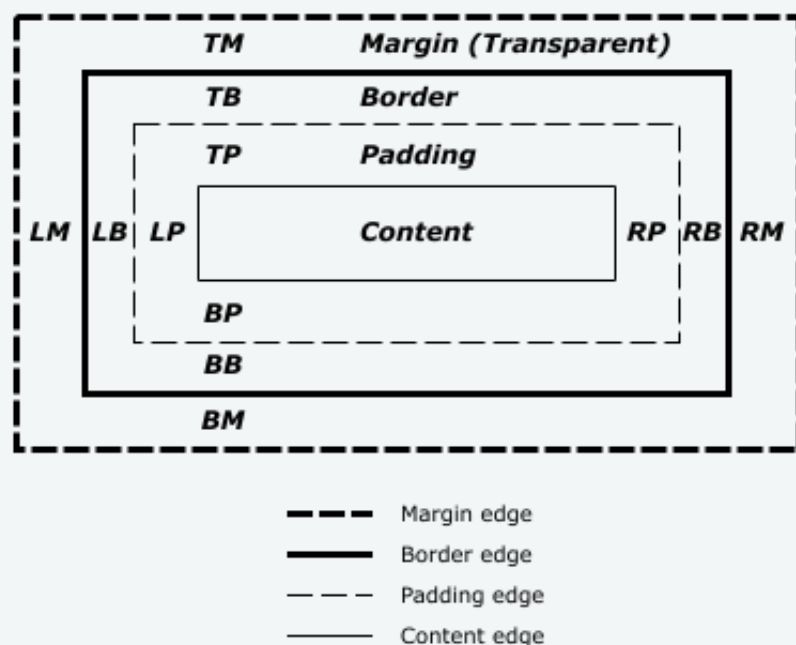


本文着重描述关于 margin，我们日常不太容易发现的“坑”。

盒模型

接触过 CSS 的人应该都知道 CSS 的盒模型：



由内容边缘（Content edge）包围形成的是内容盒（Content Box），类推还有内边距盒（Padding Box）、边框盒（Border Box）、外边距盒（Margin Box）。

其中内容盒、内边距盒、边框盒的背景由 `background` 属性决定，而外边距盒的背景是透明的。

CSS margin 属性

关于 margin 属性，有几点可能跟我们的直觉不相符：

- 如果 margin 的值是百分比，则是相对于父元素的**内容盒宽度**来计算的，即使 margin-top 和 margin-bottom 也是如此。因此即使父元素的高宽不相等，子元素的 margin 元素指定了相同的百分比值，则子元素各个方向的 margin 计算值都是相等的。
- margin-top 和 margin-bottom 值对**行内非替换元素（non-replaced inline element）**是无效的。因此我们可以指定 img 元素的 margin-top 和 margin-bottom，而非替换行内元素（如 i，span 等）设置 margin-top 和 margin-bottom 却不会产生效果。

相邻的 margin (Adjoining margin)

如果两个垂直方向上的 margin，它们中间没有其他垂直 margin，但它们之间不一定相接触，我们就说这两个 margin 是**垂直毗连** (vertical-adjacent) 的，包括以下四种情况，满足其中之一即可：

- 父元素的 top margin 和第一个子元素的 top margin
- 父元素的 bottom margin 和最后一个子元素的 bottom margin
- 元素的 bottom margin 和与这个元素相邻的兄弟元素的 top margin
- 如果一个元素，它没有生成 BFC、没有包含正常流的子元素、`min-height` 是0、`height` 是0 或者 auto，则它的 top margin 和 bottom margin 也是垂直毗连的

如果两个 margin 满足以下三个条件，我们就说这两个 margin 是**相邻** (adjoining) 的：

1. 这两个 margin 是**垂直毗连**的，即满足上面四种情况之一
2. margin 的两个元素都是正常流的块级元素，并且在同一个 BFC 中
3. 两个 margin 之间没有行盒 (line box)、清除浮动后的空隙 (**clearance**)、padding 和 border

margin 折叠 (Collapsing margins)

margin 折叠，即**相邻**的 margin 有可能会被折叠成一个。

比如元素 #a 指定了 margin-bottom 为 10px，而它下方的元素 #b 指定了 margin-top 为 20px，如**这样**：

HTML ▾

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-
width">
  <title>JS Bin</title>
</head>
<body>

