目录

[1、CSS选择器区分 3](#_Toc418007250)

[2、PNG半透明图片的问题 3](#_Toc418007251)

[3、IE6下的圆角 3](#_Toc418007252)

[4、IE6背景闪烁 3](#_Toc418007253)

[5、最小高度 3](#_Toc418007254)

[6、最大高度 4](#_Toc418007255)

[7、100% 高度 4](#_Toc418007256)

[8、最小宽度 4](#_Toc418007257)

[9、最大宽度 5](#_Toc418007258)

[10、双边距Bug 5](#_Toc418007259)

[11、清除浮动 6](#_Toc418007260)

[12、浮动层错位 6](#_Toc418007261)

[13、躲猫猫bug 6](#_Toc418007262)

[14、绝对定位元素的1像素间距bug 6](#_Toc418007263)

[15、3像素间距bug 7](#_Toc418007264)

[16、IE下z-index的bug 7](#_Toc418007265)

[17、Overflow Bug 7](#_Toc418007266)

[18、横向列表宽度bug 7](#_Toc418007267)

[19、列表阶梯bug 7](#_Toc418007268)

[20、垂直列表间隙bug 8](#_Toc418007269)

[21、IE6调整窗口大小的 Bug 8](#_Toc418007270)

[22、文本重复Bug 8](#_Toc418007271)

[23、Set Position: relative 8](#_Toc418007272)

[24、span标签的高度问题 8](#_Toc418007273)

[25、小于20px超出范围 9](#_Toc418007274)

[26、margin 9](#_Toc418007275)

[27、background 11](#_Toc418007276)

[28、font-size 11](#_Toc418007277)

[29、line-height 11](#_Toc418007278)

[30、float 11](#_Toc418007279)

[31、position 12](#_Toc418007280)

[32、border 12](#_Toc418007281)

[附录： 12](#_Toc418007282)

[1.通用区分方式 12](#_Toc418007283)

[2.各种浏览器独立支持的hack 14](#_Toc418007284)

[3.IE特有的html条件注释使用规则 14](#_Toc418007285)

[4.非IE可见的写法 15](#_Toc418007286)

[5.以下为于页面模块的常用命名 15](#_Toc418007287)

### 1、CSS选择器区分

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。

.content { color: red; }

div>p .content { color: blue; }

### 2、PNG半透明图片的问题

使用DD\_belatedPNG.js插件解决透明问题，但ie6只支持带有href属性的a标签的hover效果，其他均不支持hover。或者通过背景图与png图片融合，切成jpg的图片。

### 3、IE6下的圆角

IE6不支持CSS3的圆角属性，性价比最高的解决方法就是用图片圆角来替代，或者放弃IE6的圆角。

### 4、IE6背景闪烁

如果你给链接、按钮用CSS sprites作为背景，你可能会发现在IE6下会有背景图闪烁的现象。造成这个的原因是由于IE6没有将背景图缓存，每次触发hover的时候都会重新加载，可以用JavaScript设置IE6缓存这些图片：

document.execCommand(“BackgroundImageCache”,false,true);

### 5、最小高度

IE6 不支持min-height属性，但它却认为height就是最小高度。

最小高度问题，设置较小高度标签（一般小于10px），在ie6，ie7，遨游中高度超出自己设置高度 解决方案：给超出高度的标签设置overflow:hidden;或者设置行高line-height小于你设置的高度。 备注：这种情况一般出现在我们设置小圆角背景的标签里。出现这个问题的原因是ie8之前的浏览器都会给标签一个最小默认的行高的高度。即使你的标签是空的，这个标签的高度还是会达到默认的行高。

ie6内容撑开高度。

#container {min-height:200px; \_height:200px;}

### 6、最大高度

//直接使用css

height: expression( this.scrollHeight > 500 ? "500px" : "auto" );

//写成函数来运行  
function setMaxHeight(elementId, height){

var container = document.getElementByIdx\_x(elementId);

container.style.height = (container.scrollHeight > (height – 1)) ? height + “px” : “auto”;

}

//函数示例

setMaxHeight(‘container1′, 200);

setMaxHeight(‘container2′, 500);

### 7、100% 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%;。

但是body定义100%后滚动条会消失，无法滚动，建议使用下面的css

div{ \_width:expression(eval(document.documentElement.clientWidth)); \_height:expression(eval(window.screen.availHeight)); }

