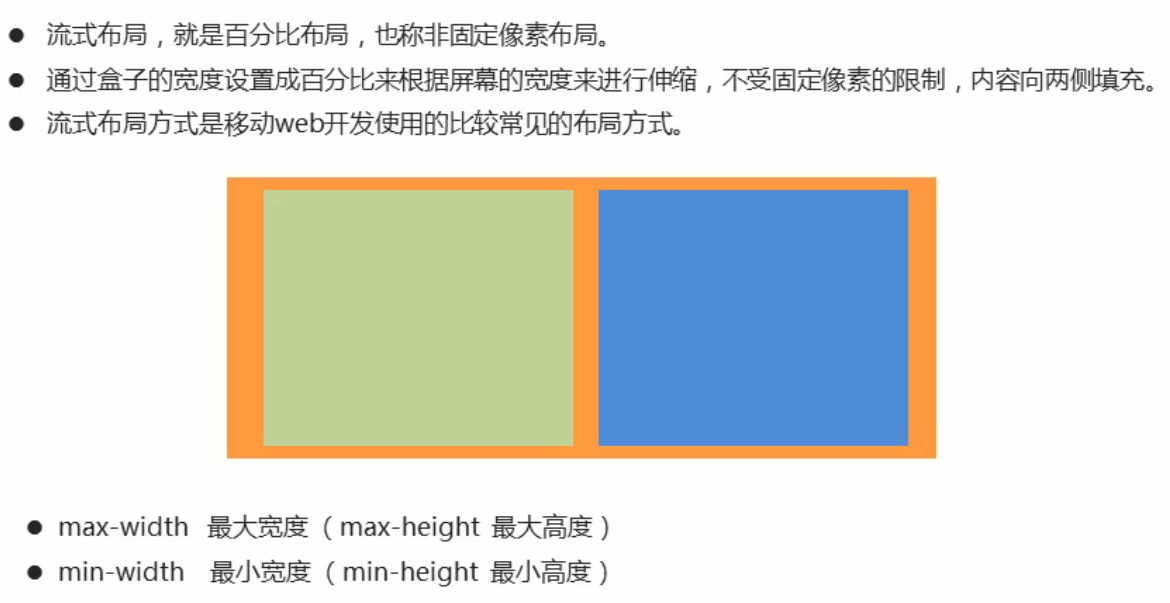
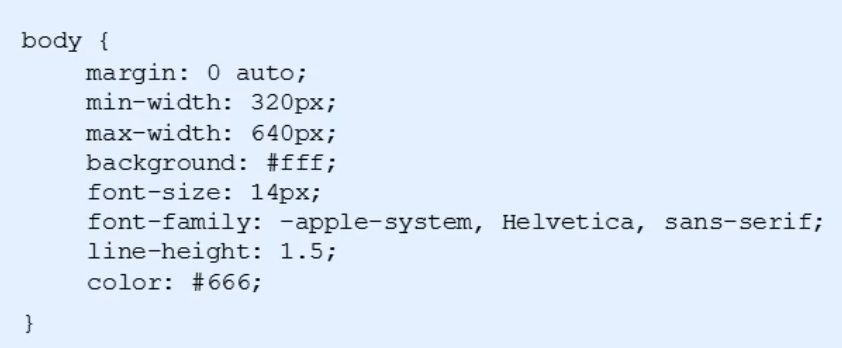
### 流式布局（百分比布局）



#### 设计步骤：

###### 1.常用初始化样式：

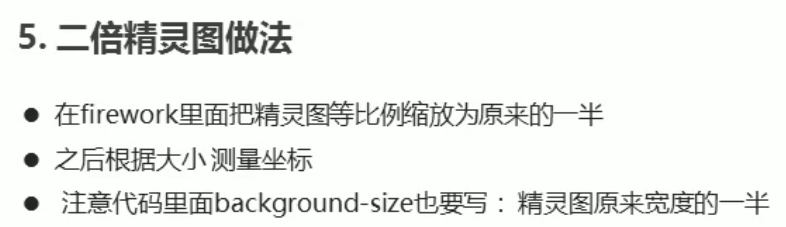


###### 2.APP布局

百分比布局针对的是宽度，通常设置一个宽度100%的大盒子，然后里面放x%+y%+z%=100%的小盒子，然后多个大盒子构成一个页面。设计过程要注意精确设置宽度与百分比宽度的结合，比如如果不希望某个盒子因为页面变化而变化这时候就需要设置精确宽度。然后如果使用了精确宽度也要注意使用定位，防止因为页面的变化导致盒子错位。

