[左右浮动 clearfix 1](#_Toc411511672)

[display:inline-block 3](#_Toc411511673)

[Margin 与 Padding 的区别 4](#_Toc411511674)

[border 属性 7](#_Toc411511675)

[<span>标签总结说明 7](#_Toc411511676)

[html dl dt dd标签元素语法结构与使用 7](#_Toc411511677)

[i标签 7](#_Toc411511678)

[CSS vertical-align 属性 8](#_Toc411511679)

[position：层级 优先级 8](#_Toc411511680)

[CSS网页布局基础 10](#_Toc411511681)

[浮动布局 13](#_Toc411511682)

[理解浮动 13](#_Toc411511683)

[清除浮动 13](#_Toc411511684)

[案例：横向两列布局 13](#_Toc411511685)

# 左右浮动 clearfix

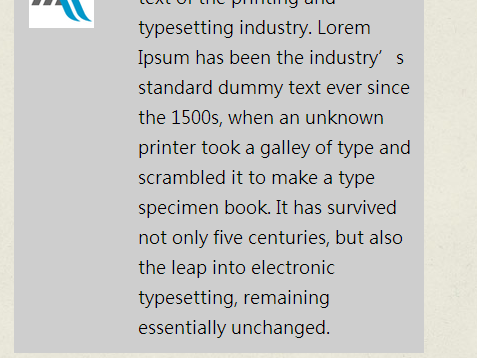
清除浮动方法：  
1、clear：both;（给浮动元素的后面的元素使用）。一个元素比如左浮动，除了它自己向左浮动，还对紧邻的元素产生影响， 并且只对它紧邻的元素和父级包含块产生影响。 只有该元素受到了浮动的影响才来清浮动。

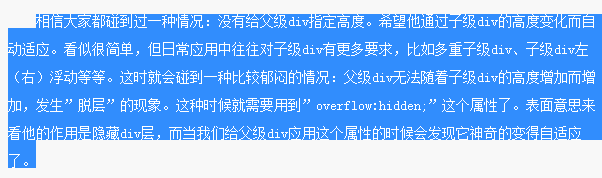
当父包含块缩成一条时，用clear both方法无效，它一般用于紧邻后面的元素清除浮动。

**Div的高度一般不要设置。**

2、width:100%;overflow:hidden;（给浮动元素的包裹层使用，常用于因未设置高度而看不见的包裹层。）







浮动：浮动的框可以左右移动，直至它的外边缘遇到包含框或者另一个浮动框的边缘。

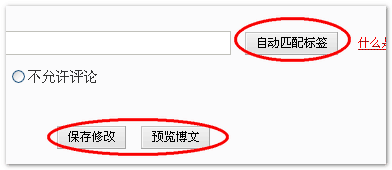
正是因为浮动的这种特性，导致本属于普通流中的元素浮动之后，包含框内部由于不存在其他普通流元素了，也就表现出高度为0（高度塌陷）。在实际布局中，往往这并不是我们所希望的，所以需要闭合浮动元素，使其包含框表现出正常的高度。

无定位值的absolute元素就是个连实际宽度也没有的float浮动元素就可以了。

先回答下开始悬而位答的问题：position为absolute的元素如果没有设置left, top等值与left:0;top:0;的的效果为何不一样？相信您应该知道答案了，例如一个div中有个absolute属性元素，其没有left或是top值，其会像个普通的inline-block属性元素一样静静地呆在这个div里面，但是一旦设置了left:0;top:0;对不起，这个absolute元素立马变身，直接从DOM tree里面脱离，独立于文档流，结果相对于最近的relative属性的祖先标签定位（如果没有，就body定位）。由于我们平时使用absolute都离不开left，top之类的值，所以才会概念不清。

# display:inline-block

display:inline-block某种意义上的作用就是包裹(wrap)，而浮动也有类似的效果。举个常见例子，或许您有实现宽度自适应按钮的经验，实现宽度自适应的关键就是要让按钮的大小自适应于文字的个数，这就需要按钮要自动包裹在文字的外面。我们用什么方法实现呢？一就是display:inline-block；二就是float。例如我们要实现新浪博客中的“发表文章”之类的宽度自适应按钮（如下图）:

