



《鸿蒙北向应用开发基础》之

容器组件介绍（上）



CONTENTS

1 PART ONE
容器组件概述

2 PART TWO
线性布局Row/Column

3 PART THREE
弹性布局Flex

4 PART FOUR
列表布局List

- ◆ 掌握线性布局、Flex弹性布局的特点及其属性的应用；
- ◆ 掌握List组件的应用；



务实创新 极致透明

01 容器组件概述



容器组件是一种比较特殊的组件，它可以包含其他的组件，而且按照一定的规律布局，帮助开发者生成精美的页面。容器组件除了放置基础组件外，也可以放置容器组件，通过多层布局的嵌套，可以布局出更丰富的页面。



02 线性布局Row/Column

2.1 线性布局介绍

2.2 自适应拉伸

2.3 自适应缩放

2.4 定位方式

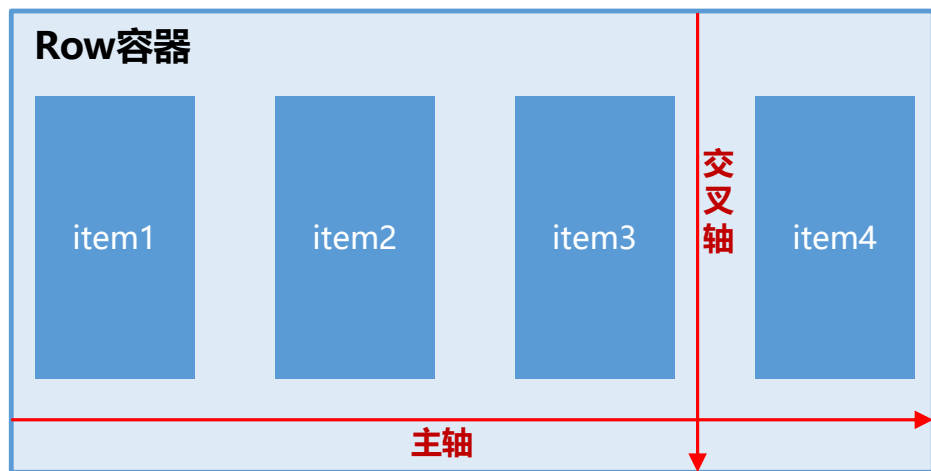
2.1 线性布局概述

务实创新 极致透明

线性布局是最常用的布局方式，它通过线性容器Row和Column来构建，容器的子组件默认沿各自的主轴排列。

- **Row**：行容器，主轴为水平方向，子组件默认在水平方向上排列；
- **Column**：列容器，主轴为垂直方向，子组件默认在垂直方向上排列。

Row容器和Column容器子组件的排列方式及轴线方向如下图所示：



线性布局在构建时可通过space、justifyContent、alignItems属性来调整子组件之间的间距及排列方式。

space

子组件在主轴上的间距，达到在排列方向上的等间距效果

justifyContent

子组件在主轴上的排列方式，实现布局自适应均分能力，取值FlexAlign类型

alignItems

子组件在交叉轴上的排列方式，交叉轴为**垂直方向**取值**VerticalAlign**类型，**水平方向**取值**HorizontalAlign**类型

(1) space间距

务实创新 极致透明

在Row容器中，space设置子组件在水平方向的间距；在Column容器中，space设置子组件在垂直方向的间距

```
build() {
```

```
  Row({space:25}){
```

```
    Text('item1').backgroundColor('#17A8E6').fontSize(22)
```

```
    Text('item2').backgroundColor('#17A8E6').fontSize(22)
```

```
    Text('item3').backgroundColor('#17A8E6').fontSize(22)
```

```
    Text('item4').backgroundColor('#17A8E6').fontSize(22)
```

```
  }
```

```
}
```



Row容器

```
build() {
```

```
  Column({space:35}){
```

```
    Text('item1').backgroundColor('#17A8E6').fontSize(22)
```

