**1. 默认的内外边距不同**

问题：

各个浏览器默认的内外边距不同

解决：

\*{margin:0;padding:0;}

**2. 水平居中的问题**

问题：

**设置** text-align: center

ie6-7文本居中，嵌套的块元素也会居中

ff /opera /safari /ie8文本会居中，嵌套块不会居中

解决：

块元素设置

1、margin-left:auto;margin-right:auto

2、margin:0 auto;

3、<div align=”center”></div>

**3. 垂直居中的问题**

**问题：**

在浏览器中想要垂直居中，设置vertical-align:middle; 不起作用。**例如：**ie6下**文本与文本输入框对不齐，需设置**vertical-align:middle，但是文本框的内容不会垂直居中

**解决：**

给容器设置一个与其高度相同的行高

line-height:与容器的height一样

**4. 关于高度问题**

问题：

如果是动态地添加内容，高度最好不要定义。浏览器可以自动伸缩，然而如果是静态的内容，高度最好定好。

如果设定了高度，内容过多时，ie6下会自动增加高度、其他浏览器会超出边框

解决：

1.设置overflow:hidden;

2.高度自增height:auto!important;height:100px;

**5. IE6 默认的div高度**

问题：

ie6默认div高度为一个字体显示的高度，所在ie6下div的高度大于等于一个字的高度，因此在ie6下定义高度为1px的容器，显示的是一个字体的高度

解决：

**为这个容器设置下列属性之一**

1、设置overflow:hidden;

2、设置line-height:1px;

3、设置zoom:0.08

**6. IE6 最小高度(宽度)的问题**

**问题：**

ie6不支持min-height、min-width属性，默认height是最小高度，width是最小宽度。

**解决：**

使用ie6不支持但其余浏览器支持的属性!important。

设置属性min-height:200px; height:auto !important; height:200px;

**7. td高度的问题**

问题：

table中td的宽度都不包含border的宽度，但是oprea和ff中td的高度包含了border的高度

解决：

设置line-height和height一样。在ie中如果td中的没有内容，那么border将不会显示

**8. div嵌套p时，出现空白行**

问题：

div中显示<p>文本</p>，ff、oprea、Chrome：top和bottom都会出现空白行，但是在ie下不会出现空白行。

解决：

设置p的margin:0px，再设置div的padding-top和padding-bottom

**9. IE6-7图片下面有空隙的问题**

**问题：**

块元素中含有图片时，ie6-7中会出现图片下有空隙

**解决：**

1、在源代码中让</div>和<img>在同一行

2、**将图片转换为块级对象**display:block;

3、**设置图片的垂直对齐方式** vertical-align:top/middle/bottom

**4、改变父对象的属性，**如果父对象的宽、高固定，图片大小随父对象而定，那么可以对父元素设置： overflow:hidden;

**5、设置图片的浮动属性**  float:left;

**10. IE6双倍边距的问题**

**问题：**

**ie6中设置浮动，同时又设置margin时，会出现双倍边距的问题**

**例**float:left;width:100px;margin:0 100px;

**解决：**

**设置**display:inline;

**11. IE6 width为奇数，右边多出1px的问题**

**问题：**

父级元素采用相对定位，且宽度设置为奇数时，子元素采用绝对定位，在ie6中会出现右侧多出1像素

**解决：**

将宽度的奇数值改成偶数

**12. IE6两个层之间3px的问题**

**问题：**

**左边层采用浮动，右边没有采用浮动，这时在ie6中两层之间就会产生3像素的间距**

**解决：**

1、右边层也采用浮动 float

2、左边层添加属性 margin-right:-3px;

**13. IE6 子元素绝对定位的问题**

**问题：**

**父级元素使用padding后，子元素使用绝对定位，不能精确定位**

**解决：**

在子元素中设置 \_left:-20px; \_top:-1px;