背景图片为：http://www.zhangxinxu.com/study/image/sina_gray_btn.png

***display:inline-block方法***

CSS代码如下：

.btn1{**display:inline-block;** background:url(http://www.zhangxinxu.com/study/image/sina\_gray\_btn.png) no-repeat left top; padding-left:3px; color:#000000; font-size:12px; text-decoration:none;}

.btn1 cite{display:block; line-height:26px; padding:0 13px 0 10px; background:url(http://www.zhangxinxu.com/study/image/sina\_gray\_btn.png) no-repeat right top;}

HTML代码如下：

<a href="javascript:void(0);" class="btn1"><cite>inline-block方法</cite></a>

结果如下图：

inline-block方法实现的宽度自适应按钮

然而，float无法等同于display:inline-block，其中原因之一就是浮动的方向性，display:inline-block仅仅一个水平排列方向，就是从左往右，而float可以从右往左排列，这就是两者的差异

# [Margin 与 Padding 的区别](http://www.cnblogs.com/tianyue3107/archive/2009/04/22/1441358.html)

1.**Margin**

   用来设置页面中一个元素所占空间的边缘到相邻元素之间的距离.

   如果提供全部四个参数值，将按上－右－下－左的顺序作用于四边。  
   如果只提供一个，将用于全部的四边。  
   如果提供两个，第一个用于上－下，第二个用于左－右。  
   如果提供三个，第一个用于上，第二个用于左－右，第三个用于下。

   在IE4+，margin属性不可用于td和tr对象。

   用法:body { margin: 36pt 24pt 36pt; }

2.**Padding**

   用来设置元素内容到元素边界的距离。

   如果提供全部四个参数值，将按上－右－下－左的顺序作用于四边。  
   如果只提供一个，将用于全部的四条边。  
   如果提供两个，第一个用于上－下，第二个用于左－右。  
   如果提供三个，第一个用于上，第二个用于左－右，第三个用于下。  
   内联对象要使用该属性，必须先设定对象的[height](http://www.cnblogs.com/tianyue3107/admin/mk:@MSITStore:E:/CSS/%E6%95%99%E6%9D%90/CSS%202.0%20%E4%B8%AD%E6%96%87%E6%89%8B%E5%86%8C.chm::/css2/c_height.html)或[width](http://www.cnblogs.com/tianyue3107/admin/mk:@MSITStore:E:/CSS/%E6%95%99%E6%9D%90/CSS%202.0%20%E4%B8%AD%E6%96%87%E6%89%8B%E5%86%8C.chm::/css2/c_width.html)属性，或者设定[position](http://www.cnblogs.com/tianyue3107/admin/mk:@MSITStore:E:/CSS/%E6%95%99%E6%9D%90/CSS%202.0%20%E4%B8%AD%E6%96%87%E6%89%8B%E5%86%8C.chm::/css2/c_position.html)属性为absolute。  
   不允许负值。

   用法:body { padding: 36pt 24pt 36pt; }

何时应当使用margin：  
需要在border外侧添加空白时。  
空白处不需要背景（色）时。  
上下相连的两个盒子之间的空白，需要相互抵消时。如15px + 20px的margin，将得到20px的空白。

何时应当时用padding：  
需要在border内测添加空白时。  
空白处需要背景（色）时。  
上下相连的两个盒子之间的空白，希望等于两者之和时。如15px + 20px的padding，将得到35px的空白。

效果预览：

核心分析：

上面那个类似于商城热销商品邮戳效果其实也是由负margin实现的，把以往使用的relative/absolute这样的“危险”属性代替为相对“温和”的margin属性。核心在于小的邮戳图片利用负margin-top向上移动，达到类似absolute的叠加效果。

CSS代码：

|  |  |
| --- | --- |
| 1  2  3  4 | .demoSpecItem{width:212px; min-height:178px; \_height:178px;}  .demoSpecItem .demoItemPic{display:block; text-align:center;}  .demoSpecItem .demoItemPic img{padding:5px; border:1px solid #ccc;}  .demoSpecItem .demoIcon{display:block; margin:-185px 0 0 150px;} |

HTML代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <div class="demoSpecItem">    <a href="javascript:void(0)" class="demoItemPic">      <img width="200" height="166" src="toygersKittens.jpg" alt="toygers kittens" />    </a>    <span class="demoIcon"><img width="54" height="54" src="buyInSoon.png" alt="icon" /></span>  </div> |

