# 概念 & 模型

<div class=”box”>

<div></div>

<div></div>

<div></div>

</div>

大的div（开启了flex布局的元素）称为”flex container”；

里面的直接子元素div称为”flex items”。

只需要：

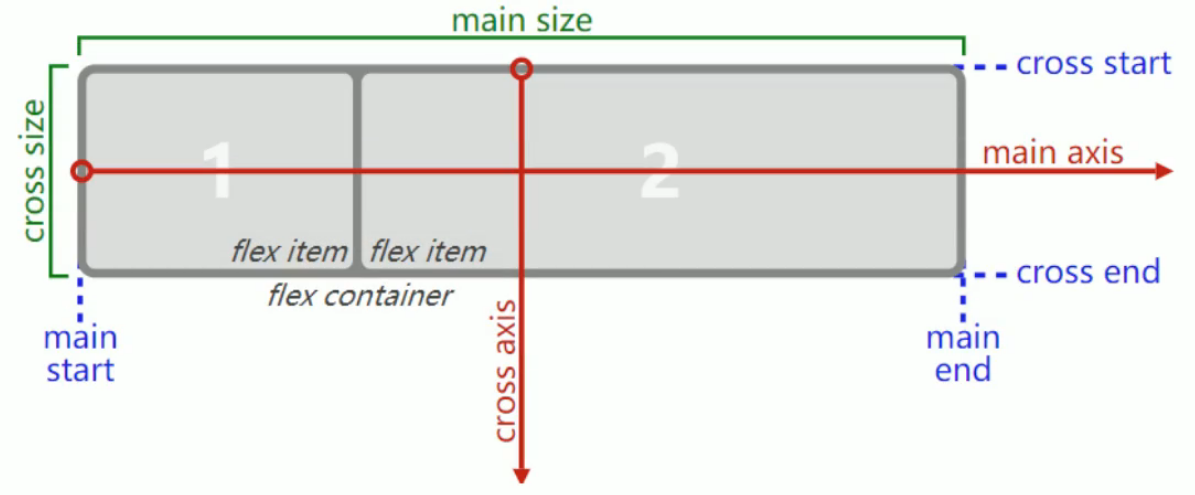
.box{

display = flex; //box为块级元素

// display = inline-flex; //box为行内元素

}

即可开启flex布局。



轴的默认情况如上图所示。

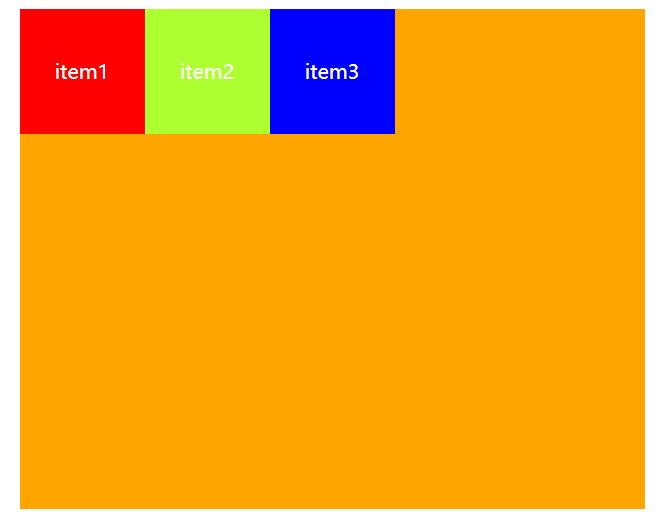
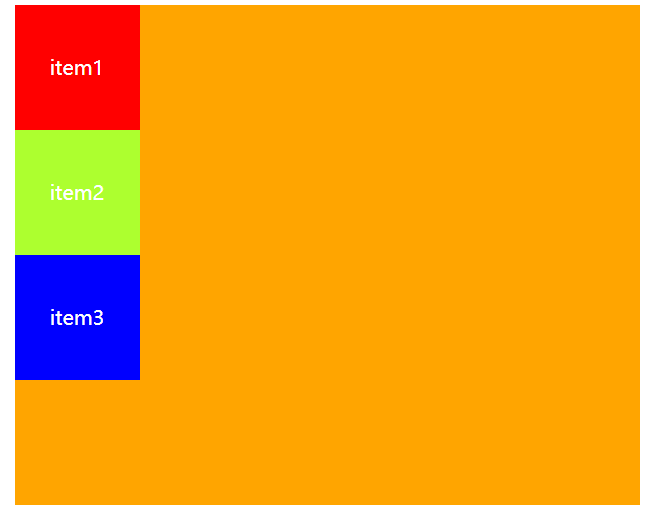
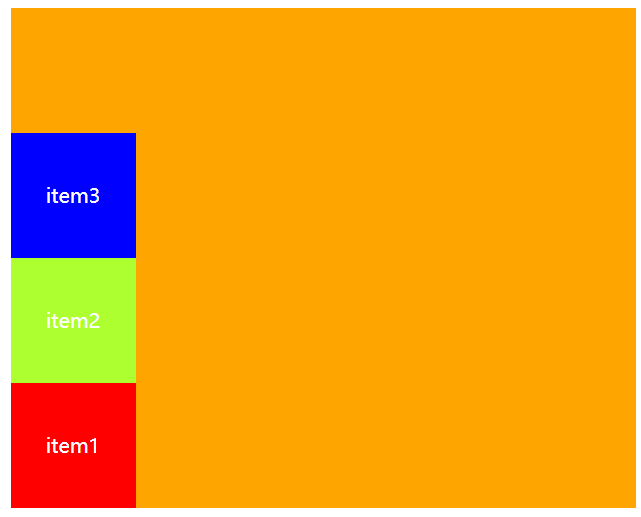
# flex container属性

