

如表 6- 1 所示，以下 6 属性，是作用在父容器上的。

表 6- 1 作用在父容器上的 6 个弹性布局属性汇总

值	描述
flex-direction	指定弹性容器中子元素排列方式
flex-wrap	设置弹性盒子的子元素超出父容器时是否换行
flex-flow	flex-direction 和 flex-wrap 的简写
justify-content	设置弹性盒子元素在主轴（横轴）方向上的对齐方式
align-items	设置弹性盒子元素在侧轴（纵轴）方向上的对齐方式
align-content	修改 flex-wrap 属性的行为，类似 align-items，但不是设置子元素对齐，而是设置行对齐

1.1.1 flex-direction 属性

flex-direction 属性是用来指定弹性容器中子元素排列方式，它决定了主轴的方向。它有如下 4 个属性值,如表 6- 2 所示。他们所表现出的效果如图 6- 5 所示

表 6- 2 flex-direction 属性值

序号	值	描述
1	row (默认值)	主轴为水平方向，起点在左端
2	row-reverse	主轴在水平方向，起点在右端
3	column	主轴为垂直方向，起点在上沿
4	column-reverse	主轴为垂直方向，起点在下沿

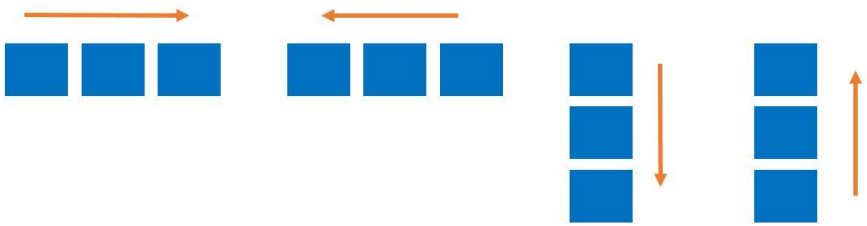


图 6- 5 不同排列方式所带来的效果

1.1.2 flex-wrap 属性

flex-wrap 属性规定 flex 容器是单行或者多行，同时横轴的方向决定了新行堆叠的方向。也就是说用来设置弹性盒子的子元素超出父容器时是否换行。属性值如表 6- 3 所示。

表 6- 3 flex-wrap 属性值

值	描述
nowrap (默认值)	当容器宽度不够时，每个项目会被挤压宽度。
wrap	换行，并且第一行在容器最上方
wrap-reverse	换行，并且第一行在容器最下方。

(1) nowrap。如所示当容器宽度不够时，每个项目会被挤压宽度；效果如图 6- 6 所示。

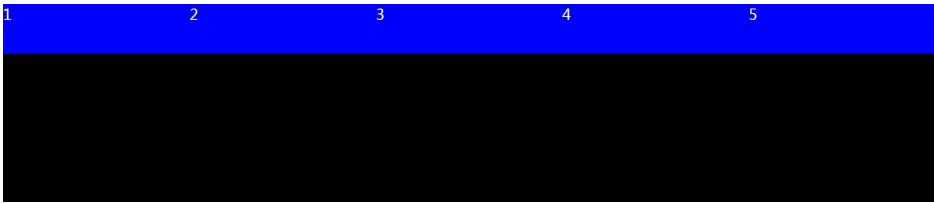


图 6- 6 “flex-wrap:nowrap” 使用效果

(2) wrap: 换行，第一行在上方。效果如图 6- 7 所示。



图 6- 7 “flex-wrap:wrap” 使用效果

(3) wrap-reverse: 换行，第一行在下方。效果如图 6- 8 所示。

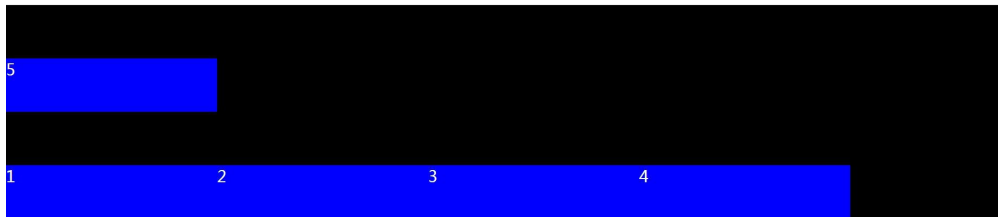


图 6- 8 “flex-wrap: wrap-reverse” 使用效果

1.1.3 、flex-flow 属性

flex-flow 是 flex-direction 和 flex-wrap 的缩写形式，默认值为：flex-flow: row nowrap; 语法如下：

flex-flow: <flex-direction> || <flex-wrap>

flex-direction: row (默认值) row-reverse column column-reverse;
flex-wrap: nowrap (默认值) wrap wrap-reverse;