**14. 显示手型cursor:hand**

**问题：**

**ie6/7/8、opera 都支持**

**但是safari 、 ff 不支持**

**解决：**

**写成 cursor:pointer; (所有浏览器都能识别)**

**15. IE6-7 line-height失效的问题**

**问题：**

**在ie中img与文字放一起时， line-height不起作用**

**解决：**

都设置成float

**16. td自动换行的问题**

问题：

Table宽度固定，td自动换行

解决：

设置Table的table-layout:fixed，td的word-wrap:break-word

**17. 子容器浮动后，父容器扩展问题**

问题：

子容器都float以后，父容器没有设定高度,父容器将不会扩展

解决：

只需要添加一个clear:both的div，代码如下：

<div style="border:1px solid#333;width:204px">

<divstyle="width:100px;border:1px solid #333; float:left; ">子容器a</div>

<divstyle="width:100px;border:1px solid #333; float:left;">子容器b</div>

<divstyle="clear:both"></div>

</div>

**18. 透明png图片会带背景色**

问题：

在ie6下透明的png图片会带一个背景色

解决：

background-image: url(icon\_home.png);

background-repeat: no-repeat;

\_filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(src='icon\_home.png');

\_background-image: none;

**19. list-style-position默认值的问题**

问题：

ie下list-style-position默认为inside, firefox默认为outside

解决：

css中指定为outside即可解决兼容性问题

**20. list-style-image准确定位的问题**

**问题：**

**li前设置图片时，图片与其后的文字对齐问题**

**解决：**

1、采用背景定位 和 字符缩进的方法

background:url() no-repeat left center;text-index:16px;

2、采用相对定位方法

li 设置list-style:url();

li的子元素position:relative;top:-5px;

**21. ul标签默认值的问题**

**问题：**

**ul标签在**ff中默认是有padding值的,而在ie中只有margin有值

**解决：**

定义ul{margin:0;padding:0;}就能解决大部分问题

**22. IE中li指定高度后，出现排版错误**

问题：

在ie下如果为li指定高度可能会出现排版错位

解决：

设置line-height

**23. ul或li浮动后，显示在div外**

问题：

div中的ul或li设置float以后，都不在div中

解决：

必须在ul标签后加<div style="clear:both"></div>来闭合外层div

**24. ul浮动后，margin变大**

问题：

ul设置 float后，在ie中margin将变大

解决：

设置ul的display:inline，li的list-style-position:outside

**25. li浮动后，margin变大**

问题：

li设置 float后，在ie中margin将变大

解决：

设置li的display:inline

**26. 嵌套使用ul、li的问题**

问题：

ie的bug，嵌套使用ul、li时，里层的li设置float以后，外层li不设置float, 里面的ul顶部和它外面的li总是有一段间距

解决：

设置里面的ul的zoom:1

**27. IE6-7 li底部有3px的问题**

**问题：**

这个bug产生的充要条件是li的子元素浮动并且li设置了以下CSS属性之一：width、height、zoom、padding-top、padding-bottom、margin-top、margin-bottom。

**解决：**

1、div设置clear:left|both，这时li不能设置width、height、zoom。

2、li设置float:left，这时li可以设置width、height、zoom。

3、li设置clear:left|both，这时li不能设置width、height、zoom。

4、IE6/IE7的这个Bug可以通过给li中的div设置vertical-align:top|middle|bottom解决。

**28. IE6 垂直列表间隙的问题**

**问题：**

li中有a且设置display:block时，ie6中列表间会出现空隙

**解决：**

1、li中加display:inline;

2、li使用float 然后 clear:both;

3、给包含的文本末尾添加一个空格

4、设置width

**29. IE6 列表背景颜色失效的问题**

**问题：**

当父元素设置position:relative;时，在ie6中第一个ul、ol、dl的背景颜色失效

**解决：**

ul、ol、dl 都设置为position:relative;

**30. IE6-7 列表背景颜色失效的问题2**

**问题：**

做横向导航栏时，ul设置为float且有背景色，li设置为float。ie6-7背景颜色失效

**解决：**

很多ie的bug都可以通过触发layout来解决 ul添加属性

1、height:1%;

2、float:left;

3、zoom:1;

**31. 列表不能换行的问题**

**问题：**

**li设置为浮动，后面的li紧随其后，不能换行**

**解决：**

1、为这个ul定义合适的宽高

2、给包含这个ul 的父div定义合适的宽高。

**32. li中的内容以省略号显示**

**问题：**

li中内容超过长度时，想以省略号显示， 此方法适用于ie6-7-8、opera、safari浏览器

ff浏览器不支持

**解决：**

li{width:200px;white-space:nowrap;text-overflow:ellipsis;