## flex-direction

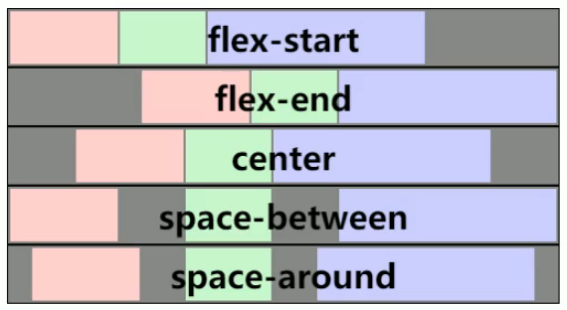
flex布局中不再有分块级元素、行内元素等。

flex items默认都是沿着main axis (主轴)从main start开始往main end方向排布。

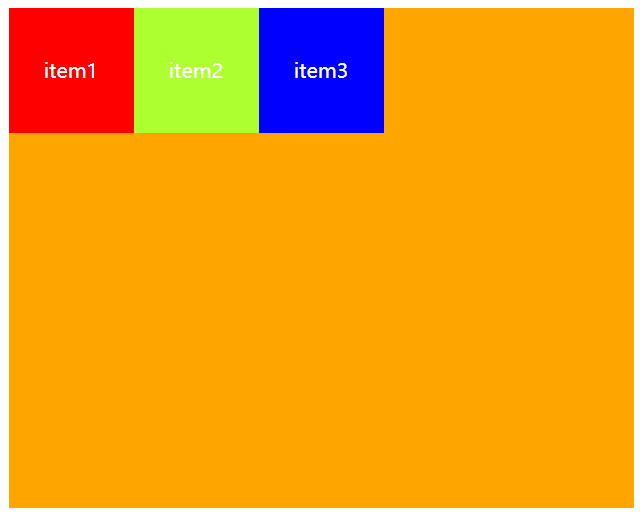