注意：如果使用百分比宽度最好不要使用绝对定位，只有设置精确宽度才能使用。

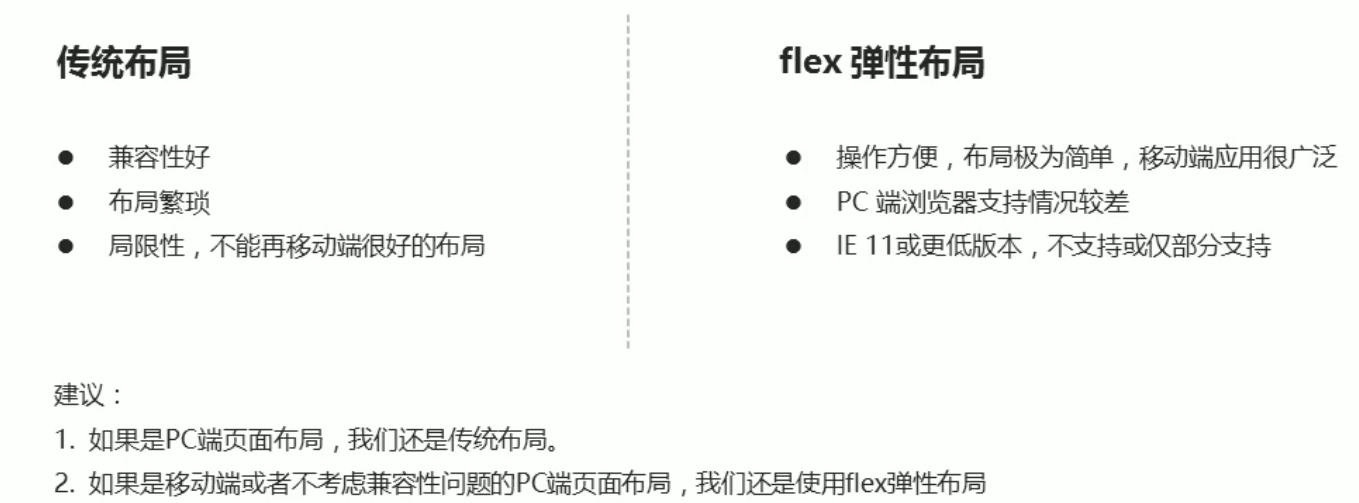
二倍精灵图的取图：



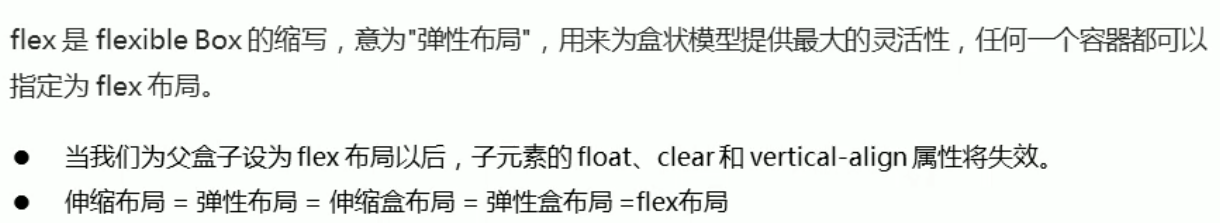
### Flex布局

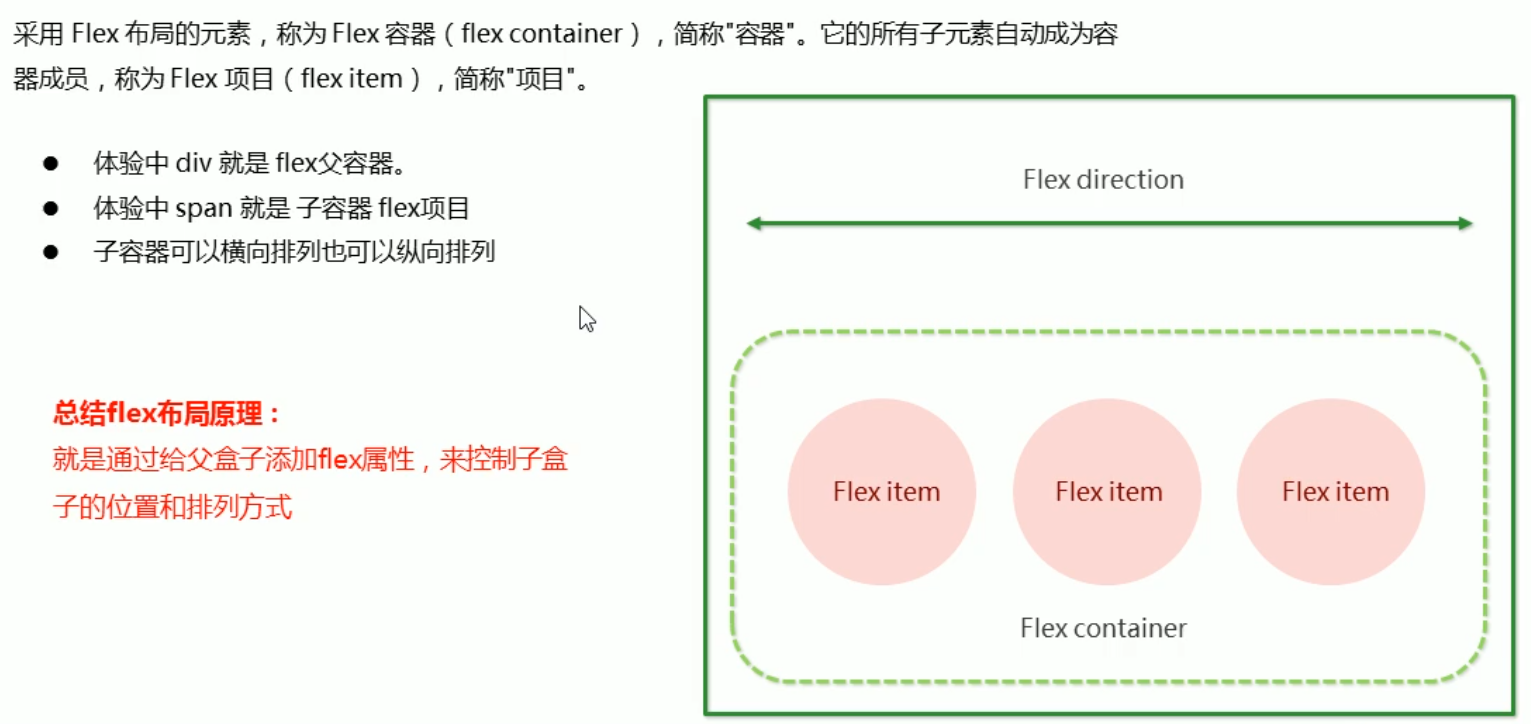
父元素添加display: flex;

