

# 常见老版本浏览器兼容处理

IE6支持重定义中的!important,

例如:

```
.yuanxin {color:#e00!important;}      .yuanxin {color:#000;}
```

你将会发现定义了样式class="yuanxin"时, 在IE下, 字体显示为红色 (#e00)。但不支持同一定义中的!important。

例如:

```
.yuanxin {color:#e00!important;color:#000}
```

此时在IE6下不支持, 你将会发现定义了样式class="yuanxin"时, 字体显示为黑色 (#000)。

**不包括如下这种形式的同一定义中的!important。**

```
#pageOver{height:expression(document.documentElement.offsetHeight)!important;
height:100%;}
```

此种形式的定义, IE6中是可以解释到important的。

## 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:

```
#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- width: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; //实现同一行排列的效果
display: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 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}
HTML代码
<div id="box">
    <div id="left"></div>
    <div id="right"></div>
</div>
```

## 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padding 时。

例：

```
<div id="box">
<p>p对象中的内容</p>
</div>
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