

Margin 的特性

margin 始终是透明的。

margin 通过使用单独的属性，可以对上、右、下、左的外边距进行设置。即：margin-top、margin-right、margin-bottom、margin-left。

也可以使用简写的外边距属性同时改变所有的外边距：margin: top right bottom left;(eg: margin:10px 20px 30px 40px) 记忆方式是元素周围正上方顺时针“上右下左”记忆。

并且规范还提供了省略的数值写法，基本如下：

1、如果 margin 只有一个值，表示上右下左的 margin 同为这个值。例如：margin:10px; 就等于 margin:10px 10px 10px 10px;a

2、如果 margin 只有两个值，第一个值表示上下 margin 值，第二个值为左右 margin 的值。例如：margin:10px 20px; 就等于 margin:10px 20px 10px 20px;

3、如果 margin 有三个值，第一个值表示上 margin 值，第二个值表示左右 margin 的值，第三个值表示下 margin 的值。例如：margin:10px 20px 30px; 就等于 margin:10px 20px 30px 20px;

4、如果 margin 有四个值，那这四个值分别对应上右下左这四个 margin 值。例如：margin:10px 20px 30px 40px;

Margin 垂直外边距合并

外边距合并指的是，当两个垂直外边距相遇时，它们将形成一个外边距。合并后的外边距的高度等于两个发生合并的外边距的高度中的较大者。

垂直外边距合并问题常见于第一个子元素的 margin-top 会顶开父元素与父元素相邻元素的间距，而且只在标准浏览器下 (Firefox、Chrome、Opera、Safari) 产生问题，IE 下反而表现良好。例子可以查看下面代码(IE 下表现“正常”，标准浏览器下查看出现“bug”)：

示例：

```
.top{width:160px; height:50px; background:#ccf;}  
.middle{width:160px; background:#cfc;}  
.middle .firstChild{margin-top:20px;}  
<div class="top"></div>  
<div class="middle">
```

```
<div class="firstChild">我其实只是想和我的父元素隔开点距离。</div>

<div class="secondChild"></div>

</div>
```



如果按照 CSS 规范，IE 的“良好表现”其实是一个错误的表现，因为 IE 的 `hasLayout` 渲染导致了这个“表现良好”的外观。而其他标准浏览器则会表现出“有问题”的外观。好了，如果你读过了上面 W3Shcool 的 CSS 外边距合并的文章后，就很容易讨论这个问题了。这个问题发生的原因是根据规范，一个盒子如果没有上补白(`padding-top`)和上边框(`border-top`)，那么这个盒子的上边距会和其内部文档流中的第一个子元素的上边距重叠。

再说了白点就是：父元素的第一个子元素的上边距 `margin-top` 如果碰不到有效的 `border` 或者 `padding` 就会不断一层一层的找自己“领导”(父元素，祖先元素)的麻烦。只要给领导设置个有效的 `border` 或者 `padding` 就可以有效的管制这个目无领导的 `margin` 防止它越级，假传圣旨，把自己的 `margin` 当领导的 `margin` 执行。

对于垂直外边距合并的解决方案上面已经解释了，为父元素例子中的 `middle` 元素增加一个 `border-top` 或者 `padding-top` 即可解决这个问题。

margin 在块元素、内联元素中的区别

HTML（这里说的是 html 标准，而不是 xhtml）里分两种基本元素，即 `block` 和 `inline`。顾名思义，`block` 元素就是以“块”表现的元素（`block-like elements`），`inline` 元素即是以“行”表现的元素（`character level elements and text strings`）。二者表现的主要差别在于，在页面文档中 `block` 元素另起一行开始，并独占一行。`inline` 元素则同其他 `inline` 元素共处一行。

