|  |  |
| --- | --- |
| **值** | **描述** |
| 长度 | 通过距离升高（正值）或降低（负值）元素。'0cm'等同于'baseline' |
| 百分值 – % | 通过距离（相对于1line-height1值的百分大小）升高（正值）或降低（负值）元素。'0%'等同于'baseline' |
| baseline | 默认。元素的基线与父元素的基线对齐。 |
| sub | 降低元素的基线到父元素合适的下标位置。 |
| super | 升高元素的基线到父元素合适的上标位置。 |
| top | 把对齐的子元素的顶端与line box顶端对齐。 |
| text-top | 把元素的顶端与父元素内容区域的顶端对齐。 |
| middle | 元素的中垂点与 父元素的基线加1/2父元素中字母x的高度 对齐。 |
| bottom | 把对齐的子元素的底端与line box底端对齐。 |
| text-bottom | 把元素的底端与父元素内容区域的底端对齐。 |
| inherit | 采用父元素相关属性的相同的指定值。 |

1. 首先**关于数值**，见下面的示例：

.test{vertical-align:-2px;}

我的理解为，元素相对于基线向下偏移两像素，这个常常用来修复单选框/复选框与12像素文字大小不对齐的问题。这个没有什么好说的。

2. 再者**关于百分数值**，百分值与数值，以我的眼光看去，代表了不同的思想，以及心态。前者代表着灵活，自由与开放，后者有严谨，精确，安稳之意。CSS中支持百分值的属性还不少，例如width/height,line-height,font-size,这里的vertical-align属性也是其一。提到百分值，必然牵扯到相对于那个属性（或值），例如宽度百分比都是相对于父块状元素的宽度值的，font-size的百分值是相对于向外的第一含有font-size属性的层的font-size大小而言的，而这里的vertical-align，有些不拘一格，是相对于此标签继承的line-height值决定的。例如，如下示例代码：

.test{vertical-align:-10%;}

假设这里的.test的标签继承的行高是20px，则这里的vertical-align:-10%所代表的实际值是：-10% \* 20 = 2(像素)。不过事情没有这么简单，CSS中的line-height是个非常精深的属性，建议您看看我之前写的这篇文章：[css行高line-height的一些深入理解及应用](http://www.zhangxinxu.com/wordpress/?p=384)，所以遇到像IE6/IE7这样吃三鹿奶粉长大的浏览器肯定会出点篓子的。什么篓子呢，就是“***IE6/IE7浏览器下的vertical-align的百分比值不支持小数line-height。***”您可以看看我处理的以小数值line-height继承的vertical-align百分比显示对比图[参展对象：IE6(观众扔鸡蛋)和Firefox3.6(美女送鲜花)]：



如果您现在使用的就是IE6或是IE7或是IE6/7内核的浏览器，同时相对这个bug有所见识，您可以狠狠地点击这里：[IE6/7 vertical-align百分值与line-height小数值bug](http://www.zhangxinxu.com/study/201005/verticle-align-percent-bug-demo.html)

要是您不想见到让您生气的IE下的糟糕表现，您可以狠狠地点击这里：[非小数line-height修复demo](http://www.zhangxinxu.com/study/201005/verticle-align-percent-value-demo.html)。

3. 最后关于其他属性，像是bottom，middle等，我是懒得说什么了，为何？可以看看下面这张兼容性表：

**兼容性(vertical-align)**

| **Internet Explorer** | | | | **Firefox** | | | | | **Safari** | | | | **Opera** | | | **Chrome** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5.5 | 6.0 | 7.0 | 8.0 | 1.0 | 1.5 | 2.0 | 3.0 | 3.5 | 1.3 | 2.0 | 3.1 | 4.0 | 9.2 | 9.5 | 10.0 | 2.0 |
| Buggy | Buggy | Buggy | Buggy | Buggy | Buggy | Buggy | Buggy | Buggy | Buggy | Buggy | Buggy | Buggy | Full | Full | Full | Buggy |

这是老外前辈整出的vertical-align兼容性表，虽然我不清楚Opera浏览器一栏的full是个什么意思，但是”buggy”知道，就是臭虫成群，bug成堆的意思，我想，做过简单研究的都会注意到不同浏览器下的差异确实明显。

不过vertical-align的这些属性值也不是一无是处，有些属性，例如text-bottom和middle有时会用来修正一些样式表现或是实现特定布局。这个后面再说~~不过有一点知识有必要补一补就是，baseline,middle,top,bottom到底指什么，看下面这张图，我是找了张图ps了好一会儿：



