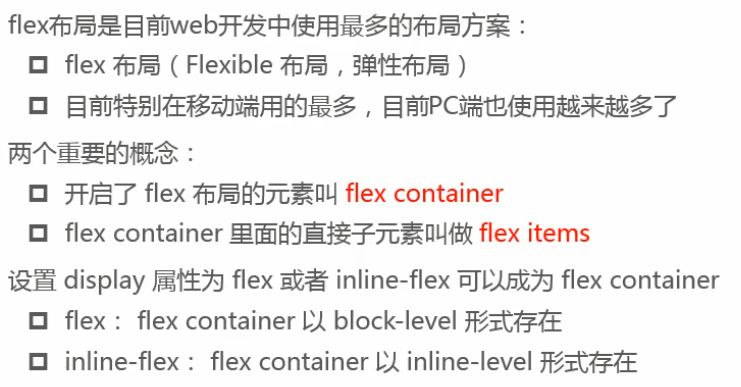
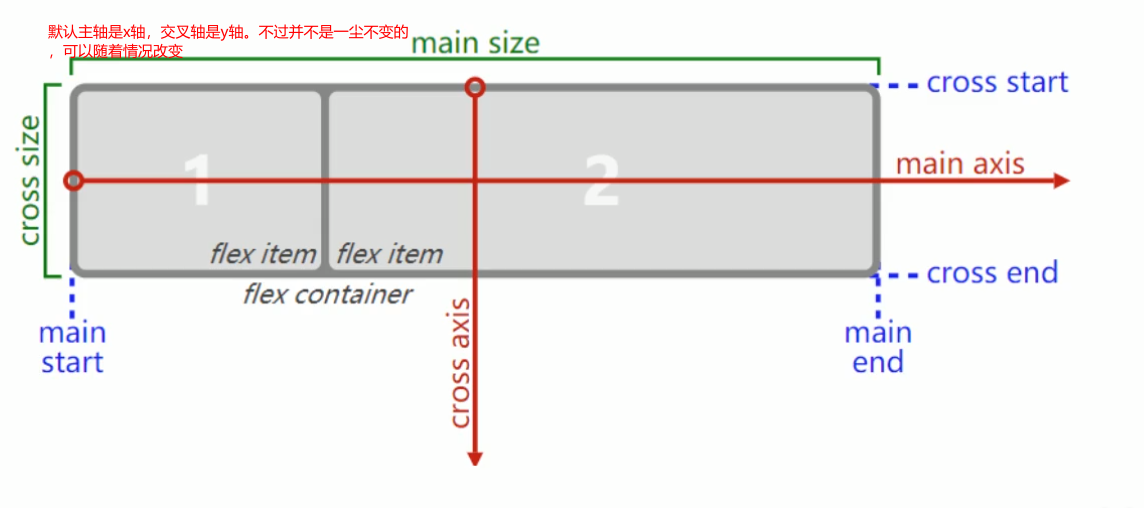
1. 基本认识



1. flex布局模型





1. flex container属性

一旦父元素开启了display:flex，那么它的子元素不论之前是块级元素还是行级元素现在的排布方式默认都是沿着main axis（主轴）从main start 到main end。

**（1）flex-direction：决定主轴的方向**

①row：默认方式，主轴从左到右

②row-reverse：将主轴方向反转，将会改变元素排布方式。主轴从右到左

③column：将主轴由原来的行形式改为列形式。主轴从上到下

④column-reverse：列形式取反，主轴从下到上。

**（2）justify-content：决定flex items主轴的对齐方式**

①flex-start：默认值，与main start对齐

②flex-end：与main end对齐

③center：居中对齐

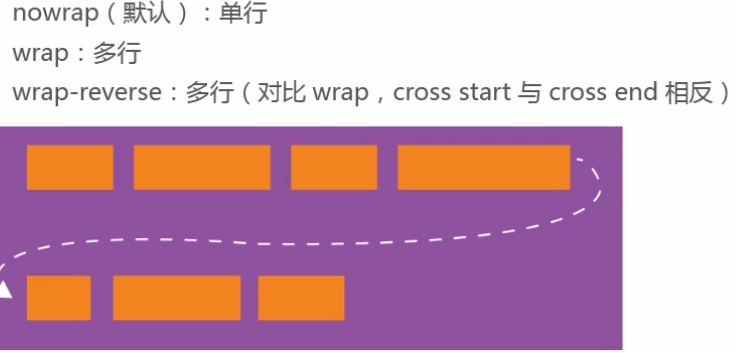
④space-between：flex items之间的距离相等，与main start，main end两端对齐。

