你真的了解margin吗？你知道margin有什么特性吗？你知道什么是垂直外边距合并？margin在块元素、内联元素中的区别？什么时候该用padding而不是margin？你知道负margin吗？你知道负margin在实际工作中的用途吗？常见的浏览器下margin出现的bug有哪些？……

写css，你少不了与margin打交道，而对于这个平时我们最常用的css属性我们并非十分了解。介于此我打算写下这篇文章，一来是自己工作中的总结，也是对自己知识的一次梳理。

**Margin是什么**

CSS 边距属性定义元素周围的空间。通过使用单独的属性，可以对上、右、下、左的外边距进行设置。也可以使用简写的外边距属性同时改变所有的外边距。——W3School

边界，元素周围生成额外的空白区。“空白区”通常是指其他元素不能出现且父元素背景可见的区域。——CSS权威指南

我比较喜欢使用“外边距”这个词来解释margin（同理padding可以称之为“内边距”，但是我又恰恰喜欢称呼padding为“补白”或者“留白”），我们可以很清楚的了解到margin的最基本用途就是控制元素周围空间的间隔，从视觉角度上达到相互隔开的目的。

**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。

外边距的 margin-width 的值类型有：auto | length | percentage

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;

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;

在实际应用中，个人不推荐使用三个值的margin，一是容易记错，二是不容易日后修改，一开始如果写成margin:10px 20px 30px;日后需求改动为上10px，右30px，下30px，左20px，你不得不还是得把这个margin拆开为margin:10px 30px 30px 20px;费力且不讨好，不如一开始就老老实实的写成margin:10px 20px 30px 20px;来的实在，不要为了现在节省俩个字节而让日后再次开发的成本上升。

**垂直外边距合并问题**

别被上面这个名词给吓倒了，简单地说，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。你可以查看[W3Shool CSS外边距合并](http://www.w3school.com.cn/css/css_margin_collapsing.asp)了解这个基本知识。

实际工作中，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(FirfFox、Chrome、Opera、Sarfi)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”)：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | <html xmlns="<http://www.w3.org/1999/xhtml>">  <head>  <title>垂直外边距合并</title>  <style>  .top{width:160px; height:50px; background:#ccf;}  .middle{width:160px; background:#cfc;}  .middle .firstChild{margin-top:20px;}  </style>  </head>    <body>  <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>  </body>  </html> |

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。

一般说来这个问题解释到这里，大多数文章就不会再深入下去了，但作为一名实战开发者，最求的是知其然知其所以然，原本使用margin-top就是为了与父元素隔开距离，而按照你这么一个解法，其实是一种“修复”，为了“弥补修复”这个父子垂直外边距合并这个CSS规范“Bug”，而强制在父元素上使用border-top和padding-top，不舒服，也不容易记住，下次再发生这样的情况还是会忘记这条准则，而且在页面设计稿里如果不需要border-top加个上边框，这么一加反而画蛇添足，为以后修改留下隐患。

为什么一定要用border-top,padding-top去为了这么一个所谓的标准规范而多写这么一行代码呢？答案你可以参考另外一篇文章[用Margin还是用Padding](http://hicss.net/use-margin-or-padding/)里找到答案。

**用Margin还是用Padding**

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。

个人认为：**margin是用来隔开元素与元素的间距；padding是用来隔开元素与内容的间隔。margin用于布局分开元素使元素与元素互不相干；padding用于元素与内容之间的间隔，让内容（文字）与（包裹）元素之间有一段“呼吸距离”。**

这里我截取了部分另外一篇文章的内容，详细内容请见[用Margin还是用Padding](http://hicss.net/use-margin-or-padding/)

**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，他们被称为可置换元素（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。

或许有朋友对非置换元素（non-replaced element）有点疑惑，稍微帮助大家理解一下。非置换元素，W3C 中没有给出明确的定义，但我们从字面可以理解到，非置换元素对应着置换元素（replaced element），也就是说我们搞懂了置换元素的含义，就懂了非置换元素。置换元素，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，置换元素（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，非置换元素（non-replaced element）就是除了 img, input, textarea, select, object 等置换元素以外的元素。

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的话要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，那么你只能使用这三个属性：line-height，fong-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？这都说不准，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置换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置换内联元素（non-replaced inline element）不起作用。**