  <div id="a"></div>
  <div id="b"></div>

</body>
</html>
```

CSS ▾

```
#a {
  height: 10px;
  background: blue;
  margin-bottom: 10px;
}

#b {
  height: 10px;
  background: red;
  margin-top: 20px;
}
```

Output

Run with JS Auto-run JS ☐



元素 #a 的 margin-bottom 和元素 #b 的 margin-top 在位置上重叠了，它们之间的距离是 20px，即元素 #b 的 bottom margin 长度，这就是 margin 折叠现象。关于这个现象，可以这么理解：

margin 定义的是它与其他盒子之间的最小间距。其中元素 #a 指定了 margin-bottom 为 10px，就表明它下方的元素 #b 与它至少要有 10px 的距离，它指定的是一个最小值，因此实际的距离可以比这个大。

元素 #a 下方的元素 #b 也设置了 margin-top 为 20px，如果不折叠，则他们之间就有 30px 的距离。如果折叠成了一个 20px 的距离，则对元素 #a 来说，它的 margin-bottom 要求至少要有 10px 的距离，是满足的，而对于元素 #b 来说，它的 margin-top 要求至少要有 20px 的距离，也是满足的。

而 margin 折叠的存在，其实是为了可以在视觉上显得更美观，也更贴近设计师的预期。

margin 折叠规则

并不是所有的 margin 都可以折叠，需要满足以下条件：

- 垂直**相邻**的 margin 才有可能折叠，水平 margin 永远不折叠
- 根元素（即 html 元素）的 margin 永远不折叠
- 如果一个元素，它的 top margin 和 bottom margin 是**相邻**的，并且有清除浮动后的空隙（[clearance](#)），这个元素的 margin 可以跟兄弟元素的 margin 折叠，但是折叠后的 margin 不能跟父元素的 bottom margin 折叠。

需要注意的是，margin 并不是只能折叠一次，多个满足要求的 margin 都可以进行折叠形成一个折叠后的 margin（collapsed margin）。

并且假如这个折叠后的 margin 是由 margin A 等折叠而来的，如果有 margin X 跟 margin A 是**相邻**的，则我们也认为 margin X 跟这个折叠后的 margin **相邻**。

折叠后的 margin 大小

当两个或者两个以上的 margin 折叠后，margin 的值计算如下：

- 如果 margin 都是正数，则取他们当中的最大值
- 如果 margin 中有正有负，则取最大的正数加上最小的负数（如最大的 margin 是 20px，最小的 margin 是 -20px，则他们计算后的值是 0）
- 如果 margin 中都是负数，则取他们当中的最小值

几道思考题

浮动、定位元素的 margin 不会和其他任何元素的 margin 发生重叠，包括它的子元素。

