额外知识补充

王红元 coderwhy

目录
content

CSS属性 - vertical-align

Name:	***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 <percentage> and <length> the absolute length, otherwise as specified

This property affects the vertical positioning inside a line box of the boxes generated by an inline-level element.

深入理解vertical-align – line boxes

This property affects the vertical positioning inside a line box of the boxes generated by an inline-level element.

- **官方文档的翻译：vertical-align会影响 行内块级元素 在一个 行盒 中垂直方向的位置**
- **思考：一个div没有设置高度的时候，会不会有高度?**
  - 没有内容，没有高度
  - 有内容，内容撑起来高度
- **但是内容撑起来高度的本质是什么呢?**
  - 内容有行高（line-height），撑起来了div的高度
- **行高为什么可以撑起div的高度?**
  - 这是因为line boxes的存在，并且line-boxes有一个特性，包裹每行的inline level
  - 而其中的文字是有行高的，必须将整个行高包裹进去，才算包裹这个line-level
- **那么，进一步思考：**
  - 如果这个div中有图片，文字，inline-block，甚至他们设置了margin这些属性呢?

深入理解vertical-align – 不同情况分析

- 情况一：只有文字时，line boxes如何包裹内容？（注意：红色是包裹的div，下面也都一样）

div元素中的文本

- 情况二：有图片，有文字时，line-boxes如何包裹内容?

- 情况三：有图片，有文字，有inline-block（比图片要大）如何包裹内容?

深入理解vertical-align – 不同情况分析

- 情况四：有图片，有文字，有inline-block（比图片要大）而且设置了margin-bottom，如何包裹内容?

- 情况五：有图片、文字、inline-block（比图片要大）而且设置了margin-bottom并且有文字，如何包裹内容?

vertical-align的baseline

- **结论：line-boxes一定会想办法包裹住当前行中所有的内容。**
- **但是，但是为什么对齐方式千奇百怪呢?**
  - 你认为的千奇百怪，其实有它的内在规律
  - 答案就是baseline对齐
- **我们来看官方vertical-align的默认值：没错，就是baseline**

Name:	***vertical-align***
Value:	baseline | sub | super | top | text-top | middle | bottom | text-bottom | <percentage> | <length> | inherit
Initial:	baseline

- **但是baseline都是谁呢?**
  - 文本的baseline是字母x的下方
  - Inline-block默认的baseline是margin-bottom的底部（没有，就是盒子的底部）
  - Inline-block有文本时，baseline是最后一行文本的x的下方
- **一切都解释通了**

vertical-align的其他值

- **现在，对于不同的取值就非常容易理解了**
  - baseline(默认值)：基线对齐（你得先明白什么是基线）
  - top：把行内级盒子的顶部跟line boxes顶部对齐
  - middle：行内级盒子的中心点与父盒基线加上x-height一半的线对齐
  - bottom：把行内级盒子的底部跟line box底部对齐
  - <percentage>：把行内级盒子提升或者下降一段距离（距离相对于line-height计算\元素高度）， 0%意味着同baseline一样
  - <length>：把行内级盒子提升或者下降一段距离，0cm意味着同baseline一样

- **解决图片下边缘的间隙方法:**
  - 方法一: 设置成top/middle/bottom
  - 方法二: 将图片设置为block元素

CSS属性 - transform

- CSS transform属性允许你旋转，缩放，倾斜或平移给定元素。
- Transform是形变的意思，transformer就是变形金刚

- 常见的函数transform function有：
  - 平移：translate(x, y)
  - 缩放：scale(x, y)
  - 旋转：rotate(deg)
  - 倾斜：skew(deg, deg)
- 通过上面的几个函数，我们可以改变某个元素的形变

位移 - translate

- **平移：translate(x, y)**
- **值个数**
  - 一个值时，设置x轴上的位移
  - 二个值时，设置x轴和y轴上的位移
- **值类型：**
  - 数字：100px
  - 百分比：参照元素本身（ refer to the size of bounding box ）