#### 传统布局与flex布局

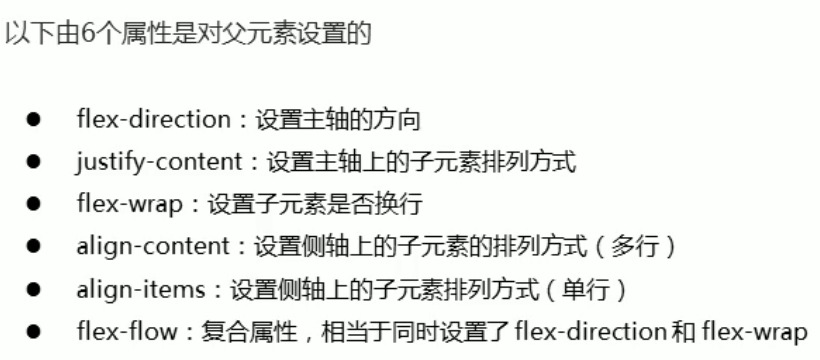


#### Flex布局原理

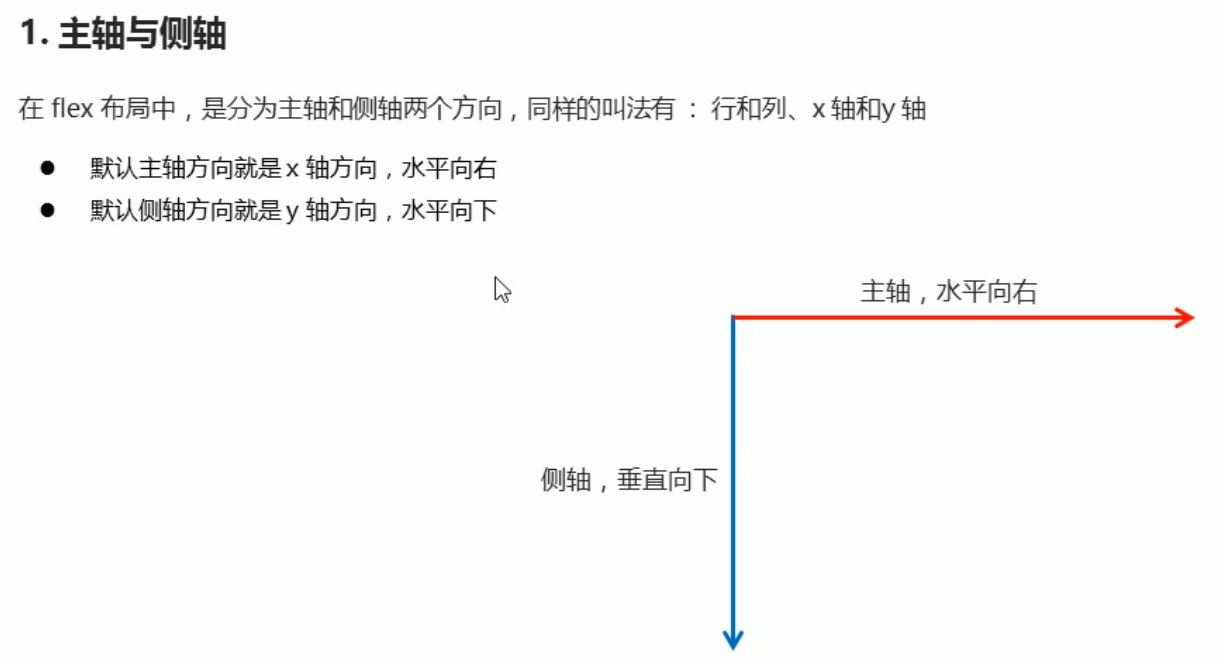




#### Flex布局父项常见属性



###### 设置主轴方向flex-direction

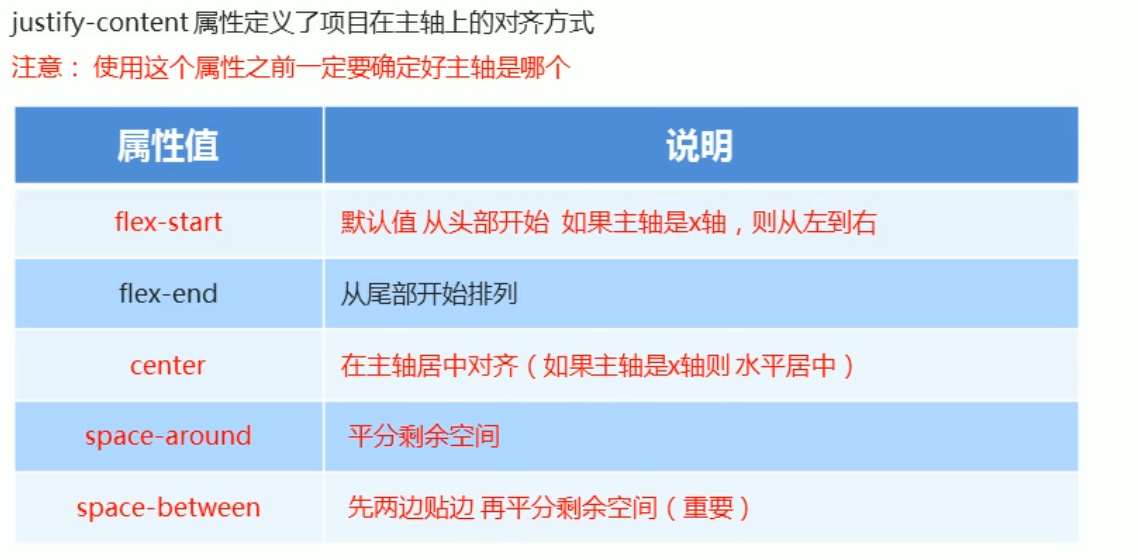


主轴与侧轴是可以变换的：



元素即flex-item是根据主轴排列的。

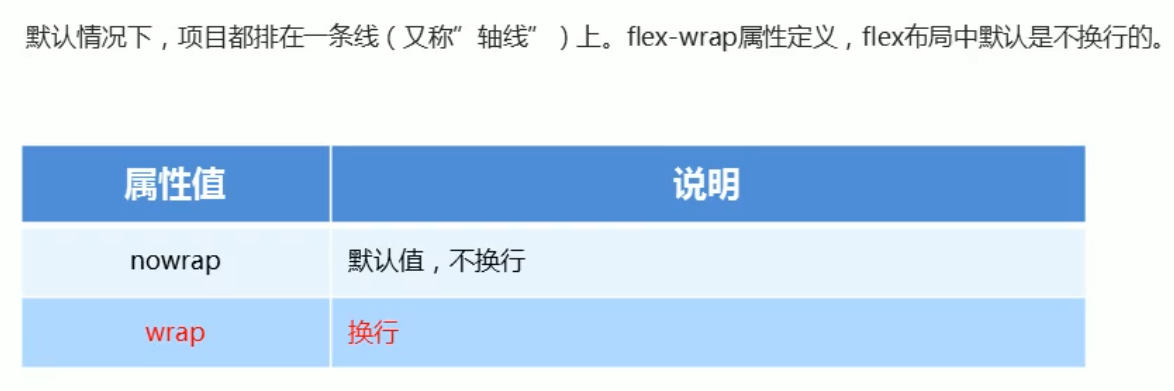
###### 设置主轴上子元素的排列方式justify-content（重点）



justify-content: flex-end需要与flex-direction: row-reverse区别开，前者是贴着尾部对齐，排列顺序不变，即主轴方向是没变的；而后者改变了主轴方向，虽然元素也是贴着尾部对其，但元素排列顺序变了。

space-around是擅自给盒子加了外边距，使盒子与外边距构成一个大盒子，再将这几个大盒子平均分配空间，这样就会有第一个和第二个盒子不贴边的情况；space-between是先使第一个和最后一个盒子贴边，然后再分配剩余空间。

###### 设置子元素是否换行flex-wrap（重要）



###### 设置（单行）侧轴子元素排列方式align-items（重要）



设置单行元素垂直居中（主轴为x轴）：justify-content: center;align-items: center;

