fixed, 注意顺序 */
}
</style>
```

注意:

*+html 对 IE7 的 HACK 必须保证 HTML 顶部有如下声明：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

二、万能 float 闭合

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.

以下为引用的内容：

```
<style>
/* 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 */
</style>
```

三、其他兼容技巧

1, FF 下给 div 设置 padding 后会导致 width 和 height 增加, 但 IE 不会.(可用!important 解决)
2, 居中问题.
1).垂直居中.将 line-height 设置为 当前 div 相同的高度, 再通过 vertical-align: middle.(注意内容不要换行.)
2).水平居中. margin: 0 auto;(当然不是万能)
3, 若需给 a 标签内内容加上 样式, 需要设置 display: block;(常见于导航标签)
4, FF 和 IE 对 BOX 理解的差异导致相差 2px 的还有设为 float 的 div 在 ie 下 margin 加倍等问题.
5, ul 标签在 FF 下面默认有 list-style 和 padding . 最好事先声明, 以避免不必要的麻烦. (常见于导航标签和内容列表)
6, 作为外部 wrapper 的 div 不要定死高度, 最好还加上 overflow: hidden.以达到高度自适应.
7, 关于手形光标. cursor: pointer. 而 hand 只适用于 IE.

1 针对 firefox ie6 ie7 的 css 样式
现在大部分都是用!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！找到一个针
对 IE7 不错的 hack 方式就是使用“*+html”，现在用 IE7 浏览一下，应该没有问题了。
现在写一个 CSS 可以这样：

以下为引用的内容：

