

Flex布局

Flex 布局->基本语法

什么是Flex布局: Flex是Flexible Box的缩写，意为"弹性布局"，用来为盒状模型提供最大的灵活性。

->任何一个容器都可以指定为Flex布局;

```
.box{
    display: flex;
}
```

->行内元素也可以使用Flex布局;

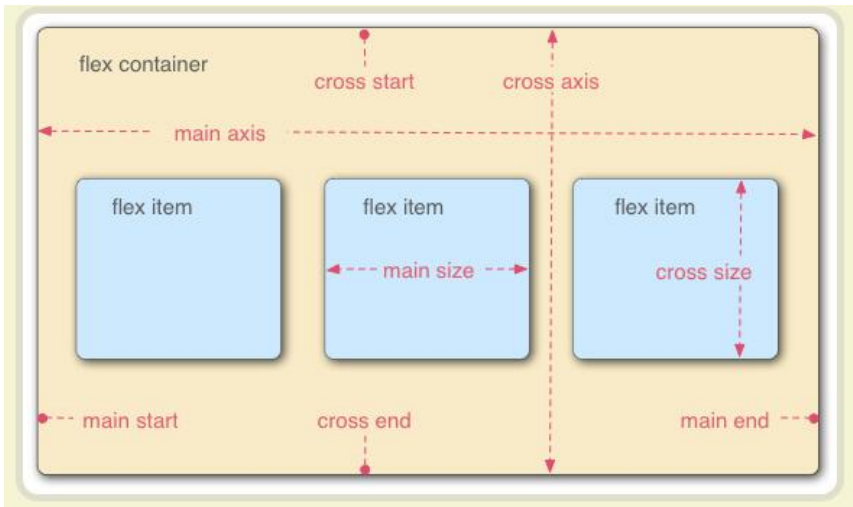
```
.box{
    display: inline-flex;
}
```

->Webkit内核的浏览器，必须加上-webkit前缀;

```
.box{
    display: -webkit-flex; /* Safari */
    display: flex;
}
```

注意: 设为Flex布局以后，子元素的float、clear和vertical-align属性将失效;

基本概念: 采用Flex布局的元素，称为Flex容器（flex container），简称"容器"。它的所有子元素自动成为容器成员，称为Flex项目（flex item），简称"项目"。



->容器默认存在两根轴：水平的主轴（main axis）和垂直的交叉轴（cross axis）；

->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；

容器的属性: ->以下六个属性设置在容器上

- flex-direction
- flex-wrap
- flex-flow
- justify-content

- align-items
- align-content

1)flex-direction:决定主轴的方向(即项目的排列方向)



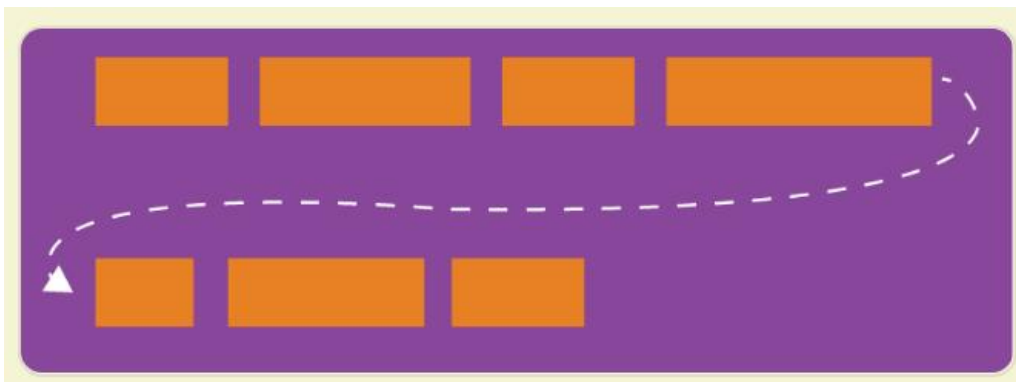
语法:

```
.box{
  flex-direction: row | row-reverse | column | column-reverse;
}
```

它可能有4个值:

- row (默认值) : 主轴为水平方向, 起点在左端。
- row-reverse : 主轴为水平方向, 起点在右端。
- column : 主轴为垂直方向, 起点在上沿。
- column-reverse : 主轴为垂直方向, 起点在下沿。

2)flex-wrap:定义了我们的弹性盒子的子元素的换行方式 默认是不换行.