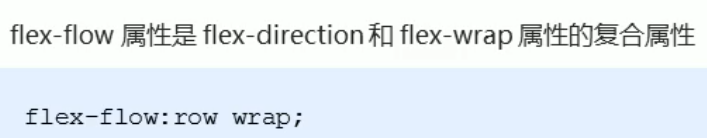
###### 设置（换行/多行）侧轴子元素排列方式align-content



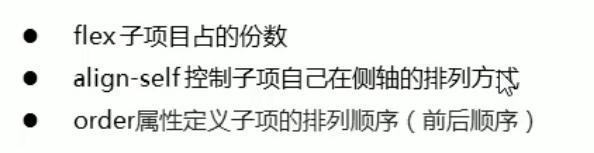


一定要非常注意：align-content适用于换行条件，如上图第二个情况不算是多行。

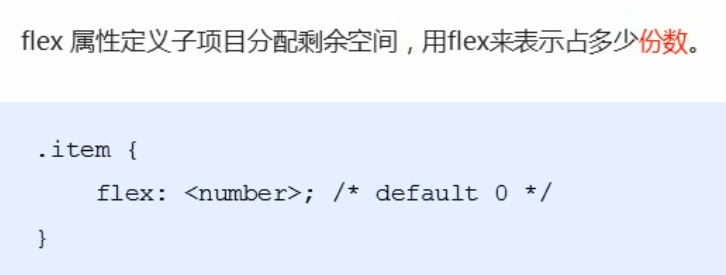
###### flex-flow



### Flex布局子项常见属性



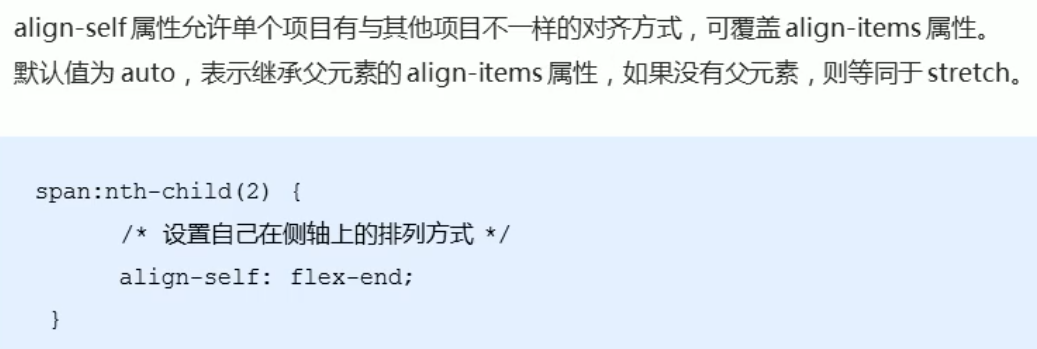
###### 设置份数flex属性（重要）



假设京东的搜索栏，左右侧固定，中间可以直接添加flex: 1；分配剩余空间，然后页面变化中间也能自适应变化。

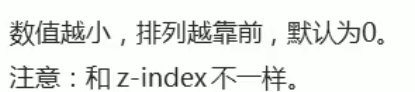
如果多个孩子都写了flex: 1；那么剩余部分一共分成了写了该语句的孩子数，然后每个孩子平均分配剩余部分。

###### 控制子项自己在侧轴上的排列方式align-self



用于对单独的子项进行特殊变换。

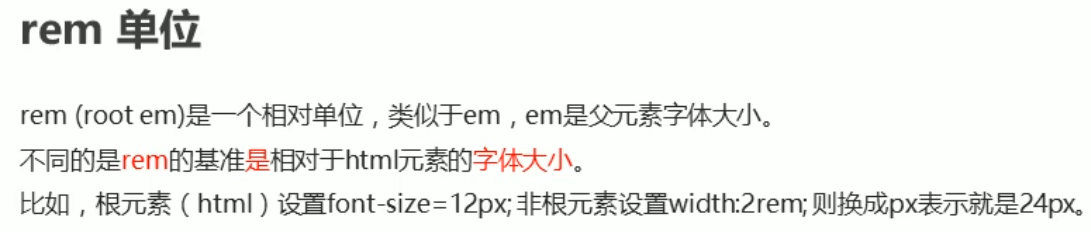
###### 定义项目排列顺序order



### Rem适配布局

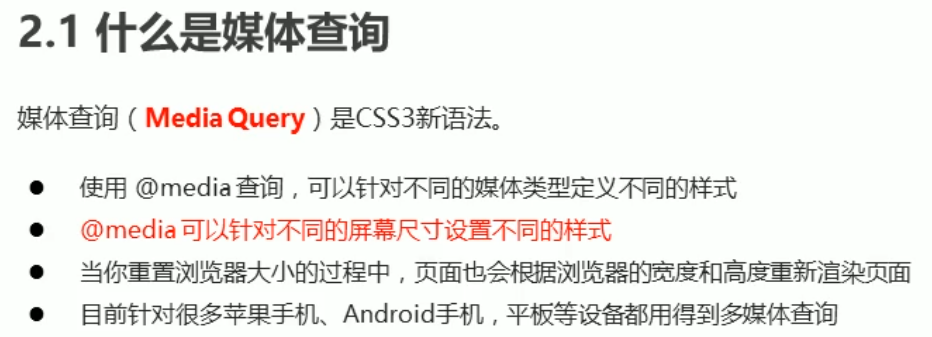
以前的百分比布局和flex布局都是针对宽度自适应来设计的，高度是固定死的。而rem能实现对高度的自适应。