```
#1 { color: #333; } /* Moz */
* html #1 { color: #666; } /* IE6 */
```

+html #1 { color: #999; } / IE7 */

那么在 firefox 下字体颜色显示为#333，IE6 下字体颜色显示为#666，IE7 下字体颜色显示为#999。

2 css 布局中的居中问题
主要的样式定义如下：

body {TEXT-ALIGN: center;}
#center { MARGIN-RIGHT: auto; MARGIN-LEFT: auto; }
说明：
首先在父级元素定义 TEXT-ALIGN: center;这个的意思就是在父级元素内的内容居中；对于 IE 这样设定就已经可以了。
但在 mozilla 中不能居中。解决办法就是在子元素定义时候设定时再加上“MARGIN-RIGHT: auto;MARGIN-LEFT: auto; ”
需要说明的是，如果你想用这个方法使整个页面要居中，建议不要套在一个 DIV 里，你可以依次拆出多个 div，
只要在每个拆出的 div 里定义 MARGIN-RIGHT: auto;MARGIN-LEFT: auto; 就可以了。

3 盒模型不同解释

#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-}

4 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,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; //实现同一行排列的的效果 diplay:table;

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;}

6 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 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。

7 清除浮动

.hackbox{ display:table; //将对象作为块元素级的表格显示}或者.hackbox{ clear:both;}
或者加入:after（伪对象）,设置在对象后发生的内容，通常和 content 配合使用，IE 不支持此伪对象，非 Ie 浏览器支持，
所 以并不影响到 IE/WIN 浏览器。这种的最麻烦的.....#box:after{ content: "."; display: block; height: 0; clear: both; visibility: hidden;}

8 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}
HTML 代码

9 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.

10 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。

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：

p 对象中的内容

CSS：#box {background-color:#eee; }
#box p {margin-top: 20px;margin-bottom: 20px; text-align:center; }
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。

六、CSS 兼容要点分析 IE vs FF

CSS 兼容要点：

DOCTYPE 影响 CSS 处理

FF: div 设置 margin-left, margin-right 为 auto 时已经居中, IE 不行

FF: body 设置 text-align 时, div 需要设置 margin: auto(主要是 margin-left,margin-right) 方可居中

FF: 设置 padding 后, div 会增加 height 和 width, 但 IE 不会, 故需要用 !important 多设一个 height 和 width

FF: 支持 !important, IE 则忽略, 可用 !important 为 FF 特别设置样式

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

cursor: pointer 可以同时在 IE FF 中显示游标手指状， hand 仅 IE 可以

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 XHTML+CSS 兼容性解决方案小集

使用 XHTML＋CSS 构架好处不少，但也确实存在一些问题，不论是因为使用不熟练还是思路不清晰，我就先把一些我遇到的问题写在下面，省的大家四处找^^

1、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;

2、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}

，关于这个/**/是什么我也不太明白，只知道 IE5 和 firefox 都支持但 IE6 不支持，如果有人理解的话，请告诉我一声，谢了！:）

3、ul 标签在 Mozilla 中默认是有 padding 值的,而在 IE 中只有 margin 有值所以先定义

ul{margin:0;padding:0;}

就能解决大部分问题

4、关于脚本，在 xhtml1.1 中不支持 language 属性，只需要把代码改为

< type="text/java">

就可以了

七、10 个你未必知道的 CSS 技巧

1、CSS 字体属性简写规则

一般用 CSS 设定字体属性是这样做的：

以下为引用的内容：
font-weight:bold;

font-style:italic;

font-varient:small-caps;

font-size:1em;

line-height:1.5em;

font-family:verdana,sans-serif;

但也可以把它们全部写到一行上去：

font: bold italic small-caps 1em/1.5em verdana,sans-serif;

真不错！只有一点要提醒的：这种简写方法只有在同时指定 font-size 和 font-family 属性时才起作用。而且，如果你没有设定 font-weight, font-style, 以及 font-varient ，他们会使用缺省值，这点要记上。

2、同时使用两个类

一般只能给一个元素设定一个类（Class），但这并不意味着不能用两个。事实上，你可以这样：

...

同时给 P 元素两个类，中间用空格格开，这样所有 text 和 side 两个类的属性都会加到 P 元素上来。如果它们两个类中的属性有冲突的话，后设置的起作用，即在 CSS 文件中放在后面的类的属性起作用。

补充：对于一个 ID，不能这样写 ... 也不能这样写

3、CSS border 的缺省值

通常可以设定边界的颜色，宽度和风格，如：

border: 3px solid #000

这位把边界显示成 3 像素宽，黑色，实线。但实际上这里只需要指定风格即可。

如果只指定了风格，其他属性就会使用缺省值。一般地，Border 的宽度缺省是 medium，一般等于 3 到 4 个像素；缺省的颜色是其中文字的颜色。如果这个值正好合适的话，就不用设那么多了。

4、CSS 用于文档打印

许多网站上都有一个针对打印的版本，但实际上这并不需要，因为可以用 CSS 来设定打印风格。

也就是说，可以为页面指定两个 CSS 文件，一个用于屏幕显示，一个用于打印：

<link type="text/css" rel="stylesheet" href="/blog/stylesheet.css" media="screen" /> <link

type="text/css" rel="stylesheet" href="printstyle.css" media="print" />

第 1 行就是显示，第 2 行是打印，注意其中的 media 属性。

但应该在打印 CSS 中写什么东西呢？你可以按设计普通 CSS 的方法来设定它。设计的同时就可以把这个 CSS 设成显示 CSS 来检查它的效果。也许你会使用 display: none 这个命令来关掉一些装饰图片，再关掉一些导航按钮。要想了解更多，可以看“打印差异”这一篇。

5、图片替换技巧

一般都建议用标准的 HTML 来显示文字，而不要使用图片，这样不但快，也更具可读性。但如果你想用一些特殊字体时，就只能用图片了。

比如你想整个卖东西的图标，你就用了这个图片：

<h1></h1>

这当然可以，但对搜索引擎来说，和正常文字相比，它们对 alt 里面的替换文字几乎没有兴趣这是因为许多设计者在这里放许多关键词来骗搜索引擎。所以方法应该是这样的：

<h1>Buy widgets</h1>

但这样就没有特殊字体了。要想达到同样效果，可以这样设计 CSS：

h1 { background: url(/blog/widget-image.gif) no-repeat; height: image height text-indent: -2000px }

注意把 image height 换成真的图片的高度。这里，图片会当作背景显示出来，而真正的文字由于设定了-2000 像素这个缩进，它们会出现在屏幕左边 2000 点的地方，就看不见了。但这对于关闭图片的人来说，可能全部看不到了，这点要注意。

6、CSS box 模型的另一种调整技巧

这个 Box 模型的调整主要是针对 IE6 之前的 IE 浏览器的，它们把边界宽度和空白都算在元素宽度上。比如：

#box { width: 100px; border: 5px; padding: 20px }

这样调用它：

...

这时盒子的全宽应该是 150 点，这在除 IE6 之前的 IE 浏览器之外的所有浏览器上都是正确的。但在 IE5 这样的浏览器上，它的全宽仍是 100 点。可以用以前人发明的 Box 调整方法

来处理这种差异。

但用 CSS 也可以达到同样的目的，让它们显示效果一致。

#box { width: 150px } #box div { border: 5px; padding: 20px }

这样调用：

...

这样，不管什么浏览器，宽度都是 150 点了。

7、块元素居中对齐

如果想做个固定宽度的网页并且想让网页水平居中的话，通常是这样：

#content { width: 700px; margin: 0 auto }

你会使用 来围上所有元素。这很简单，但不够好，IE6 之前版本会显示不出这种效果。改 CSS 如下：

body { text-align: center } #content { text-align: left; width: 700px; margin: 0 auto }

这会把网页内容都居中，所以在 Content 中又加入了

text-align: left 。

8、用 CSS 来处理垂直对齐

垂直对齐用表格可以很方便地实现，设定表格单元 vertical-align: middle 就可以了。但对 CSS 来说这没用。如果你想设定一个导航条是 2em 高，而想让导航文字垂直居中的话，设定这个属性是没用的。

CSS 方法是什么呢？对了，把这些文字的行高设为 2em：line-height: 2em ，这就可以了。

9、CSS 在容器内定位

CSS 的一个好处是可以把一个元素任意定位，在一个容器内也可以。比如对这个容器：

#container { position: relative }

这样容器内所有的元素都会相对定位，可以这样用：

...

如果想定位到距左 30 点，距上 5 点，可以这样：

#navigation { position: absolute; left: 30px; top: 5px }

当然，你还可以这样：

margin: 5px 0 0 30px

注意 4 个数字的顺序是：上、右、下、左。当然，有时候定位的方法而不是边距的方法更好些。

10、直通到屏幕底部的背景色

在垂直方向是进行控制是 CSS 所不能的。如果你想让导航栏和内容栏一样直通到页面底部，用表格是很方便的，但如果只用这样的 CSS：

#navigation { background: blue; width: 150px }

较短的导航条是不会直通到底部的，半路内容结束时它就结束了。该怎么办呢？

不幸的是，只能采用欺骗的手段了，给这较短的一栏加上个背景图，宽度和栏宽一样，并让它的颜色和设定的背景色一样。

body { background: url(/blog/blue-image.gif) 0 0 repeat-y }

此时不能用 em 做单位，因为那样的话，一旦读者改变了字体大小，这个花招就会露馅，只能使用 px。

HTML 的 SPAN 和 DIV 的区别
SPAN 和 DIV 的区别在于，DIV(division)是一个块级元素，可以包含段落、标题、表格，乃至诸如章节、摘要和备注等。而 SPAN 是行内元素，SPAN 的前后是不会换行的，它没有结构的意义，纯粹是应用样式，当其他行内元素都不合适时，可以使用 SPAN。

SPAN 标记有一个重要而实用的特性，即它什么事也不会做，它的唯一目的就是围绕你的 HTML 代码中的其它元素，这样你就可以为它们指定样式了。在此例中，标识符允许你将一个段落分成不同的部分。　　还有一个标识符具有类似的功能， DIV 也被用来在 HTML 文件中建立逻辑部分。但与 SPAN 不同， 工作于文本块一级，它在它所包含的 HTML 元素的前面及后面都引入了行分隔。

SPAN 标记有一个重要而实用的特性，即它什么事也不会做，它的唯一目的就是围绕你的 HTML 代码中的其它元素，这样你就可以为它们指定样式了。在此例中，标识符允许你将一个段落分成不同的部分。　　还有一个标识符具有类似的功能， DIV 也被用来在 HTML 文件中建立逻辑部分。但与 SPAN 不同， 工作于文本块一级，它在它所包含的 HTML 元素

的前面及后面都引入了行分隔。

div 布局时不能居中、居中失灵的解决
一般情况下，div 不能居中，div 居中失灵是因为页面没有声明引起的：加上<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
即可

使用 margin-left:auto;margin-right:auto; 可以让你的 div 居中对齐。如
.style{margin-left:auto;margin-right:auto;}
缩写形式为：
.style{margin:0 auto;}
数字 0 表示上下边距是 0。可以按照需要设置成不同的值。

查看下面的例子：

以下为引用的内容：

