揭开外边距折叠Collapsing margins的面纱

外边距折叠，是属于盒模型部分的一个重要概念。在刚刚开始接触CSS的时候，对外边距折叠的认识，只有一句话：垂直方向上相邻的两个外边距(margin)会合并成一个，最终的大小，是两个中较大的。

51CTO推荐专题：[深入浏览器核心 高效CSS的秘诀](http://developer.51cto.com/art/201006/207317.htm" \t "_blank)

什么是外边距折叠，以及还有其他什么情况下会发生，都不清楚。后来，在实际应用中发现，外边距折叠不是这么简单的，其中包含了很多其他情况下的折叠规则。  
  
**一、概念**

Collapsing margin，指的是毗邻的两个或多个外边距 (margin) 会合并成一个外边距，即外边距折叠。其中所说的 margin 毗邻，可以归结为以下两点：

1. 这两个或多个外边距没有被非空内容、padding、border或clear分隔开。

2. 这些margin都处于普通流(in-flow，非浮动元素，非定位元素)中。

**二、margin折叠的规则**

**1. 两个或多个毗邻的普通流中的块元素垂直方向上的margin会折叠**

**A. 两个或多个**

说明其数量必须是大于一个，又说明，折叠是元素与元素间相互的行为，不存在A和B折叠，B没有和A折叠的现象了。

**B. 毗邻**

是指没有被非空内容、padding、border或clear分隔开，说明其位置关系。注意一点，在没有被分隔开的情况下，一个元素margin-top会和它普通流中的第一个子元素(非浮动元素等)的margin-top相邻；只有在一个元素的height是”auto”的情况下，它的margin-bottom才会和它普通流中的最后一个子元素(非浮动元素等)的margin-bottom相邻。文章不够，例子来凑：

1. HTML code**<div** style="border:1px solid red; width:100px;"**>**
2. **<div** style="margin:50px 0; background-color:green; height:50px; width:50px;"**>**
3. **<div** style="margin:20px 0;"**>**
4. **<div** style="margin:100px 0;"**>**B**</div>**
5. **</div>**
6. **</div>**
7. **</div>**

截图：

[](http://images.51cto.com/files/uploadimg/20100816/1341140.jpg)

以上测试用例中，margin会把B的包含块撑开。

如果一个元素的height特性的值不是 auto，那么它的margin-bottom和它的子元素的margin-bottom不算是相邻，因此，不会发生折叠。margin-top 没有此限制，所以是 100px，margin-bottom 没有折叠，所以只有 50px。

**C. 垂直方向**

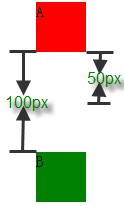
是指具体的方位，只有垂直方向的margin才会折叠，也就是说，水平方向的margin不会发生折叠的现象。

**D. 折叠后margin的计算**

参与折叠的margin都是正值。例：

1. HTML code**<div** style="height:50px; margin-bottom:50px; width:50px; background-color: red;"**>**A**</div>**
2. **<div** style="height:50px; margin-top:100px; width:50px; background-color: green;"**>**B**</div>**

截图：

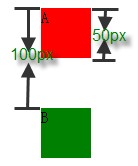
[](http://images.51cto.com/files/uploadimg/20100816/1341141.jpg)

可见，在margin都是正数的情况下，会取其中margin较大的值为最终 margin值。

参与折叠的margin中有正值，有负值。如果，毗邻的margin中有正值，同时存在负值会怎样呢？有正有负，先取出负margin中绝对值最大的，然后，和正marin值最大的magin，相加。请看下面的例子：

1. HTML code**<div** style="height:50px; margin-bottom:-50px; width:50px; background-color: red;"**>**A**</div>**
2. **<div** style="height:50px; margin-top:100px; width:50px; background-color: green;"**>**B**</div>**

截图：

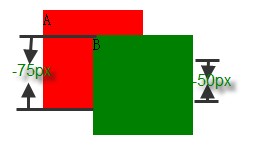
[](http://images.51cto.com/files/uploadimg/20100816/1341142.jpg)

上面的例子最终的margin是 100+(-50)=50px。

参与折叠的margin都是负值。最后，如果毗邻的两个margin都是负值又会怎样呢？当margin都是负值的时候，取的是其中绝对值较大的，然后，从0位置，负向位移。看例子：

