

# CSS 基线之道

## line-height 到底有多 height?

### 行距、行间距傻傻分不清

行距 = 行间距 + 字体大小

行距：相邻文本行间上一个文本行基线和下一文本行基线间的距离。

行间距：指相邻文本行间上一个文本行下行线(ascent)和下一文本行上行线(descent)间的距离。

在 CSS 中，CSS 属性 line-height 则是用于设置真实的行距。从 W3C Rec 中看出，line-height 就是行距，而 line-height 的字面意思即为“行高”，推导结果 CSS 中行高即是行距。

这里我们了解到**行高**，**行距**和**行间距**的区别了。那接下来要介绍 line-height 的一个重要特性——垂直居中性。

行距 (leading) = 行间距 (line-space) + 字体大小 (font Size)。字形大小我们可以通过 font-size 来设置，而 line-height 就更不用说了，问题是行间距所占的空间是怎样分配的呢？

方案 1: 不是说行间距就是上一行的文字的底到下一行文字的顶间的距离吗？那直接分配到 A 位置就好了。



方案 2: 如果方案 1 的分配方案合理，那么分配到 B 位置就也是 OK 的。



方案 3: 一边倒的分配方案太不美观了吧！不如将行间距对半开，然后分别分配到上下两端不就 OK 了吗！



CSS 采用的就是方案 3。这是引用了 Half-leading 的概念，Half-leading = 行距/2.

## line-height 属性

normal

Tells user agents to set the used value to a "reasonable" value based on the font of the element. The value has the same meaning as `normal`. We recommend a used value for 'normal' between 1.0 to 1.2. The computed value is 'normal'.

$$14/10 = 1.4$$

$$45/40 = 1.125$$

$$92/80 = 1.15$$

average: 1.225 约为 1.2

不同浏览器的 normal 值不相同;

同一个浏览器下, font-size 值不同, normal 值也会有变化;

同一浏览器下, font-size 值相同, font-family 值不同, normal 值也会有变化;

normal 的平均值确实是在 1.0~1.2 之间 (含两端), 但具体到特定浏览器、font-family 和 font-size 时, normal 的实际值可能会大于 1.2。

<length>

指定的长度用于计算线盒高度。负值是非法的。

设置固定值, 单位可以是 px、pt。好处是简单——设置是什么, line-height 的实际高度就是什么。坏处是子元素默认情况下会继承父容器的 line-height 属性, 若子元素的 font-size 大于父容器的 font-size 属性值, 那么子元素的文本行会十分密集, 降低可读性。所以我们一般采用相对 font-size 实际大小来设置 line-height 值的方式, 如默认 normal 方式。

<percentage>

属性的计算值是这个百分比乘以元素的计算字体大小。负值是非法的。

既然采用副作用那么大, 那采用这个相对值就万事大吉了吧! 非也, 首先我们要弄清楚这个的参考系是啥, 另外还要明白子元素的 line-height 到底继承的了哪个值, 是值还是父容器实际的 line-height 值。

1. 的参考系的确是 font-size;
2. 子元素继承的是父容器实际的 line-height 值。也就是说父容器设置为 font-size:20px;line-height:200%;, 那么子元素继承来的 line-height 值为 40px, 而不是 200%。因此又回到方式的问题上了。

<number>

属性的使用值是这个数字乘以元素的字体大小。负值是非法的。计算值与指定值相同。

和方式一样，以 font-size 作为参考系，以相对值的方式设置 line-height。唯一的不同就是子元素继承的是父容器的值，参考系自动变更为子元素的 font-size。

其实 line-height:1.2em;和 line-height:1.2;是等价的。若想将参考系改为根元素的 font-size，则需要采用 CSS3 新增的 line-height:1.2rem 单位了。