这是因为浮动元素脱离了正常流，所以它和其他相邻元素就不处在同一个流中，自然不**相邻**；又因为浮动元素的内容盒会形成一个新的 BFC，所以浮动元素跟子元素不处在同一个 BFC 中，因此它们的 margin 也不能折叠。定位元素同理可得。

inline-block 的元素不会和其他元素的 margin 发生折叠，包括它的子元素。

因为 margin 折叠只会发生在块级元素上，因此 inline-block 元素的 margin 不会和兄弟元素折叠，又因为 inline-block 的内容盒会形成一个新的 BFC，所以 inline-block 元素本身也不会和子元素的 margin 发生折叠

margin 折叠的几个栗子

栗子1

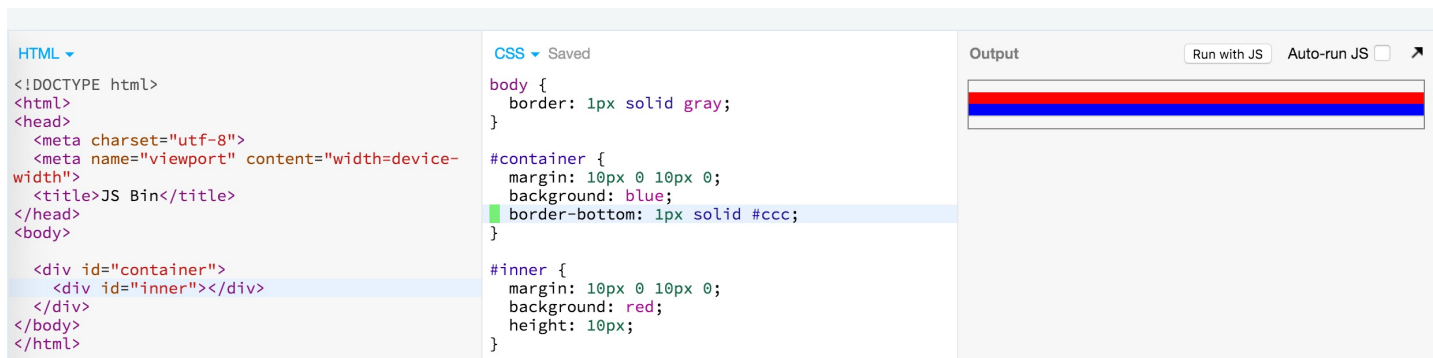
如果两个 margin 满足以下三个条件，我们就说这两个 margin 是**相邻** (adjoining) 的：

- 两个 margin 之间没有行盒 (line box)、清除浮动后的空隙 (clearance)、padding 和边框

针对这个条件，我们通过增加 padding 的方式来阻止 margin 的折叠：

| | | |
|--|--|---------------|
| HTML | CSS | Output |
| <pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <meta name="viewport" content="width=device-width"> <title>JS Bin</title> </head> <body> <div id="container"> <div id="inner"></div> </div> </body> </html></pre> | <pre>body { border: 1px solid gray; } #container { margin: 10px 0 10px 0; background: blue; } #inner { margin: 10px 0 10px 0; background: red; height: 10px; }</pre> | |

如果 #container 没有下边框，则 #container 的 bottom margin 和 #inner 的 bottom margin 是**相邻**的，因此它们折叠了，并且 #inner 撑开了 #container 元素，所以可以看到 #container 元素的高度变成了 10px，且显示的是 #inner 的红色背景



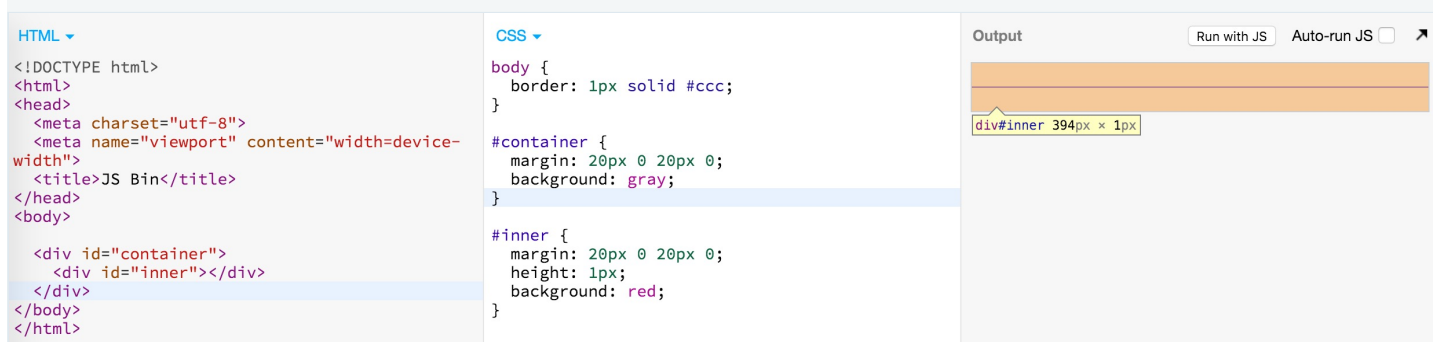
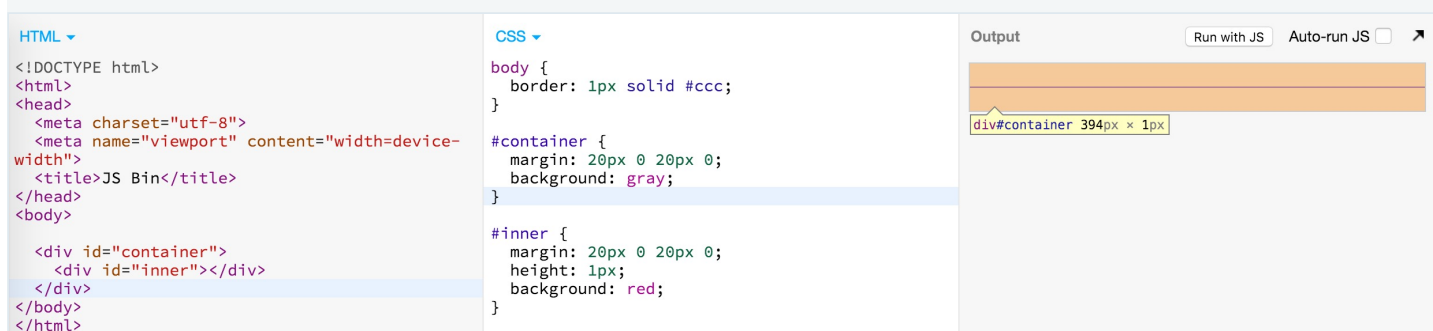
当给 #container 添加一个下边框，两个 margin 之间就边框的阻隔，他们就不**相邻**了，因此不能折叠。所以可以看到 #container 被撑开成了 20px，其中 10px 是 #inner 的高度，还有 10px 是 #inner 的 bottom margin，并且由于 margin 是透明的，因此 #container 露出了部分蓝色的背景。