1. HTML code**<div** style="height:100px; margin-bottom:-75px; width:100px; background-color: red;"**>**A**</div>**
2. **<div** style="height:100px; margin-left:50px; margin-top:-50px; width:100px; background-color: green;"**>**B**</div>**

截图：

[](http://images.51cto.com/files/uploadimg/20100816/1341143.jpg)

相邻的margin要一起参与计算，不得分步计算。以上举的例子都是兄弟节点在垂直方向的相邻，要注意，相邻的元素不一定非要是兄弟节点，父子节点也可以，即使不是兄弟父子节点也可以相邻。举一个复杂点儿的例子，我们要计算的是A和B之间的margin最终应该是多少：

1. HTML code**<div** style="margin:50px 0; background-color:green; width:50px;"**>**
2. **<div** style="margin:-60px 0;"**>**
3. **<div** style="margin:150px 0;"**>**A**</div>**
4. **</div>**
5. **</div>**
6. **<div** style="margin:-100px 0; background-color:green; width:50px;"**>**
7. **<div** style="margin:-120px 0;"**>**
8. **<div** style="margin:200px 0;"**>**B**</div>**
9. **</div>**
10. **</div>**

有人可能说了，算A和B之间的margin，分别算A和其父元素的折叠，然后与其父元素的父元素的折叠，这个值算出来之后，应该是90px。依此法算出B的为80px；然后，A和B折叠，margin 为90px。

对吗？错了，大错特错。错在哪里了呢？请注意，多个margin相邻折叠成一个margin，所以计算的时候，应该取所有相关的值，而不能分开，分步来算。以上例子中，A和B之间的margin折叠产生的margin，是6个相邻margin折叠的结果。分别是：

◆正值：50px，150px，200px

◆负值：-60px，-100px，-120px

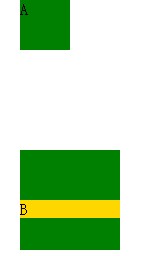
根据有正有负时的计算规则，正值的最大值为 200px，负值中绝对值最大的是-120px，所以，最终折叠后的margin应该是 200 + (-120) = 80px。  
可以打开浏览器来验证自己的猜想。

**2. 浮动元素/inline-block的元素/绝对定位元素的margin不会和垂直方向上的其他元素的margin折叠**

浮动元素的margin在垂直方向上也不会发生 margin折叠，即使和它相邻的子元素，也不会。看例子：

1. HTML code**<div** style="margin-bottom:50px; width:50px; height:50px; background-color:green;"**>**A**</div>**
2. **<div** style="margin-top:50px; width:100px; height:100px; background-color:green; float:left;"**>**
3. **<div** style="margin-top:50px; background-color:gold;"**>**B**</div>**
4. **</div>**

截图：

[](http://images.51cto.com/files/uploadimg/20100816/1341144.jpg)

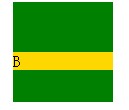
可见，两个绿色的块儿之间，相距100px，而若B和它的浮动包含块发生margin折叠的话，金色的条应该位于绿色块的最上方，显然，没有发生折叠。  
将float:leftt分别换成 “display:inline-block”和”position:absolute”测试后两种情况，截图与上同。

**3. 创建了块级格式化内容的元素，不和它的子元素发生margin折叠**

关于块级格式化内容，请参照：http://www.w3.org/TR/CSS2/visuren.html#block-formatting，其中浮动元素，绝对定位元素，以及 overflow值不是’visible’的元素，都符合。以 “overflow:hidden” 的元素为例：

1. HTML code**<div** style="margin-top:50px; width:100px; height:100px; background-color:green; overflow:hidden;"**>**
2. **<div** style="margin-top:50px; background-color:gold;"**>**B**</div>**
3. **</div>**

截图：

[](http://images.51cto.com/files/uploadimg/20100816/1341145.jpg)

若B和它的 “overflow:hidden” 包含块发生margin折叠的话，金色的条应该位于绿色块的最上方，显然，没有发生折叠。

**4. 元素自身的margin-bottom和margin-top相邻时也会折叠**

自身margin-bottom和margin-top相邻，只能是自身内容为空，垂直方向上border、padding为0。例：

1. HTML code**<div** style="border:1px solid red; width:100px;"**>**
2. **<div** style="margin-top:100px;margin-bottom:50px;"**></div>**
3. **</div>**

截图：

[](http://images.51cto.com/files/uploadimg/20100816/1341146.jpg)

以上代码运行后，我们得到的是一个正方形，说明，最终的margin是100px，而不是150px，也就是说发生了margin折叠。