## strut - 幽灵空白节点

“幽灵空白节点”实际上也是一个盒子，不过是个假想盒，名叫“strut”，是一个存在于每个“行框盒子”前面，同时具有该元素的字体和行高属性的0宽度的内联盒。

## 基线

字母x的下边缘（线）就是我们的基线

line-height行高的定义就是两基线的间距

vertical-align的默认值就是基线

x-height，指的是字母x的高度，术语描述就是基线和等分线（meanline）（也称作中线，midline）之间的距离



vertical-align:middle。这里的middle是中间的意思。注意，跟上面的median（中线）不是一个意思。在CSS世界中，middle指的是基线往上1/2 x-height高度。我们可以近似理解为字母x交叉点那个位置。

强调：

* line-height不能影响内联替换元素，有时候看到的表象的影响其实是影响了行框盒子里的“幽灵空白节点”
* 替换元素和内联非替换元素在一起时，由于同属内联元素，因此，会共同形成一个“行框盒子”，line-height在这个混合元素的“行框盒子”中扮演的角色是决定这个行盒的最小高度。例子：明明文字设置了line-height为20px，但是，如果文字后面有小图标，最后“行框盒子”高度却是21px或是22px。这种现象背后最大的黑手其实是vertical-align属性
* 对于块级元素，line-height对其本身是没有任何作用的，我们平时改变line-height，块级元素的高度跟着变化实际上是通过改变块级元素里面内联级别元素占据的高度实现的。

**display:inline**的内联元素的**line-height**对最终内容盒子的高度没有显式影响，因为行高影响的是行框，并不是内联元素的高度，高度由font-size和font-family影响。而**display:inline-block**有，因为其内部盒子是块状盒子，块状盒子的内容高度由line-height、padding、border等共同决定****

### 百分比值

凡是百分比值，均是需要一个相对计算的值，例如，margin和padding是相对于宽度计算的，line-height是相对于font-size计算的，而这里的vertical- align属性的百分比值则是相对于line-height的计算值计算的。

# 高度的决定因素

高度对于我来说也是很魔幻的，譬如line-height能决定块状元素的高度，却完全不能左右内联元素的高度。font-size也是只对内联元素高度有影响，却不是唯一的决定因素。

* 块状元素：line-height、height、padding、border共同作用
* 内联元素：内联盒子共同作用后的行框盒子结果，每个内联盒子，如果是纯文本，又是font-size和font-family作用的结果