-o-text-overflow:ellipsis; overflow:hidden; }

**33. 超链接访问过后hover样式不出现的问题**

**问题：**

**点击超链接后，hover、active样式没有效果**

**解决：**

改变CSS属性的排列顺序: L-V-H-A

**34. 禁用中文输入法的问题**

**问题：**

**不能在输入框中输入汉字**

**解决：**

只在ie系列和ff中有效

ime-mode:disabled **(但可以粘贴)**

禁用粘贴：

onpaste="return false"

**35. 除去滚动条的问题**

**问题：**

**隐藏滚动条**

**解决：**

**1、只有ie6-7支持**<body scroll="**no**">

**2、除ie6-7不支持** body{overflow:hidden}

**3、所有浏览器** html{overflow:hidden}

**36. 让层显示在FLASH之上**

**问题：**

**想让层的内容显示在flash上**

**解决：**

把FLASH设置透明

1、<param name=" wmode " value="transparent" />

2、<param name="wmode" value="opaque"/>

**37. 去除链接虚线边框的问题**

**问题：**

**当点击超链接后，ie6/7/8 ff会出现虚线边框 ,而**opera、safari没有虚线边框

**解决：**

ie6/7中 设置为a {blr:expression\_r(this.onFocus=this.blur()) }

ie8 和 ff 都不支持expression 在ie8 、ff中设置为 :focus { outline: none; }

**38. css滤镜的问题**

**问题：**

css滤镜只在ie中有效，Firefox, Safari(WebKit), Opera只能够设置透明，它们不支持滤镜filter，无法实现图片切换中间变换的效果，只能通过透明度来设置。

**解决：**

**ff中设置透明度** -moz-opacity:0.10; opacity:0.6;

ie中只设置filter:alpha(opacity=50);时，ie6-7失效，还要设置

1、zoom:1; 2、width:100%;

**39. IE6背景闪烁的问题**

**问题：**

链接、按钮用CSS sprites作为背景，在ie6下会有背景图闪烁的现象。原因是:IE6没有将背景图缓存，每次触发hover的时候都会重新加载

**解决：**

可以用JavaScript设置ie6缓存这些图片：

document.execCommand("BackgroundImageCache",false,true);

**40. 出现重复文字的问题**

**问题：**

**<div style="width:400px">**

**<div style="float:left"></div>**

**<!– \_ –>**

**<div style="float:right;width:400px">↓这就是多出来的那只猪</div>**

**</div>**

**解决：**

**1、 改变结构，不出现【一个容器包含2两个具有“float”样式的子容器】的结构。**

**2、减小第二个容器的宽度，使父容器宽度减去第二个容器宽度的值大于3**

3、**去掉所有的注释。**

**4、**修正注释的写法。<!--[if!IE]>这里是注释内容<![endif]-->

5、在第二个容器后面加一个或者多个<divstyle="clear"></div>来解决。

**41. ff、chrome绝对定位无效**

问题：

在IE给td设置position:relative，然后给它包含的一个容器使用position:absolute进行定位是有效的，但在FF和Chrome下却不可以。

解决：

设置td的display:block。

**42. IE6 绝对定位的问题**

**问题：**

<div style="position:relative;border:1px solid orange;text-align:center;">

<div style="position:absolute;top:0;left:0;

background:#CCC;">dovapour</div>

<a href="#" title="vapour的blog">内容</a>

</div>

**解决：**

left的定位错误问题

1、给父层设置zoom:1触发layout。

2、给父层设置宽度width

bottom的定位错误问题

1、给父层设置zoom:1触发layout。

2、给父层设置高度height

**43. 子容器宽度大于父容器宽度时，内容超出**

问题：

子DIV的宽度和父DIV的宽度都已经定义，在IE6中如果其子DIV的宽度大于父DIV的宽度，父DIV的宽度将会被扩展，在其他浏览器中父DIV的宽度将不会扩展，子DIV将超出父DIV