#### Rem基础



#### 媒体查询

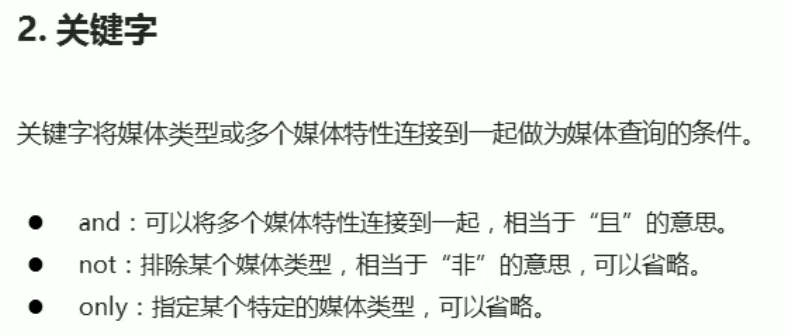
###### 什么是媒体查询

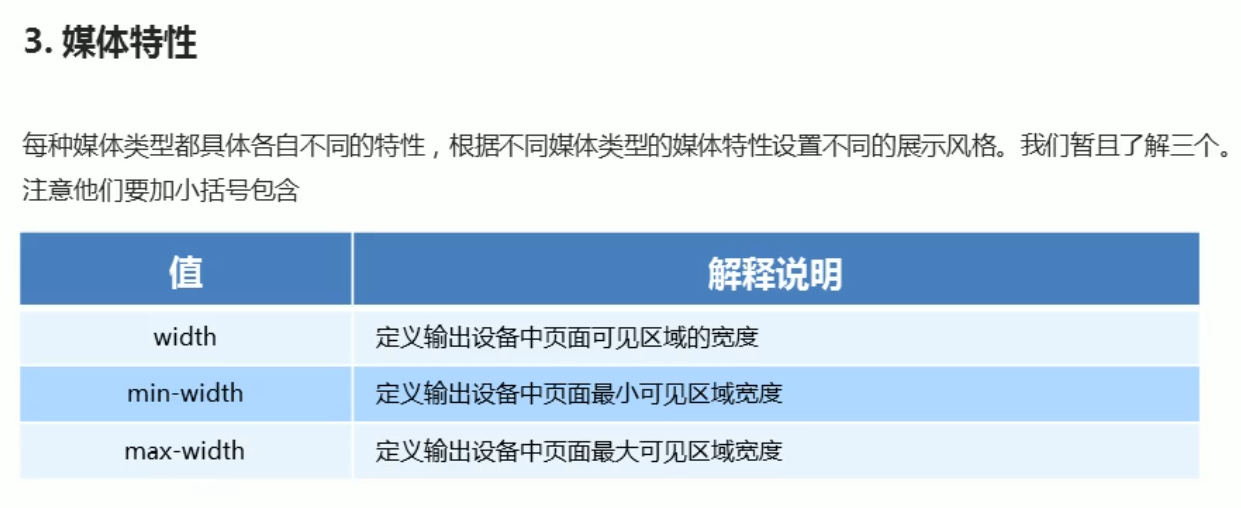


###### 语法

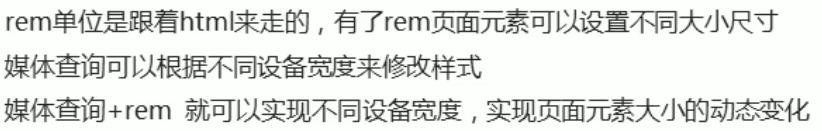




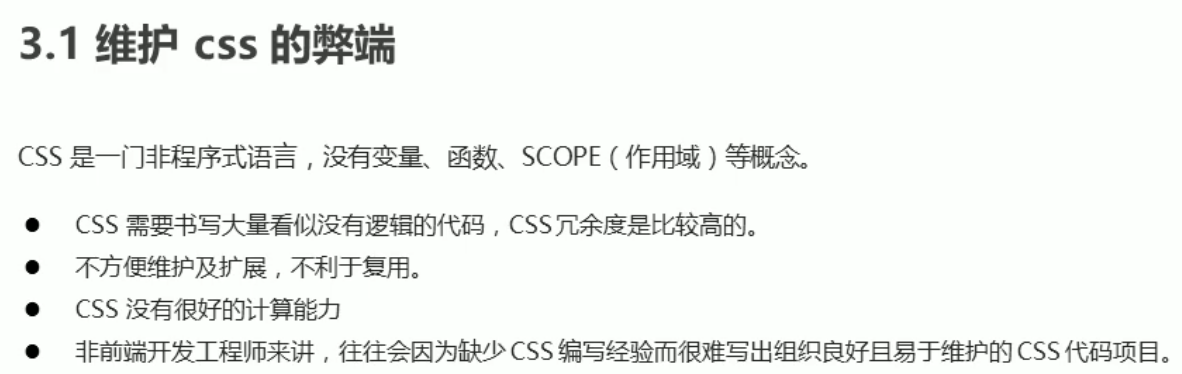




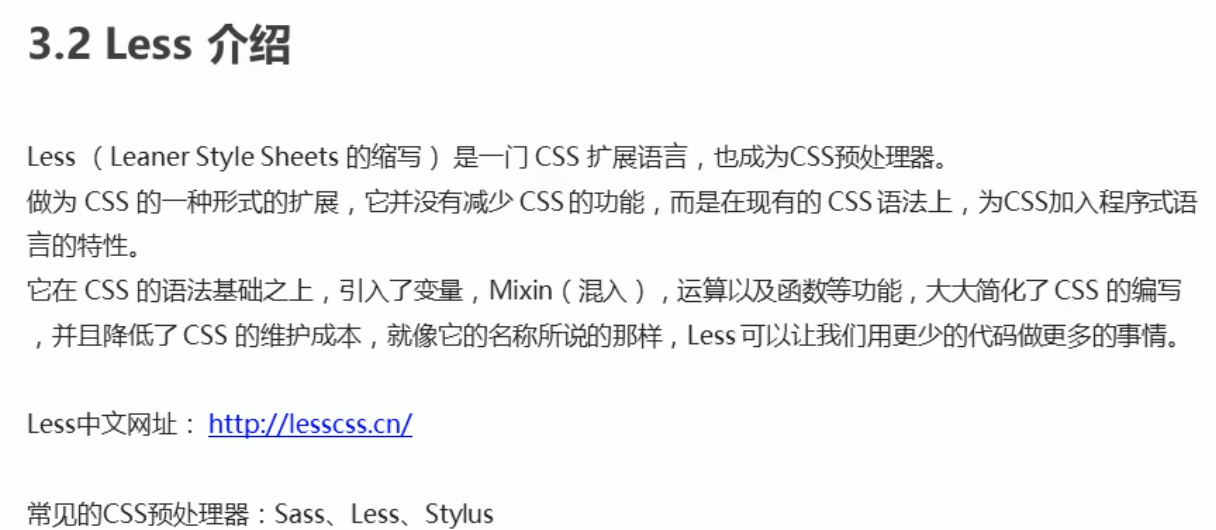
#### 媒体查询+rem实现元素动态大小适配



#### Less基础

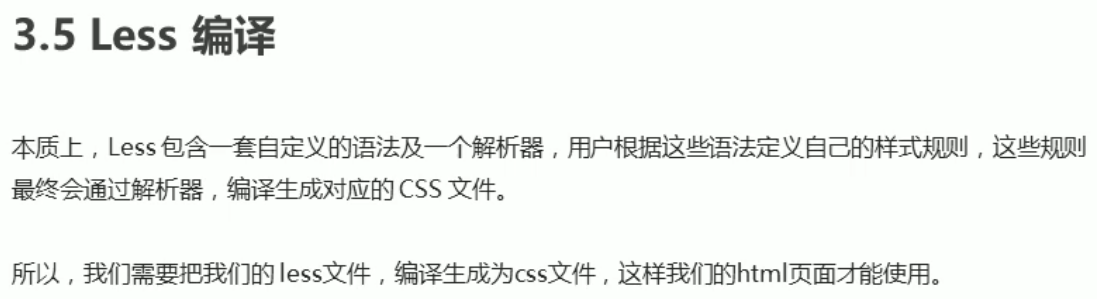