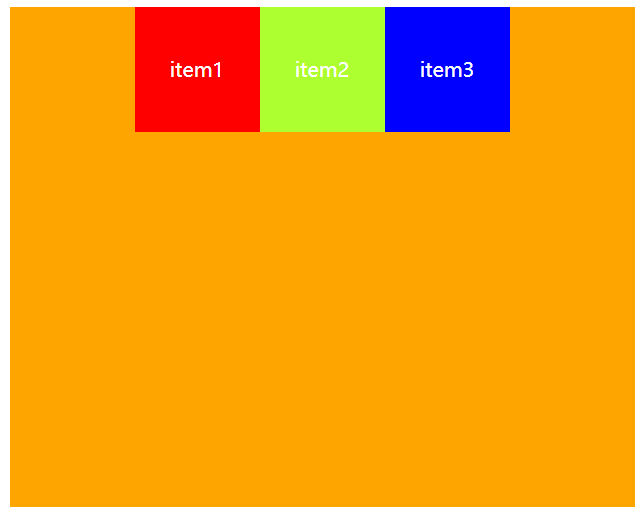
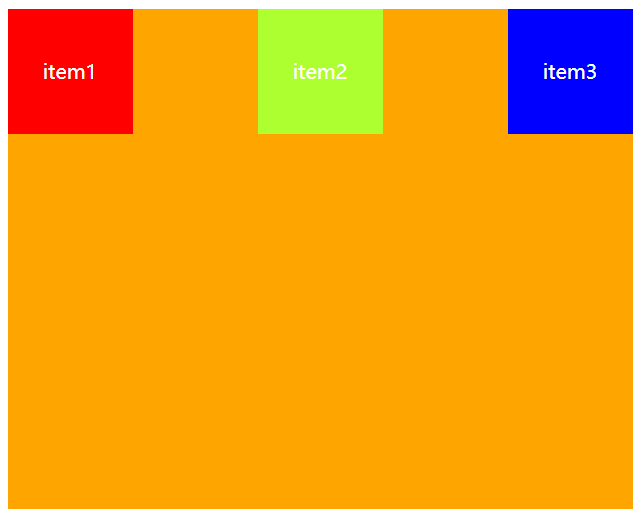
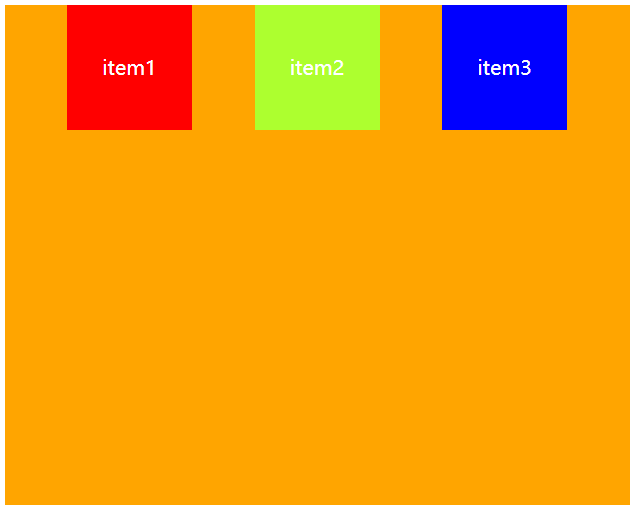
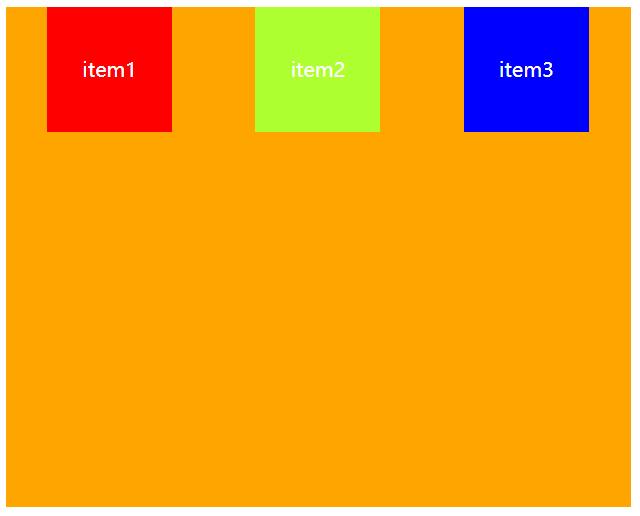
flex-direction用来决定主轴的方向：