1.1.4 justify-content 属性

justify-content 属性定义了项目在主轴上的对齐方式。justify-content 属性值如表 6- 4 所示。为方便演示效果，我们在 html 代码部分，将盒子减少为 3 个。

表 6- 4 justify-content 属性值

值	描述
flex-start (默认)	子元素位于主轴起点。
flex-end	子元素位于主轴终点。
center	居中。
space-between	两端对齐，项目之间的间隔都相等。(开头和最后的项目，与父容器边缘没有间隔)。
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。(开头和最后的项目，与父容器边缘有一定的间隔)。

(1) flex-start。子元素左对齐。效果如图 6- 9 所示。



图 6- 9 “justify-content: flex-start” 使用效果

(2) flex-end。子元素右对齐。效果如图 6- 10 所示。



图 6- 10 “justify-content: flex-end” 使用效果

(3) center。子元素居中。效果如图 6- 11 所示。

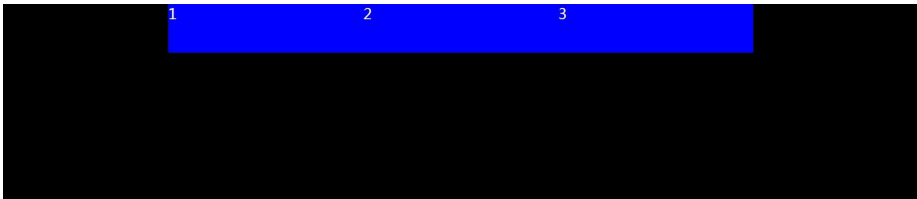


图 6- 11 “justify-content: center” 使用效果

(4) space-between。子元素两端对齐，且两端没空隙。效果如图 6- 12 所示。

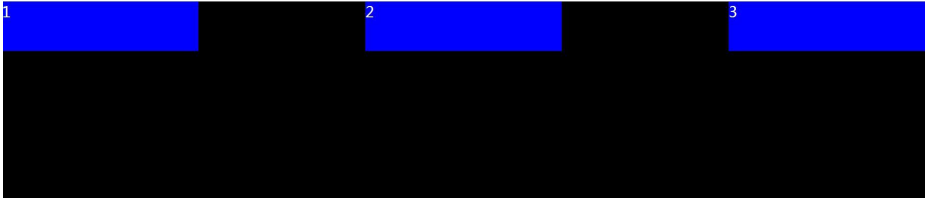


图 6- 12 “justify-content: space-between” 使用效果

(5) space-around。每个子元素两侧的间隔相等。且两端空隙，是中间的一半。效果如图 6- 13 所示。

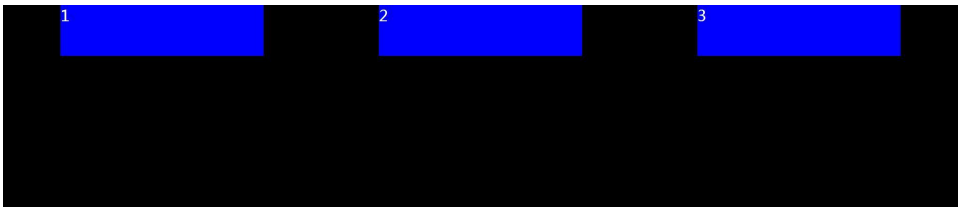


图 6- 13 “justify-content: space-around” 使用效果

1.1.5 align-items 属性

align-items 属性定义项目在交叉轴上如何对齐。如表 6- 5 所示。

表 6- 5

值	描述
flex-start	交叉轴的起点对齐
flex-end	交叉轴的终点对齐
center	交叉轴的中点对齐

baseline	项目的第一行文字的基线对齐。(文字的行高、字体大小会影响每行的基线)
stretch (默认值)	如果项目未设置高度或设为 auto，将占满整个容器的高度。