**利用负margin制作鳞片式导航：**

效果预览：

* [**鳞片导航**](javascript:void(0))
* [**前端开发**](javascript:void(0))
* [**用户体验**](javascript:void(0))
* [**实战为主**](javascript:void(0))

核心分析：

鳞片式导航可以说是使用负margin技术最佳场合之一。如上方的例子，试着想象不使用负margin，单纯的使用图片来制作这类鳞片式导航，想象下切图、定位、点击后图片切换、中间接壤部分的遮罩、维护开发的工作量，光想想这些就不寒而栗。而利用负margin这里仅仅添加一个margin-left即可达到前后遮盖的效果。

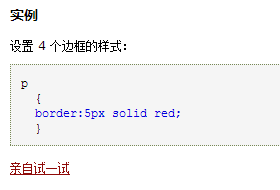
CSS代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | .demoNav .demoList{float:left; width:154px; height:53px; margin:0 0 0 -20px;}  .demoNav .demoList.firstList{margin:0;}  .demoNav .demoList a{display:block; line-height:53px; padding:0 0 0 25px; color:#005590; background:url("flakeNavTab.png") left top no-repeat; font-weight:bold; text-decoration:none;}  .demoNav .demoList a:hover{background:url("flakeNavTabHover.png") left top no-repeat;}  .demoNav .demoList.current a{background:url("flakeNavTabCurrent.png") left top no-repeat;}  .demoNav .demoList.current a:hover{background:url("flakeNavTabCurrent.png") left top no-repeat;}/\* fix ie6 hover issue \*/ |

HTML代码：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <div id="demoNav" class="demoNav">    <ul class="clearfix">      <li class="demoList firstList current"><a href="javascript:void(0)">鳞片导航</a></li>      <li class="demoList"><a href="javascript:void(0)">前端开发</a></li>      <li class="demoList"><a href="javascript:void(0)">用户体验</a></li>      <li class="demoList"><a href="javascript:void(0)">实战为主</a></li>    </ul>  </div> |

# border 属性



# <span>标签总结说明

span在[html](http://www.divcss5.com/html/" \t "_blank)中常用的布局标签,与[div标签](http://www.divcss5.com/html/h121.shtml" \t "_blank)区别在于，span随内容而占用高宽空间（紧贴内容），而一对div标签却占用一行。

# [html dl dt dd标签元素语法结构与使用](http://www.divcss5.com/html/h88.shtml)



**dl 内容块**

# i标签

定义和用法

<i> 标签显示斜体文本效果。

<i> 标签和基于内容的样式标签 [<em>](http://www.w3school.com.cn/tags/tag_em.asp) 类似。它告诉浏览器将包含其中的文本以斜体字（italic）或者倾斜（oblique）字体显示。如果这种斜体字对该浏览器不可用的话，可以使用高亮、反白或加下划线等样式。

**提示：**<i> 标签一定要和结束标签 </i> 结合起来使用。

# CSS vertical-align 属性

定义和用法

vertical-align 属性设置元素的垂直对齐方式。

说明

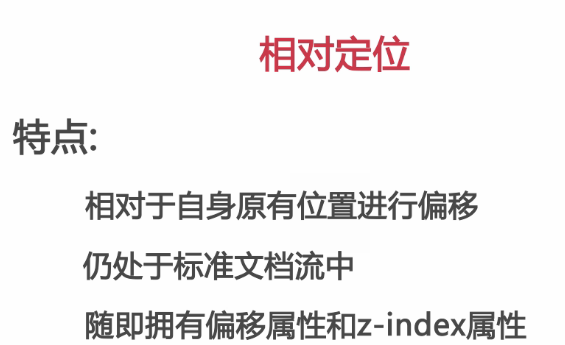
该属性定义行内元素的基线相对于该元素所在行的基线的垂直对齐。允许指定负长度值和百分比值。这会使元素降低而不是升高。在表单元格中，这个属性会设置单元格框中的单元格内容的对齐方式。

可能的值

|  |  |
| --- | --- |
| **值** | **描述** |
| baseline | 默认。元素放置在父元素的基线上。 |
| sub | 垂直对齐文本的下标。 |
| super | 垂直对齐文本的上标 |
| top | 把元素的顶端与行中最高元素的顶端对齐 |
| text-top | 把元素的顶端与父元素字体的顶端对齐 |
| middle | 把此元素放置在父元素的中部。 |
| bottom | 把元素的顶端与行中最低的元素的顶端对齐。 |
| text-bottom | 把元素的底端与父元素字体的底端对齐。 |
| length |  |
| % | 使用 "line-height" 属性的百分比值来排列此元素。允许使用负值。 |
| inherit | 规定应该从父元素继承 vertical-align 属性的值。 |

# position：层级 优先级

通过设置position来实现绝对定位布局。：带有遮罩层效果的提示框.固定层效果.全屏广告. 

