

## 问题一

不同浏览器的标签默认的外补丁和内补丁不同

问题症状：随便写几个标签，不加样式控制的情况下，各自的margin 和padding差异较大。

碰到频率：100%

解决方案：css里 `*{margin:0;padding:0;}`

## 问题二

块属性标签float后，又有横行的margin情况下，在ie6显示margin比设置的大

问题症状：常见症状是ie6中后面的一块被顶到下一行

碰到频率：90%（稍微复杂点的页面都会碰到，float布局最常见的浏览器兼容问题）

解决方案：在float的标签样式控制中加入 `display:inline`;将其转化为行内属性

## 问题三

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

问题症状：ie6、7和遨游里这个标签的高度不受控制，超出自己设置的高度

碰到频率：60%

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。

圆角兼容性

解决方案：

```
border-radius: 6px 0 0 6px;
-moz-border-radius: 6px 0 0 6px;
-webkit-border-radius: 6px 0 0 6px;
behavior: url(PIE.htc);
```

## 问题四

行内属性标签，设置`display:block`后采用float布局，又有横行的margin的情况，ie6间距bug（类似第二种）

问题症状：ie6里的间距比超过设置的间距

碰到几率：20%

解决方案：在`display:block`;后面加入`display:inline;display:table`;

备注：行内属性标签，为了设置宽高，我们需要设置`display:block`;(除了input标签比较特殊)。在用float布局并有横向的margin后，在ie6下，他就具有了块属性float后的横向margin的bug。不过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:table`。

## 问题五

图片默认有间距

问题症状：几个img标签放在一起的时候，有些浏览器会有默认的间距，加了问题一中提到的通配符也不起作用。

碰到几率：20%

解决方案：使用float属性为img布局

备注：因为img标签是行内属性标签，所以只要不超出容器宽度，img标签都会排在一行里，但是部分浏览器的img标签之间会有个间距。去掉这个间距使用float是正道。

去除IE图片默认样式

解决方案：`outline: none;border: none`;

## 问题六

标签最低高度设置`min-height`不兼容

问题症状：因为`min-height`本身就是一个不兼容的css属性，所以设置`min-height`时不能很好的被各个浏览器兼容

碰到几率：5%

解决方案：如果我们要设置一个标签的最小高度200px，需要进行的设置为：`{min-height:200px; height:auto !important; height:200px; overflow:visible;}`

备注：在B/S系统前端开时，有很多情况下我们又这种需求。当内容小于一个值（如300px）时。容器的高度为300px；当内容高度大于这个值时，容器高度被撑高，而不是出现滚动条。这时候我们就会面临这个兼容性问题。

## 问题七

透明度的兼容css设置

解决方案：

```
filter:alpha(opacity=50);
-moz-opacity:0.5;
-khtml-opacity: 0.5;
opacity: 0.5;
```

# 47种浏览器常见问题

## JavaScript

### 1. HTML对象获取问题

Firefox: `document.getElementById(“idName”);`  
ie: `document.idname`或者`document.getElementById(“idName”)`.

解决办法：统一使用`document.getElementById(“idName”);`

### 2. const问题

说明:Firefox下, 可以使用const关键字或var关键字来定义常量;

IE下, 只能使用var关键字来定义常量.

解决方法：统一使用var关键字来定义常量.

### 3. event.x与event.y问题

说明:IE下, event对象有x, y属性, 但是没有pageX, pageY属性;

Firefox下, event对象有pageX, pageY属性, 但是没有x, y属性.

解决方法：使用`mX(mX = event.x ? event.x : event.pageX);`来代替IE下的event.x或者Firefox下的event.pageX.

### 4. window.location.href问题

说明:IE或者Firefox2.0.x下, 可以使用`window.location`或`window.location.href`;

Firefox1.5.x下, 只能使用`window.location`.

解决方法：使用`window.location`来代替`window.location.href`.

### 5. frame问题

以下面的frame为例：

(1) 访问frame对象：

IE:使用`window.frameId`或者`window.frameName`来访问这个frame对象. `frameId`和`frameName`可以同名。

Firefox: 只能使用`window.frameName`来访问这个frame对象.

另外，在IE和Firefox中都可以使用`window.document.getElementById(“frameId”)`来访问这个frame对象.

(2) 切换frame内容：

在 IE和Firefox中都可以使用`window.document.getElementById(“testFrame”).src = “xxx.html”` 或`window.frameName.location = “xxx.html”` 来切换frame的内容.

