1. flex+rem知识引入(https://juejin.im/post/58e3a5a0a0bb9f0069fc16bb)

总结，flex不支持的有:

·Flexbox不支持 ::first-line 和 ::first-letter 这两种伪元素

·vertical-align对Flexbox中的子元素是没有效果的

·float 和 clear 属性对Flexbox中的子元素是没有效果的，也不会使子元素脱离文档流

·多栏布局（column-\*）在 Flexbox 中也是失效的，就是说我们不能使用多栏布局在Flexbox 排列其下的子元素（鱼和熊掌不可得兼嘛）

·Flexbox下的子元素不会继承父级容器的宽

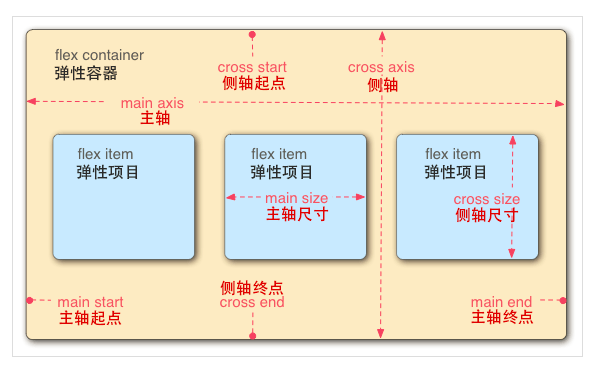
1. 归纳整理(相信看了上面内容，对flex布局有了大体认知。现在就对上面内容再细抠一下)
2. 任何一个元素都可以指定为flex布局。

语法: display:flex||inline-box;

//flex:不会被浮动的元素遮盖，不会垂直外边距坍塌等等。(块级元素)

//inline-box:即 一个被行列化后的 Flexbox。它不会独占一行，但是可以设置宽和高。(行内元素)

加上上面语句，元素会变成伸缩容器，其中子元素会变成伸缩项。



上图是一个横向的，文本流从左到右的弹性布局图示。这里有几个概念

弹性容器(Flex container或伸缩容器，简称容器), 指一个声明为 flex 或者 inline-flex 的元素，是所有 Flex items 的父元素。

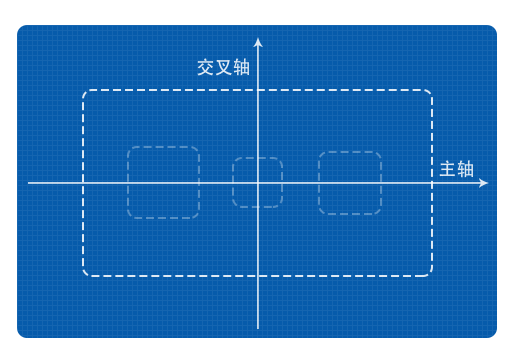
弹性项目(Flex item或伸缩项，简称项目), 指 flex 容器内的直接孩子元素(注：如果父级设置了display:flex,那只作用于父级下的第一级子元素)；弹性容器中的直接文本项默认为匿名弹性项目。

主轴(main axis)和侧轴(cross axis，又名交叉轴), 类似于二维中的 X 轴和 Y 轴，主轴由属性 flex-direction 来定义（注意:主轴不一定是x轴），与主轴垂直的轴则是侧轴。

*说明:主侧轴的切换十分简单，当主轴设定的时候，它的垂直面，就默认被设定成了侧轴。* 主轴的开始位置（与边框的交叉点）叫做main start，结束位置叫做main end；交叉轴的开始位置叫做cross start，结束位置叫做cross end。  
 项目默认沿主轴排列。单个项目占据的主轴空间叫做main size，占据的交叉轴空间叫做cross size。

注:主轴确立项目主体排列方向，侧轴确立如果元素沿一条轴线排列不下时，该如何换行

(来个简图：)



1. 属性(小贴士:可以自己画下主轴方向，便于理解)---详细效果请戳文首链接

※容器：

1. flex-direction//定义排列的方向

* row 默认值，主轴为水平，从左向右排列
* row-reverse 主轴为水平，从右向左排列
* column 主轴为垂直方向，从上向下排列
* column-reverse 主轴为垂直方向，从下向上排列

(2)flex-wrap 定义是否新行显示，以及新行的排列方向(即设置了侧轴方向)

* + - * nowrap 默认值，显示在同一行，不换行
      * wrap 一行显示不了时使用新行显示
      * wrap-reverse 效果是wrap反方向

(3)flex-flow 是 flex-direction 和 flex-wrap 的简写方式

使用方法 flex-flow: <‘flex-direction’> || <‘flex-wrap’>

默认值为： row nowrap

注意:可以只写一个属性值，系统会根据属性值类型来判定是设置了哪一个属性。

(4)justify-content 定义弹性项目在主轴上的对齐方式，类似水平居中(只有在有空隙时，才可以看到效果)