解决：

设置overflow:hidden，子DIV将不会超出父DIV。

**44. float的div闭合的问题**

问题：

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，问题出在其他浏览器中。原因是NOTfloatC并非float标签，必须将float标签闭合。

解决：

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;}

**45. 单选框、复选框与后面的文字对不齐**

问题：

单选框、复选框与后面的文字对不齐。

解决：

.align{font-size:12px;}

.align input{ display:block; float:left;}

.align label{ display:block; float:left;padding-top:3px; \*padding-top:5px;}

**需注意的问题：**

**1. 设置padding后高度和宽带都会增加**

说明：

除了ie5.5，其他所有浏览器中，设置padding以后高度和宽带都会增加

**2. 使用XHTML 1.0Transitional后，div宽度**

说明：

在使用XHTML 1.0Transitional以后div宽度都不包含border的宽度了，设置宽度的时候需要注意下。

**3. 外层相对定位，内层绝对定位**

说明：

ie6下，外层div的postion:relative，并设置text-align，内层div的postion: absolute，这时内层的位置是相对于text-align而言的

例如：

<div style="position:relative;border:1px solid orange;text-align:center;zoom:1"> position:relative

<div style="position:absolute;top:0;left:0;background:#CCC;">position:absolute</div>

</div>

**4. &nbsp; 显示的大小不一致**

说明：

默认字本显示问题，导致&nbsp;显示的大小不一致，在ie下比较小一点，其他的浏览器都一致，当你使用了&nbsp;造成问题时请注意。

**5. 边框重叠说明**

说明：

为 table、td 都指定了边框后，然后使用border-collapse:collapse让边框重叠，可以看出在发生重叠时，Firefox是用 td 覆盖 table 的，而 IE 是用 table 覆盖 td 的。使用时候需要注意。

**6. 设置td padding的说明**

说明：

设置td的padding以后高度和宽带都会增加,padding-left和padding-right的效果都一样增加了td的宽带，但是padding-top和padding-bottom的效果不一样。最好不要使用td的padding-top和padding-bottom

**7. ul设置的说明**

说明：

ul一般设置：list-style-type:none;margin:0px;padding:0px；li一般设置：list-style-type:none;list-style-position:outside

**8. 使一个层垂直居中于浏览器中**

说明：

使用百分比绝对定位,与外补丁负值的技巧,负值的大小为其自身宽度高度除以二

div {  
position:absolute; top:50%; lef:50%; margin:-100px 0 0 -100px;

width:200px; height:200px; border:1px solidred;  
}

**9. 万能 float 闭合**

说明：

可以用这个解决多个div对齐时的间距不对，将以下代码加入Global CSS 中,给需要闭合的div加上 class=”clearfix” 即可

<style>  
  
.clearfix:after { content:".";display:block;height:0; clear:both;visibility:hidden;  
}  
.clearfix {  
 display:inline-block;  
}  
  
.clearfix {display:block;}  
  
  
</style>

**10. 触发layout**

说明：

IE6中很多Bug都可以通过触发layout得到解决.下列的CSS属性或取值会让一个元素获得layout：

position:absolute 绝对定位元素的包含区块(containingblock)就会经常在这一方面出问题

float:left|right 由于layout元素的特性，浮动模型会有很多怪异的表现

display:inline-block 当一个内联级别的元素需要layout的时候就往往符用到它，这也可能也是这个CSS属性的唯一效果----让某个元素有layout

width: 除auto外的任何值

height: 除auto外的任何值

zoom: 除auto外的任何值

**11、如何使连续长字段自动换行**

**ff最新版本word-wrap:break-word;就可以了**

**ff旧版本 还要使用javascript完成文字换行**

**<style type="text/css">**

**div {**

**width:300px;**

**word-wrap:break-word;**

**border:1px solid red;**

**}**

**</style>**

<scripttype="text/javascript">

function toBreakWord(intLen){

varobj=document.getElementByIdx\_x("ff");

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.getElementByIdx\_x && !document.all) toBreakWord(37)