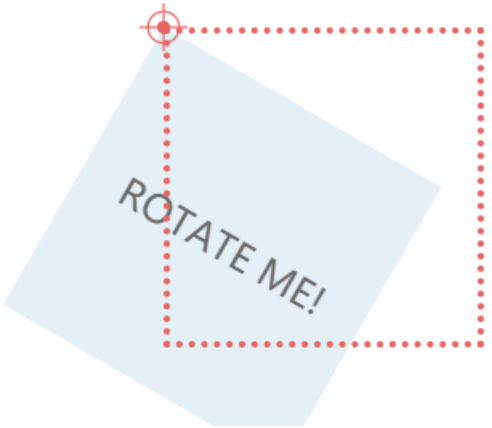
缩放 - scale

- **缩放：scale(x, y)**
- **值个数**
  - 一个值时，设置x轴上的缩放
  - 二个值时，设置x轴和y轴上的缩放
- **值类型：**
  - 数字：
    - ✓ 1：保持不变
    - ✓ 2：放大一倍
    - ✓ 0.5：缩小一半
  - 百分比：不支持百分比

transform-origin

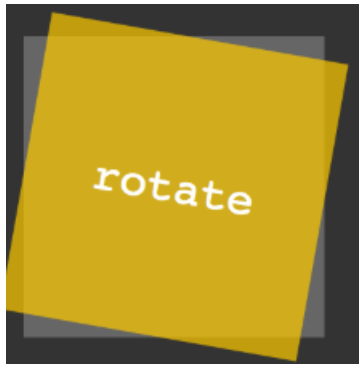
- **transform-origin：变形的原点**
- **一个值：**
  - 设置x轴的原点
- **两个值：**
  - 设置x轴和y轴的原点
- **必须是<length>，<percentage>，或 left, center, right, top, bottom关键字中的一个**
  - left, center, right, top, bottom关键字
  - length：从左上角开始计算
  - 百分比：参考元素本身大小

```
transform-origin: top left;
```



缩放 - rotate

- **旋转：rotate(deg)**
- **值个数**
  - 一个值时，表示旋转的角度
- **值类型：**
  - deg：旋转的角度
  - 正数为顺时针
  - 负数为逆时针
- **注意：旋转的原点受transform-origin的影响**



倾斜 - skew

- **旋转：skew(x, y)**
- **值个数**
  - 一个值时，表示x轴上的倾斜
  - 二个值时，表示x轴和y轴上的倾斜
- **值类型：**
  - deg：旋转的角度
  - 正数为顺时针
  - 负数为逆时针
- **注意：旋转的原点受transform-origin的影响**

过渡动画 - transition

- **transition CSS 属性是 transition-property，transition-duration，transition-timing-function 和 transition-delay 的一个简写属性。**
- transition-property：指定应用过渡属性的名称
  - 可以写all表示所有可动画的属性
  - 属性是否支持动画查看文档
- transition-duration：指定过渡动画所需的时间
  - 单位可以是秒（s）或毫秒（ms）
- transition-timing-function：指定动画的变化曲线
  - https://developer.mozilla.org/zh-CN/docs/Web/CSS/transition-timing-function
- transition-delay：指定过渡动画执行之前的等待时间

```
<single-transition>#

where
<single-transition> = [ none | <single-transition-property> ] || <time> || <timing-function> || <time>
```

关键帧动画

- **之前我们学习了transition来进行过渡动画，但是过渡动画只能控制首尾两个值：**
  - 从关键帧动画的角度相当于只是定义了两帧的状态：第一帧和最后一帧。
  - 如果我们希望可以有更多状态的变化，可以直接使用关键帧动画。
- **关键帧动画使用@keyframes来定义多个变化状态，并且使用animation-name来声明匹配：**
  - 1.使用 @keyframes创建一个规则
  - 2. @keyframes中使用百分比定义各个阶段的样式
  - 3. 通过animation将动画添加到属性上
- **另外，也可以使用from和to关键字：**
  - from相当于0%
  - to相当于100%

animation属性

- **CSS animation 属性是 animation-name，animation-duration, animation-timing-function，animation-delay，animation-iteration-count，animation-direction，animation-fill-mode 和 animation-play-state 属性的一个简写属性形式。**
- animation-name：指定执行哪一个关键帧动画
- animation-duration：指定动画的持续时间
- animation-timing-function：指定动画的变化曲线
- animation-delay：指定延迟执行的时间
- animation-iteration-count：指定动画执行的次数，执行infinite表示无限动画
- animation-direction：指定方向，常用值normal和reverse
- animation-fill-mode：执行动画最后保留哪一个值
  - none：回到没有执行动画的位置
  - forwards：动画最后一帧的位置
  - backwards：动画第一帧的位置
- animation-play-state：指定动画运行或者暂停（在JavaScript中使用，用于暂停动画）