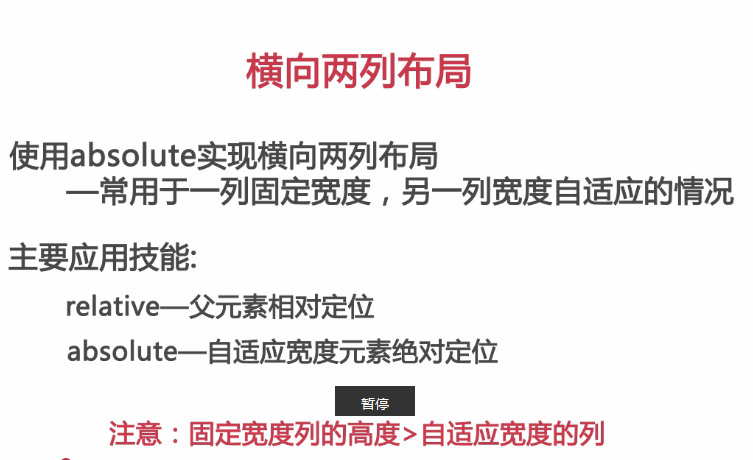


设置z轴 相对定位会产生层堆叠

绝对定位。

特点：会从标准文档流中脱离。

父级relative 子级absolute 其实子级就突出来了 并且脱离了标准文档。会根据跟的偏移量 根据父级的位置偏移（没有父级 就根据浏览器屏幕偏移）





# [CSS网页布局基础](http://fy98.com/2014/11/30/css-layout/)

By [Frank Fan](http://fy98.com/about)

 发表于 2014-11-30

**文章目录**

1. [1. CSS基础](http://fy98.com/2014/11/30/css-layout/#CSS基础)
2. [2. 浮动布局](http://fy98.com/2014/11/30/css-layout/#浮动布局)
   1. [2.1. 理解浮动](http://fy98.com/2014/11/30/css-layout/#理解浮动)
   2. [2.2. 清除浮动](http://fy98.com/2014/11/30/css-layout/#清除浮动)
   3. [2.3. 案例：横向两列布局](http://fy98.com/2014/11/30/css-layout/#案例：横向两列布局)

CSS基础

网页布局分类

1. 流式布局
2. 浮动布局
3. 绝对定位布局

需要理解：

* 标准文档流
* 盒子模型
* float属性
* position属性

W3C标准是指由万维网联盟指定的一系列标准，包括：

* 结构化标准语言（HTML和XML）
* 表现标准语言（CSS）
* 行为标准语言（DOM和ECMAScript）

W3C倡导结构、样式和行为分离。

在CSS中，存在3中定位机制：

1. 标准文档流（Normal flow）
2. 浮动（Floats）
3. 绝对定位（Absolute positioning）

标准文档流  
特点：  
从上到下、从左到右输出文档内容，  
由块级元素和行级元素组成，可以实现一列布局。

块级元素的特点是：  
从左到右撑满页面，独占一行；  
触碰到页面边缘时，会自动换行。

常见的块级元素有 div、ul、li、dl、dt、p 等

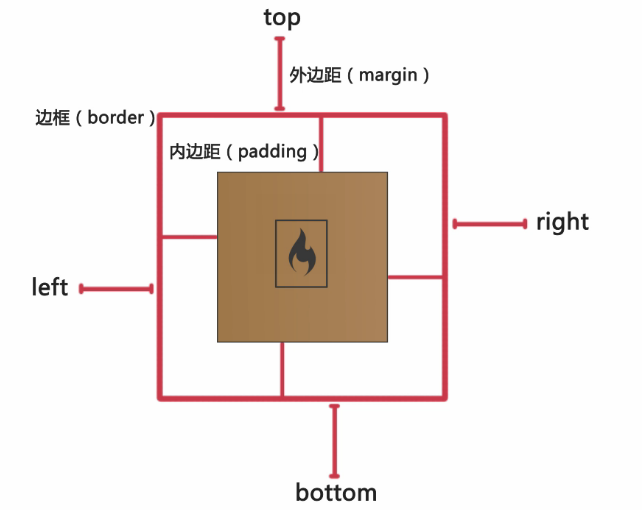
行级元素的特点：  
能在同一行内显示，  
不会改变HTML文档结构

常见的行级标签有 span、strong、img、input 等

块级元素和行级元素都是盒子模型。

盒子模型是网页布局的基石，盒子模型由4部分组成：

* 边框（border）
* 外边距（margin）
* 内边距（padding）
* 内存（content）

[](http://images.cnitblog.com/blog/282019/201412/061903416238106)