栗子2：

如果两个 margin 满足以下三个条件，我们就说这两个 margin 是**相邻** (adjoining) 的：

- margin 的两个元素都是正常流的块级元素，并且在同一个 BFC 中

我们通过创建新的 BFC来阻止 margin 的折叠：



如上图 #container 元素和 #inner 元素同属于一个 BFC 中，#container 的 top margin 和 #inner 的 top margin 折叠，bottom margin 同理。

但如果让 #container 跟 #inner 处在不同的 BFC 中，则 top margin 和 bottom margin 都不会折叠，如：

| HTML | CSS | Output |
|--|---|--------|
| <pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <meta name="viewport" content="width=device-width"> <title>JS Bin</title> </head> <body> <div id="container"> <div id="inner"></div> </div> </body> </html></pre> | <pre>body { border: 1px solid #ccc; } #container { margin: 20px 0 20px 0; background: gray; overflow: hidden; } #inner { margin: 20px 0 20px 0; height: 1px; background: red; }</pre> | |

给 #container 元素增加一个 `overflow: hidden` 属性，让它的内容盒生成一个独立的 BFC，而 #inner 处于这个独立的 BFC 中，因此 #container 和 #inner 就处于两个不同的 BFC 中了，所以他们的 margin 不能折叠。

栗子3：

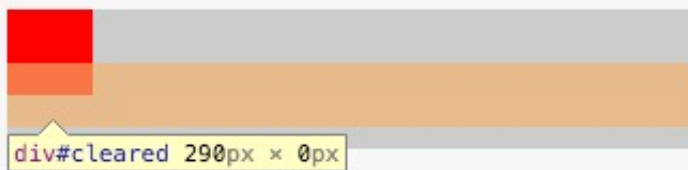
- 如果一个元素，它本身的 top margin 和 bottom margin 是**相邻的**，并且有清除浮动后的空隙（**clearance**），这个元素的 margin 可以跟兄弟元素的 margin 折叠，但是折叠后的 margin 不能跟父元素的 bottom margin 折叠。

| HTML | CSS | Output |
|---|---|--------|
| <pre><!DOCTYPE html> <html> <head> <meta charset="utf-8"> <meta name="viewport" content="width=device-width"> <title>JS Bin</title> </head> <body> <div id="container"> <div id="floated"></div> <div id="cleared"></div> <div