如果需要将frame中的参数传回父窗口(注意不是opener, 而是parent frame)，可以在frame中使用parent来访问父窗口。例如：

```
parent.document.form1.filename.value=“Aqing”;
```

### 6. 模态和非模态窗口问题

说明:IE下, 可以通过`showModalDialog`和`showModelessDialog`打开模态和非模态窗口;Firefox下则不能.

解决方法：直接使用`window.open(pageURL, name, parameters)`方式打开新窗口。

如果需要将子窗口中的参数传递回父窗口, 可以在子窗口中使用`window.opener`来访问父窗口.

例如: `var parWin = window.opener; parWin.document.getElementById(“Aqing”).value = “Aqing”;`

### 7. firefox与IE的父元素(parentElement)的区别

IE: `obj.parentElement`

firefox: `obj.parentNode`

解决方法：因为firefox与IE都支持DOM,因此使用obj.parentNode是不错选择。

#### 8. document.formName.item(" itemName" ) 问题

问题说明：IE下，可以使用document.formName.item(" itemName" ) 或document.formName.elements [ "elementName" ]; Firefox 下，只能使用document.formName.elements[ "elementName" ]。

解决方法：统一使用document.formName.elements[ "elementName" ]。

#### 9. 集合类对象问题

问题说明：IE下，可以使用 () 或 [] 获取集合类对象；Firefox下，只能使用 [] 获取集合类对象。

解决方法：统一使用 [] 获取集合类对象。

#### 10. 自定义属性问题

问题说明：IE下，可以使用获取常规属性的方法来获取自定义属性，也可以使用getAttribute() 获取自定义属性；Firefox下，只能使用getAttribute() 获取自定义属性。

解决方法：统一通过getAttribute() 获取自定义属性。

#### 11. input.type属性问题

问题说明：IE下input.type属性为只读；但是Firefox下input.type属性为读写。

解决办法：不修改input.type属性。如果必须要修改，可以先隐藏原来的input，然后在同样的位置再插入一个新的input元素。

#### 12. event.srcElement问题

问题说明：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：使用srcObj = event.srcElement ?event.srcElement : event.target;

如果考虑第8条问题，就改用myEvent代替event即可。

#### 13. body载入问题

问题说明：Firefox的body对象在body标签没有被浏览器完全读入之前就存在；而IE的body对象则必须在body标签被浏览器完全读入之后才存在。

[注] 这个问题尚未实际验证，待验证后再来修改。

[注] 经验证，IE6、Opera9以及FireFox2中不存在上述问题，单纯的JS脚本可以访问在脚本之前已经载入的所有对象和元素，即使这个元素还没有载入完成。

#### 14. 事件委托方法

问题说明：IE下，使用document.body.onload = inject; 其中function inject()在这之前已被实现；在Firefox下，使用document.body.onload = inject();

解决方法：统一使用document.body.onload=new Function(' inject()' ); 或者document.body.onload = function() { /\* 这里是代码 \*/ }

[注意] Function和function的区别。

#### 15. Table操作问题

问题说明：ie、firefox以及其它浏览器对于 table 标签的操作都各不相同，在ie中不允许对table和tr.innerHTML赋值，使用js增加一个tr时，使用appendChild方法也不管用。

解决方法：//向table追加一个空行：

```
var row = otable.insertRow(-1);var cell = document.createElement( "td" );cell.innerHTML = " ";cell.className = "XXXX";row.appendChild(cell);[注] 由于俺很少使用JS直接操作表格，这个问题没有遇见过。建议使用JS框架集来操作table，如jQuery。
```

#### 16. 对象宽高赋值问题

问题说明：FireFox中类似obj.style.height = imgObj.height的语句无效。

## CSS

#### 1. cursor:hand VS cursor:pointer

firefox不支持hand，但ie支持pointer

解决方法：统一使用pointer

#### 2. innerText在IE中能正常工作，但在FireFox中却不行。

需用textContent。

解决方法：

```
1 if (indexOf "Explorer" 1
2 document.getElementById('element').innerText = "my text";
3 else
4 document.getElementById('element').textContent = "my text";
5
```

