前言：为什么Vertical-align垂直居中这么难用！！！

多么优美的方法，但使过程中，99%的人使终会犯错，到底Vertical-align有哪些要求呢？我们先从理论上来分析一下。

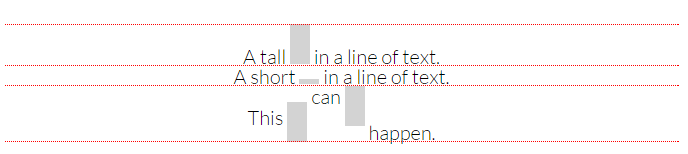
开始

在遇到将元素在垂直方向上对齐的需求时，CSS提供了多种方法，有时候我用float解决，有时候我使用position:absolute解决，还有时候采用的方法是手动调整margin和padding。  
其实我并不喜欢这些解决方法，float会让元素的顶端对齐而且需要手动清楚浮动。绝对定位让元素脱离了正常流，这样这些元素就不会再影响到周围的元素。在元素的padding和margin固定的情况下，一点小的改变也很有可能对布局造成影响。  
接下来登场的就是本文的主角了：Vertical-align。通常来说使用这个属性进行布局是一种hack行为，因为它本来并不是被用于这个目的。它用在文本和与文本相邻元素的垂直方向上的对齐问题。然而，你也可以使用Vertical-align在不同的上下文中对元素进行灵活的，细粒度的排布。元素的尺寸无需知晓，元素任然处于正常流当中，因此元素的变化会影响到周围的元素，这使得Vertical-align是非常有价值的排布方案的选择。

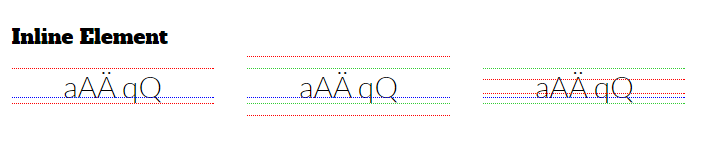
Vertical-align的奇特之处

Vertical-align有时候会变的面目狰狞。它似乎有一些谜一样的规则。举例来说，有时候你改变了Vertical-align的值，但是该元素垂直方向上没有发生改变反而别的元素改变了位置。  
一些资料在谈到Vertical-align这个问题的时候深入程度不够，特别是当人们想使用这个属性布局的时候。这些资料更多的将精力集中在了一个误区上，就是尝试将元素内的一切都垂直对齐。他们给出了一下在简单情况下的例子来解释Vertical-align这个属性而忽略了一些复杂奇怪的方面。

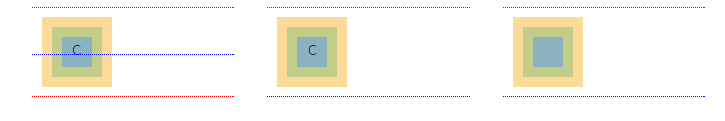
使用Vertical-align的要求

Vertical-align被用于垂直对齐inline元素，也就是display值为inline和inline-block的元素。inline-table的元素不在本文的讨论范围内。  
inline元素基本上值得就是文本  
inline-block元素就像它的名字一样，同时具备inline元素和block元素的特点，这样的元素有padding,margin,border,width,height。其中高度有可能是由元素的内容决定的。  
inline元素一个挨着一个的摆放在行内，当行内元素太多的情况下，一个新行会被创建出来，这些行也叫做line--box。它将行内的所有内容都包裹了起来。根据行内内容的不同，line-box的尺寸也会不同，在接下来的图里面，红线代表了line-box的上下边界  
在这些line-box内，Vertical-align属性负责对齐一些独立的元素。那么，这些元素是要和谁对齐呢？

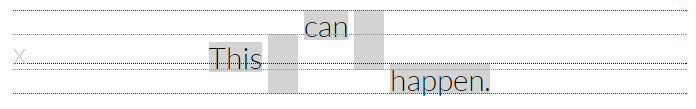
关于边界和baseline

对于垂直对齐这个知识点来说最重要的就是涉及元素的baseline。有时候元素的盒模型的上下边界也会变的很重要。下面上图  
  
如图所示有三行文字。行高的上下边界是红线。文字的上下边界是绿色的线，蓝色的线就是baseline了，左边文字的高度与行高是一致的，因此绿线和红线重合了，中间的行高是文字大小的两倍，而在右边，行高是文字大小的二分之一。  
行内元素的外边缘在行高的上边缘和下边缘这个范围内对齐，如果行高小于文字的高度也无所谓。关于baseline的定义还是直接给出标准的链接

