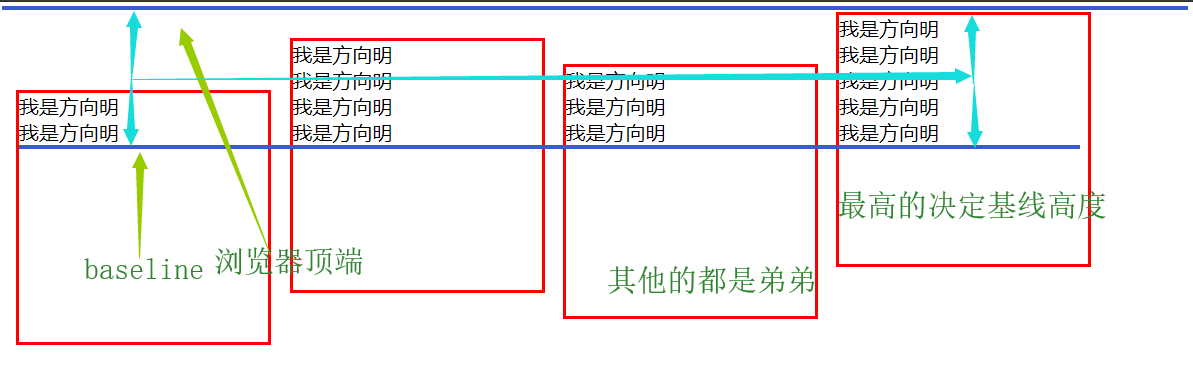
### 关于vertical-align踩的坑

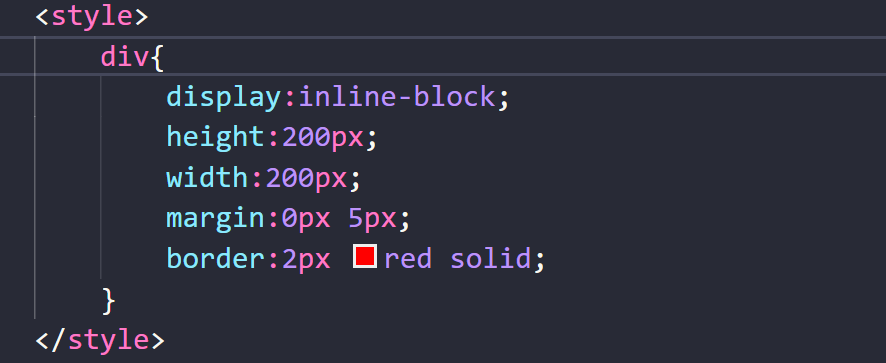
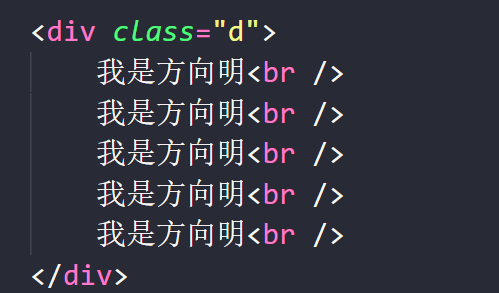
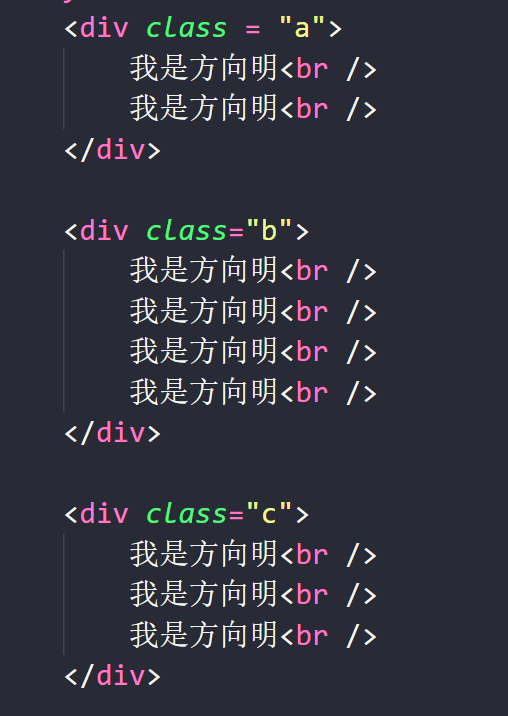
这个问题遇到很多次了，今天又遇到了，虽然有了之前的经验，这个问题一看就明白了，但是今天也做个笔记吧。首先需要声明的是，这个属性是应用与某个子元素的，而非父元素。比如说这样一个场景：

有许多行内元素的div，这些div都是position:relative；当这些div中没有其他行内元素或行内块状元素的时候，你把他们都相对移动到了一个想要的完美的地方。这时候你打算想这些div中加入一些行内元素或者行内块状元素，比如说不同长度文本，然而加入文本之后你会发现之前完美的布局崩了，而且高度越大的文本就越崩的离谱。