#### 3. CSS透明

IE: filter:progid:DXImageTransform.Microsoft.Alpha(style=0,opacity=60)。

FF: opacity:0.6。

opacity 透明，子元素会继承透明属性。解决方式：

(1)使用 background:rgba(0,0,0,.6) //IE8及以下无效果。

(2)使用定位，背景色与子元素处于同级关系。

#### 4. css中的width和padding

在IE7和FF中width宽度不包括padding，在Ie6中包括padding。

#### 5. FF和IEBOX模型解释不一致导致相差2px

```
box.style{width:100;border 1px;}
```

ie理解为box.width = 100

ff理解为box.width = 100 + 1\*2 = 102 //加上边框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，IE不能识别!important这个属性，但别的浏览器可以识别。所以在IE下其实解释成这样：

```
div{margin: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;

#### 6. 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}

#### 7. ul和ol列表缩进问题

消除ul、ol等列表的缩进时，样式应写成: list-style:none;margin:0px;padding:0px;

经验证，在IE中，设置margin:0px可以去除列表的上下左右缩进、空白以及列表编号或圆点，设置padding对样式没有影响；在Firefox中，设置margin:0px仅仅可以去除上下的空白，设置padding:0px后仅仅可以去掉左右缩进，还必须设置list-style:none才能去除列表编号或圆点。也就是说，在IE中仅仅设置margin:0px即可达到最终效果，而在Firefox中必须同时设置margin:0px、padding:0px以及list-style:none三项才能达到最终效果。

#### 8. 元素水平居中问题

FF: margin:0 auto;

IE: 父级{ text-align:center; }

#### 9. 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。

#### 10. 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

```
1 #imfloat{
2     float:left;
3     width:5px /*IE下理解为10px*/
4     display:inline; /*IE下再理解为5px*/
5 }
```

#### 11. 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：

```
1 #box{ width: 80px; height: 35px;}html>body #box{ width: auto; height: auto; min-width:
2 80px; min-height: 35px;}
```

#### 12. 页面的最小宽度

如上一个问题，IE不识别min，要实现最小宽度，可用下面的方法：

```
#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。

#### 13. DIV浮动IE文本产生3像素的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。

```
1 <style>
2 #box{ float:left; width:800px;}
3 #left float:left width 50%
4 #right{ width:50%;
5 *html #left margin-right -3px //这句是关键
```

```

6      </style>
7      <div id="box">
8        <div id="left"></div>
9        <div id="right"></div>
10     </div>

```

#### 14. 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。

#### 15. 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,为了让高度能自适应,要在wrapper里面加上overflow:hidden; 当包含float的box的时候,高度自适应在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：

```

1      <div id="page">
2      <div id="left">
3      <div id="center"></div>
4      <div id="right">
5      </div>
6      </div>
7      </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：

```

1      <div id="page">
2      <div id="bg" style="float:left; width:100%; height:100px; background-color:blue;">
3      <div id="left">
4      <div id="center"></div>
5      <div id="right">
6      </div>
7      </div>
8      </div>
9      </div>
10     </div>
11     </div>

```

再嵌入一个float left而宽度是100%的DIV解决之。

或者另一种方法：用选择器（：after）在page之后插入一个空标签，并清除浮动

```

1      <div id="page">
2      <div id="bg" style="float:left; width:100%; height:100px; background-color:blue;">
3      <div id="left">
4      <div id="center"></div>
5      <div id="right">
6      </div>
7      </div>
8      </div>
9      </div>
10     </div>
11     </div>

```

#### 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。

```

1      /* Clear Fix */
2      .clearfix:after { content:"."; display:block; height:0; clear:both; visibility:hidden; }
3      .clearfix { }
4      .clearfix { }
5      /* Hide from IE Mac */
6      .clearfix { }
7      /* End hide from IE Mac */
8      /* 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}

#### 16. 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padding时。

例：

```

1      #box {background-color:#eee; }
2      #box p {margin-top: 20px;margin-bottom: 20px; text-align:center; }
3      <div id="box">
4      <p>对象中的内容</p>
5      </div>

```

解决技巧：在P对象上下各加2个空的div对象CSS代码{height:0px;overflow:hidden;} 或者为DIV加上border属性。

#### 17. IE6下图片下有空隙产生

解决这个BUG的技巧有很多,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。

## 18. 对齐文本与文本输入框

加上vertical-align:middle;

```
1 <style type="text/css">
2 <!--
3 input
4 width:200px;
5 height:30px;
6 border:1px solid red;
7 vertical-align middle
8 }
9 -->
10 </style>
```

经验证，在IE下任一版本都不适用，而ff、opera、safari、chrome均OK！

## 19. LI中内容超过长度后以省略号显示

此技巧适用与IE、Opera、safari、chrom浏览器，FF暂不支持。

```
1 <style type="text/css">
2 <!--
3 li
4 width:200px;
5 white-space nowrap
6 text-overflow:ellipsis;
7 -o-text-overflow ellipsis
8 overflow: hidden;
9
10
11 -->
12 </style>
```

