[**浮动元素的两端对齐**](http://www.ddcat.net/blog/?p=1199)

当我们使用float来使元素并排显示的时候，可以使用margin来控制元素之间的距离，而在很多版式里（例如产品图片的列表），需要浮动的元素达到两端对齐的效果，如图1所示。

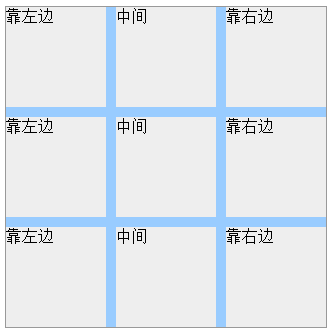
[](http://www.ddcat.net/web/css/float-justify/float-justify-01-v.png)

图1 两端对齐的版式

单纯使用float:left或者float:right，而不添加额外的class区分元素的位置，似乎是无法实现靠左/右的效果。

首先列出结构：

<div class="justify">

<ul>

<li>靠左边</li>

<li>中间</li>

<li>靠右边</li>

<li>靠左边</li>

<li>中间</li>

<li>靠右边</li>

<li>靠左边</li>

<li>中间</li>

<li>靠右边</li>

</ul>

</div>

思路是：li不需要单独的class来设定左右，而是通过扩大ul的宽度和高度，使其溢出父层，再通过父层的overflow:hidden来隐藏多出容器的部分。

CSS

.justify {

width:320px; /\* 1行3个li，li之间距离10px = 100px\*3 + 10px\*2 \*/

overflow:hidden;

margin:10px;

border:1px solid #999999;

}

.justify ul {

width:330px; /\* 可容纳下3列的宽度 \*/

margin-bottom:-10px;/\* 隐藏掉最下面一行的margin-bottom \*/

overflow:hidden;

zoom:1;/\* 触发IE的Layout \*/

}

\*+html .justify ul {margin-bottom:0;}/\* 针对IE7中最后1行li的margin-bottom失效 \*/

.justify li {

display:inline;float:left;list-style:none;

width:100px;height:100px;margin:5px;

background:#EEEEEE;}

其实，用图2来解释更直观一些：

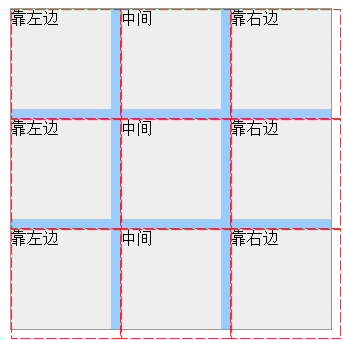
[](http://www.ddcat.net/web/css/float-justify/float-justify-02-v.png)

图2 实现原理

提示：IE7内最后一行li的margin-bottom应该会被IE吃掉，所以ul就不需要margin-bottom了。

此处需注意的是调整div的宽度的同时也要调整ul的宽度。

**或者不设定ul的宽度而是用负marign值来达到扩大ul宽度的效果。**

非浮动和定位的块级元素，其width值为auto（默认值）时，会尽量充满父元素的内容框，也就是说当没有border/padding/margin时，width是100%。  
计算公式：  
margin-left+border-left+padding-left+width+padding-right+border-right+margin-rigth=包含块的宽度（div的width）  
因此如果设定ul的margin-right为-10px，其他padding/margin/border都为0，则：  
ul的width=320px（div的width）-(-10px)=330px  
代码如下：

.justify {......}

.justify ul {

margin:0 -10px -10px 0;

overflow:hidden;

zoom:1;/\* 触发IE的Layout \*/

}

.justify li {......}

这样的优点在于，只要调整最外层div的宽度，就可以实现2列或者4列的效果。

[——查看示例文件——](http://www.ddcat.net/web/css/float-justify/float-justify.html)