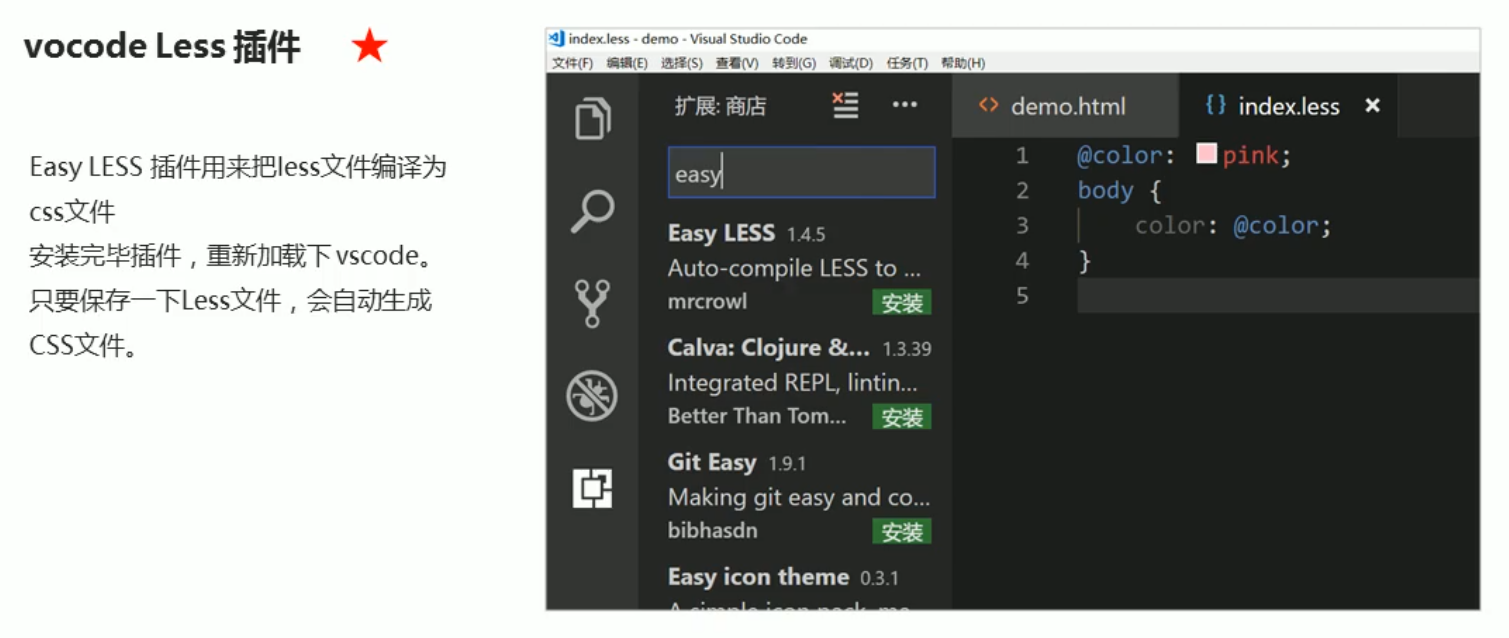
而less能解决以上问题。



Less语法：@变量名: 值;

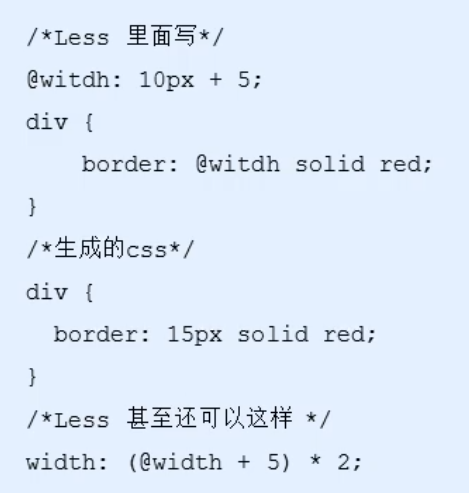
然而less并不能直接引入html，需要先编译成css文件才能引入。

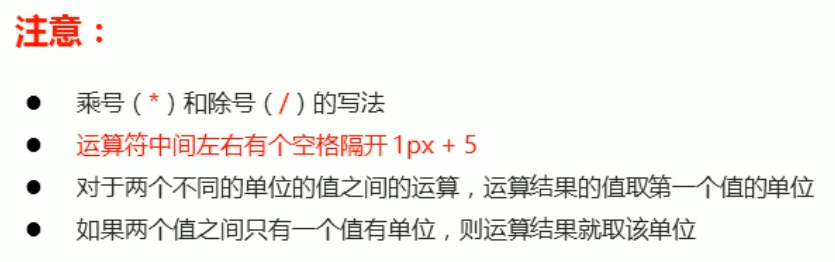




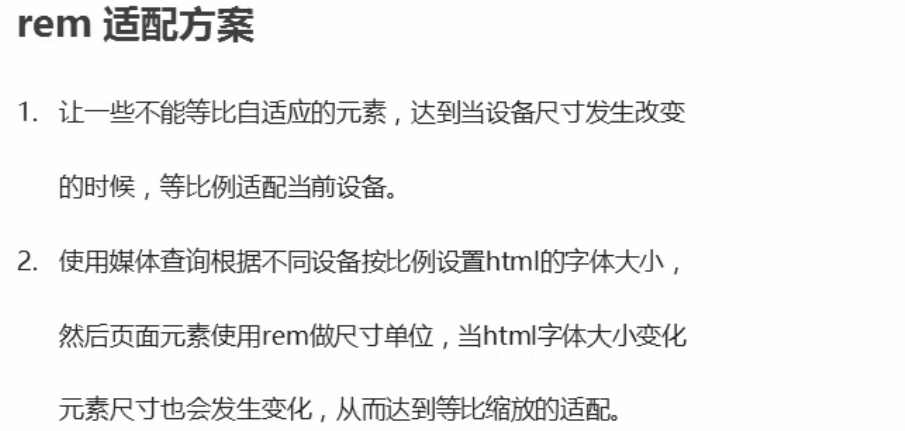
###### Less运算







#### Rem的适配方案



有两种方案：  
less+media媒体查询+rem

Flexible.js+rem

###### less+media媒体查询+rem



###### Flexible.js+rem

由于第一种适配方案设置媒体查询比较繁琐，因此开发出了第二种方案。

第一种设置媒体查询时，页面大小的变化是离散的，而利用flexible.js的话变化就是连续的。也就是说使用第一种方法，页面的font-size只有页面大小到某个节点才会跳跃式变化，而第二种是实时与页面保持一致的。





可以实现输入px自动换算成rem

该插件使用时必须手动修改html字体大小：

进入settings.js;

搜索cssrem.rootFontSize

(该插件第一种适配方法也适用)