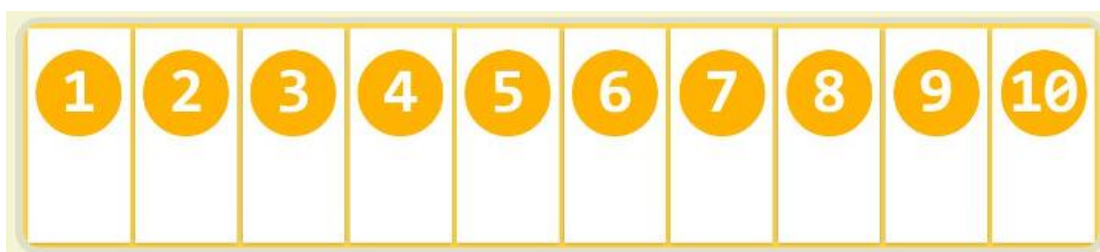


语法:

```
.box{
  flex-wrap: nowrap | wrap | wrap-reverse;
}
```

它可能取三个值:

(1) nowrap (默认) : 不换行,该情况下弹性子项可能会溢出容器.



(2) `wrap` : 换行, 第一行在上方, 该情况下弹性子项溢出的部分会被放置到新行, 子项内部会发生断行.



(3) `wrap-reverse` : 换行, 第一行在下方



3) `flex-flow`: 属性是 `flex-direction` 属性和 `flex-wrap` 属性的简写形式, 默认值为 `row nowrap`.

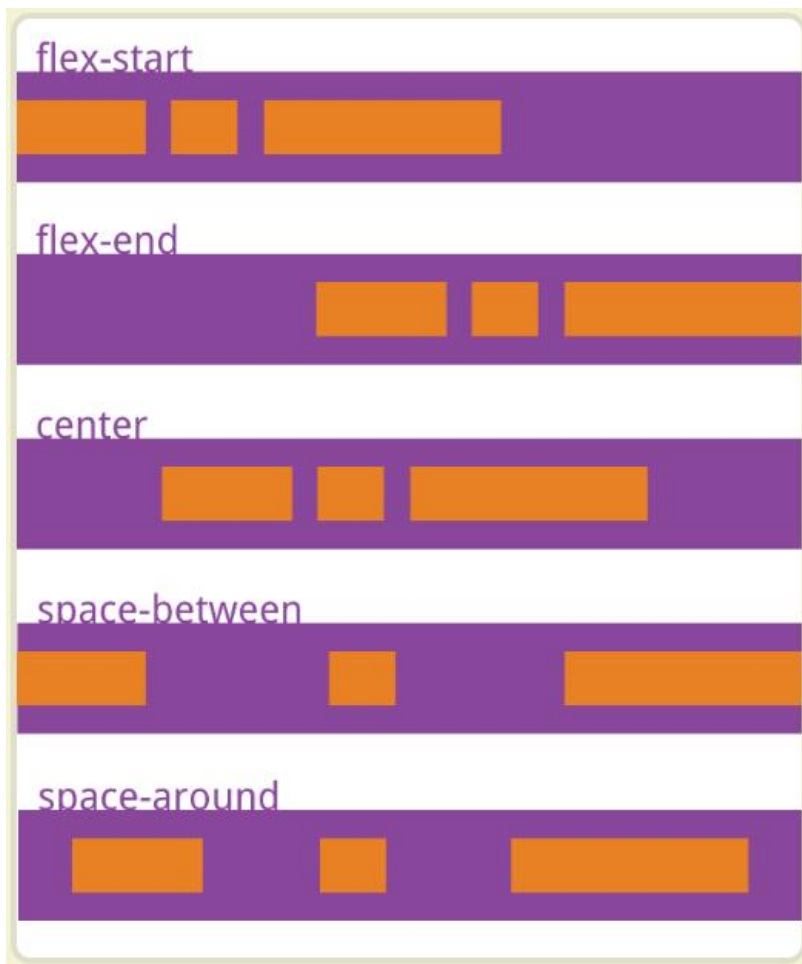
语法:

```
.box{
  flex-flow: <flex-direction> || <flex-wrap>;
}
```

*4) `justify-content`: 定义了项目在主轴上的对齐方式(内容对齐 (*justify-content*) 属性应用在弹性容器上 (父元素上), 把弹性项沿着弹性容器的主轴线 (*main axis*) 对齐).