### 8、最小宽度

同max-height和max-width一样，IE6也不支持min-width。

//直接使用css  
#container {min-width:200px; \_ width:200px;\_overflow:hidden;\_ white-space:nowrap; }

//写成函数来运行  
function setMinWidth(elementId, width){

var container = document.getElementByIdx\_x(elementId);

container.style.width = (container.clientWidth < width) ? width + “px” : “auto”;

}

//函数示例  
setMinWidth(‘container1′, 200);

setMinWidth(‘container2′, 500);

### 9、最大宽度

//直接使用css

.a{width:expression(this.clientWidth > 950？"950px": "auto" ); }

//写成函数来运行

function setMaxWidth(elementId, width){

var container = document.getElementByIdx\_x(elementId);

container.style.width = (container.clientWidth > (width – 1)) ? width + “px” : “auto”;

}

//函数示例

setMaxWidth(‘container1′, 200);

setMaxWidth(‘container2′, 500);

### 10、双边距Bug

当元素float并且margin时，IE6会错误的把浮动方向的margin值双倍计算。解决办法display:inline;

### 11、清除浮动

如果你想用div(或其他容器)包裹一个浮动的元素，你会发现必须给div(容器)定义明确的height、width、overflow之中一个属性（除了auto值）才能将浮动元素严实地包裹。

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

.clearFix{ \*zoom: 1; }

### 12、浮动层错位

当内容超出外包容器定义的宽度时，在IE6中容器会忽视定义的width值，宽度会错误地随内容宽度增长而增长。一般情况下记得要清除浮动，并且width和height正好包裹内容。

浮动层错位问题在IE6下没有真正让人满意的解决方法，虽然可以使用overflow:hidden;或overflow:scroll;来修正，但hidden容易导致其他一些问题，scroll会破坏设计；JavaScript也没法很好地解决这个问题。所以建议是一定要在布局上避免这个问题发生，使用一个固定的布局或者控制好内容的宽度（给内层加width）。

### 13、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。

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%;

### 14、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。

### 15、3像素间距bug

在IE6中，当文本(或无浮动元素)跟在一个浮动的元素之后，文本和这个浮动元素之间会多出3像素的间隔。

给浮动层添加 display:inline 和 -3px 负值margin

给中间的内容层定义 margin-right 以纠正-3px

### 16、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。

### 17、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。解决方法就是给父元素加上 position:relative;。

### 18、横向列表宽度bug

如果你使用float:left;把横向摆列，并且 内包含的（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，只需要给定义同样的float:left;即可。

### 19、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。

### 20、垂直列表间隙bug

当我们使用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（

）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。

### 21、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。

### 22、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;。

### 23、Set Position: relative

给元素指定Position: relative能够解决很多奇怪的问题，比如对齐和显示不正确等。对于 Position的属性要小心设置，特别是设为absolute。

### 24、span标签的高度问题

这个在做图片圆角的时候很见，background-position会出现溢出，只要在span标签下加上 line-height:0;font-size:0; 就可以解决了。

### 25、小于20px超出范围

在ie6下宽高小于20px不可显示，最小显示宽高为20px。

通过overflow:hidden;解决

### 26、margin

1）默认内外补丁，各个浏览器对标签默认的内外补丁都是不同的，若想在正常情况下设置我们想要的样式，就 必须把那些存在内外补丁差异的标签初始化，初始化的程序代码（是经过网友们一支修改过的）如下：。

程序代码

html, body, div, dl, dt, dd, ul, ol, li, h1, h2, h3, h4, h5, h6,

pre, form, fieldset, input, textarea, select, p, blockquote, th, td, legend, img {

padding: 0;

margin: 0;

}

fieldset, img, a, abbr, acronym {

border: none 0;

}

table {

border-collapse: collapse;

border-spacing: 0;

}

ol, ul {

list-style: none;

}

address, caption, cite, code, dfn, em, strong, th, var {

font-weight: normal;

font-style: normal;

}

caption, th {

text-align: left;

}

h1, h2, h3, h4, h5, h6 {

font-weight: normal;

font-size: 100%;

}

select, input { vertical-align:middle; }

a { text-decoration:none;}

.clearFix:after{ clear: both; display: block; visibility: hidden; height: 0; line-height: 0; font-size: 0; content: ""; }