* 默认值为row，从左向右
* row-reverse，从右到左
* column，从上到下
* column-reverse，从下到上

## justify-content



以下图片中轴为默认情况。

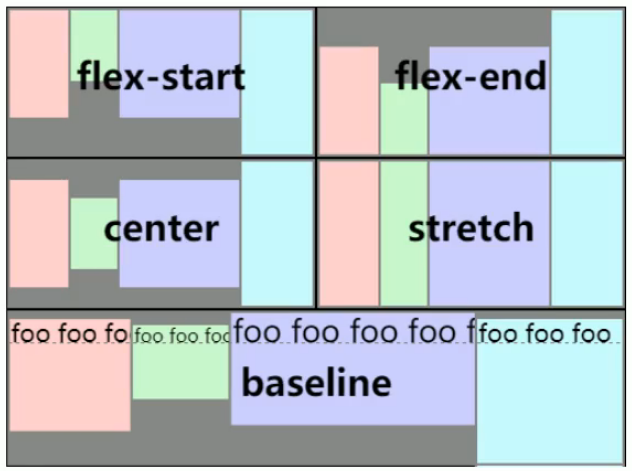
* flex-start，默认值，与main start对齐
* flex-end，与main end对齐
* center，居中
* space-between，flex items之间的距离相等，与main start、main end两端对齐
* space-evenly，flex items之间的距离相等，flex items与main start、main end之间的距离 等于 flex items之间的距离
* space-around，space-evenly，flex items之间的距离相等，flex items与main start、main end之间的距离 是 flex items之间的一半

## align-items

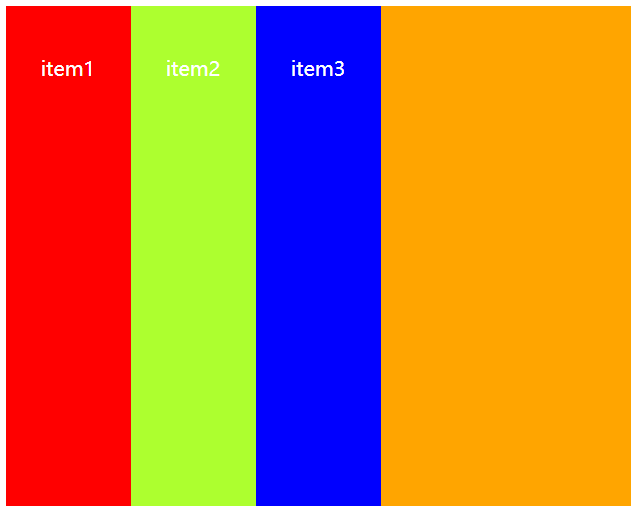
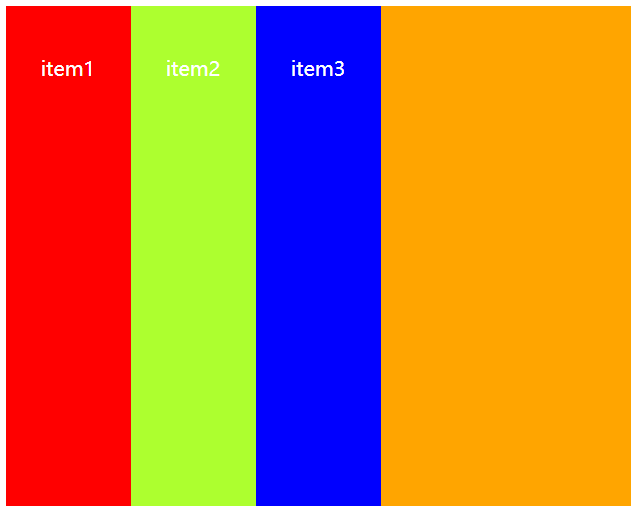
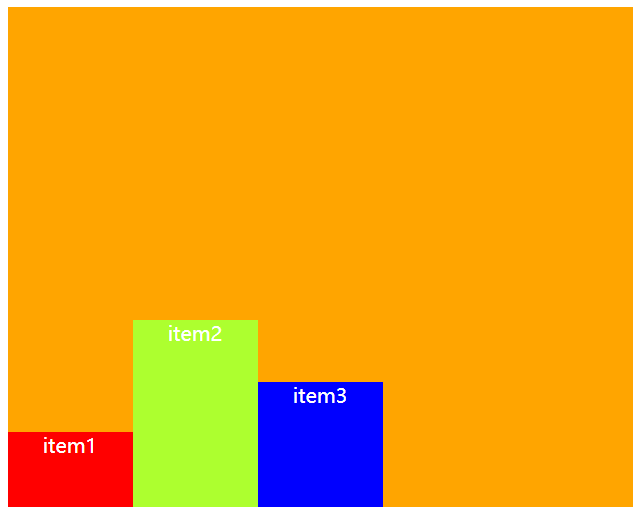
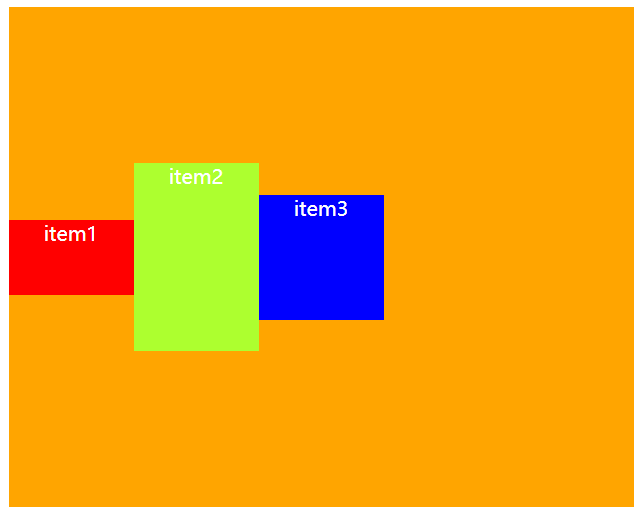
在之前，元素如果没有设置高度，则元素的高度由内容撑开，