语法:

```
.box{
  justify-content: flex-start | flex-end | center | space-between | space-around;
}
```



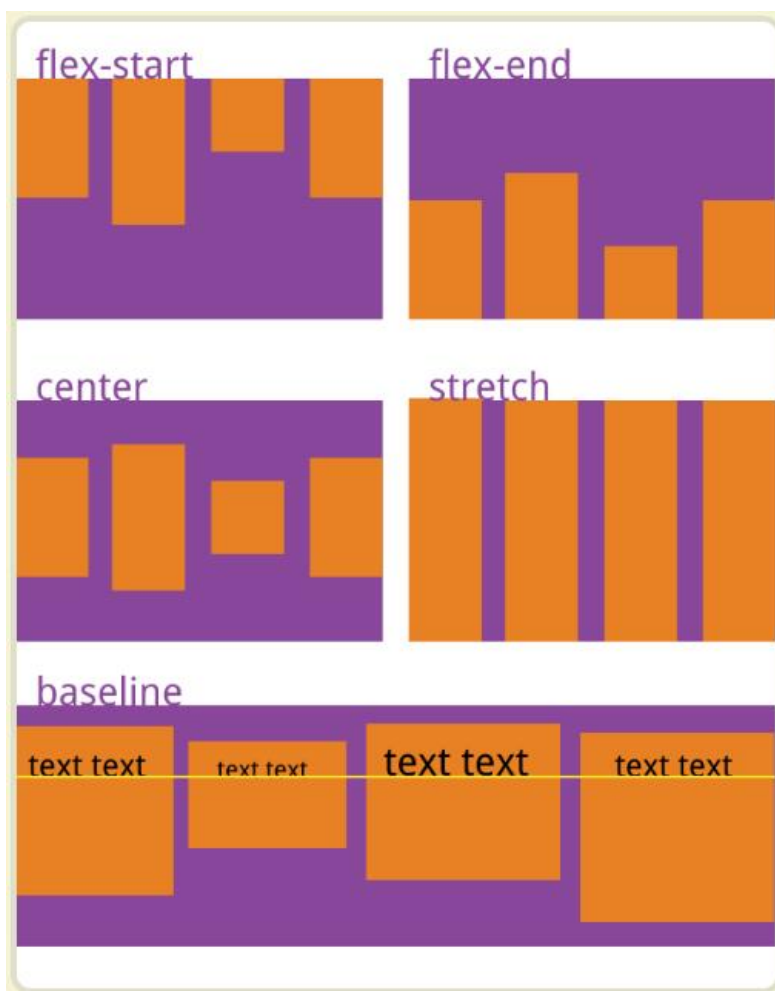
它可能取5个值，具体对齐方式与轴的方向有关。下面假设主轴为从左到右。

- flex-start（默认值）：左对齐.
- flex-end：右对齐.
- center：居中.
- space-between：两端对齐，项目之间的间隔都相等.
- space-around：每个项目两侧的间隔相等。所以，项目之间的间隔比项目与边框的间隔大一倍.

*5)align-items:属性定义项目在交叉轴上如何对齐.

语法:

```
.box{  
  align-items: flex-start | flex-end | center | baseline | stretch;  
}
```