```
<!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" lang="zh-cn">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>居中 div 演示效果</title>
<style type="text/css">
.align-center{
      margin:0 auto;          /* 居中 这个是必须的,, 其它的属性非必须 */
       width:500px;             /* 给个宽度 顶到浏览器的两边就看不出居中效果了 */
       background:red;         /* 背景色 */
       text-align:center;    /* 文字等内容居中 */
}
</style>
</head>

<body>
<div class="align-center">居中 div 演示效果</div>
</body>
</html>
```

注意，需要加上上面的那句

才能生效，要是不想要这句的话，也可以给 body 加一个属性：

body{text-align:center;}

另外，让 div 内的内容（包括文字及图片）居中的代码是： text-align:center;

解决 Flash 在 Div 中无法居中
以下为引用的内容：

```
<style>
/* -----页面统一设置----- */
html,body{ margin:0px; padding:0px; background:url(images/boydBg.gif) repeat-x;}
img{ border:0px; }
/* -----开始顶部统一设置----- */
#head{ width:1000px; height:364px; text-align:center; }
#logo{ height:110px; background-color:#0099CC; background:url(images/home01.jpg) repeat-x;
text-align:right; padding-right:420px;}
#nav{ height:54px;}
#banner{ width:950px; height:200px; text-align:center;}
/* -----顶部设置完毕----- */