根据 WCAG2.0(万维网内容可存取性指南)规定“段落中的行距至少要 1.5 倍”，那么是否在 body 那设置一个就一劳永逸呢？请看

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style type="text/css">
    body{
      font-size: 16px;
      line-height: 1.5;
    }
    h1 {font-size:32px;}
    p {font-size:16px;}
    #footer {font-size:12px;}
  </style>
</head>
<body>
  <h1>深入理解 line-height 和 vertical-align</h1>
  <p>In my dual profession as an educator and health care provider, I have
worked with numerous children infected with the virus that causes AIDS. The
relationships that I have had with these special kids have been gifts in my
life. They have taught me so many things, but I have especially learned that
great courage can be found in the smallest of packages. Let me tell you about
Tyler. </p>
  <div id="footer">bed and whispered, "I might die soon.
I' m not scared. When I die, please dress me in red. Mom promised she' s
coming to heaven, too. I' ll be playing when she gets there, and I want to
make sure she can find me." </div>
</body>
</html>
```

# 深入理解line-height和vertical-align

In my dual profession as an educator and health care provider, I have worked with numerous children infected with the virus that causes AIDS. The relationships that I have had with these special kids have been gifts in my life. They have taught me so many things, but I have especially learned that great courage can be found in the smallest of packages. Let me tell you about Tyler.

bed and whispered, "I might die soon. I'm not scared. When I die, please dress me in red. Mom promised she's coming to heaven, too. I'll be playing when she gets there, and I want to make sure she can find me."

看对于 h1 标题栏而言，行距太多了。于是得出如下配置：

```
body{line-height:1.5;}  
h1,h2,h3,h4,h5,h6{line-height:1.2;}
```

## 深入理解line-height和vertical-align

行距确实减小了。

### vertical-align 到底如何对齐呢？

#### vertical-align 属性

'vertical-align'

Value: baseline | sub | super | top | text-top | middle | bottom | text-bottom | <percentage> | <length> | inherit

Initial: baseline

Applies to: inline-level and 'table-cell' elements

Inherited: no

Percentages: refer to the 'line-height' of the element itself

Media: visual

Computed value: for and the absolute length, otherwise as specified

<length>:设置相对于 baseline 的距离, 正数表示位于 baseline 的上方, 负数表示位于 baseline 的下方;

<percentage>:设置以 line-height 为参考系, 相对于 baseline 的距离, 正数表示位于 baseline 的上方, 负数表示位于 baseline 的下方;

baseline:默认值。元素的基线与父元素的基线对齐;

top:把元素 line box 上边框对齐父元素的 line box 上边框;

text-top:把元素 line box 上边框对齐父元素的 ascent(即 content top edge);

super:升高元素的基线到父元素合适的上标位置;

middle:把元素 line box 中垂点与父元素基线 + x-height/2 的高度对齐;

sub:降低元素的基线到父元素合适的下标位置;

text-bottom:把元素 line box 下边框对齐父元素的 descent(即 content bottom edge);

bottom:把元素 line box 下边框对齐父元素的 line box 下边框;

inherit:继承父元素的对齐方式。

怎么这么多规则要记啊? 我记性不好难道到时还要挨个查吗? 其实归纳一下就 OK 了!

1. 对齐操作必定涉及操作元素和参考系元素, 而 vertical-align 的值全是指参考系元素的位置, 而操作元素则以 baseline 或 linebox 上中下作对齐;
2. 默认对齐方式为 baseline, 数量值均是相对于 baseline 而言。

**vertical-align 仅对 inline-level 和 table-cell 元素有效**

**注意: vertical-align 仅对 inline-level 和 table-cell 元素有效。**

### 1. 默认对齐方式——baseline