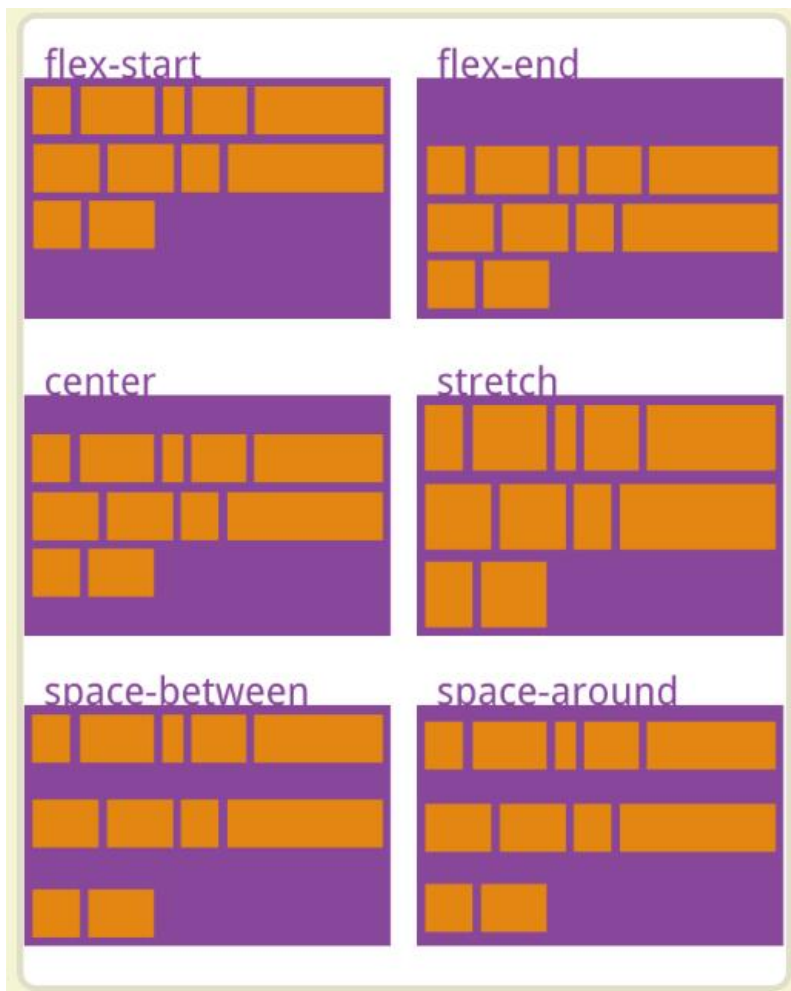
它可能取5个值。具体的对齐方式与交叉轴的方向有关，下面假设交叉轴从上到下。

- flex-start：弹性盒子元素的侧轴（纵轴）开始位置（默认最上边）紧靠住该行的侧轴起始边界。
- flex-end：弹性盒子元素位于符合自的底边（纵轴底部对齐）
- center：弹性盒子元素在该行的侧轴（纵轴）上居中放置。（如果该行的尺寸小于弹性盒子元素的尺寸，则会向两个方向溢出相同的长度）
- baseline:基于我们内容的对于基线对齐，应用于文字对齐，该基线对齐
- stretch（默认值）：弹性盒子元素的侧轴（纵轴）开始位置（默认最上边）紧靠住该行的侧轴起始边界

*6)align-content:属性定义了对多根轴线的对齐方式.如果项目只有一根轴线，该属性不起作用.

语法:

```
.box{
  align-content: flex-start | flex-end | center | space-between | space-around | stretch;
}
```



该属性可能取6个值。

- flex-start : 与交叉轴的起点对齐。
- flex-end : 与交叉轴的终点对齐。
- center : 与交叉轴的中点对齐。
- space-between : 与交叉轴两端对齐，轴线之间的间隔平均分布。
- space-around : 每根轴线两侧的间隔都相等。所以，轴线之间的间隔比轴线与边框的间隔大一倍。
- stretch (默认值) : 轴线占满整个交叉轴。

项目的属性:

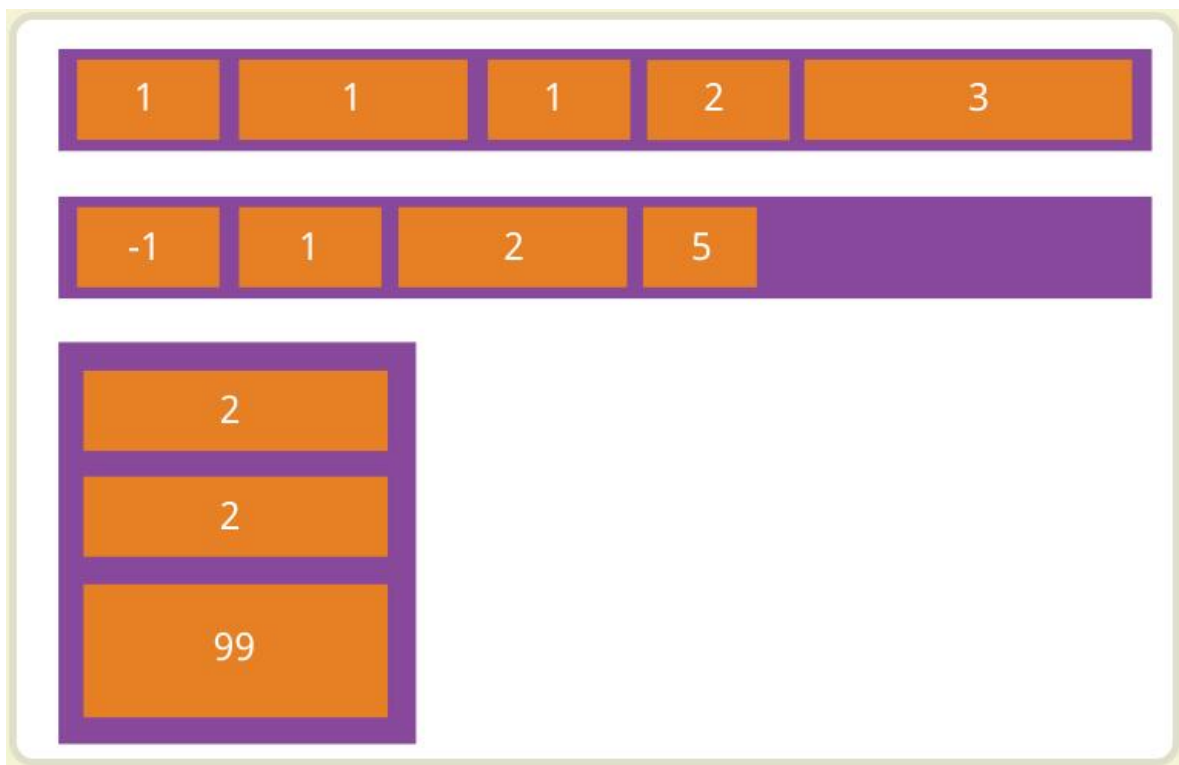
- order
- flex-grow
- flex-shrink
- flex-basis
- flex
- align-self