</style>
</head>

<body>
<!-- 顶部开布局 -->
<div id="head">
    <div id="banner"><iframe marginwidth="0" marginheight="0" src="images/banner.swf"
frameborder="0" width="950" scrolling="No" height="200"></iframe></div>
</div>
```

#banner{margin:0 auto;} -----用这个动画是距中了

#head{margin:0 auto;} ------用这个整个页面都距中啦~~

FireFox 和 IE 浏览器对于 DIV+CSS 设计的区别及对策

1、实际像素
IE/Opera：对象的实际宽度 = (margin-left) + width + (margin-right)
Firefox/Mozilla：对象的实际宽度= (margin-left) + (border-left-width) + (padding- left) + width + (padding-right) + (border-right-width) + (margin-right)
所以排列好及列的表格时 ie 和 ff 显示宽度稍有区别

2、水平居中
问题：div 里的内容，ie 默认为 center，而 ff 默认 left

解决：mairgin:0px auto;

3、高度问题
问题：如果设置了一个 DIV 的高度，当 DIV 里实际内容大于所设高度，ie 会自动拉伸以适应 DIV 容器大小，ff 会固定 DIV 的告诉，超过部分超出 DIV 底线以外，出现和下面的内容重叠的现象
解决：控制恰当的高度，或者不写，让浏览器自动调节高度，或者设置 overflow:hidden;

4、clear:both;
问题：如果上面用 float 控制了 n 列 DIV，下面 ie 会自动检测自动排列，ff 则可能很不老实，到处乱动
解决：float 结束后的下一个标签加 clear:both;以结束 float 的控制

5、最大/小宽度问题
问题：min-width,max-width 只是 ff 的命令，如何让 ie 实现同样的效果
解决：ie 不认识 min-和 max-，实际 ie 认为 min-width、max-width 和 width 效果一样，可以用下面方法解决
#cctext{
min-width: 700px;
max-width: 1000px;
width:expression(document.body.clientWidth<700 ? "700px" : document.body.clientWidth>1000 ? "1000px" : "auto");
}

6、!important 支持
问题：ff 支持 ie6.0 不支持
解决：无。ie 会忽略。

7、游标状态
问题：cursor:hand;仅 ie 支持，显示手指状态
解决：使用 cursor:pointer;ie 和 ff 都支持

8、单位问题
问题：任何距离的数值 ie 可以不加单位，ff 必须要求写单位（0 除外）
解决：写全单位如 padding:0px;

查看文章
/zz/div+css - Firefox 和 IE 浏览器兼容问题 - padding-right 在 IE6 下导致抖动 2008-01-11 17:33
由于 IE6 的盒模型计算缺陷，padding-right 在特定的情况会下导致抖动。
[出现抖动的代码]

最近在写一个 Div+Css 布局的网站首页，以前写的页面都统一 width:900px;不存在什么布局呀 什么 float 的设置。现在没办法呀！要对页面进行切割，分块。幸好，有点 css 的基础，

不过用起来就比较郁闷了。在 ie 和 firefox 下，相同的属性值往往会有不同的显示效果。没办法 要兼容浏览器 只得一个一个 bug 去找。但完全兼容浏览器是无法做到的，所以也只能将个大概的效果展示出来，不要偏差太大太明显就行。
首先，需要提出的是 float 这个浮动属性，这是 div+css 布局的关键所在。float:left;float:right;是常用的浮动属性。为了使 div 能在一行排列，不得补用到它们。呵呵！不就是个 float 吗？这有什么好提的，要浮动我就 float 一下呀。哎！话是这么说，但真正用到时却出问题了。在 ie 下，只要前一个 div 有 float:left;后面的 div 宽度不超过（总 body 的宽度）-（前一个 div 的宽度），后面的 div 就自动浮动，并排列在同一行。ie 和 firefox 下，这点效果是一样。好，接着往下，下一行也这样布局，就分两栏吧。

```
<style type="text/css">
#div1{}{width:200px;height:80px;float:left;border:1px solid blue;}
#div2{}{width:600px;height:80px;border:1px solid blue;}
</style>
<div>
<div id="div1"></div>
<div id="div2"></div>
</div>
<div style="clear:both;margin-top:20px;" id="div3">
</div>
```

这样的布局，在 ie 下和火狐下的显示效果就不同了 在 firefox 下的 margin-top:20px;没效果。。。。

在 ie 下 div3 的 margin-top 有效果，而在 firefox 下却没效果。为什么？翻来覆去的想，终究没有想明白。没办法了，后来我有改 css.偶然，给 div2 加上了 float:left 属性，然后 margin-top:20px 都有效果了。郁闷吧 呵呵 不过还是没弄明白为什么会这样，问题解决了就 ok 了！

下一个问题是 ul 的问题。在 ie 下和 firefox 下注意了 ul{margin-left:10px;}的效果是不同的，这样设置的效果才会相同
ul{padding:0;margin-left:10px;}在火狐下默认的 padding!=0 所以需要这样来设置下 ...
呵呵 愚昧的个人见解!
div+css 扩展框问题浮动下降 IE 与 Firefox 兼容性整理 2008-08-14 10:57div+css 扩展框问题浮动下降 IE 与 Firefox 兼容性整理
图文混排 容易导致 扩展框问题.

```
<div><img src="images/index_1.jpg" />扩展框问题</div>
```

这样排版容易导致 扩展框问题.
尽量定义宽高给定值

* 浮动下降问题[size=+0]
加上 {float：left;} 即可```
IE6 的双倍边距 BUG
解决办法是加上 display:inline
IE6 下为什么图片下方有空隙产生
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.
IE6 下这两个层中间怎么有间隙
这个 IE 的 3PX BUG 也是经常出现的，解决的办法是给.right 也同样浮动 float:left 或者相对 IE6 定义.left
如何对齐文本与文本输入筐
遇到此种问题，设置文本框的 vertical-align:middle 就可以了
为什么 FF 下文本无法撑开容器的高度
[size=+0]
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;
}
```

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 value="transparent" />或者<param value="opaque" />

怎样使一个层垂直居中于浏览器中

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.

方法二:首先在父级元素定义 TEXT-ALIGN: center;这个的意思就是在父级元素内的内容居中；对于 IE 这样设定就已经可以了。
但在 mozilla 中不能居中。解决办法就是在子元素定义时候设定时再加上“MARGIN-RIGHT: auto;MARGIN-LEFT: auto; ”
需要说明的是，如果你想用这个方法使整个页面要居中，建议不要套在一个 DIV 里，你可以依次拆出多个 div，
只要在每个拆出的 div 里定义 MARGIN-RIGHT: auto;MARGIN-LEFT: auto; 就可以了。

```
{
width:300px;
margin-left:auto;
margin-right:auto;
}
```

针对 firefox ie6 ie7 的 css 样式

现在大部分都是用!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！找到一个针
对 IE7 不错的 hack 方式就是使用“*+html”，现在用 IE7 浏览一下，应该没有问题了。
现在写一个 CSS 可以这样：

#1 { color: #333; } /* Moz */
* html #1 { color: #666; } /* IE6 */
+html #1 { color: #999; } / IE7 */
那么在 firefox 下字体颜色显示为#333，IE6 下字体颜色显示为#666，IE7 下字体颜色显示为#999。

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 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。

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.