在flex布局中，如果items没有设置高度，则会在cross axis上拉伸至cross size（填充flex container）

align-items决定了flex items在cross axis上的对齐方式。



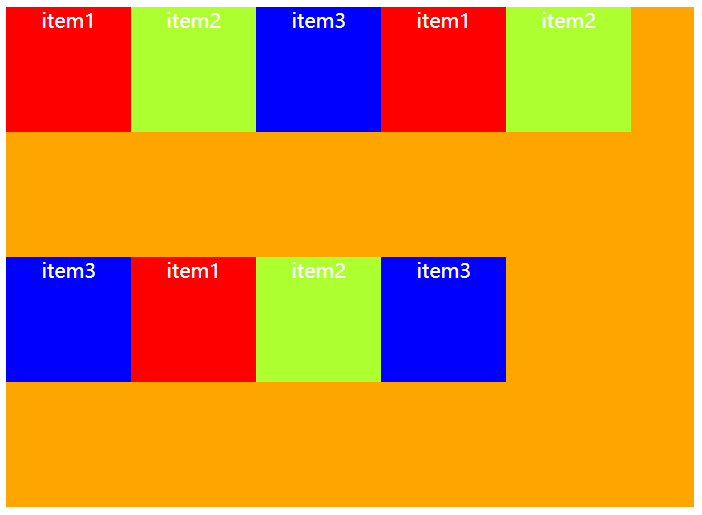
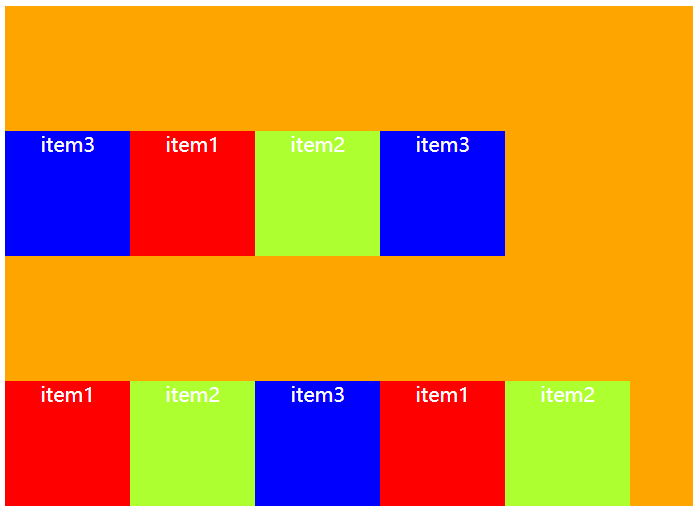
以下图片中轴为默认情况。

* normal，默认值，前提是没有设置高度，否则为指定的高度，在弹性布局中，效果和stretch一样
* stretch，当flex items在cross axis方向的size为auto时，会自动拉伸至填充flex container
* flex-start，与cross start对齐
* flex-end，与cross end对齐
* center，居中对齐
* baseline，与(文字的)第一行基准线对齐