.clearFix{ \*zoom: 1; }

input::-moz-focus-inner { border:none; padding:0; }

2）横向双倍margin：块属性标签float后，又对其设置横行margin情况下，在ie6显示margin比设置的大，通常是标准浏览器的二倍，常见症状是ie6中后面的一块被顶到下一行，解决办法是将其display:inline。那么行内属性标签设置display:block后采用float布局，也有横行的margin的情况，ie6间距bug，解决办法是在display:block;后面加入display:inline;display:table;因为：行内属性标签，为了设置宽高，我们需要设置display:block;(除input、img、textarea标签比较特殊)。在用float布局并有横向的margin后，在ie6下，它就具备了块属性float后的横向margin的bug。但因为它本身就是行内属性标签，所以我们再加上display:inline的话，它的高宽就不可设了。这时候我们还需要在display:inline后面加入display:talbe。方可解决。

3）父子标签之间控制间距最好用padding，因为我们经常会与上这样的情况，子标签margin时父标签也跟着margin了，magin会存在合并的情况，因此父子关系最好用padding，兼容性好，兄弟之间用margin。

### 27、background

1）在我们用a标签的伪类做tab切换效果时，发现凡是对有背景（或颜色或背景图）的标签使用了position定位控制其位置后，切换时在IE6和遨游下就会遗留背景。其他浏览器正常。

2）在background-position的x或y轴上使用百分比控制时，IE6的计算方式也不同。例如banner部分每小格的下方都有一个小箭头，为了拓展性效果好，我们尝试过使用背景图的百分比控制它在无论多宽的小格子下方都能居中，但是标准浏览器都是以当前格子的为基准的，IE6却是以当前至最后一个这一长短距离为基准的。

### 28、font-size

IE6内核不支持单数。

### 29、line-height

IE6中行高也能撑开高度，例如当前标签height：20px;设置行高30px；标准浏览器中缩显示的是我们所设的样式，但IE中此标签的高就会是30px，解决办法是设置overflow:hidden;并且IE6中标签的默认行 高也能通过此方法解决。

### 30、float

1）IE6浮动后横向margin换行的问题已在上述中说过。

2）假设两块div，第一块浮动而第二块没有浮动，IE6下第二块就会跑到第一块边上，并且二者之间还留有间距，但标准浏览器中是第二块重合于第一块。解决办法，将第二块设置为清除浮动。

3）父子关系的标签，子标签浮动导致父标签不再包裹子标签，对父标签设置clearfix或overflow：hidden;

### 31、position

1）、使用position定位做重叠但超出其父标签范围的效果时，做完后一定要清除浮动，否则要么不显示，要么下面内容会乱，总之清除浮动很重要。

2）、absolute，父子关系的标签，父标签使用overflow：hidden；后正常浏览器子标签都不显示了，IE6下还是显示的状态。

### 32、border

1）、a标签嵌套下的img标签,ie和ff会带有边框。解决办法：a, img {border:none;}这条命令我们一般都写在初始化里，一是因为样式上一般不需要这样的效果，再者a标签中的字体颜色尚且解析不同，边框的颜色也一样不同。

2）、去掉input边框问题，正常浏览器border:none;便可解决，但是ie6的默认CSS样式，涉及到border的有border-style:inset;border-width:2px;必须对border的width和style进行全部设置方可消除IE6下的border。因为浏览器是先解析自身的默认CSS，再解析开发者书写的CSS，所以当有border-width或border-color设置的时候才会令IE6去解析border-style:none;目前所遇情况都能解释。因此想要去除input的border就只有使用border:0 none;虽然但看命令有些说不过去，但是这也是浏览器解析存在问题时较好的解决办法。

|  |
| --- |
|  |

# 附录：

### 1.通用区分方式

IE6、IE7能识别\*，标准浏览器（如FF）不能识别\*；

IE6能识别\*，但不能识别 !important；

IE7能识别\*，也能识别 !important；

IE8能识别\0，不能识别\*，+,\_,\* 加!important;

FF不能识别\*，但能识别 !important；

例如style=”\*width:10px!important; width:20px;”，其在IE7下宽度为10px,在IE6下宽度是20px.