(1) flex-start。交叉轴的起点对齐。效果如图 6- 14 所示。

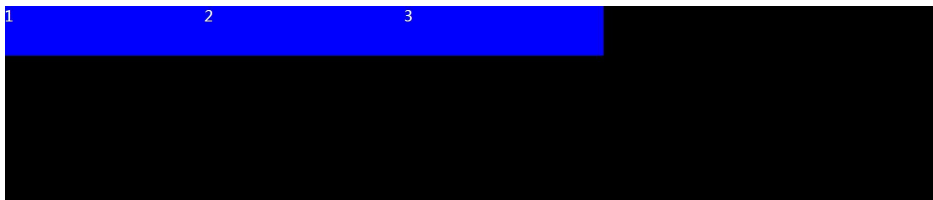


图 6- 14 flex-start 效果

(2) flex-end。交叉轴的终点对齐。效果如图 6- 15 所示。

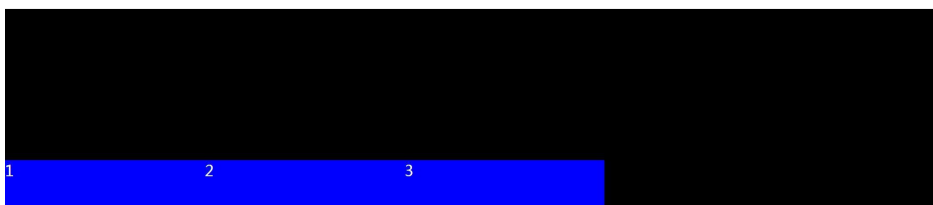


图 6- 15 flex-end 效果

(3) center。交叉轴的中点对齐。效果如图 6- 16 所示。



图 6- 16 center 效果

(4) baseline。项目的第一行文字的基线对齐。(文字的行高、字体大小会影响每行的基线)。在这里，我们把数字 1 设置成了 50px。因为字体变大，基线下移（红色为基线），所有 div 基线对齐。如图 6- 17 所示

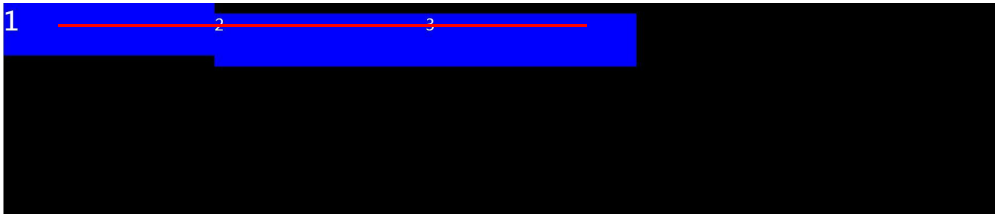


图 6- 17 Baseline 效果

(5) stretch (默认值)，拉升。子项目不需要指定高度（我们在 css 中的代码中，注释了高度）。效果如图 6- 18 所示。

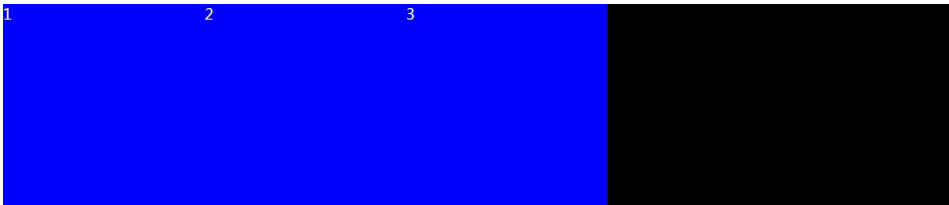


图 6- 18 stretch 效果

1.1.6 align-content 属性

align-content 属性定义了控制容器内多行在交叉轴上的排列方式。这里注意关键词多行，要实现 align-content 效果，首先我们要修改以下属性为多行。

```
flex-wrap:wrap;
```

