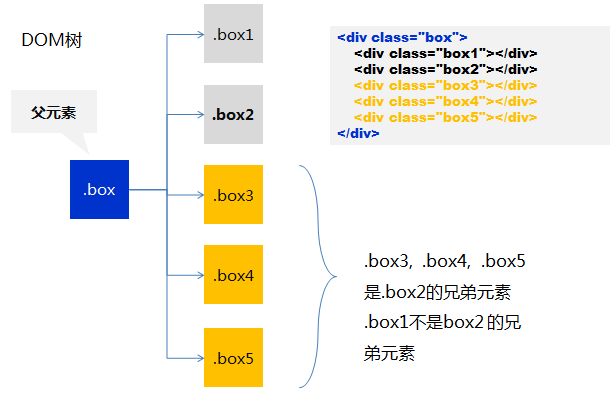
# 复习

## 兄弟元素



## 浮动



清除浮动是清除浮动的影响，而不是清除浮动本身，浮动还在，只不过不对其他元素产生作用

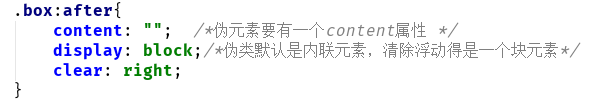
如果父元素未设置高度，子元素的浮动会造成父元素高度塌陷。

为父元素设置 overflow: hidden。防止高度塌陷。

注意！！！上述方案有”潜在风险”！！！

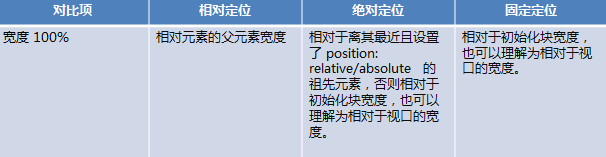
如果浮动元素的内容溢出，会被“隐藏掉”。

理想解决方案：通过为父元素添加一个伪元素，并为其设置清除浮动解决。



## 定位





# CSS2文本

## 字体颜色

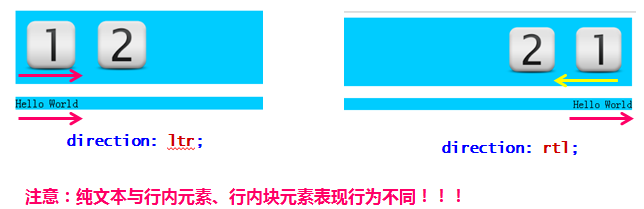
* color\_name 规定颜色值为颜色名称的颜色（比如 red）。
* hex\_number 规定颜色值为十六进制值的颜色（比如 #ff0000）。
* rgb\_number 规定颜色值为 rgb 代码的颜色（比如 rgb(255,0,0)）。
* 透明色 rgba(255, 0, 0, .8) a一般写为点几而不写为零点几

## 文本方向direction

ltr 默认。一行中文本、行内元素、行内块元素 排列方向从左到右。

rtl一行中文本、行内元素、行内块元素 排列方向从右到左。

inherit 规定应该从父元素继承 direction 属性的值。



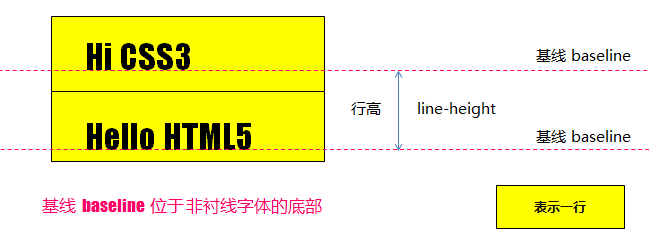
**Direction 使用技巧**

* **快速实现两个按钮的对调，而无需JS。**



## line-height

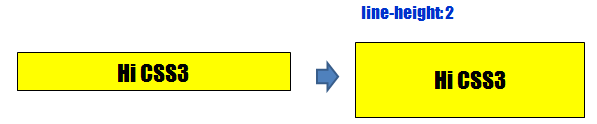
* 定义了该元素中基线之间的最小距离而不是最大距离



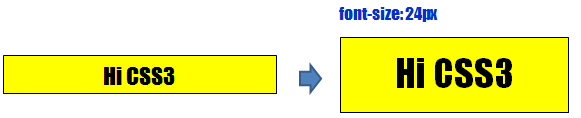
* 行高可选值
  + normal 默认。设置合理的行间距。
  + *number* 设置数字，此数字会与当前的字体尺寸相乘来设置行间距。
  + *length* 设置固定的行间距。
  + *%* 基于当前字体尺寸的百分比行间距。
  + inherit 规定应该从父元素继承 line-height 属性的值。
* 行高是根据字体大小计算的！！！
* 规范要求，行高 = 字体大小\*1.2 不同浏览器实现有差异