inline-block元素

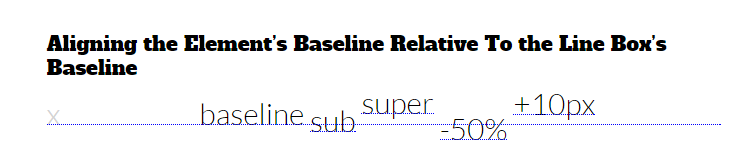
  
从左到右的三幅图片都是inline-block元素，不同的是，左面包含着没有脱离正常流的内容c，中间的除了没有脱离正常流的内容以外还加了overflow:hidden,右面的没有内容但是内容区还有高度。红线代表了margin-box的边界。黄色代表的是border，绿色的是padding，蓝色的是content，蓝色的线代表的还是baseline。  
inline-block元素的外边缘就是margin-box的边缘。  
inline-block元素的baseline的位置要看该元素有没有处于正常流之内的内容。  
(1)在有处于正常流内容的情况下，inline-block元素的baseline就是最后一个作为内容存在的元素的baseline，这个元素的baseline的确定就要根据他自身来确定了。  
(2)在overflow属性不为visible的情况下，baseline就是margin-box的下边界了。  
(3)第三种情况下baseline还是margin-box的下边界。

line-box

  
关于line-box的图上面已经给过了，这次我将文字部分高亮显示。line-box的上边界与最高元素的上边界对齐，下边界与最低元素的下边界对齐。  
W3C标准中并没有定义line-box的baseline的位置。这一点很让人困惑，baseline的位置需要满足vertical-align属性的值以及让line-box的高度最小等条件，是一个很灵活的参数。  
line-box的baseline是不可见的，但是可以很轻松的将它可视化出来，在行的开头添加一个字母，比如'x'，这个字母的下边界默认就是baseline的位置。围绕着baseline在line-box中形成了文字盒。文字盒可以被认为是没有和任何元素对齐的line-box中的inline元素，它的告诉与它的父元素的font-size的值相同。因此，文字盒仅仅包含非格式化的line-box的文本，文字盒的边界由绿线来表示。因为文字盒是紧挨着baseline的，所以baseline的位置发生变化的话，文字盒的位置也会跟着改变(这里所说的文字盒在标准中被叫做strut)。  
总结起来的话有以下两点：

* 有一个区域叫做line-box，垂直方向上的对齐都是发生在这个区域里面，它有baseline，有文字盒，有上下边界。
* inline元素也有baseline和上下边界，inline元素是需要对齐的对象。

Vertical-align属性的值

  
有关Vertical-align各个取值的说明读者还是上MDN看吧。也可以看看天镶大神的博客上的文章

Vertical这个属性到底是怎么起作用的呢？

对齐一个Icon

我想将一个icon与文字对齐仅仅使icon的vertical-align属性的值设置为middle似乎不能产生令人满意的结果，例子如下图  


<!-- left mark-up --><spanclass="icon middle"></span>Centered?<!-- right mark-up --><spanclass="icon middle"></span><spanclass="middle">Centered!</span><styletype="text/css">.icon{ display: inline-block;            /\* size, color, etc. \*/}.middle{ vertical-align: middle; }</style>

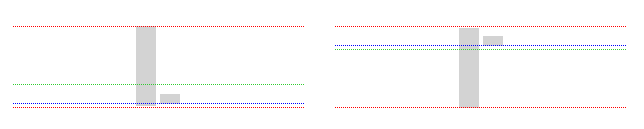
下面给出上图的辅助线版本  
  
原因就是左面的文字根本就没有发生对齐行为，它还是对齐于line-box的baseline。而vertical-align对齐的点是baseline加上半个x的距离(half of the x-height)。因此文字的最高点超过了icon的高度。  
而右面的例子，文字与icon都对齐于一个中点，文字的baseline稍微下移，位于line-box的baseline的下方。结果是很好的达到了icon与文字对齐的效果。

line-box的baseline的移动问题

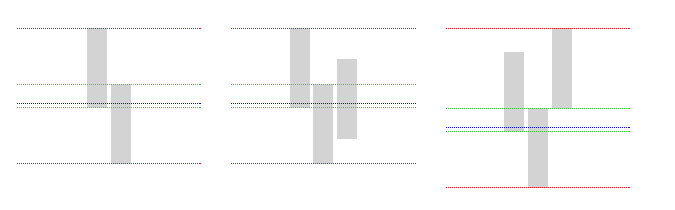
这是一个Vertical-align的坑，line-box中的所有元素都会影响到baseline的位置。假设，一个元素按某种方式垂直对齐了，但是这种对齐方式会引起baseline的移动，又因为大部分的垂直对齐方式(除了top和bottom)和baseline有关，因此这个元素的垂直方向对齐的行为会引起该line-box内其他元素位置的调整。  
下面还是一些例子