## flex-wrap

flex-wrap决定了flex container是单行还是多行。

默认情况下，如果items超出container的宽度，flex布局会对items进行压缩后放在同一行显示

* nowrap，默认值，不换行
* wrap，换行
* wrap-reverse，换行，交换cross start与cross end

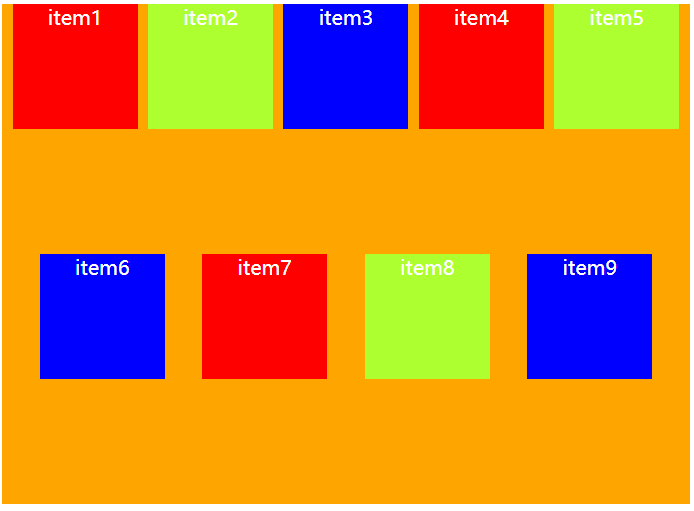
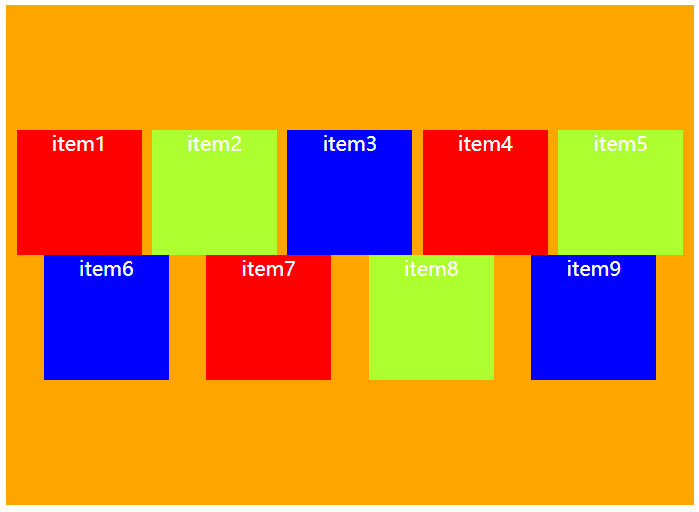
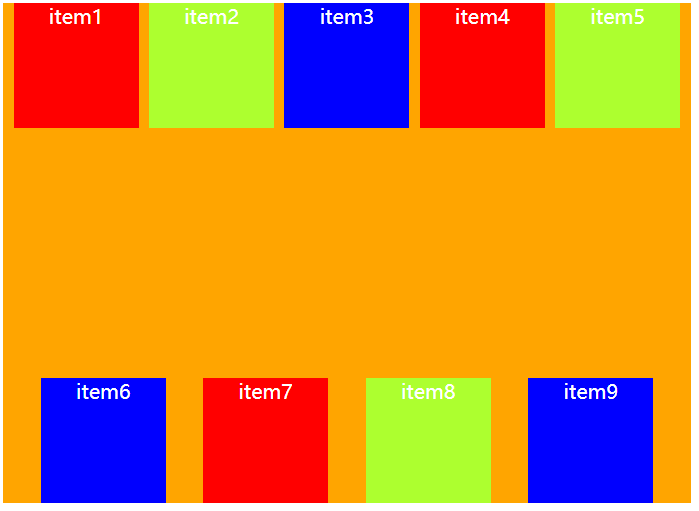
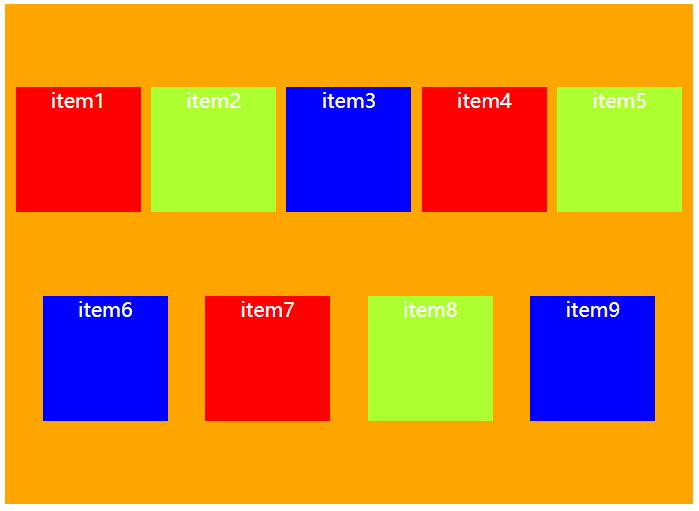
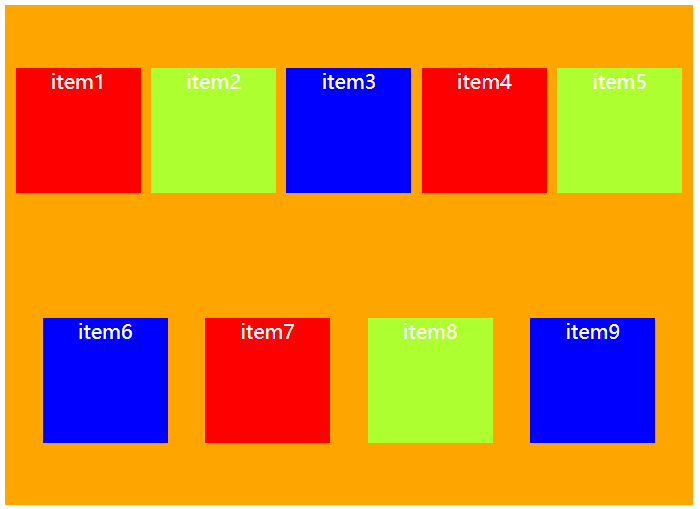
## flex-flow