`block` 元素（块元素）大致有：`P`|`H1`|`H2`|`H3`|`H4`|`H5`|`H6`|`UL`|`OL`|`PRE`|`DL`|`DIV`|`NOSCRIPT`|`BLOCKQUOTE`|`FORM`|`HR`|`TABLE`|`FIELDSET`|`ADDRESS`(随着 html5 标准的推进，一些元素将被废除，而一些新的元素将被引入)注意的是并非所有的 `block` 元素的默认 `display` 属性都是 `block`，像 `table` 这种 `display:table` 的元素也是 `block` 元素。

`inline` 元素（内联元素）大致有：`#PCDATA`（即文本）|`TT`|`I`|`B`|`BIG`|`SMALL`|`EM`|

STRONG | DFN | CODE | SAMP | KBD | VAR | CITE | ABBR | ACRONYM | A | IMG | OBJECT |
BR | SCRIPT | MAP | Q | SUB | SUP | SPAN | BDO | INPUT | SELECT | TEXTAREA | LABEL |
BUTTON

其中有类特殊的元素：如 `img|input|select|textarea|button|label` 等，他们被称为可替换元素（Replaced element）。他们区别一般 inline 元素（相对而言，称 non-replaced element）是：这些元素拥有内在尺寸(intrinsic dimensions),他们可以设置 width/height 属性。他们的性质同设置了 `display:inline-block` 的元素一致。

或许有朋友对非替换元素（non-replaced element）有点疑惑，稍微帮助大家理解一下。非替换元素，W3C 中没有给出明确的定义，但从字面可以理解到，非替换元素对应着替换元素（replaced element），也就是说我们搞懂了替换元素的含义，就懂了非替换元素。替换元素，W3C 中给出了定义：

“An element that is outside the scope of the CSS formatter, such as an image, embedded document, or applet”

从定义中我们可以理解到，替换元素（replaced element）主要是指 `img, input, textarea, select, object` 等这类默认就有 CSS 格式化外表范围的元素。进而可知，非替换元素（non-replaced element）就是除了 `img, input, textarea, select, object` 等替换元素以外的元素。

`margin` 在块级元素下，他的性能可以完全体现，上下左右任你设定。且记住块级元素的 `margin` 的参照基准是前一个元素即相对于自身之前的元素有 `margin` 距离。如果元素是第一个元素，则就是相对于父元素的 `margin` 距离（但第一个元素相对于父元素 `margin-top` 而父元素又没有设定 `padding-top/border-top` 的话需要印证上面的垂直外边距合并的知识）

`margin` 也能用于内联元素，这是规范所允许的，但是 `margin-top` 和 `margin-bottom` 对内联元素（对行）的高度没有影响，并且由于边界效果(`margin` 效果)是透明的，他也没有任何的视觉影响。

这是因为边界应用于内联元素时不改变元素的行高度，如果你要改变内联元素的行高即类似文本的行间距，那么你能使用这三个属性：`line-height`, `font-size`, `vertical-align`。请记住，这个影响内联元素高度的是 `line-height` 而不是 `height`，因为内联元素是一行行的，定一个 `height` 的话，那这到底是整段 inline 元素的高呢？还是 inline 元素一行的高呢？这都不准，所以统一都给每行定一个高，只能是 `line-height` 了。

`margin-top/margin-bottom` 对内联元素没有多大实际效果，不过 `margin-left/margin-right` 还是能够对内联元素产生影响的。应用 `margin:10px 20px 30px 40px;`，左边这个 css 如果写

在 `inline` 元素上，他的效果大致是，上下无效果，左边离他相邻元素或者文本距离为 40px，右边离他相邻元素或者文本距离为 20px。你可以自行尝试一番。

最后在内联元素中还有上文我们提到的非可替换 `inline` 元素（`non-replaced element`），这些个元素 `img|input|select|textarea|button|label` 虽然是内联元素，但 `margin` 依旧可以影响到他的上下左右！

总结下来 `margin` 属性可以应用于几乎所有的元素，除了表格显示类型（不包括 `table-caption`, `table` and `inline-table`）的元素，而且垂直外边距对非替换内联元素（`non-replaced inline element`）不起作用。