```
<div style="font-size:14px;">
  <span id="obj" style="font-size:40px;">line-height x vertical-align</span>
  x for reference frame
</div>
```

# line-height x vertical-align x for reference frame

这里 x for reference frame 作为参考系, 而它的 baseline 就是 span#obj 所要对齐的 baseline 了。

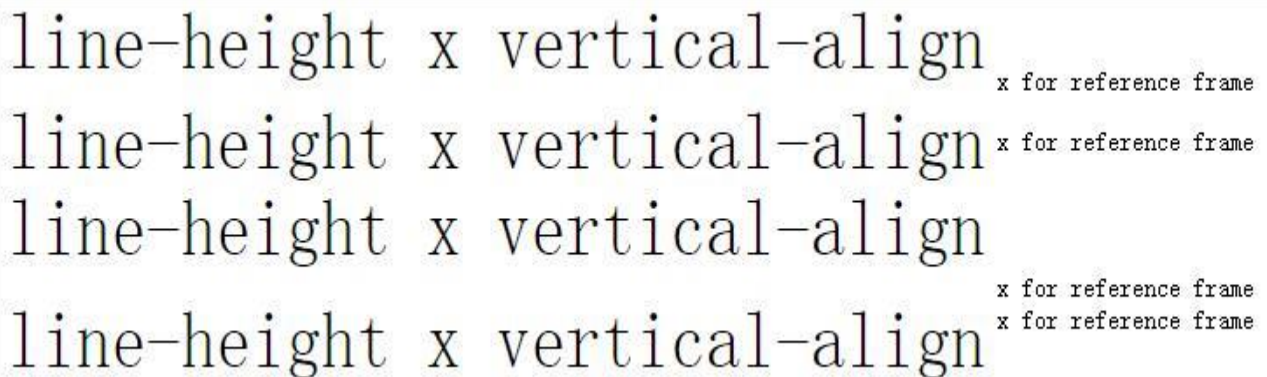
那么在 baseline 的基础上的设置<length>和<percentage>

```
<div style="font-size:14px;">
```

```

    <span id="obj" style="font-size:40px;vertical-align:10px;">line-height x
vertical-align</span>
    x for reference frame
</div>
<div style="font-size:14px;">
    <span id="obj" style="font-size:40px;vertical-align:-10px;">line-height x
vertical-align</span>
    x for reference frame
</div>
<div style="font-size:14px;line-height:1;">
    <span id="obj" style="font-size:40px;vertical-align:50%;">line-height x
vertical-align</span>
    x for reference frame
</div>
<div style="font-size:14px;line-height:1;">
    <span id="obj" style="font-size:40px;vertical-align:-50%;">line-height x
vertical-align</span>
    x for reference frame
</div>

```



line-height x vertical-align x for reference frame

line-height x vertical-align x for reference frame

line-height x vertical-align

line-height x vertical-align x for reference frame

2. top——把元素 line box 上边框对齐父元素的 line box 上边框  
我们将上面的示例稍微改一下

```

<span style="font-size:14px;">
    <span id="obj" style="font-size:40px;vertical-align:top;">line-height x
vertical-align</span>
    x for reference frame
</span>

```

# line-height x vertical-align x for reference frame

确实不同了，但这无法证明是元素的 line box 上边框对齐父元素的 line box 上边框哦。  
那么我们改改代码看看

```
<body style="margin:0 10px;padding:0;">
  <div style="border:solid 1px blue;font-size:14px;line-height:1;">
    <span id="parent" style="background:red;line-height:1;">
      <span id="obj" style="background:yellow;font-size:40px;vertical-align:top;line-height:1;">line-height x vertical-align</span>
      x for reference frame
    </span>
  </div>
</body>
```

## line-height x vertical-align x for reference frame

通过 line-height:1 使 line box 与 content box/area 的高度一致，虽然 span#parent 和 span#obj 的上边框对齐，但还不能说明什么。

```
<body style="margin:0 10px;padding:0;">
  <div style="position:relative;top:100px;;border:solid 1px blue;font-size:14px;line-height:1;">
    <span id="parent" style="background:red;line-height:1;">
      <span id="obj" style="background:yellow;font-size:40px;vertical-align:top;line-height:1;margin-top:100px;padding-top:100px;background-clip:content-box;">line-height x vertical-align</span>
      x for reference frame
    </span>
  </div>
</body>
```

## line-height x vertical-align x for reference frame

没有任何变化。那改变 line-height 又如何呢？