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.
关于容器的包涵关系
很多时候，尤其是容器内有平行布局，例如两、三个 float 的 div 时，宽度很容易出现问题。在 IE 中，外层的宽度会被内层更宽的 div 挤破。一定要用 Photoshop 或者 Firework 量取像素级的精度。

1.写两句代码来控制一个属性，区别 Firefox 与 IE：
backgroundrange; *background:green;
//这一句代码写出来时，你用 Firefox 浏览，会发现背景是橙色的，而 IE 里却是绿色的，很简单，因为 Firefox 不能识别*，而 IE6，IE7 都可以识*,标准浏览器(如 Firefox,Opera,Netscape)

不能识别*;。
2.写两句代码来控制一个属性，区别 IE7 与 IE6：
background:green !important;background:blue;
//这一句代码写出来时，你用 IE7 浏览，会发现，写了该代码的区域背景是绿色的，如果用 IE6 浏览，却是蓝色的，这是因为 IE7 能识别!important*，一但识别了，就执行，忽略了后面的那一句，但 IE6 却不能识别!important，所以前面部分跳过，直接执行了后半部份.
3.写三句代码来控制一个属性，区别 Firefox，IE7，IE6：
backgroundrange;*background:green !important;*background:blue;
//这一句会使在 Firefox 在，背景呈橙色，IE7 中为绿色，IE6 中为蓝色，道理和前面一样，Firefox 不能识别*，所以后面两句都不执行，直接执行第一句，IE7 当然也能执行第一行代码，但是因为第二句，他也能识别，所以就执行了第二句代码，把前面的效果给过滤了，而最后一句，IE7 是不能识别的。IE6 不能识别!imprtant，本来运行了第一句代码了，第二句不能识别，那就理所当然的执行了最后一句。
注：IE 都能识别*;标准浏览器(如 Firefox,Opera,Netscape)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;

div+css 在 FireFox 里居中的问题。2008-07-15 15:31 尽管有 CSS 的 vertical-align 特性，但是并不能有效解决未知高度的垂直居中问题（在一个 DIV 标签里有未知高度的文本或图片的情况下）。

标准浏览器如 Mozilla, Opera, Safari 等.，可将父级元素显示方式设定为 TABLE(display: table;) ，内部子元素定为 table-cell (display: table-cell),通过 vertical-align 特性使其垂直居中，但非标准浏览器是不支持的。
非标准浏览器只能在子元素里设距顶部 50%，里面再套个元素距顶部-50% 来抵消。

解决代码如下：
CSS:

```
body {padding: 0; margin: 0;}body,html{height: 100%;}
#outer {height: 100%; overflow: hidden; position: relative;width: 100%; background:ivory;}
#outer[id] {display: table; position: static;}
#middle {position: absolute; top: 50%;} /* for explorer only*/
#middle[id] {display: table-cell; vertical-align: middle; position: static;}
#inner {position: relative; top: -50%;width: 400px;margin: 0 auto;} /* for explorer only */
div.greenBorder {border: 1px solid green; background-color: ivory;}

<div id="outer">
<div id="middle">
<div id="inner" class="greenBorder">
</div>
</div>
</div>
```

以上 CSS 代码的优点是没有 hacks，采用了 IE 不支持的 CSS2 选择器#value[id]。

CSS2 选择器#value[id]相当于选择器#value，但是 Internet Explorer 不支持这种类型的选择器。同样地.value[class]，相当于.value,这些只有标准浏览器能读懂。

div+css 实现 Firefox 和 IE6 兼容的垂直居中 2008 年 03 月 10 日 星期一 02:06div+css 实现 Firefox 和 IE6 兼容的垂直居中

Firefox 中使用 display: table-cell; vertical-align: middle;可以实现 div 垂直居中，而 IE6 中则需要借助 IE6 中 css 的特点实现垂直居中。为了实现 Firefox 和 IE6 兼容的垂直居中，还需要 借助于!important 标记。Firefox 支持!important 标记，而 IE6 忽略!important 标记，因此可以使用! important 标记区别 Firefox 和 IE6。

[示例代码]

```
<html>
    <body>
        <div style="display: table-cell; vertical-align: middle; height: 200px; border: 1px solid red;">
            <p>垂直居中，Firefox only</p>
            <p>垂直居中，Firefox only</p>
            <p>垂直居中，Firefox only</p>
        </div>
        <div style="border: 1px solid red; height: 200px; position: relative;">
            <div style="position: absolute; top: 50%;">
                <div style="position: relative; top: -50%;">
                    <p>垂直居中，IE6 only</p>
                    <p>垂直居中，IE6 only</p>
                    <p>垂直居中，IE6 only</p>
                </div>
            </div>
        </div>
        <div style="border: 1px solid red; height: 200px; position: relative; display: table-cell; vertical-align: middle;">
            <div style="position: static !important; position: absolute; top: 50%;">
                <div style="position: relative; top: -50%;">
                    <p>垂直居中，Firefox IE6 only</p>
                    <p>垂直居中，Firefox IE6 only</p>
                    <p>垂直居中，Firefox IE6 only</p>
```

IE VS FireFox

CSS 兼容要点：DOCTYPE 影响 CSS 处理

FF: div 设置 margin-left, margin-right 为 auto 时已经居中, IE 不行

FF: body 设置 text-align 时, div 需要设置 margin: auto(主要是 margin-left,margin-right) 方可居中