1)order:属性定义项目的排列顺序。数值越小，排列越靠前，默认为0(可以为负数)。

<!-- 设置"margin"值为"auto"值，自动获取弹性容器中剩余的空间。所以设置垂直方向margin值为"auto"，可以使弹性子元素在弹性容器的两上轴方向都完全集中。-->

语法:

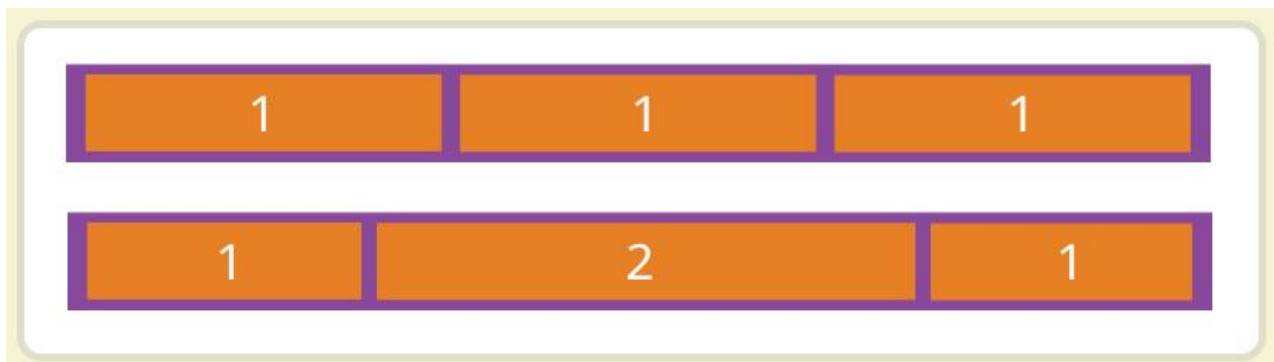
```
.item {
  order: <integer>;
}
```



2) **flex-grow**:属性定义项目的放大比例，默认为0，即如果存在剩余空间，也不放大。

语法:

```
.item {
  flex-grow: <number>; /* default 0 */
}
```

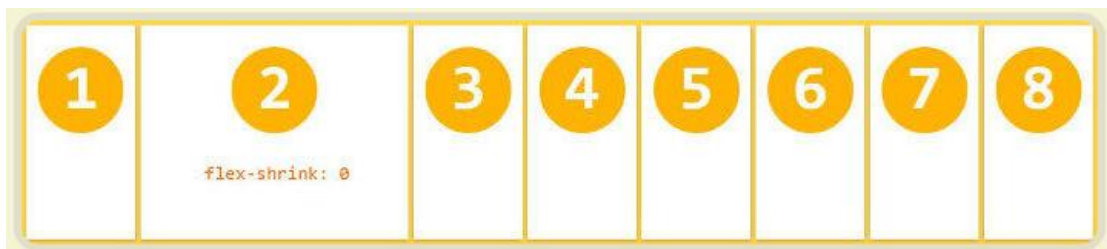


如果所有项目的flex-grow属性都为1，则它们将等分剩余空间（如果有的话）。如果一个项目的flex-grow属性为2，其他项目都为1，则前者占据的剩余空间将比其他项多一倍。

