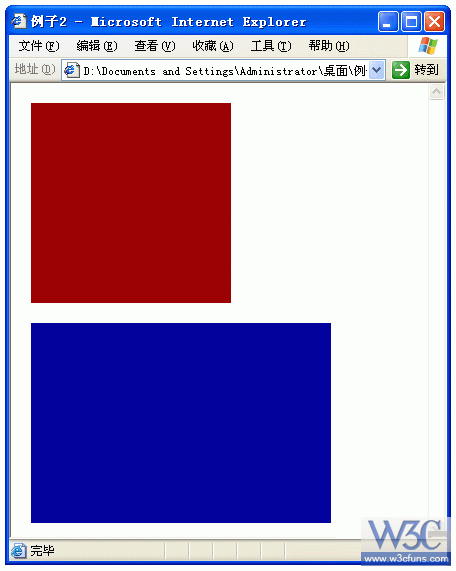
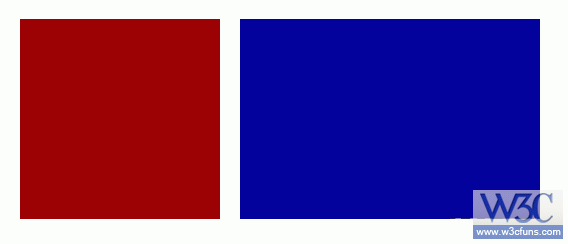
**【第二课】浮动**

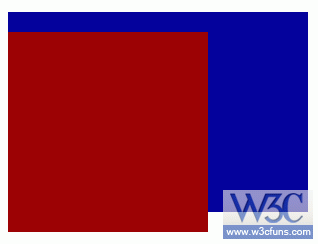
**页面布局有两种方式**  
**1）浮动Float   
2）定位Position**  
  
今天就通过一个小小的练习，通过引导的方式，让大家理解Float的含义。  
  
**【例子】**  
**要求：**  
1）两个方块，一个红色#900，一个蓝色#009；  
2）红色方块宽度和高度均为200像素，蓝色方块 宽度为300像素，高度为200像素；  
3）红色方块和蓝色方块上外边距(margin-top)和左外边距(margin-left)均为20像素；  
  
页面效果如下：



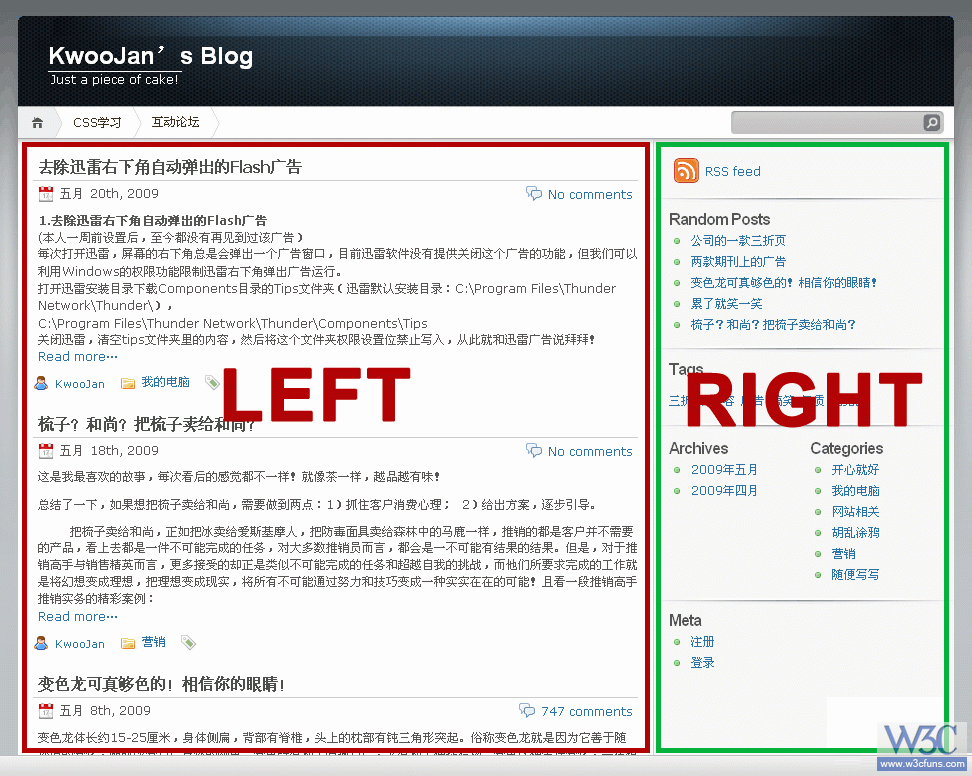
**源代码：** http://www.w3cfuns.com/static/image/filetype/rar.gif [【第二课】浮动 实例.rar](http://www.w3cfuns.com/forum.php?mod=attachment&aid=NTA5fDQxYmM3ZGE5fDE0MDk3NTk2NjZ8MHwyNDg%3D) (1.29 KB, 下载次数: 1706)   
  