FF: 设置 padding 后, div 会增加 height 和 width, 但 IE 不会, 故需要用 !important 多设一个 height 和 width

FF: 支持 !important, IE 则忽略, 可用 !important 为 FF 特别设置样式

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

cursor: pointer 可以同时在 IE FF 中显示游标手指状， hand 仅 IE 可以

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 XHTML+CSS 兼容性解决方案小集

使用 XHTML+CSS 构架好处不少，但也确实存在一些问题，不论是因为使用不熟练还是思路不清晰，我就先把一些我遇到的问题写在下面。

1.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 css xhtml xml Example Source Code Example Source Code [www.52css.com]
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 css xhtml xml Example Source Code Example Source Code [www.52css.com]
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;

2.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 css xhtml xml Example Source Code Example Source Code [www.52css.com]
div{width:300px!important;width /**/:340px;margin:0 10px 0 10px}

关于这个/**/是什么我也不太明白，只知道 IE5 和 firefox 都支持但 IE6 不支持，如果有人理解的话，请告诉我一声，谢了!:）

3.ul 标签在 Mozilla 中默认是有 padding 值的,而在 IE 中只有 margin 有值所以先定义

div css xhtml xml Example Source Code Example Source Code [www.52css.com]
ul{margin:0;padding:0;}

就能解决大部分问题

4.关于脚本，在 xhtml1.1 中不支持 language 属性，只需要把代码改为

div css xhtml xml Example Source Code Example Source Code [www.52css.com]

div+css 编辑网页 firefox 下错位的解决方法{clear 和 float 属性}好不容易编辑好的网页导航，在 FIREFOX 下测试发现错位的厉害，毫无美感可言{原因找出，原来是当时测试的一个代码，测试完了忘记改了过来，额的个神，用了我一个多小时}

没办法，做为使用比较多的浏览器之一，不能不考虑适应它。出现错位混乱的原因主要是因为 IE 和 FIREFOX 的标准不同，某些属性的释义出现相差所致。

没办法只好边学边用了。clear 和 float 属性
在 css 文件里定义的容器时加入 clear 属性来控制页面
clear 属性说明：该属性的值指出了不允许有浮动对象的边。
注意这里要注意和 float 属性的区分
float：该属性的值指出了对象是否及如何浮动。
当该属性不等于 none 引起对象浮动时，对象将被视作块对象(block-level)，即 display 属性等于 block。也就是说，浮动对象的 display 特性将被忽略。
clear 属性支持的参数有下面 4 个：
none : 允许两边都可以有浮动对象
both : 不允许有浮动对象
left : 不允许左边有浮动对象
right : 不允许右边有浮动对象

而 float 属性支持的参数有下面 3 个：
none : 对象不浮动
left : 对象浮在左边
right : 对象浮在右边 DOCTYPE 影响 CSS 处理

FF: div 设置 margin-left, margin-right 为 auto 时已经居中, IE 不行

FF: body 设置 text-align 时, div 需要设置 margin: auto(主要是 margin-left,margin-right) 方可居中

FF: 设置 padding 后, div 会增加 height 和 width, 但 IE 不会, 故需要用 !important 多设一个 height 和 width

FF: 支持 !important, IE 则忽略, 可用 !important 为 FF 特别设置样式

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

cursor: pointer 可以同时在 IE FF 中显示游标手指状， hand 仅 IE 可以

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

XHTML+CSS 兼容性解决方案小集
使用 XHTML＋CSS 构架好处不少，但也确实存在一些问题，不论是因为使用不熟练还是思路不清晰，我就先把一些我遇到的问题写在下面，省的大家四处找^^

1.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;

2.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}，关于这个/**/是什么我也不太明白，只知道 IE5 和 firefox 都支持但 IE6 不支持，如果有人理解的话，请告诉我一声，谢了！:）

3.ul 标签在 Mozilla 中默认是有 padding 值的,而在 IE 中只有 margin 有值所以先定义

ul{margin:0;padding:0;}就能解决大部分问题

4.关于脚本，在 xhtml1.1 中不支持 language 属性，只需要把代码改为

<script type="text/javascript">

1、div+css 如何使网站兼容不同字体

使用 utf-8 内码进行编写

(右键-编码-选择一种其它国家的编码-查看效果)

asp 代码如下:

```
<%
function chinese2unicode(Str)
dim i
dim Str_one
dim Str_unicode
for i=1 to len(Str)
Str_one=Mid(Str,i,1)
Str_unicode=Str_unicode&chr(38)
Str_unicode=Str_unicode&chr(35)
Str_unicode=Str_unicode&chr(120)
Str_unicode=Str_unicode& Hex(ascw(Str_one))
Str_unicode=Str_unicode&chr(59)
next
Response.Write Str_unicode
end function
%>
```

使用方法:

```
<%=chinese2unicode("显示的内容") %>
```

2、关于 div+css 兼容 IE 和 firefox 的问题

好不容易建立的 div+css 网页完全符合 W3C 标准。在 ie7 下显示非常完美，用 firefox 一测试，乖乖一探糊涂惨不忍睹。经过一番研究发现兼容很

简单。

1、增加　　div {overflow: hidden;},目的就是让 div 自动适应内容高度
2、横排的 div 外套 div

另外 设定

```
ul {
        margin: 0px;
        padding: 0px;
}
```

是消除 li 前面的空格

3、div+css 兼容性问题

CSS 兼容要点：DOCTYPE 影响 CSS 处理

FF: div 设置 margin-left, margin-right 为 auto 时已经居中, IE 不行

FF: body 设置 text-align 时, div 需要设置 margin: auto(主要是 margin-left,margin-right) 方可居中