* flex-start 默认值，与主轴起始位置对齐，通常为左对齐
* flex-end 从结束位置对齐，通常为右对齐
* center 居中对齐
* space-between 两端对齐，平均间隔 //子容器沿主轴均匀分布，位于首尾两端的子容器与父容器相切。
* space-around 每个子元素都有相等的外边距，相邻元素外边距不会叠加

//子容器沿主轴均匀分布，位于首尾两端的子容器到父容器的距离是子容器间距的一半。

(5)align-items 定义弹性项目在侧轴上的对齐方式(类似垂直居中)

* + - * stretch 默认值，弹性项目在没指定高度或者高度为 auto 的情况下，则高度会占满空间
      * flex-start 侧轴起点开始对齐
      * flex-end 从侧轴结束位置对齐
      * center 相对侧轴居中对齐
      * baseline 与基线对齐(基线对齐，这里的 baseline 默认是指首行文字，即 first baseline，所有子容器向基线对齐，交叉轴起点到元素基线距离最大的子容器将会与交叉轴起始端相切以确定基线。)

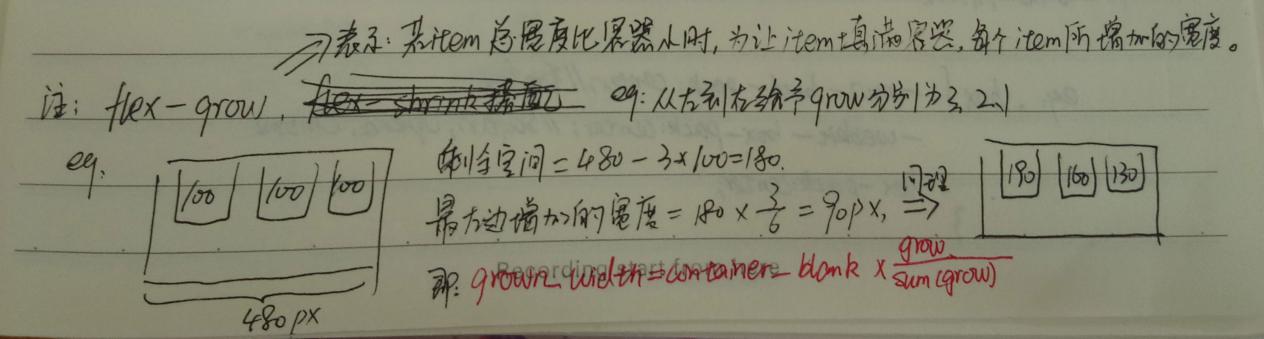
(6)align-content 定义多行在侧轴的对齐方式，当只有一行时，不起作用

* stretch 默认值，各行会伸展以占满整个纵轴空间
* flex-start 与交叉轴起点对齐
* flex-end 与交叉轴起点终齐
* center 相对纵轴中间对齐
* space-between (同上)
* space-around (同上)