行高对元素大小影响

**如果元素未指定高度，行高会导致元素高度的变化**

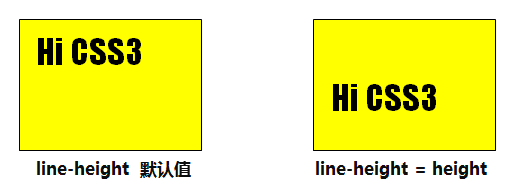


**如果元素未指定高度，字体的变化，会导致行高变化，进而会导致元素高度的变化**



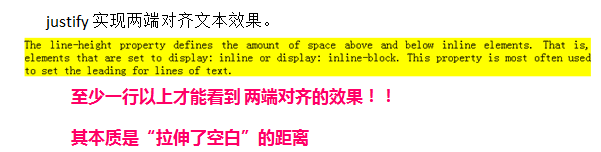
**行高小技巧**

**设置 line-height = height 一行中的文本，行内元素，行内块元素会在垂直方向居中，父元素中只能是一行**



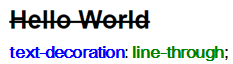
## text-align

* 规定一行中**文本内容**，**行内元素**，**行内块元素**的对齐方式
* 可选值
  + left 把文本排列到左边。默认值：由浏览器决定。
  + right 把文本排列到右边。
  + center 把文本排列到中间。
  + justify 实现两端对齐文本效果。
  + inherit 规定应该从父元素继承 text-align 属性的值。



## text-decoration

* 文本的修饰。
  + none 默认。定义标准的文本。
  + underline 定义文本下的一条线。
  + overline 定义文本上的一条线。
  + line-through 定义穿过文本下的一条线。
  + inherit 规定应该从父元素继承text-decoration属性的值

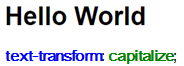
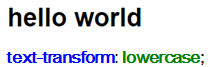
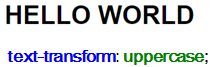
  

<a>元素 默认设置了text-decoration: underline;

我们可以设置text-decoration: none; 取消下划线

## text-transform

* 控制文本的大小写
  + none 默认。定义带有小写字母和大写字母的标准的文本。
  + capitalize 文本中的**每个单词以大写字母开头**。
  + uppercase 定义仅有大写字母。
  + lowercase 定义无大写字母，仅有小写字母。
  + inherit 规定应该从父元素继承 text-transform 属性的值。

## word-spacing

* 该属性定义元素中字之间插入多少空白符。针对这个属性，“字” 定义为由空白符包围的一个字符串。如果指定为长度值，会调整字之间的通常间隔；所以，normal 就等同于设置为 0。允许指定负长度值，这会让字之间挤得更紧。
  + normal 默认。规定字符间没有额外的空间。
  + *length* 定义字符间的固定空间（允许使用负值）。
  + inherit 规定应该从父元素继承 letter-spacing 属性的值。

## letter-spacing

* 该属性定义了在文本字符框之间插入多少空间。由于字符字形通常比其字符框要窄，指定长度值时，会调整字母之间通常的间隔。因此，normal 就相当于值为 0。
  + normal 默认。规定字符间没有额外的空间。
  + *length* 定义字符间的固定空间（允许使用负值）。
  + inherit 规定应该从父元素继承 letter-spacing 属性的值。

## text-indent

* 首行缩进
  + *length* 定义固定的缩进。默认值：0。
  + *%* 定义基于父元素宽度的百分比的缩进。
  + inherit规定应该从父元素继承text-indent属性的值
* **实际开发中我们使用em，确保一个很好的比例!!!**