flex-flow是flex-direction || flex-wrap的简写，可以省略，顺序任意。

例如：flex-flow: row-reserve wrap;

## align-content

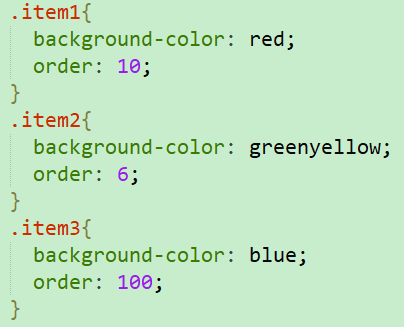
align-content决定多行flex items在cross axis上的对齐方式，用法与justify-content类似。

* stretch，默认值，与align-items的stretch类似，将交叉轴N等分
* flex-start，与cross start对齐
* flex-end，与cross end对齐
* center，居中对齐
* space-between，flex items之间的距离相等，与cross start、cross end两端对齐
* space-evenly，flex items之间的距离相等，flex items与cross start、cross end之间的距离等于flex items之间的距离
* space-around，flex items之间的距离相等，flex items与cross start、cross end之间的距离是flex items之间距离的一半

# flex items属性

## order

可以设置任意整数(正整数、负整数、0)，值越小就越排在前面，默认值是0。

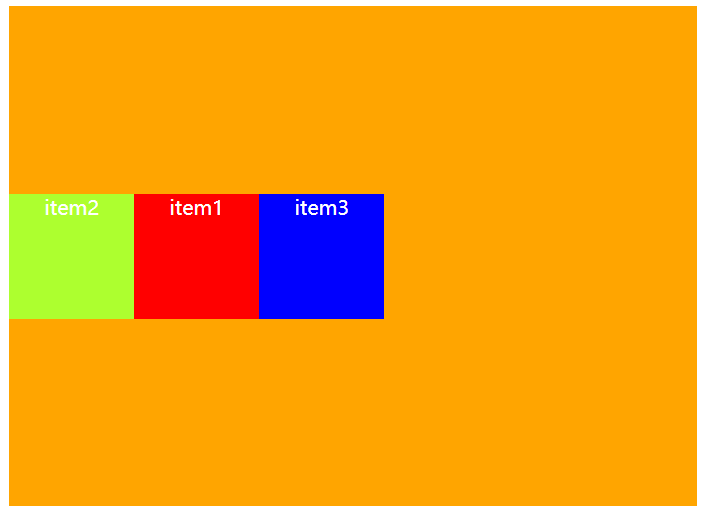
## align-self

flex items可以通过align-self覆盖flex container设置的align-items

* auto，默认值，遵从flex container的align-items设置；
* stretch、flex-start、flex-end、center、baseline，效果跟align-items一致；