```
<body style="margin:0 10px;padding:0;">
```



```

<div style="border:solid 1px blue;font-size:14px;line-height:1;">
  <span id="parent" style="background:red;line-height:2;">
    <span style="display:inline-block;vertical-align:top;background:green;">
      <span id="obj" style="background:yellow;font-size:40px;line-
height:2;">line-height x vertical-align</span>
    </span>
    x for reference frame
  </span>
</div>
</body>

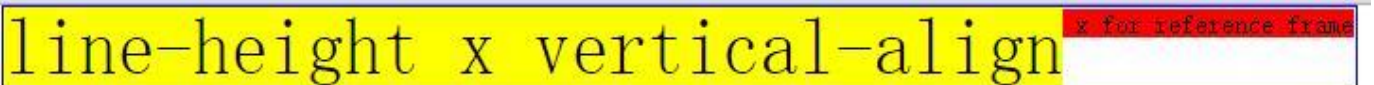
```



为了让 span#obj 的 Half-leading 清晰可见，特意添加一个 display:inline-block 的 inline box 包裹着 span#obj。而 span#parent 也增大了 Half-leading 的高度。现在可以我们清晰看到确实是 span#obj 的 line box 的上边框对齐父元素的 line box 上边框。

（同理证明了 vertical-align:bottom 是把元素 line box 下边框对齐父元素的 line box 下边框；）

**注意：**chrome 下若外层 div 不添加 font-size:14px;line-height:1;属性，将导致 span#parent 上有条空白间隙



原因十分简单，那是因为 span#parent 的对齐方式是 baseline，参考系是 div 的 baseline，而 div 的 line-height 为 normal，有空白间隙就是当然的事了。通过 JS 就可以看清楚了。

```

var div = document.getElementsByTagName('div')[0]
console.log(div.childNodes[0].nodeType) // 显示 3，就是 TextNode

```

其实除了在 div 上设置 line-height:1 之外，我们还可以在 span#parent 上设置 vertical-align:top 来解决。

```

<body style="margin:0 10px;padding:0;">
  <div style="border:solid 1px blue;font-size:14px;">
    <span id="parent" style="background:red;line-height:1;vertical-
align:top;">

```

```

    <span id="obj" style="background:yellow;font-size:40px;vertical-align:top;line-height:1;">line-height x vertical-align</span>
    x for reference frame
  </span>
</div>
</body>

```



3. text-top——把元素的 line box 上边框对齐父元素的 ascent (即 content top edge)

```

<body style="margin:0 10px;padding:0;">
  <div style="position:relative;top:100px;border:solid 1px blue;font-size:14px;line-height:1;"> <span id="parent" style="background:red;line-height:2;">
    <span id="obj" style="background:yellow;vertical-align:text-top;font-size:2px;line-height:1;">*</span>
    x for reference frame
  </span>
</div>
</body>

```



```

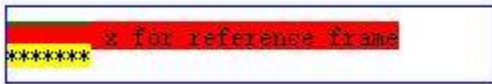
<body style="margin:0 10px;padding:0;">
  <div style="position:relative;top:100px;border:solid 1px blue;font-size:14px;line-height:1;"> <span id="parent" style="background:red;line-height:2;">
    <span style="display:inline-block;vertical-align:text-top;border-top:solid 2px green;">
      <span id="obj" style="background:yellow;font-size:2px;line-height:2;">*****</span>
    </span>
  </div>
</body>

```

```

        x for reference frame
    </span>
</div>
</body>

```



4. middle——把元素 line box 中垂点与父元素基线 + x-height/2 的高度对齐

```

<body style="margin:0 10px;padding:0;">
  <div style="border:solid 1px blue;font-size:40px;line-height:1;">
    <span id="parent" style="background:red;line-height:1;">
      <span id="obj" style="padding-top:10px;display:inline-
block;background:yellow;font-size:15px;line-height:1;vertical-
align:middle;">*****</span>x for reference frame
    </span>
  </div>
</body>

```



### 注意

当元素的 `display:inline-block/inline-table` 等对应的是 `atomic inline-level box` 时，其 `line box` 高度为 `margin box` 的高度。若元素对应的是 `inline box`，则其最小高度为 `line-height`，最大则由子盒子决定。