# CSS2字体

## font-family

* font-family 规定元素的字体系列（字体族）。
* font-family 可以把多个字体名称作为一个“回退”系统来保存。如果浏览器不支持第一个字体，则会尝试下一个。
* 例如
  + Microsoft YaHei, tahoma, arial, Hiragino Sans GB, sans-serif

有两种主要类型的字体

1. 衬线字体(serif)：在字的笔划开始及结束的地方有额外的装饰，而且笔划的粗细会因直横的不同而有不同。
2. 非衬线字体(Sans Serif) : 没有额外的装饰，笔划粗细大致差不多。

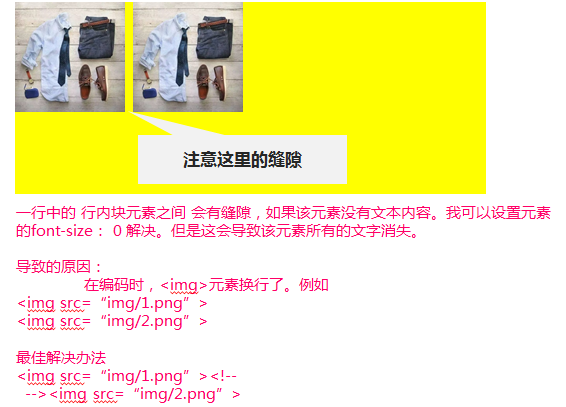


## font-size

* 设置字体大小
* 默认值：medium。一般浏览器默认是16px
* smaller 把 font-size 设置为比父元素更小的尺寸。
* larger 把 font-size 设置为比父元素更大的尺寸。
* *length* 把 font-size 设置为一个固定的值。
* *%* 把 font-size 设置为基于父元素的一个百分比值。
* inherit 规定应该从父元素继承字体尺寸。

注意：font-size 会影响 line-height 属性，进而影响一行中基线的位置。

**font-size使用技巧**



## font-style

* 字体样式
* normal 默认值。浏览器显示一个标准的字体样式。
* italic 浏览器会显示一个斜体的字体样式。
* oblique 浏览器会显示一个倾斜的字体样式。
* inherit 规定应该从父元素继承字体样式。

italic和oblique都是向右倾斜的文字，区别在于italic是指斜体字，而oblique是倾斜的文字，对于没有斜体的字体应该使用Oblique属性值来实现倾斜的文字效果.

## font-variant

* 小型大写字母字体
* normal 默认值。浏览器会显示一个标准的字体。
* small-caps 浏览器会显示小型大写字母的字体。
* inherit 规定应该从父元素继承 font-variant 属性的值。

小写字母转大写字母，转换后字母大小不变

## font-weight

* 字体的粗细
* normal 默认值。定义标准的字符。
* bold 定义粗体字符。
* bolder 定义更粗的字符。
* lighter 定义更细的字符。
* 100 ~900 定义由粗到细的字符。400 等同于 normal，而 700 等同于 bold。
* inherit 规定应该从父元素继承字体的粗细。

**实际开发中，我们推荐使用关键字方式。因为700 和 900的粗细根本看不出区别**

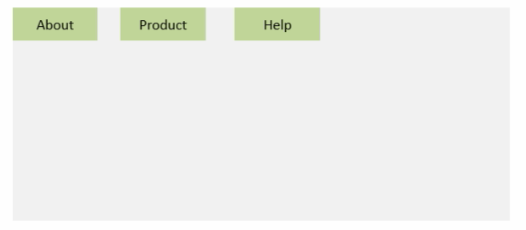
## font 简写

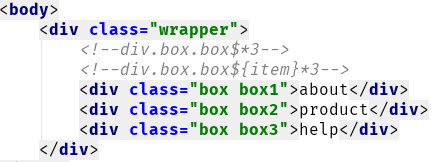
* 常见写法 font：字体样式 小型大写字母 字体粗细 字体大小/行高 字体族(多个)