FF: 设置 padding 后, div 会增加 height 和 width, 但 IE 不会, 故需要用 !important 多设一个 height 和 width

FF: 支持 !important, IE 则忽略, 可用 !important 为 FF 特别设置样式

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

cursor: pointer 可以同时在 IE FF 中显示游标手指状， hand 仅 IE 可以

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 XHTML+CSS 兼容性解决方案小集

使用 XHTML+CSS 构架好处不少，但也确实存在一些问题，不论是因为使用不熟练还是思路不清晰，我就先把一些我遇到的问题写在下面。

1.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 css xhtml xml Example Source Code Example Source Code [www.52css.com]
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 css xhtml xml Example Source Code Example Source Code [www.52css.com]
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;

2.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 css xhtml xml Example Source Code Example Source Code [www.52css.com]
div{width:300px!important;width /**/:340px;margin:0 10px 0 10px}

关于这个/**/是什么我也不太明白，只知道 IE5 和 firefox 都支持但 IE6 不支持，如果有人理解的话，请告诉我一声，谢了！:）

3.ul 标签在 Mozilla 中默认是有 padding 值的,而在 IE 中只有 margin 有值所以先定义

div css xhtml xml Example Source Code Example Source Code [www.52css.com]
ul{margin:0;padding:0;}

就能解决大部分问题

4.关于脚本，在 xhtml1.1 中不支持 language 属性，只需要把代码改为

div css xhtml xml Example Source Code Example Source Code [www.52css.com]
<script type="text/javascript">

4、div+css 兼容问题解决方案（IE5/5.5/6/7/FF）

之前找了几个不同版本的可独立运行的 IE 浏览器，正好拿来试试页面的兼容性问题。不试不知道，在 IE6 和 FF 中没问题的页面在 IE5 和 IE5.5 中乱

成一团，一直听说 IE5 是 WEB 标准制作的一个“钉子户”，现在不得不相信了。

既然有问题，那就找找解决的方法咯，在网上一搜，相关的文章还是不少的，觉得最直接的方法还是“IE 条件注释”，很方便的就能为 IE 的不

同版本写样式。但这样就得为每个版本写一个样式，不利于文件的优化。

找了一些相关的 CSS HACK 后，觉得应该可以把 IE5/IE5.5/IE6/FF 的 HACK 写到一起的，经过测试，终于找到了一个不错的方法，下面我们来看看

怎么实现：

大家都知道用!important 声明可以提升指定样式规则的应用优先权，如下面的例子：

E1{

background-color: red !important;/*提升优先权*/

background-color: blue;

}

但这样写在 IE 中会有个问题，看过我的《关于 CSS 样式表优先级》和《关于 CSS 样式表优先级补遗》，你会知道在 IE6 和 FF 中用! important 声明

可以提高优先级别，但在 IE6 中的!important 声明并不是绝对的，它会被之后的同名属性定义所替换。也就是说在上面的例子中，IE6 所应用的

是最后一个背景色的值，即“blue”；而在 FF 中背景色的值为“red”。根据这一点，我们就可以把 FF 和 IE 的样式分离开。

OK ，解决了 FF 和 IE 的问题，现在来解决 IE 自己的问题。

看过了嘟嘟的《绕过 IE6 支持 IE5 的别一种写法-IE 也支持">"》后有感而发，使用“＞”IE 是否真的可以认得？我们来看个例子：

E1{

background-color: red;

＞background-color: blue;

}

在 FF 中得到的是背景色红色，而在 IE 中得到的背景色是蓝色，根据样式重定义的规则，如果浏览器可以识别“＞”，则应该得到的蓝色的背景

，因此可以知道“＞”只有 IE 可以识别，这点是很重要的哦！在后面大家就会知道了。（注：我测试过其它的一些符号，如“~”、“`”、“

＜”等，都只有 IE 可以识别，在此为了感谢嘟嘟，推荐使用“＞”）

我们再来看个例子：

E1{

＞/*IE only*/background-color: black;

＞/*IE only*/background-color /*IE5.5*/: green;

}

这个例子在 IE6 中得到了黑色的背景；而在 IE5.5 中得到的绿色的背景；在 IE5 中也得到了黑色的背景。这就说明了第二句定义只有 IE5.5 能识别

，这是个很早就公布的 HACK，可以在网上找到相关的资料，要注意的就是在属性名之后是有一个空格的。到此我们已经把 FF、IE5.5、IE6 分离

出来了，那 IE5 呢？其实现在我们只要把 IE5 跟 IE6 分开就 OK 了，来看看例子：

E1{

＞/*IE only*/background-color: red;/*IE5*/

}

E1/*IE5.5+*/{

＞/*IE only*/background-color: black;

}

这里我们又用到一个 HACK，就是“E1/**/{}”，这个定义在 IE5 以上的版本才能识别出来。这个例子得到的结果是，在 IE5 中的背景色为红色；

在 IE5 以上版本中得到的是黑色背景。

终于把不同版本的浏览器都分离出来了，这样我们就可以为不同的浏览器定义不同的样式了。来看个完整的例子：

E1{

width: 500px;

height: 50px;

background-color: red !important;/*FF*/

background-color: blue;/*IE5*/

text-align:center;