## 20. 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

```
1 DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
2 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"
3 <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
4 <style type="text/css">
5 <!--
6 html
7 scrollbar-face-color:#f6f6f6;
8 scrollbar-highlight-color #fff
9 scrollbar-shadow-color:#eeeeee;
10 scrollbar-3dlight-color #eeeeee
11 scrollbar-arrow-color:#000;
12 scrollbar-track-color #fff
13 scrollbar-darkshadow-color:#fff;
14
15 -->
16 </style>
```

## 21. 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,解决的技巧也有很多:

例如:overflow:hidden zoom:0.08 line-height:1px

## 22. 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" />

## 23. 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。

## 24. 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,解决技巧是改变CSS属性的排列顺序: L-V-H-A

Code:

```
1 <style type="text/css">
2 <!--
3 a:link
4 a:visited {}
5 a:hover
6 a:active {}
7 -->
8 </style>
```

## 25. 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;}。

## 26. 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。

## 27. 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:

```
1
2 height:auto!important;
3     200px
4 min-height:200px;
5
```

## 28. IE和FireFox 对空格的尺寸解释不同

FireFox为4px, IE为8px; FireFox对div与div之间的空格是忽略的,但是IE是处理的。因此在两个相邻div之间不要有空格跟回车,否则可能造成不同浏览间之间格式不正确,比如著名的3px偏差(多个img标签连着,然后定义float: left;结果在firefox里面正常,而IE里面显示的每个img都相隔了3px。我把标签之间的空格都删除都没有作用。解决方法是在img外面套li,并且对li定义margin: 0; 避免方式:在必要的时候不要无视 list 标签)而且原因难以查明。

## 29. 条件注释

```
1 link "stylesheet" "text/css" "css.css"
2
3 if IE 7
4 <link rel="stylesheet" type="text/css" href="ie7.css" />
5 endif
6
7 if lte IE 6
8 <link rel="stylesheet" type="text/css" href="ie.css" />
9 endif
```

lte — 小于等于

lt — 小于

gte — 大于等于

gt — 大于

! — 不等于

## 30. 强制渲染

```
1 meta Compatible //这句话的意思是强制使用IE7模式
2 来解析网页代码!
3
4 meta Compatible 8
5
6 meta Compatible 1 //Google Chrome Frame也可以
7 让IE用上Chrome的引擎
8
9 meta Compatible 7 EmulateIE7 IE7 mode meta
10 //强制IE8使用IE7模式来
11 解析
12
13 meta Compatible 5 IE6 mode IE5 mode meta
14 //强制IE8使用IE6或IE5模式来解析
15
16 meta Compatible 5 8 //一个特定版本的IE支持所要求
17 的兼容性模式多于一种
```

## 31. js兼容文件

使IE5, IE6兼容到IE7模式(推荐)

```
1 if lt IE 7
2 <script src="http://ie7-js.googlecode.com/svn/version/2.0(beta)/IE7.js"
3 type="text/javascript"></script>
4 endif
5 使IE5,IE6,IE7兼容到IE7模式
6
7 <![if lt IE 8]>
8 <script //ie7-js.googlecode.com/svn/version/2.0(beta)/IE8.js"
9 type="text/javascript"></script>
10 <![endif]->
11 IE8 IE9
12
13 if lt IE 9
14 <script src="http://ie7-js.googlecode.com/svn/version/2.1(beta4)/IE9.js"></script>
15 endif
```

## 32. 浏览器识别符

```
1 p IE6
2 *html p{ color:red; } IE6 专用
3 p
4 p{ *color:red; } IE6,7 专用
5 html p
6 p{+color: red;} IE7 专用
7 Body p IE6
8 p{ color:red\9; } IE8
9
10 Firefox: -moz-
11
12 Opera: -o-
13
```

如果在ie8里浏览乱了把<head>标签下的<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />换成

<meta http-equiv="x-ua-compatible" content="ie=7" />, 这样虽然在ie8里浏览, 实际是用ie7的模式浏览  
再者建适合几个不同浏览器的css样式

```
<link href="css/index.css" rel="stylesheet" type="text/css" />
```

```
<!--[if IE 6]>
```

```
<link rel="stylesheet" type="text/css" href="css/indexIE6.css" media="screen" />
```

这是用链接多个样式表来解决兼容性问题的

以下是不同浏览器的区别, 或许对你有帮助

#### 1. 区别IE和非IE浏览器

```
#tip {  
background:blue; /*非IE 背景蓝色*/  
background:red \9; /*IE6、IE7、IE8背景红色*/  
}
```

#### 2. 区别IE6, IE7, IE8, FF

【区别符号】: 「\9」、「\*」、「\_」

【示例】:

```
#tip {  
background:blue; /*Firefox 背景变蓝色*/  
background:red \9; /*IE8 背景变红色*/  
*background:black; /*IE7 背景变黑色*/  
_background:orange; /*IE6 背景变橘色*/  
}
```