# 扩展练习

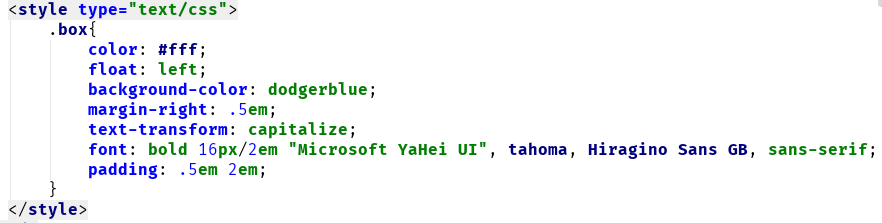
效果图：







另一种写法



但这种写法需要三个div中的文字宽度大致相同，这个练习中三个文字about、product、help宽度有差异，在表现上背景宽度就不一致



但如果宽度大致一致，如数量一样的中文或英文可以使用这个，而且更加简便

# vertical-align

* 垂直对齐 vertical-align 是指在一行内，行内元素在垂直方向的对齐位置。

以下所有关于垂直的对齐，参照物都是行内的文本，我们通过img演示各个值的差异



默认行内元素在垂直方向都是基于元素的“基线”对齐，如同设置了

img{ vertical-align: baseline; }

**基线位于一行非衬线字体小写字母的底部**

注意：行高会影响基线的位置，而行高默认是基于字体大小计算（line-height 默认 为字体大小的 1.2 倍，不同浏览器可能计算有差异），因此字体大小也会影响基线的位置。

img { vertical-align: top; }

位于父元素**一行**的顶部

**此时我们需要设置父元素的高度大于图片高度，下图利用文字高度撑起父元素高度，才能看到效果，或者设置足够高的行高也可以**



img{ vertical-align: bottom; }

位于父元素**一行**的底部

**此时我们需要为父元素设置“足够”的高度，才能看到效果**



## img{ vertical-align: middle; }

**位于同一行的非衬线字体小写字母的 1/2 处，字母是直接写在父元素中的，不是元素中的，如父元素中有一个<span>hello</span>，不会和 hello对齐**

图片的中线的基线就是同一行的非衬线字体小写字母的 1/2 处



左图为没有设置vertical-align，此时图片撑起父元素高度，元素基于基线对齐，由于基线与父元素底边有距离，所以图片和父元素底边有距离

当给图片设置vertical-align后，基线不再对图片产生影响，图片的中线的基线就是同一行的非衬线字体小写字母的 1/2 处，此时如果不给父元素设置高度，同时图片又是一行中最高的元素，那么图片撑起父元素高度，同时图片又基于文字1/2处对齐，视觉上造成图片居中



而当父元素有足够的高度，图片就不居中了，下图利用文字字体大小撑起父元素高度，图片还是基于文字1/2处对齐，但不居中

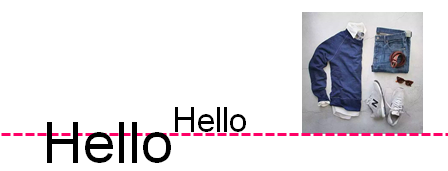


**vertical-align说白了就是相当于一行中小写字母1/2对齐（这是根本），如果需要图片居中，必须保证图片是一行中最高的元素，而且父元素不能设置高度，必须使图片撑起父元素高度**

img{ vertical-align: text-top; }

位于同一行的sup元素文本的基线对齐

**为父元素设置足够的高度，在图片同一行中，添加 <sup>Hello</sup>**



img{ vertical-align: text-bottom; }

位于同一行的sub元素文本的基线对齐

**为父元素设置足够的高度，在同一行中，添加 <sub>Hello</sub>**



## 垂直对齐扩展

* 默认vertical-align只能设置在inline或者 inline-block上，不能设置在父元素。
* **垂直对齐是指在一行内的垂直方向对齐！**
* 以下两种情况可以把vertical-align设置在父元素
  1. <td> 元素作为父元素
  2. 一个元素被显示的设置 display: table-cell

vertical-align 设置在 td

* 当td元素显示设置了
  + vertical-align : top
* 该元素所有的内容（文本及其子元素）都会该元素顶部
* **注意：请为td元素设置足够的高度，宽度**
* 当td元素显示设置了
  + vertical-align : middle (这是默认值，无需显示设置)
* 该元素所有的内容（文本及其子元素）都会该元素中间
* **注意：请为td元素设置足够的高度，宽度**
* 当td元素显示设置了
  + vertical-align : bottom
* 该元素所有的内容（文本及其子元素）都会该元素底部
* **注意：请为td元素设置足够的高度，宽度**