3) **flex-shrink**:属性定义了项目的缩小比例，默认为1，即如果空间不足，该项目将缩小

语法:

```
.item {
  flex-shrink: <number>; /* default 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

4)flex-basis:属性定义了再分配多余空间之前，项目占据的主轴空间（main size）。浏览器根据这个属性，计算主轴是否有多余空间。它的默认值为auto，即项目的本来大小

语法:

```
.item {
  flex-basis: <length> | auto; /* default auto */
}
```

它可以设为跟width或height属性一样的值（比如350px），则项目将占据固定空间。

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:

```
.item {
  flex: none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]
}
```

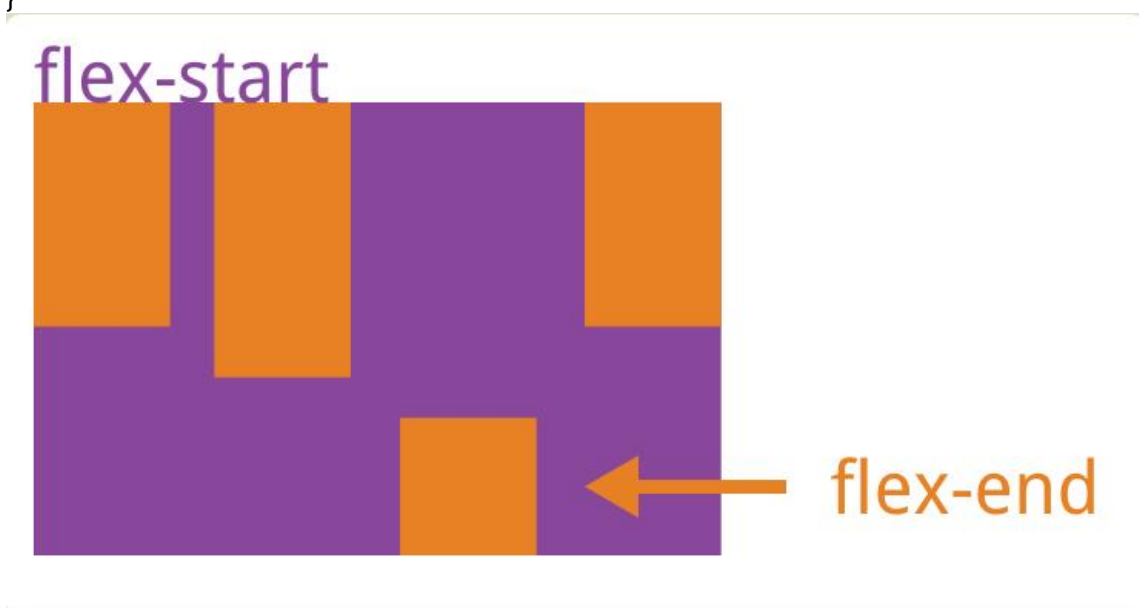
该属性有两个快捷值：auto (1 1 auto) 和 none (0 0 auto)。

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。

6)align-self:属性允许单个项目有与其他项目不一样的对齐方式，可覆盖align-items属性。默认值为auto，表示继承父元素的align-items属性，如果没有父元素，则等同于stretch

语法:

```
.item {
  align-self: auto | flex-start | flex-end | center | baseline | stretch;
}
```



- auto：如果'align-self'的值为'auto'，则其计算值为元素的父元素的'align-items'值，如果其没有父元素，则计算值为'stretch'。
- flex-start：弹性盒子元素的侧轴（纵轴）起始位置的边界紧靠住该行的侧轴起始边界。

- flex-end：弹性盒子元素的侧轴（纵轴）起始位置的边界紧靠住该行的侧轴结束边界。
- center：弹性盒子元素在该行的侧轴（纵轴）上居中放置。（如果该行的尺寸小于弹性盒子元素的尺寸，则会向两个方向溢出相同的长度）。
- baseline：内容只有一个的时候行内轴与侧轴为同一条，则该值与'flex-start'等效。其它情况下，该值将参与基线对齐。
- stretch：如果指定侧轴大小的属性值为'auto'，则其值会使项目的边距盒的尺寸尽可能接近所在行的尺寸，但同时会遵照'min/max-width/height'属性的限制