其中还有如下三种写法：

1)

width:100px; /\* FireFox及其他浏览器 \*/

width:200px\0; /\* IE8能识别\0\*/

\*width:300px!important; /\* ,IE7 既能能识别\*号，也能识别important \*/

\*width:400px; /\* IE6也能识别\*号 \*/

/\*提示：请注意书写顺序\*/

2)

width:100px; /\* FireFox及其他浏览器 \*/

width:200px\0; /\* IE8能识别\0\*/

\*width:300px; /\* IE7也能识别\*号 \*/

\_width:400px; /\* IE6能识别下划线\*/

/\*提示：请注意书写顺序\*/

3)

width:100px; /\* FireFox及其他浏览器 \*/

width:200px\0; /\* IE8能识别\0\*/

+width:300px; /\* +只识别IE7 \*/

\_width:400px; /\* IE6能识别下划线\*/

/\*提示：请注意书写顺序\*/

### 2.各种浏览器独立支持的hack

width:100px\0;/\* 支持IE8 \*/

\_width:100px; /\* 支持IE6 \*/

[width:100px; /\* 支持IE6,7 \*/

+width:100px; /\* 支持IE6,7 \*/

\*width:100px; /\* 支持IE6,7 \*/

\*width:100px!important; /\* 支持IE6,7, \*/

\*+width:100px; /\* 支持IE6,7, \*/

\*+width:100px!important;/\* 支持IE6,7, \*/

width:100px\9; /\* 支持IE6,7,8 \*/

width:100px!important; /\* 支持IE6,7,8,FF \*/

w\idth:100px; /\*IE5.x不支持 IE6、IE7、IE8、FF支持 \*/

### 3.IE特有的html条件注释使用规则

<!--[if IE 7]>

<style type="text/css">

</style>

<![endif]-->

<!--[if IE 6]>

<style type="text/css">

</style>

<![endif]-->

1)仅IE可见的写法

<!--[if IE]>此处内容只有IE可见<![endif]-->

2)仅IE6可见的写法

<!--[if IE 6]>此处内容只有IE6.0可见<![endif]-->

3)仅IE7可见的写法

<!--[if IE 7]>此处内容只有IE7.0可见<![endif]-->

4)版本区间可显示写法

<!--[if lt IE 6]>IE6以及IE6以下版本可识别<![endif]-->

<!--[if gte IE 6]>IE6以及IE6以上版本可识别<![endif]-->

<!--[if lt IE 7]>IE7以及IE7以下版本可识别<![endif]-->

<!--[if gte IE 7]>IE7以及IE7以上版本可识别<![endif]-->

PS：区间注释代码实际用起来貌似不包括本区间，这里需要注意下

### 4.非IE可见的写法

（注意：此条不符合WEB标准，但的确实用）

<!--[if !IE]>此处内容只非IE可见<![endif]-->

1)IE8降级为IE7：<meta http-equiv="X-UA-Compatible" content="IE=7">

2)解决chrome不支持12号以下字体的代码：-webkit-text-size-adjust:none;

3)li左侧ico样式：li { list-style:disc outside none; }

### 5.以下为于页面模块的常用命名

头：header

内容：content/container

尾：footer

导航：nav

侧栏：sidebar

栏目：column

页面外围控制整体布局宽度：wrapper

左右中：left right center

登录条：loginbar

标志：logo

广告：banner

页面主体：main

热点：hot

新闻：news

下载：download

子导航：subnav

菜单：menu

子菜单：submenu

搜索：search

友情链接：friendlink

页脚：footer

版权：copyright

滚动：scroll

内容：content

### 6.51vv网站常用bug修复

1）fixed处理

top处理

\_position: absolute;

\_top:expression(eval(document.documentElement.scrollTop));

bottom处理

\_position: absolute; \_margin-bottom: Xpx;

\_top: expression(eval(document.documentElement.scrollTop+document.documentElement.clientHeight-this.offsetHeight-(parseInt(this.currentStyle.marginTop,10)||0)-(parseInt(this.currentStyle.marginBottom,10)||0)));

width和height全屏处理

\_width:expression(eval(document.documentElement.clientWidth));

\_height:expression(eval(document.documentElement.clientHeight));

2）超出文字隐藏

overflow: hidden;

white-space: nowrap;

text-overflow: ellipsis;

display: block or inline-block;

width: Xpx;

3）透明处理

opacity: 0.5; filter: alpha(opacity=50);

4）评论

#vv\_comment{ width: 800px; margin: 10px auto 0; }

.talk\_box { min-height: 300px; \_height:300px; }

.cmt\_modelAdd{ display: none; }