我们刚学英语的时使用的那个英语本子每行有四条线，其中底部第二条线就是基线，是a,c,z,x等字母的底边线。下图的红色线即为基线。

**三、为什么我的vertical-align属性不起作用？**

知道了vertical-align是垂直对齐的意思，不少经验尚浅的同行会试着使用这个属性实现一些垂直方向上的对齐效果，会发现有时候可以，有时候又不起作用，不知道为什么？不急，慢慢来。

我们知道display也有很多属性值，其中以inline/inline-block/block三个最常见，这代表了页面上三种不同水平的元素。我常常会以液态/固液混合态/固态加以形象化思考，对应于现实中的事物就是：牛奶/果冻/坚果。

们都知道，每个人都有不同的嗜好，有的人喜欢吃甜食，有的人喜欢吃辣的东西，有的人不喜欢吃芹菜，有的人不喜欢吃羊肉等等。CSS中的有些元素也是这样，他们有的只对牛奶感兴趣，有的只喜欢吃坚果和果冻，而讨厌牛奶。而vertical-align呢，是个比较挑食的家伙，它只喜欢吃果冻，从小吃果冻长大，没有了果冻，它就会闹脾气，对你不理不睬。我称之为“果冻依赖型元素”，又称之为“inline-block依赖型元素”，也就是说，只有一个元素属于inline或是inline-block（table-cell也可以理解为inline-block水平）水平，其身上的vertical-align属性才会起作用。所以，类似下面的代码就不会起作用：

s

div{vertical-align:middle;}

所谓inline-block水平的元素，就是既可以“吸”又可以“咬”的元素，既可以与inline水平元素混排，又能设置高宽属性的元素。哪些元素呢，例如图片，按钮，单复选框，单行/多行文本框等HTML控件，只有这些元素默认情况下会对vertical-align属性起作用。

虽然vertical-align属性只会在inline-block水平的元素上期作用，但是其影响到的元素涉及到inline属性的元素，这里千万记住，inline水平元素受vertical-align属性而位置改变等不是因为其对vertical-align属性敏感或起作用，而是受制于整个line box的变化而不得不变化的，这个后面会较为深入的分析。

zxx://最近较忙，下面内容写于14日

**四、vertical-align属性是如何起作用的？**

这里，我使用vertical-align:middle属性作为例子，讲讲我对vertical-align是如何起作用的理解。

CSS参考手册上说vertical-align:middle是将当前元素放在父元素的中间，大致一读，似乎是那么回事，但是细细一想，不知道具体指什么，概念模糊。我们从简单的开始，一步一步探究生效的原理。

**1. 一些简单的例子**  
这里先列举几个简单的例子，方便对vertical-align:middle有了初步的直观的认识。例子内容如下，创建一个inline-block属性的元素，此元素的高度和宽度均为4像素，为了对比效果明显，我使用黑白二色，背景黑色，此4像素\*4像素的元素为白色，vertical-align属性依次为middle，bottom和text-bottom，当然，文字是必须的，否则何见对齐呢！整个实例的HTML代码都是一致的，唯一不同的就是一段vertical-align属性，测试环境均是Firefox3.6浏览器，HTML代码如下：

<span class="box">

<span class="dot"></span>

我是一段卡哇伊的文字。

</span>

***实例一：默认属性（也就是baseline）***  
参见如下的CSS代码：

.box{background:black; color:white; padding-left:20px;}

.dot{display:inline-block; width:4px; height:4px; background:white;}

结果如下图：



***实例二：bottom***  
参见如下的CSS代码：

.box{background:black; color:white; padding-left:20px;}

.dot{display:inline-block; width:4px; height:4px; background:white; vertical-align:bottom;}

结果如下图：

bottom属性的vertical-align效果 张鑫旭-鑫空间-鑫生活

***实例三：text-bottom***  
参见如下的CSS代码：

.box{background:black; color:white; padding-left:20px;}

.dot{display:inline-block; width:4px; height:4px; background:white; vertical-align:text-bottom;}

结果如下图：

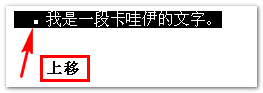
text-bottom属性的vertical-align效果 张鑫旭-鑫空间-鑫生活