⑤space-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。

⑥space-around：flex items之间的距离相等，flex items与main start，main end之间的距离是flex items之间距离的一半。

**（3）flex-wrap：决定了flex-container是单行显示子元素还是多行显示**

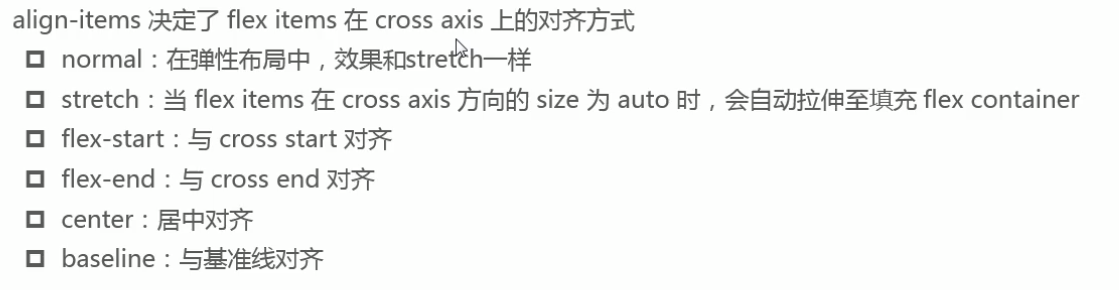
默认情况下flex items都是在一行显示，如果不够摆flex就会自动对其进行压缩。wrap-reverse基本不用。



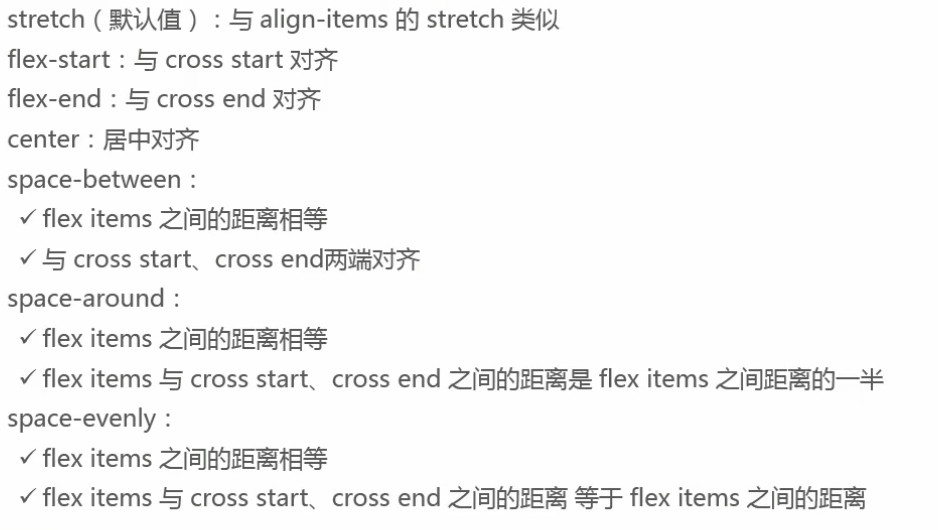


**（4）align-items：决定flex items在cross axis上的对齐方式**

注意这里的center就是垂直居中的效果，baseline如果是多行文本的话以第一行为准。



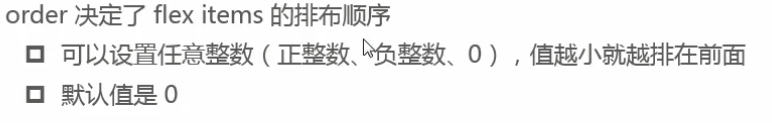
**(5)align-content：决定了多行flex items在cross axis上的对齐方式，用法与justify-content类似。**