此时子元素呈现多行排列。网页效果如图 6- 19 所示。

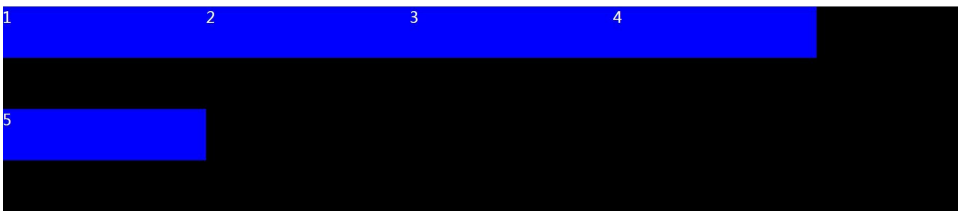


图 6- 19 “flex-wrap:wrap” 网页效果

align-content 属性主要有以下 6 个属性值，如表 6- 6 所示。

表 6- 6 align-content 属性值

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

(1) flex-start。与交叉轴的起点对齐。如图 6- 20 所示。

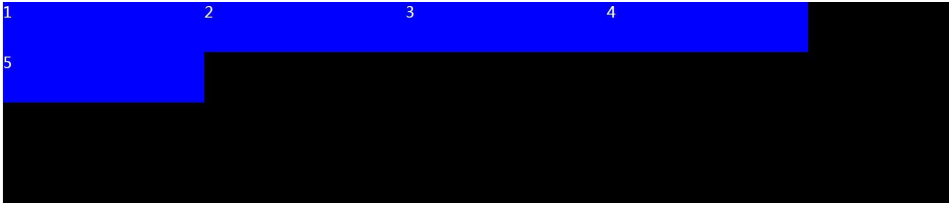


图 6- 20 “flex-start” 使用效果

(2) flex-end。与交叉轴的终点对齐。如图 6- 21 所示。

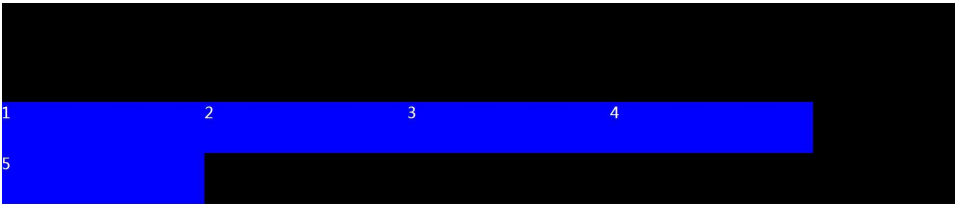


图 6- 21 “flex-end” 使用效果

(3) center。与交叉轴的中点对齐。如图 6- 22 所示。

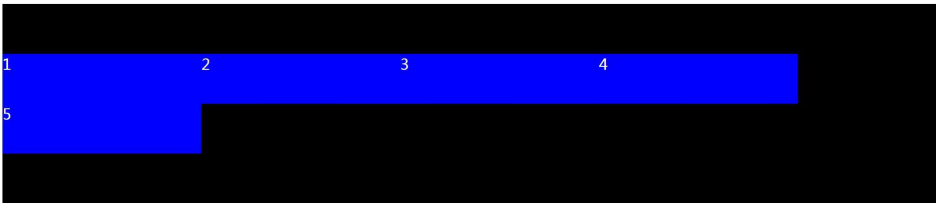


图 6- 22 “center” 使用效果

(4) space-between。与交叉轴两端对齐，轴线之间的间隔平均分布。如图 6- 23 所示。



图 6- 23 “space-between” 使用效果

(5) 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。如图 6- 24 所示。



图 6- 24 “space-around” 使用效果

(6) stretch (默认值)。子项目不需要指定高度（我们在 css 中的代码中，注释了高度）。如图 6- 25 所示。

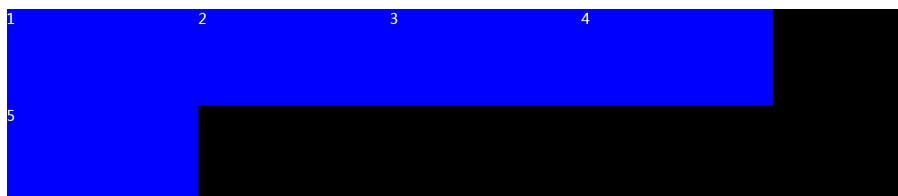


图 6- 25 “stretch” 使用效果