**12、设置滚动条颜色 只对ie系列有效 在html中 而不是设置body**

<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>

IE不支持float：inherit overflow:hidden有2个用法，一个是隐藏溢出，另一个是清除浮动。

<div>, <p>, <h1>,<form>, <ul> 和 <li>是块元素的*例子*

<span>, <a>, <label>,<input>, <img>, <strong> 和<em>是inline元素

<body oncontextmenu="returnfalse" ondragstart="return false" tstart="returnfalse" scroll="auto">

这行代码放在body中，去掉了页面鼠标右键快捷菜单，达到防止图片另存为的目的。

**javascript部分  
1. document.form.item 问题**  
问题：  
代码中存在 document.formName.item("itemName") 这样的语句，不能在FF下运行  
解决方法：  
改用 document.formName.elements["elementName"]  
  
**2. 集合类对象问题**  
问题：  
代码中许多集合类对象取用时使用()，IE能接受，FF不能  
解决方法：  
改用 [] 作为下标运算，例：  
document.getElementsByName("inputName")(1) 改为document.getElementsByName("inputName")[1]  
  
**3. window.event**  
问题：  
使用 window.event 无法在FF上运行  
解决方法：  
FF的 event 只能在事件发生的现场使用，此问题暂无法解决。可以把 event 传到函数里变通解决：  
onMouseMove = "functionName(event)"  
function functionName (e) {  
 e = e || window.event;  
 ......  
}  
  
**4. HTML对象的 id 作为对象名的问题**  
问题：  
在IE中，HTML对象的 ID 可以作为 document 的下属对象变量名直接使用，在FF中不能  
解决方法：  
使用对象变量时全部用标准的 getElementById("idName")  
  
**5. 用 idName 字符串取得对象的问题**  
问题：  
在IE中，利用 eval\_r("idName") 可以取得 id 为 idName 的HTML对象，在FF中不能  
解决方法：  
用 getElementById("idName") 代替eval\_r("idName")  
  
**6. 变量名与某HTML对象 id 相同的问题**  
问题：  
在FF中，因为对象 id 不作为HTML对象的名称，所以可以使用与HTML对象 id 相同的变量名，IE中不能  
解决方法：  
在声明变量时，一律加上 var ，以避免歧义，这样在IE中亦可正常运行  
最好不要取与HTML对象 id 相同的变量名，以减少错误  
  
**7. event.x 与 event.y 问题**  
问题：  
在IE中，event 对象有x,y属性，FF中没有  
解决方法：  
在FF中，与 event.x 等效的是 event.pageX ，但event.pageX IE中没有  
故采用 event.clientX 代替 event.x ，在IE中也有这个变量  
event.clientX 与 event.pageX 有微妙的差别，就是滚动条  
要完全一样，可以这样：  
mX = event.x ? event.x : event.pageX;  
然后用 mX 代替 event.x  
  
**8. 关于frame**  
问题：  
在IE中可以用 window.testFrame 取得该frame，FF中不行  
解决方法：  
window.top.document.getElementByIdx\_x("testFrame").src = 'xx.htm'  
window.top.frameName.location = 'xx.htm'  
  
**9. 取得元素的属性**  
在FF中，自己定义的属性必须 getAttribute() 取得  
  
**10. 在FF中没有 parentElement，parement.children 而用 parentNode，parentNode.childNodes**  
问题：  
childNodes 的下标的含义在IE和FF中不同，FF的 childNodes 中会插入空白文本节点  
解决方法：  
可以通过 node.getElementsByTagName\_r() 来回避这个问题  
问题：  
当html中节点缺失时，IE和FF对 parentNode 的解释不同，例如：  
<form>  
<table>  
<input/>  
</table>  
</form>  
FF中 input.parentNode 的值为form，而IE中 input.parentNode 的值为空节点  
问题：  
FF中节点自己没有 removeNode 方法  
解决方法：  
必须使用如下方法 node.parentNode.removeChild(node)  
  
**11. const 问题**  
问题：  
在IE中不能使用 const 关键字  
解决方法：  
以 var 代替  
  
**12. body 对象**  
FF的 body 在 body 标签没有被浏览器完全读入之前就存在，而IE则必须在 body 完全被读入之后才存在  
这会产生在IE下，文档没有载入完时，在body上appendChild会出现空白页面的问题  
解决方法：  
一切在body上插入节点的动作，全部在onload后进行  
  