id="sibling"></div> </div> </body> </html></pre> | <pre>body { border: 1px solid #ccc; } #container { margin: 50px; background: #ccc; } #floated { height: 40px; width: 40px; float: left; background: red; } #cleared { margin: 15px; clear: left; } #sibling { margin: 40px; margin-bottom: 0; }</pre> | |

给父元素 #container 设置了一个灰色背景，并且没有设置高度，因此高度会随着内容而扩展，margin 设置为 50px。

其中有一个红色的浮动元素 #floated，高宽都设置为 40px。

给 #cleared 设置了 15px 的 margin，并且元素的高度、padding、margin 都为 0，因此 #cleared 元素的 top margin 和 bottom margin 是**相邻的**。这个元素的位置如下图所示：



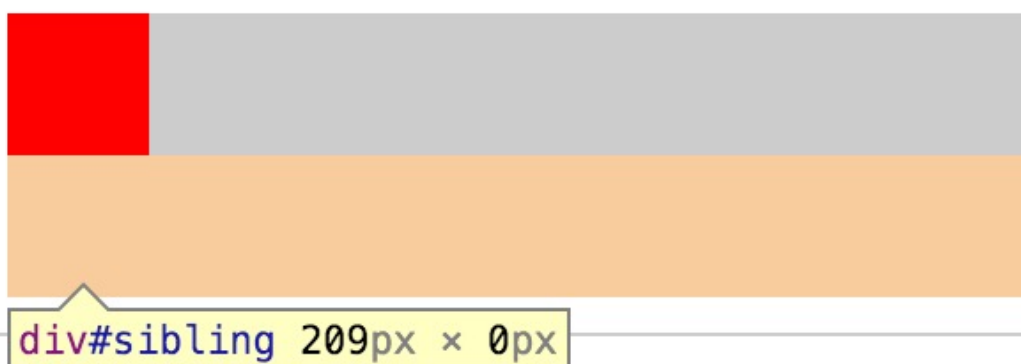
因为 #cleared 元素清除了左浮动，所以 #cleared 元素下移。

而 #cleared 元素和 #sibling 元素的 margin 折叠了，因此可以看到他们的位置是重叠的。



由于这条规则的存在，导致他们折叠后的 margin 不能跟 #container 的 bottom margin 进行折叠，因此 #container 的高度被撑开。

如果没有这条规则，他们还应该跟 #container 的 bottom margin 进行折叠，如：



以上这张图，在去掉了 #cleared 元素的 clear 属性之后，就不满足这条规则了，所以可以看到

#container 的高度就只有 40px，即红色的浮动元素的高度，而 #cleared 元素、#sibling 元素、#container 元素的 margin 都折叠成了一个。

结语

这篇文章的绝大多数内容都是从[官方规范](#)翻译而来，同时也参考也网上一些写的比较好的文章而写的一个介绍性文章，其中有部分内容并没有展开，如 BFC、clearance 等，因为这部分内容不是三言两语就可以解释清楚，我本人也需要进行更深入的学习理解，所以请读者自行查阅相关文章

参考文献

<https://www.w3.org/TR/CSS2/box.html>

<https://www.w3.org/TR/CSS2/visuren.html>

<http://www.w3cplus.com/css/understanding-bfc-and-margin-collapse.html>

<https://segmentfault.com/a/1190000003099116>

<https://segmentfault.com/a/1190000003096320>

<http://melonh.com/css/2015/04/28/understand-margin-collapse.html>