盒子模型属性设置4个值时表示 上-右-下-左 ，  
设置3个值表示 上-左右-下  
设置2个值表示 上下-左右  
设置1个值表示 上下左右

CSS优先级采用就近原则：行内样式 > 内部样式 > 外部样式

## 浮动布局

### 理解浮动

float是CSS中规定的第二种定位机制，能够实现横向多列布局。  
float: left | right | none, 浮动的特点是元素会左移或者右移，  
知道触碰到容器边缘为止。

注意：设置了浮动的元素，仍旧处于标准文档流中，也就意味着它仍旧  
占据着标准文档流中的空间，会影响到其他（特指紧挨着浮动元素的后一个元素）元素的布局。

### 清除浮动

方法1：  
clear: left | right | both属性, 常用 clear: both;

方法2：  
为元素同时设置 width: 100% 或者固定宽度 + overflow:hidden;

### 案例：横向两列布局

主要技能：

* float属性——使纵向排列的块级元素，横向排列
* margin属性——设置两列之间的间距

html

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | <body>  <div id="wrap">  <div id="header">头部</div>  <div id="mainbody">  <div class="left"></div>  <div class="right"></div>  </div>  <div id="footer">版权部分 copyright 2014</div>  </div>  </body> |

css

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | \*{ margin: 0; padding: 0;}  #wrap{background: #00C; margin: 0 auto; width: 960px;}  #header{background: #FF3300; width: 100%;}  #mainbody{background: #FC0; width: 100%; overflow: hidden;}  .left{width: 800px; height: 200px; background: #000; float: left;}  .right{width: 140px; height: 500px; background: #690; float: right;}  #footer{background-color: #639; width: 100%;} |

效果图：  
[](http://images.cnitblog.com/blog/282019/201412/062048378891094)

[css](http://fy98.com/categories/css/)

[**上一篇：**  
 javascript中的正则表达式](http://fy98.com/2014/12/01/regex-in-javascript/)

[**下一篇：**  
通过nodejs、Express框架搭建Web](http://fy98.com/2014/11/25/build-web-with-node-express-js/)

最新最早最热

* [1条评论](javascript:void(0);)
* 1

1

[织]

1月20日[回复顶转发](javascript:void(0);)

社交帐号登录:

* [微博](http://oobigo.duoshuo.com/login/weibo/)
* [QQ](http://oobigo.duoshuo.com/login/qq/)
* [人人](http://oobigo.duoshuo.com/login/renren/)
* [豆瓣](http://oobigo.duoshuo.com/login/douban/)
* [更多»](javascript:void(0))

[http://static.duoshuo.com/images/noavatar_default.png](javascript:void(0);)

窗体顶端



发布

窗体底端

[FrankFan正在使用多说](http://duoshuo.com/)

分类

* [.net1](http://fy98.com/categories/net/)
* [css3](http://fy98.com/categories/css/)
* [database1](http://fy98.com/categories/database/)
* [github6](http://fy98.com/categories/github/)
* [javascript16](http://fy98.com/categories/javascript/)
* [myworks1](http://fy98.com/categories/myworks/)
* [node1](http://fy98.com/categories/node/)
* [其他1](http://fy98.com/categories/%E5%85%B6%E4%BB%96/)
* [前端8](http://fy98.com/categories/%E5%89%8D%E7%AB%AF/)

[RSS 订阅](http://fy98.com/atom.xml)

友情链接

* [盛利的博客 · blog](http://oopsoutofmemory.github.io/)
* [刘伟（Derick）的博客](http://liuwei.so/)

Powered by [hexo](http://zespia.tw/hexo/" \o "hexo" \t "_blank) © 2015 [Frank Fan](http://fy98.com/about)

http://fy98.com/img/scrollup.png