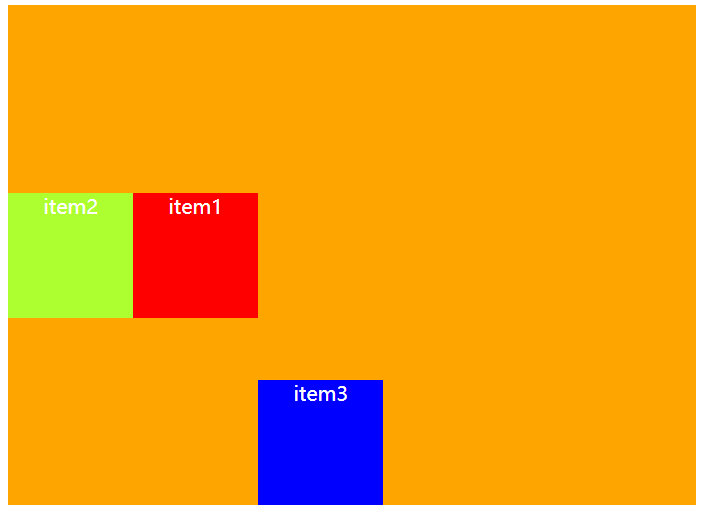
例如，设置align-items: center;

此时，效果如下图所示：



当在item3中设置align-self: flex-end;

效果如下图所示：



## flex-grow

flex-grow决定了flex items如何扩展，可以设置任意非负数字(正小数、正整数、0) ,

默认值是0，当flex container在main axis方向上有剩余size时，flex-grow属性才会有效

如果所有flex items的flex-grow总和超过1 ,每个flex item扩展的size为，flex container的剩余size \* (flex-grow) / sum

如果所有flex items的flex-grow总和不超过1 ,每个flex item扩展的size为，flex container的剩余size \* (flex-grow)

flex items扩展后的最终size不能超过(max-width) \ (max-height)

## flex-shrink

* flex-shrink决定了flex items如何收缩，可以设置任意非负数字(正小数、正整数、0)，默认值是1，当flex items在main axis方向上超过了flex container的size，flex-shrink属性才会有效
* 如果所有flex items的flex-shrink总和超过1，【每个flex item收缩的size】=【flex items超出flex container的size】\*【收缩比例】/【所有flex items的收缩比例之和】
* 如果所有flex items的flex-shrink总和sum不超过1，【每个flex item收缩的size 】=【flex items超出flex container的size】\*【sum】\*【收缩比例】/【所有flex items的收缩比例之和】————【收缩比例】=【flex-shrink】\*【flex item的base size】（base size就是flex item放入flex container之前的size）

## flex-basis

flex-basis用来设置flex items在main axis方向上的base size

* auto，默认值，显示具体的宽度数值

决定flex items最终base size的因素，从优先级高到低分别为：

* max-width \ max-height \ min-width \ min-height
* flex-basis
* width \ height
* 内容本身的size

## flex

flex是flex-grow || flex-shrink || flex-basis的简写，flex属性可以指定1个、2个、3个值。

* 单值，值必须为以下其中之一：
* 一个无单位数(<number>)：它会被当作<flex-grow>的值。例：flex: 1;
* 一个有效的宽度(width)值：它会被当作<flex- basis>的值。例：flex: 100px;
* 关键字none，auto，initial
* 双值：第一个值必须为一个无单位数,并且它会被当作<flex-grow>的值。
* 第二个值必须为以下之一：
* 一个无单位数：它会被当作<flex-shrink>的值。例：flex: 1 2;
* 一个有效的宽度值：它会被当作<flex-basis>的值。例：flex: 1 200px;
* 三值语法:
* 第一个值必须为一个无单位数，并且它会被当作<flex-grow>的值。
* 第二个值必须为一个无单位数，并且它会被当作<flex-shrink>的值。
* 第三个值必须为一个有效的宽度值，并且它会被当作<flex-basis>的值。
* 例：flex: 1 2 200px;