对比vertical-align:bottom和vertical-align:text-bottom，他们的表现似乎一样，实际上呢，这里的表现一致只是一个巧合而已，此话怎讲？要显示其差异很简单，添加一个line-height值，您就会看到不一样的地方了。

***实例四：line-height:10px + vertical-align:bottom***  
参见如下的CSS代码：

.box{background:black; color:white; padding-left:20px; line-height:10px;}

.dot{display:inline-block; width:4px; height:4px; background:white; vertical-align:bottom;}

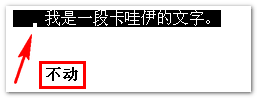
结果如下截图：  


而如果vertical-align的属性是text-bottom呢？

***实例五：line-height:10px + vertical-align:text-bottom***  
参见如下的CSS代码：

.box{background:black; color:white; padding-left:20px; line-height:10px;}

.dot{display:inline-block; width:4px; height:4px; background:white; vertical-align:text-bottom;}

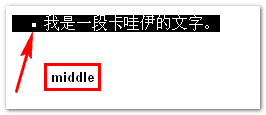
结果如下图：  


可以看到，vertical-align:text-bottom属性的那个小方点的位置没有随着line-height的改变而改变，还是与文字的底部对齐。如果您细看关于text-bottom以及bottom属性的定义，您可以找到其表现的原因：text-bottom是与父标签的文字底部对齐，在这里也就是后面的“我是一个卡哇伊的文字”这段文字对齐，而bottom是相对于父标签的底部对齐，而如果您熟悉line box模型且对高度的本质有所了解，那么您就会明白为什么line-height会改变标签的bottom的位置了。这点的了解有助于理解下面vertical-align:middle生效的过程及原理。

***实例六：middle***  
参见如下的CSS代码：

.box{background:black; color:white; padding-left:20px;}

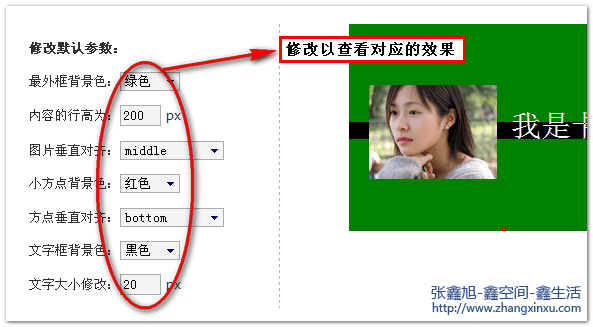
.dot{display:inline-block; width:4px; height:4px; background:white; vertical-align:middle;}

结果如下图所示：

**2. 复杂点的例子**

如果说上面的例子只是热身的话，那么现在就要开始进入实战了。在这里的实例中，我们添加一个新的元素，就是图片，我们通过观察不同属性下图片的样式表现来理解vertical-align:middle等相关属性是怎么一回事，是怎样产生页面表现的。

为了方便大家理解，同时也是为了自己加深对vertical-align的理解，我特地做了个简易的vertical-align属性实验面板，见下图：



此vertical-align实验面板页面地址如下：http://www.zhangxinxu.com/study/201005/verticle-align-test-demo.html，您还可以狠狠地点击这里：[vertical-align属性测试面板页面](http://www.zhangxinxu.com/study/201005/verticle-align-test-demo.html)

通过修改，行高以及对应元素的vertical-align属性，您会在页面上直接看到效果，同时修改文字大小的页面表现也有助于您对inline box模型的理解。由于IE6/7对inline box的解释不完善，同时对vertical-align的认识也是一坨香喷喷的鸟屎，所以请不要再IE6/7下使用此面板，会对您产生误导的，IE8，Firefox，最新的Opera浏览器，chrome。Safari浏览器都是上佳的选择。关于IE6/7的解释问题会在下集介绍。

这里说点我的认识吧：***vertical-align:middle属性的表现与否，仅仅与其父标签有关，至于我们通常看到的与后面的文字垂直居中显示那都是假象！***这就是我的认识，当然，我也可以为我的理解提供强有力的证据，我们来看vertical-align属性实验面板。

当我们将图片vertical-align:middle属性改为其他的时候，例如top，或是bottom的时候，您会发现，图片位置上移或是下移了，且居边了，但是文字在那里却纹丝不动，显然，vertical-align的本质上是个独立的个体，与后面的line水平的元素是不存在直接的关系的。两者没有必然的联系，这是一个需要认识清楚的重要的东西。