  
        大家应该注意到了，虽然红色方块的宽度并不是100%，但是蓝色并未和红色处于同一行，这就是块状元素比较“霸道”的一点，（即使块状元素的宽度不是100%，它也不允许其他元素和他同在一行）为了消除这种“霸权”，让红色和蓝色方块都处在一行，如图



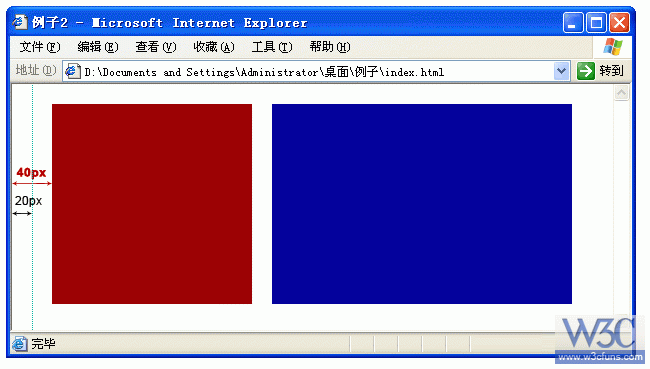
        此时就需要拿出我们的利器Float！只需要在红色方块的CSS里面加上“float:left;”，这时候在IE6中可以看到蓝色方块的确跑到红色方块的后面了，并且处于一行了，但是在FireFox中却变成了如下效果：



        这时候就需要注意了，FF中如果前面的区域浮动了，后面的那个区域很有可能就会和前面的区域发生重叠并错位。  
  
        怎么才能解决这个问题，解决这个浏览器兼容的问题，很容易，只需要在蓝色方块的CSS代码中也加入“**float:left;**”，问题就解决了，加上试试，看看在FF中蓝色方块是不是和红色方块处于一行了~  
  
        到这里，大家应该明白Float的作用了吧，就是为了消除块状元素“霸权主义”的一把利器！在布局页面时有时候是需要消除块状元素霸权主义才能布局好哟，比如KwooJan的博客中间内容部分，分为左边(LEFT)和右边(RIGHT)，就是要用上面这个方法布局的哟，如下图



细心的同学会注意到，在IE6中红色方块距离浏览器的左边距并不是CSS代码中定义的20像素，而是40像素，如下图：



其实这是IE6的一个BUG(IE6双倍边距BUG)，只要满足下面3个条件才会出现这个BUG:  
  
**1）要为块状元素；**  
**2）要左侧浮动；**  
**3）要有左外边距(margin-left)；**  
  
  
解决这个BUG很容易，只需要在相应的块状元素的[CSS](http://www.w3cfuns.cn/)属性中加入“**display:inline;**”，代码如下：

1. #redBlock{
2. width:200px;
3. height:200px;
4. background:#900;
5. margin-top:20px;
6. margin-left:20px;
7. float:left;
8. display:inline;
9. }

复制代码

现在再看看，是不是IE6和FF显示一样了呢~  
  
呵呵，这节课也比较容易吧，如果大家有不明白的可以留言，我会做进一步解释。  
下节课，我们讲讲“浮动清除(Clear)”问题！  
  
**最终代码：** http://www.w3cfuns.com/static/image/filetype/rar.gif [【第二课】浮动 实例 最终代码.rar](http://www.w3cfuns.com/forum.php?mod=attachment&aid=NTE1fDY4ZTExMDU4fDE0MDk3NTk2NjZ8MHwyNDg%3D) (1.31 KB, 下载次数: 1496)   
  
精简后的CSS代码加载更快，大家一看就明白了^\_^