※项目：

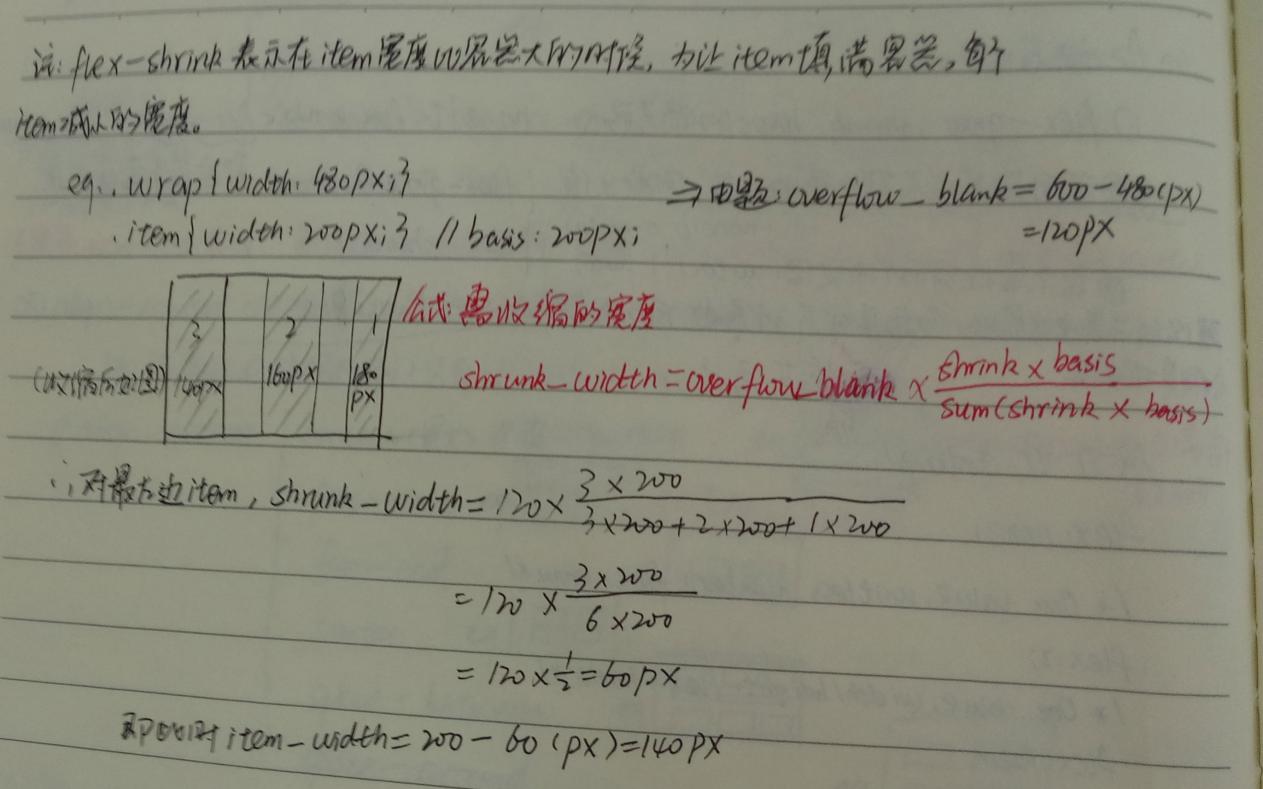
1. flex-grow 控制伸缩项在伸缩行的伸展程度，默认0，即默认不放大

语法:flex-grow：<整数>



1. flex-shrink 数字，定义弹性项目的缩小比例，默认1。值0表示不缩小

语法:flex-shrink：<整数>



注意:如果空间不足，则伸缩项将缩小。(原因:默认1)

如果所有项目的flex-shrink属性都为1，当空间不足时，都将等比例缩小。如果一个项目的flex-shrink属性为0，其他项目都为1，则空间不足时，前者不缩小。

负值对该属性无效。

<http://www.ruanyifeng.com/blog/2015/07/flex-grammar.html>

(3)flex-basis 数字，定义弹性项目的默认尺寸(或说:定义项目伸缩前尺寸)

语法:flex-basis: <length> | auto; /\*默认auto(项目本来的大小)\*/

注意:

1. length为百分比或px。
2. 设置了auto或者shrink的伸缩项，将以这个尺寸为基准进行伸缩。
3. 可以设为跟width或height属性一样的值（比如350px），则项目将占据固定空间。
4. 若缺省宽高，则flex-basis默认为对item主轴方向属性(即宽、高)的设置
5. flex属性是flex-grow, flex-shrink 和 flex-basis的简写，默认值为0 1 auto。后两个属性可选。

语法: flex: none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]

注意:该属性有两个快捷值：auto (1 1 auto) 和 none (0 0 auto)。

建议优先使用这个属性，而不是单独写三个分离的属性，因为浏览器会推算相关值。

(5)align-self 定义此项目在侧轴的对齐方式，会覆盖弹性容器 align-items 定义的对齐方式

* auto 默认为auto,表示继承父元素的align-item属性。若无父元素，则等同于stretch。
* stretch 弹性项目在没指定高度或者高度为 auto 的情况下，则高度会占满空间
* flex-start 侧轴起点开始对齐
* flex-end 从侧轴结束位置对齐
* center 相对侧轴居中对齐
* baseline 与基线对齐

1. order 数字，定义弹性项目的显示顺序，数字越小越靠前(默认全为0)
2. **缩写语法**

请参照https://juejin.im/post/58e3a5a0a0bb9f0069fc16bb最后一个图

1. 新旧对比★

<https://segmentfault.com/a/1190000003978624>([Flex布局新旧混合写法详解（兼容微信）](https://ueibo.cn/flexbu-ju-xin-jiu-hun-he-xie-fa-xiang-jie-jian-rong-wei-xin/" \t "https://segmentfault.com/a/_blank))

简介：新语法+旧语法+兼容写法(主推)

Ps:看完上面博客，你会神奇得发现，以上所讲全是新版语法，而老师所讲应该属于旧语法。所以，关于旧语法中的少许细节，请参照课堂ppt CSS3-1弹性盒模型

1. 常见布局

<http://www.ruanyifeng.com/blog/2015/07/flex-examples.html>(Flex 布局教程：实例篇)

备注：文中链接推荐用chrome打开。(时间有限，准备仓促，请见谅)