【说明】: 因为IE系列浏览器可读「\9」, 而IE6和IE7可读「\*」(米字号), 另外IE6可辨识「\_」(底线), 因此可以依照顺序写下来, 就会让浏览器正确的读取到自己看得懂得CSS语法, 所以就可以有效区分IE各版本和非IE浏览器(像是Firefox、Opera、Google Chrome、Safari等)。

#### 3. 区别IE6、IE7、Firefox (方法 1)

【区别符号】: 「\*」、「\_」

【示例】:

```
#tip {  
background:blue; /*Firefox背景变蓝色*/  
*background:black; /*IE7 背景变黑色*/  
_background:orange; /*IE6 背景变橘色*/  
}
```

【说明】: IE7和IE6可读「\*」(米字号), IE6又可以读「\_」(底线), 但是IE7却无法读取「\_」, 至于Firefox(非IE浏览器)则完全无法辨识「\*」和「\_」, 因此就可以透过这样的差异性来区分IE6、IE7、Firefox

#### 4. 区别IE6、IE7、Firefox (方法 2)

【区别符号】: 「\*」、「!important」

【示例】:

```
#tip {  
background:blue; /*Firefox 背景变蓝色*/  
*background:green !important; /*IE7 背景变绿色*/  
*background:orange; /*IE6 背景变橘色*/  
}
```

【说明】: IE7可以辨识「\*」和「!important」, 但是IE6只可以辨识「\*」, 却无法辨识「!important」, 至于Firefox可以读取「!important」但不能辨识「\*」因此可以透过这样的差异来有效区隔IE6、IE7、Firefox。

#### 5. 区别IE7、Firefox

【区别符号】: 「\*」、「!important」



【示例】：

```
#tip {
background:blue; /*Firefox 背景变蓝色*/
*background:green !important; /*IE7 背景变绿色*/
}
```

【说明】：因为Firefox可以辨识「!important」但却无法辨识「\*」，而IE7则可以同时看懂「\*」、「!important」，因此可以两个辨识符号来区隔IE7和Firefox。

## 6. 区别IE6、IE7（方法 1）

【区别符号】：「\*」、「\_」

【示例】：

```
#tip {
*background:black; /*IE7 背景变黑色*/
_background:orange; /*IE6 背景变橘色*/
}
```

【说明】：IE7和IE6都可以辨识「\*」（米字号），但IE6可以辨识「\_」（底线），IE7却无法辨识，透过IE7无法读取「\_」的特性就能轻松区隔 IE6和IE7之间的差异。

## 7. 区别IE6、IE7（方法 2）

【区别符号】：「!important」

【示例】：

```
#tip {
background:black !important; /*IE7 背景变黑色*/
background:orange; /*IE6 背景变橘色*/
}
```

【说明】：因为IE7可读取「!important;」但IE6却不行，而CSS的读取步骤是从上到下，因此IE6读取时因无法辨识「!important」而直接跳到下一行读取CSS，所以背景色会呈现橘色。

## 8. 区别IE6、Firefox

【区别符号】：「\_」

【示例】：

```
#tip {
background:black; /*Firefox 背景变黑色*/
_background:orange; /*IE6 背景变橘色*/
}
```

【说明】：因为IE6可以辨识「\_」（底线），但是Firefox却不行，因此可以透过这样的差异来区隔Firefox和IE6，有效达成CSS hack。

---

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 display: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像素的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：`.l{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中内容超过长度以后省略号显示的方法 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>

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%; left: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-align: 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 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 {margin: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- width: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 .tabdl { 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">aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
<script type="text/javascript"> /*  */ function toBreakWord(el, intLen){ var<br/>obj=document.getElementById(el); var strContent=obj.innerHTML; var strTemp=""; while(strContent.length&gt;intLen){<br/>strTemp+=strContent.substr(0,intLen)+"&amp;#10;"; strContent=strContent.substr(intLen,strContent.length); }<br/>strTemp+="&amp;#10;"+strContent; obj.innerHTML=strTemp; } if(document.getElementById &amp;&amp; !document.all)<br/>toBreakWord("ff", 37); /* ]]&gt; */ &lt;/script&gt;</pre>
```

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 onclick="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quirks mode，关于quirks mode、standards mode的相关知识，请参考：<http://www.microsoft.com/china/msdn/library/webservices/asp.net/ASPNETusStan.aspx?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及以下 -->`  
`<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