1. flex items属性

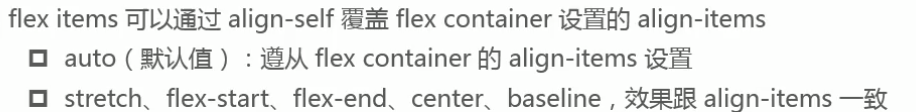
**（1）order属性：改变flex items默认的排布顺序**

**很少用order去改变默认排布顺序**

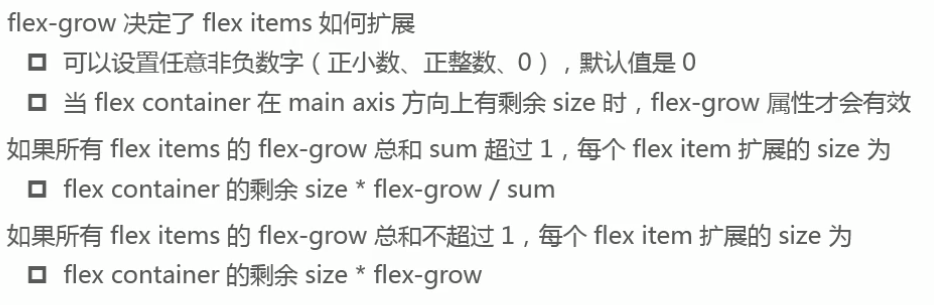


**（2）align-self：flex items在cross axis方向上的排布方式**

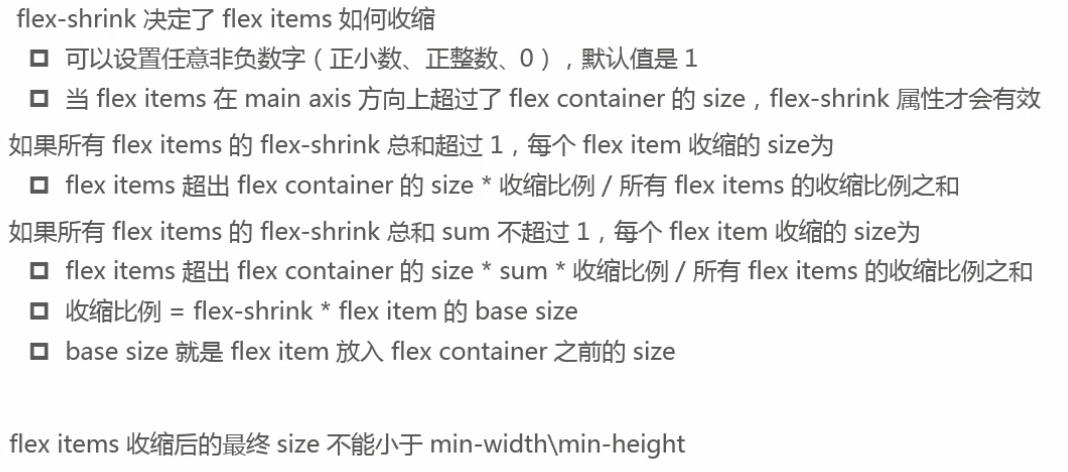
也很少用



**（3）flex-grow：根据需要适当增大flex items在主轴方向上的宽度**

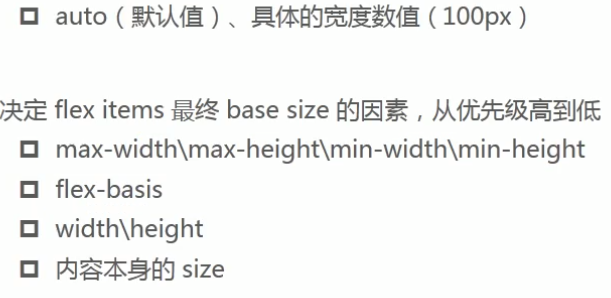


**（4）flex-shrink：收缩flex items元素**



**（5）flex-basis：决定主轴上的flex items在main axis方向上的base size**

**也很少用**



**（6）flex：是flex-grow||flex-shrink||flex-basis的缩写，可以写一个，两个或三个**