**负margin技术及其应用**

在margin所有的实际应用中，负margin技术是我学习css路上最重要一课之一，许多高级应用和页面上的疑难杂症都可以用负margin技术来实现。margin技术是那么的有用，限于篇幅我又不想草草了事，所以我决定专门为他写一篇文章，详细的说明他的效果、原理、及其应用。在此之前你可以先阅读怿飞写的[由浅入深漫谈margin属性](http://www.planabc.net/2007/03/18/css_attribute_margin/)这篇文章，大致了解“margin参考线”的概念，之后再来查看[负margin技术及其应用](http://www.hicss.net/i-know-you-do-not-know-the-negative-margin/)这篇文章。

**常见的浏览器下margin出现的bug**

林林总总写了那么多，最后总结一些浏览器中常见的margin Bug吧，以后遇到margin下的布局问题可以查看这里找到解决的方案，如果你还发现其他关于浏览器下margin的Bug你可以发表留言，核对采纳后我会及时添加进去，感谢你的分享：

**IE6中双边距Bug：**  
发生场合：当给父元素内第一个浮动元素设置margin-left（元素float:left）或margin-right（元素float:right）时margin加倍。  
解决方法：是给浮动元素加上display:inline;CSS属性；或者用padding-left代替margin-left。  
原理分析：块级对象默认的display属性值是block，当设置了浮动的同时，还设置了它的外边距就会出现这种情况。也许你会问：“为什么之后的对象和第一个对象之间就不存在双倍边距的Bug”？因为浮动都有其相对应的对象，只有相对于其父对象的浮动对象才会出现这样的问题。第一个对象是相对父对象的，而之后对象是相对第一个对象的，所以之后对象在设置后不会出现问题。为什么display:inline可以解决这个双边距bug，首先是inline元素或inline-block元素是不存在双边距问题的。然后，float:left等浮动属性可以让inline元素haslayout，会让inline元素表现得跟inline-block元素的特性一样，支持高宽，垂直margin和padding等，所以div class的所有样式可以用在这个display inline的元素上。

**IE6中浮动元素3px间隔Bug：**  
发生场合：发生在一个元素浮动，然后一个不浮动的元素自然上浮与之靠近会出现的3px的bug。  
解决方法：右边元素也一起浮动；或者为右边元素添加IE6 Hack \_margin-left:-3px;从而消除3px间距。  
原理分析：IE6浏览器缺陷Bug。

**IE6/7负margin隐藏Bug：**  
发生场合：当给一个有hasLayout的父元素内的非hasLayout元素设置负margin时，超出父元素部分不可见。  
解决方法：去掉父元素的hasLayout；或者赋hasLayout给子元素,并添加position:relative;  
原理分析：IE6/7独有的hasLayout产生问题。

**IE6/7下ul/ol标记消失bug：**  
发生场合：当ul/ol触发了haslayout并且是在ul/ol上写margin-left，前面默认的ul/ol标记会消失。  
解决方法：给li设置margin-left，而不是给ul/ol设置margin-left。  
原理分析：IE6/7浏览器Bug

**IE6/7下margin与absolute元素重叠bug：**  
发生场合：双栏自适应布局中，左侧元素absolute绝对定位，右侧的margin撑开距离定位。在IE6/7下左侧应用了absolute属性的块级元素与右边的自适应的文字内容重叠。  
解决方法：把左侧块级元素更改为内联元素，比如把div更换为span。  
原理分析：这是由于IE6/IE7浏览器将inline水平标签元素和block水平的标签元素没有加以区分一视同仁渲染了。属于IE6/7浏览器渲染Bug。

**IE6/7/8下auto margin居中bug：**  
发生场合：给block元素设置margin auto无法居中  
解决方法：出现这种bug的原因通常是没有Doctype，然后触发了ie的quirks mode，加上Doctype声明就可以了。在《打败IE的葵花宝典》里给出的方法是给block元素添加一个width能够解决，但根据本人亲测，加with此种方法是无效的，如果没有Doctype即使给元素添加width也无法让block元素居中。  
原理分析：缺少Doctype声明。

**IE8下input[button | submit] 设置margin:auto无法居中**  
发生场合：ie8下，如果给像button这样的标签（如button input[type="button"] input[type="submit"]）设置{ display: block; margin:0 auto; }如果不设置宽度的话无法居中。  
解决方法：可以给为input加上宽度  
原理分析：IE8浏览器Bug。

**IE8百分比padding垂直margin bug：**  
发生场合：当父元素设置了百分比的padding，子元素有垂直的margin的时候，就好像父元素被设置了margin一样。  
解决方法：给父元素加一个overflow:hidden/auto。  
原理分析：IE8浏览器Bug。