* 一个很高的元素，其高度占满了整个line-box，那么vertical-align对其实没有影响的，在它的top和bottom之外没有空间让其移动。但是为了满足它的vertical-align的值，line-box的baseline会发生移动，左面的高元素的取值为text-bottom，矮元素的取值为baseline。右面的高元素的取值为text-top，你会看到baseline跳上去了  
  

<!-- left mark-up --><spanclass="tall-box text-bottom"></span><spanclass="short-box"></span><!-- right mark-up --><spanclass="tall-box text-top"></span><spanclass="short-box"></span><styletype="text/css">.tall-box,  .short-box{ display: inline-block;                /\* size, color, etc. \*/}.text-bottom{ vertical-align: text-bottom; }.text-top{ vertical-align: text-top; }</style>

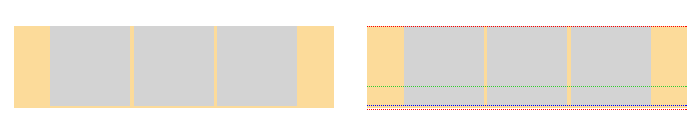
如果把高元素的vertical-align设置为其他值，也能看到类似的行为  
甚至将vertical-align设置为bottom或者是top也会让baseline发生移动。这很奇怪，因为这时候应该就没baseline什么事儿了。  


<!-- left mark-up --><spanclass="tall-box bottom"></span><spanclass="short-box"></span><!-- right mark-up --><spanclass="tall-box top"></span><spanclass="short-box"></span><styletype="text/css">.tall-box,  .short-box{ display: inline-block;               /\* size, color, etc. \*/}.bottom{ vertical-align: bottom; }.top{ vertical-align: top; }</style>

将两个更大的元素放在一个line里面，并且设置vertical-align的值让line-box的baseline移动。在满足vertical-align数值对齐的条件下，line-box的高度会自我调整，如左面的图。再增加第三个元素，第三个元素如果因为其vertical-align的设置不会超过line-box的边缘的话，它是不会影响到line-box的高度和baseline的位置的，如果它会超过line-box的边缘，那么line-box的高度和baseline的位置也会进行调整。在第二种情况下，另外两个元素的位置发生了下移。  


<!-- left mark-up --><spanclass="tall-box text-bottom"></span><spanclass="tall-box text-top"></span><!-- mark-up in the middle --><spanclass="tall-box text-bottom"></span><spanclass="tall-box text-top"></span><spanclass="tall-box middle"></span><!-- right mark-up --><spanclass="tall-box text-bottom"></span><spanclass="tall-box text-top"></span><spanclass="tall-box text-100up"></span><styletype="text/css">.tall-box{ display: inline-block;                 /\* size, color, etc. \*/}.middle{ vertical-align: middle; }.text-top{ vertical-align: text-top; }.text-bottom{ vertical-align: text-bottom; }.text-100up{ vertical-align:100%; }</style>

inline元素下方可能会有一点空隙

下面给出一个例子，如果尝试将li元素在垂直方向上进行对齐的话在，这个现象非常常见。  


<ul><liclass="box"></li><liclass="box"></li><liclass="box"></li></ul><styletype="text/css">.box{ display: inline-block;         /\* size, color, etc. \*/}</style>

正如你所见，li元素是对齐baseline的，baseline的下方会给字母的一部分留出空间，因此会产生一个空隙。解决方案就是改变line-box的baseline的位置，比如将这些li这是为vertical-align:middle

<ul><liclass="box middle"></li><liclass="box middle"></li><liclass="box middle"></li></ul><styletype="text/css">.box{ display: inline-block;            /\* size, color, etc. \*/}.middle{ vertical-align: middle; }</style>

inline元素之间的空隙造成布局效果与理想状态发生偏差  
这主要是因为inline元素本身的原因，但是因为如果要让vertical-align的值产生作用的话，inline元素是必备要求，因此了解一点也不错嘛  
这个空隙来源于inline元素之间的空格，所有的inline元素之间的空白都会变成一个空格。如果你想让inline元素在水平上紧挨着，设置它们的宽度是不行的。因为之间存在空隙，所以行的宽度不够放下两个inline元素。一行会被破坏为两行。解决方案如下图的右侧  