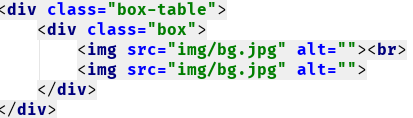
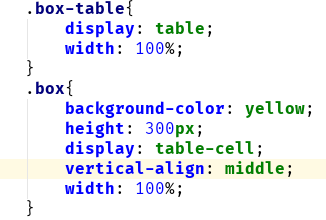
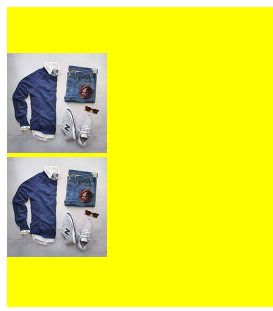
vertical-align 设置在 table-cell

display: table-cell == td

如果需要在父元素中设置内容的垂直方向居中，必须为该父元素设置display: table-cell;

当元素设置display: table-cell; 元素的默认宽度（元素不设置宽度，默认是父元素100%）会清零，需要为该元素添加父元素，并设置其父元素宽度为100%，或者指定宽度，如果先前就指定宽度，则宽度不会清零

可以把元素的所有内容（文本，子元素-inline, inline-block）都在垂直方向居中，多行也可以

如上代码，给img父元素设置了vertical-align，就需要设置display：table-cell，而给.box设置display：table-cell后，它的默认宽度会清零，需要给它加一个父元素.box-table，还需要设置父元素宽度为100%

## 使用场景

vertical-align 使用场景1： 一行内的垂直方向对齐

设置在行内元素中。如果设置 vertical-align : middle ，该行内元素在同一行中文本小写字母的1/2。

注意：不是同一行中元素内容的小写字母。

vertical-align: 只用设置行内元素（inline inline-block）

vertical-align 使用场景2： 多行内的垂直方向对齐

设置其父元素 display: table-cell,设置固定的高度，然后设置 vertical-align

当元素设置为 display: table-cell 宽度清零，需要为其指定父元素，父元素（display: table）为其指定宽度

多行垂直居中是子元素中的内容在父元素中垂直居中，但各个子元素中的内容并没有在子元素中垂直居中，如果需要在子元素中也垂直居中，还需要给子元素设置vertical-align

在局部布局中使用 display: table-cell; 以减少对整体布局的影响

# 背景

## 背景颜色：background-color: #f0f0f0;

## 背景图片：background-image: url(img/bg.jpg);

设置平铺：X轴平铺 background-repeat: repeat-x;

设置平铺：Y轴平铺 background-repeat: repeat-Y;

不平铺：background-repeat: no-repeat;

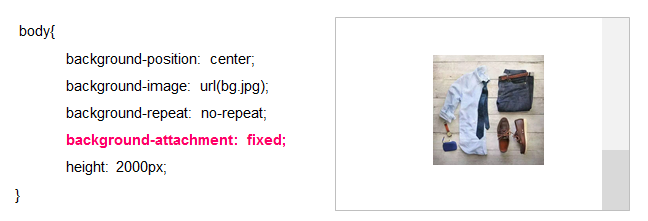
移动：background-position: 50px 50px;

background-position: top/bottom/center;

background-position:50% 50%;

background-attachment

* + scroll 默认值。背景图像会随着页面其余部分的滚动而移动。
  + fixed 当页面的其余部分滚动时，背景图像不会移动。
  + inherit 规定应该从父元素继承 background-attachment 属性的设置。



使用场景：网页的整体背景，注意必须设置在body元素上

## 雪碧图（精灵图）

为什么使用雪碧图（精灵图）：减少网络请求次数，提高网站的网页加载速度。

## 背景图vs图片：

背景图：网页加载后，或者与网页一同加载，不会影响页面加载速度

使用场景: 非重要的页面图片。例如小图片

图片：使用img元素，图片与网页同步加载，如果图片过大，影响页面加载速度，图片中的属性alt可以被搜索引擎检索到

使用场景: 重要的页面图片 网站的LOGO 商品图片

## 背景简写

background: 颜色 背景图 是否平铺 是否随页面滚动 背景图位置

* background: #00FF00 url(bgimage.gif) no-repeat fixed top;