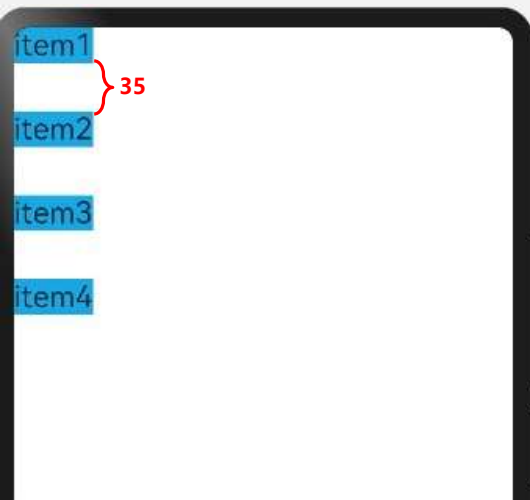
```
    Text('item2').backgroundColor('#17A8E6').fontSize(22)
```

```
    Text('item3').backgroundColor('#17A8E6').fontSize(22)
```

```
    Text('item4').backgroundColor('#17A8E6').fontSize(22)
```

```
  }
```

```
}
```



Column容器

(2) justifyContent子组件在主轴的排列方式

务实创新 极致透明

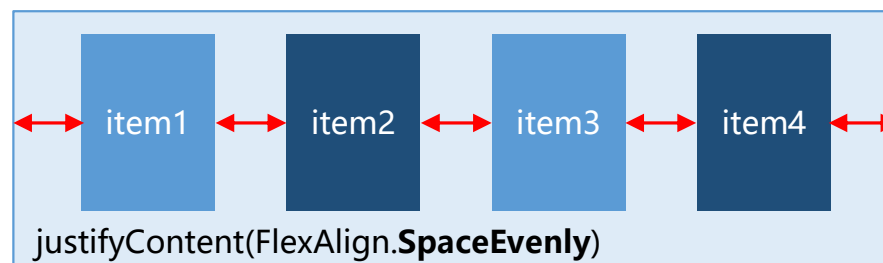
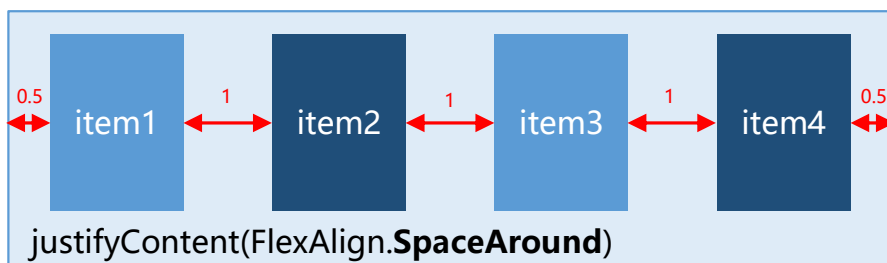
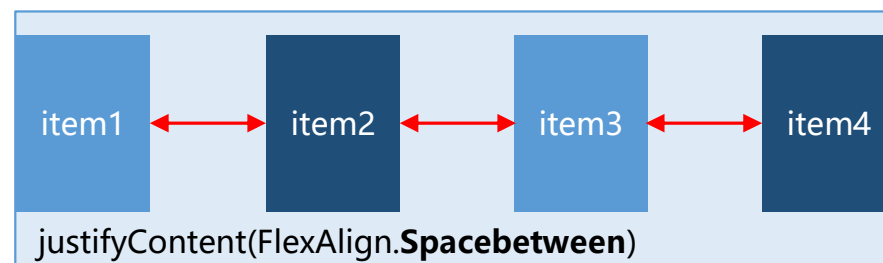
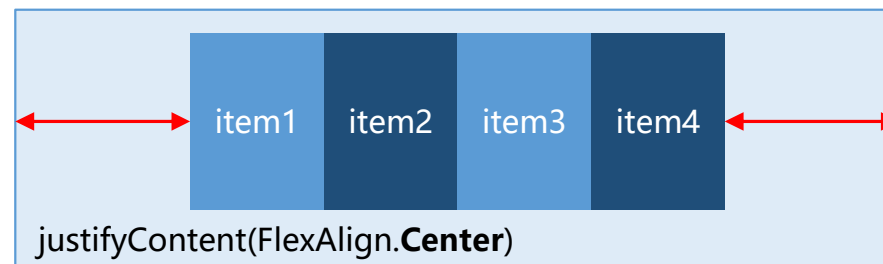
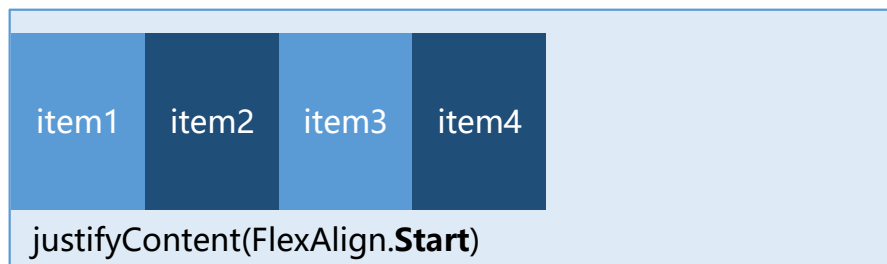
justifyContent属性控制子组件在主轴上的排列方式，实现布局自适应均分能力，取值FlexAlign类型，其值有以下6种。