这一切的罪魁祸首就是浏览器的默认文本对齐方式——基线对齐。在不设置行高时，长得高的文本块必定会决定整个行的基线高度是什么。其他长得矮小的就会顺应这个高的，最后呈现的效果是，所有同一行且在不同div的文本都会在基线处保持平齐，而其他的地方就无法得到对齐了。包含文本矮的div则为了保持文本的基线对齐，就不得不变换位置以达到让文本对齐的效果，最后呈现的样子就是div的参差不齐。

就像这样：





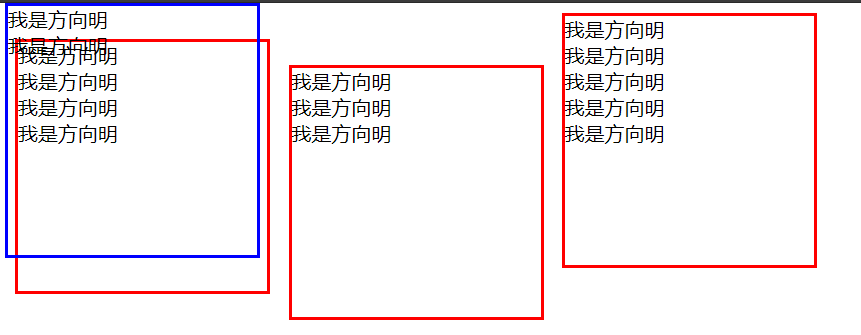
注意一定是行内元素或者行内块级元素才能出现这个问题。不是行内元素的就会自成一端，虽然基线还是存在，但是本身决定基线，也就谈不上一行的基线。

想要解决这个办法很简单，我总结的方式常用的有三种：

1. 将所有div元素设置绝对定位，定位到相应位置。
2. 将所有div设置为相对定位，之后调节每个元素的位置。强行达到对齐的效果。
3. 将每个div都设置一个合适vertical-align属性。（不是在div的父元素上！）

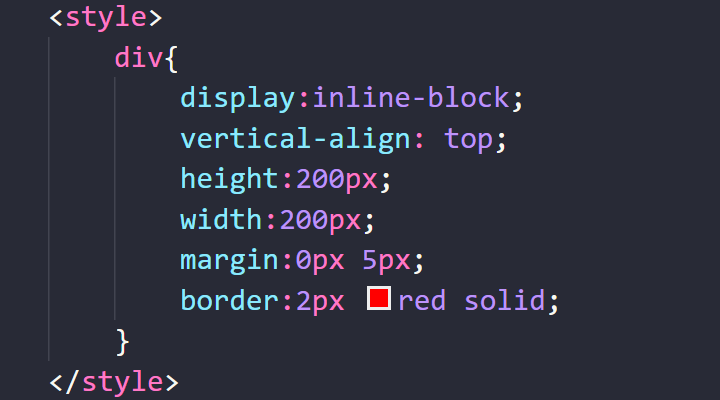
第一种方式显然是可以的，不用Sass和Less的话，除了麻烦点要设置每个div的合理定位，还要解决脱离文档流的问题，其他没什么不好的，但绝对不是最好的。让他去哪他就不许去那里，基线和边距什么的都吃屎去。（理所当然display:inline-block就无效了，没有必要的时候我也不会选择这样的方式，会让代码质量下滑，显现的死板不宜维护，而且难以和浏览器默认样式达到统一，就像蓝色框其实比最后的红色框更贴近浏览器顶部，而红色框根本就没有设置margin-top。 ）。

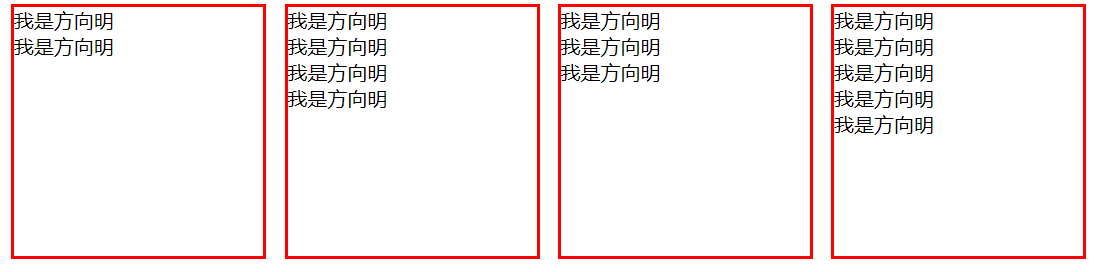




第二方法，设置relative.肯定行，能解决问题，但是谁会选择这种非常笨拙而且不计后续设计和维护后果的方法呢？

第三种方法，是我看中的最好方法，是最正确的方式。设置vertical-align:top || bottom;调换基线的位置，既然下面无法对齐，那用上面去对其很显然不会受到部分内容过高带来的影响。用决定基线的性质去决定基线显然非常合理。



（设置为bottom也能达到同样的样式效果）

以下复习一下关于vertical-align的相关知识。





Vertical-align:baseline || sub || super || top || text-top || middle || bottom || length ||

%



更多可以参考博客（快看）

http://www.cnblogs.com/starof/p/4512284.html?utm\_source=tuicool&utm\_medium=referral