**13. url encoding**  
问题：  
一般FF无法识别js中的&  
解决方法：  
在js中如果书写url就直接写&不要写&  
  
**14. nodeName 和 tagName 问题**  
问题：  
在FF中，所有节点均有 nodeName 值，但 textNode 没有 tagName 值，在IE中，nodeName 的使用有问题  
解决方法：  
使用 tagName，但应检测其是否为空  
  
**15. 元素属性**  
IE下 input.type 属性为只读，但是FF下可以修改  
  
**16. document.getElementsByName() 和 document.all[name] 的问题**  
问题：  
在IE中，getElementsByName()、document.all[name] 均不能用来取得 div 元素  
是否还有其它不能取的元素还不知道（这个问题还有争议，还在研究中）  
  
**17. 调用子框架或者其它框架中的元素的问题**  
在IE中，可以用如下方法来取得子元素中的值  
document.getElementByIdx\_x("frameName").(document.)elementName  
window.frames["frameName"].elementName  
在FF中则需要改成如下形式来执行，与IE兼容：  
window.frames["frameName"].contentWindow.document.elementName  
window.frames["frameName"].document.elementName  
  
**18. 对象宽高赋值问题**  
问题：  
FireFox中类似 obj.style.height = imgObj.height 的语句无效  
解决方法：  
统一使用 obj.style.height = imgObj.height + "px";  
**19. innerText的问题**  
问题：  
innerText 在IE中能正常工作，但是 innerText 在FireFox中却不行  
解决方法：  
在非IE浏览器中使用textContent代替innerText

**20. event.srcElement和event.toElement问题**  
问题：  
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：  
var source = e.target || e.srcElement;  
var target = e.relatedTarget || e.toElement;  
  
**21. 禁止选取网页内容**  
问题：  
FF需要用CSS禁止，IE用JS禁止  
解决方法：  
IE: obj.onselectstart = function() {return false;}  
FF: -moz-user-select:none;  
**22. 捕获事件**  
问题：  
FF没有setCapture()、releaseCapture()方法  
解决方法：  
IE:  
obj.setCapture();  
obj.releaseCapture();  
FF:  
window.captureEvents(Event.MOUSEMOVE|Event.MOUSEUP);  
window.releaseEvents(Event.MOUSEMOVE|Event.MOUSEUP);  
if (!window.captureEvents) {  
 o.setCapture();  
}else {  
 window.captureEvents(Event.MOUSEMOVE|Event.MOUSEUP);  
}  
if (!window.captureEvents) {  
 o.releaseCapture();  
}else {  
 window.releaseEvents(Event.MOUSEMOVE|Event.MOUSEUP);  
}

**一.用div+css网站布局的好处**

1：表现和内容相分离

　　将设计部分剥离出来放在一个独立样式文件中，HTML文件中只存放文本信息。

2：提高搜索引擎对网页的索引效率

　　用只包含结构化内容的HTML代替嵌套的标签，搜索引擎将更有效地搜索到你的网页内容，并可能给你一个较高的评价。

3：提高页面浏览速度

　　对于同一个页面视觉效果，采用CSS+DIV重构的页面容量要比TABLE编码的页面文件容量小得多，前者一般只有后者的1/2大小。

4：易于维护和改版

　　你只要简单的修改几个CSS文件就可以重新设计整个网站的页面。从以上的描述来看，采用CSS+DIV对网站重构可以大大提升网站用户与搜索引擎的友好度,CSS+DIV所以成为目前网页布局主流.

**二.用表格table对seo影响**

1、简介：为何使用表格排版是不明智的?

　　表格之所以存在于 HTML 中，只是为了一个目的：显示表格状的数据。然而此后的 border="0" 使得设计师可以将图片和文本放在这无形的网格中。迄今为止，表格仍然主导着视觉丰富的网站的设计方式，但它却阻碍了一种更好的、更有亲和力的、更灵活的，而且功能更强大的网站设计方法。

2、表格带来的问题

　　把格式数据混入你的内容中,这使得文件的大小无谓地变大，而用户访问每个 页面时都必须下载一次这样的格式信息。