}

E1/*IE5.5+*/{

＞/*IE only*/background-color: black;/*IE6*/

＞/*IE only*/background-color /*IE5.5*/: green;

}

需要注意的是，在上面例子中“background-color”定义的顺利不能改变，即 FF-IE5-IE6-IE5.5。对于 IE 的定义在属性前要加“＞”，因为

“E1/**/{}”这个 HACK 在 FF 中可以识别。也许你会想，上面的例子不是可以写成：

E1{

width: 500px;

height: 50px;

background-color: red;/*FF*/

＞background-color: blue;/*IE5*/

text-align:center;

}

E1/*IE5.5+*/{

＞/*IE only*/background-color: black;/*IE6*/

＞/*IE only*/background-color /*IE5.5*/: green;

}

这样不就又可以省下几个字节？是没错，可是 HACK 不是标准，如果滥用 HACK，那只会离标准越来越远！

6、div+css 实现 Firefox 和 IE6 兼容的垂直居中

Firefox 中使用 display: table-cell; vertical-align: middle;可以实现 div 垂直居中，而 IE6 中则需要借助 IE6 中 css 的特点实现垂直居中。为

了实现 Firefox 和 IE6 兼容的垂直居中，还需要 借助于!important 标记。Firefox 支持!important 标记，而 IE6 忽略!important 标记，因此可以使

用! important 标记区别 Firefox 和 IE6。

[示例代码]

```
<html>
    <body>
        <div style="display: table-cell; vertical-align: middle; height: 200px; border: 1px solid red;">
            <p>垂直居中，Firefox only</p>
            <p>垂直居中，Firefox only</p>
            <p>垂直居中，Firefox only</p>
        </div>
        <div style="border: 1px solid red; height: 200px; position: relative;">
            <div style="position: absolute; top: 50%;">
                <div style="position: relative; top: -50%;">
                    <p>垂直居中，IE6 only</p>
                    <p>垂直居中，IE6 only</p>
                    <p>垂直居中，IE6 only</p>
                </div>
            </div>
        </div>
        <div style="border: 1px solid red; height: 200px; position: relative; display: table-cell; vertical-align: middle;">
            <div style="position: static !important; position: absolute; top: 50%;">
                <div style="position: relative; top: -50%;">
                    <p>垂直居中，Firefox IE6 only</p>
                    <p>垂直居中，Firefox IE6 only</p>
                    <p>垂直居中，Firefox IE6 only</p>
                </div>
            </div>
        </div>
    </body>
</html>
```

7、div+css 的浏览器兼容问题

水平居中，Firefox 使用 margin-left: auto; margin-right: auto; IE6 使用 text-align: center;

垂直居中，Firefox 中使用 display: table-cell; vertical-align: middle;可以实现 div 垂直居中，而 IE6 中则需要借助 IE6 中 css 的特点实现垂

直居中。

!important 标记，Firefox 支持!important 标记，IE6 忽略!important 标记

使用 div+css 建站的朋友一定都知道 CSS 兼容的问题。由其是初学者在 IE 环境下好不容易写出了一个像模像样的网站来，一拿到 FF 下就变样了。所以在这里建议新手写代码时切忌要在 IE 和 FF 两个环境下去测试。今天教大家五招方法完全可以解决浏览器兼容问题。

这里大家要知道 CSS 不兼容的原因是因为各浏览器给 CSS 默认属性值不一样造成的。

第一招：给常用 CSS 规定属性值。

```
body,div,dl,dt,dd,ol,h1,h2,h3,h4,h5,h6,form,input,p,th,td{margin:0;padding:0;}
img{border:0px;}
ul {margin:0px;padding:0px;}/
ul li {list-style:none;}
```

上面的建站常用代码就相是格式化 CSS 样式，让各浏览器按照我们设置的属性值渲染网页

第二招：IE 和 FF 下对象居中问题

IE 下大家应该知道只要设置 body{text-align:center;}这样就可以居中显示。

但是这样的方法在 FF 不行的。这里就需要给修改成 body:{text-align:center;margin:0px auto;}Margin 的意思就是上下距离为 0 像素，左右为自动。所以 FF 就会居中显示。

第三招：垂直居中（仅只用于一行）

比如说一个高 30px 的 div，问题默认是会显示在左上角，如果想垂直居中对其可以加个 line-height:30px;样式。如果你想让他居下方则在修改 line-height:30px;数值越大越局下，为了防止撑破层，还需要再给一个样式 overflow:hidden;(超出的部分不显示)

第四招：给每一个块对象设置三个样式

width:**px;height:**px;overflow:hidden;即便高、宽是属性值是自动那么也需要去设置这三个样式。目的就是解决浏览器默认值的问题。

第五招：针对 IE6、IE7、FF 的 css 样式(这一招在特殊情况下经常用到)

原来建设网站经常使用!important 来设置优先权，但有了 IE7 之后就不行了。下面给大家个可以解决 IE6、IE7、FF 各个 CSS 优先权的方法

```
#1 { color: #333; } /* FF 环境  */
* html #1 { color: #666; } /* IE6 环境  */
*+html #1 { color: #999; } /* IE7 环境  */
```

上面的书写顺序一定不能去改变。

这样子网页在 FF 下显示#333，IE6 下显示#666，IE7 下显示#999;