- 1 justifyContent(FlexAlign.**Start**): 子组件整体在主轴方向**首端对齐**
- 2 justifyContent(FlexAlign.**Center**): 子组件整体在主轴方向**居中对齐**
- 3 justifyContent(FlexAlign.**End**): 子组件整体在主轴方向**尾部对齐**。
- 4 justifyContent(FlexAlign.**Spacebetween**): 子组件在主轴方向**两端对齐**，相邻子组件之间距离相同。
- 5 justifyContent(FlexAlign.**SpaceAround**): 子组件在主轴方向排列，**相邻子组件之间距离相同**。第一个子组件到主轴开始位置和最后一个子组件到主轴结束位置的距离是相邻子组件之间距离的一半。
- 6 justifyContent(FlexAlign.**SpaceEvenly**): 子组件在主轴方向均匀分配，相邻元素之间的距离、第一个子组件到主轴开始位置、最后一个子组件到主轴结束位置的**距离都一样**。

justifyContent子组件在主轴的排列方式

务实创新 极致透明

Row 容器内子组件在主轴上的排列情况如下图所示：



Row容器—主轴为水平方向

务实创新 极致透明

```
build() {  
    Column({ space: 15 }) {
```

```
        // 水平方向，左对齐
```

```
        Row() {
```

```
            Text('item1').backgroundColor('#17A8E6').fontSize(22)
```

```
            Text('item2').backgroundColor('#BFBFBF').fontSize(22)
```

```
            Text('item3').backgroundColor('#17A8E6').fontSize(22)
```

```
            Text('item4').backgroundColor('#BFBFBF').fontSize(22)
```

```
        }.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
        .justifyContent(FlexAlign.Start)
```

```
        // 水平方向，居中对齐
```

```
        Row() {
```

```
            Text('item1').backgroundColor('#17A8E6').fontSize(22)
```

```
            Text('item2').backgroundColor('#BFBFBF').fontSize(22)
```

```
            Text('item3').backgroundColor('#17A8E6').fontSize(22)
```

```
            Text('item4').backgroundColor('#BFBFBF').fontSize(22)
```

```
        }.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
        .justifyContent(FlexAlign.Center)
```

```
        // 水平方向，右对齐
```

```
        Row() {
```

```
            Text('item1').backgroundColor('#17A8E6').fontSize(22)
```

```
            Text('item2').backgroundColor('#BFBFBF').fontSize(22)
```

```
            Text('item3').backgroundColor('#17A8E6').fontSize(22)
```

```
            Text('item4').backgroundColor('#BFBFBF').fontSize(22)
```

```
        }.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
        .justifyContent(FlexAlign.End)
```

JustifyContentRow.ets

```
// 水平方向，两端对齐
```

```
Row() {
```

```
    Text('item1').backgroundColor('#17A8E6').fontSize(22)
```

```
    Text('item2').backgroundColor('#BFBFBF').fontSize(22)
```

```
    Text('item3').backgroundColor('#17A8E6').fontSize(22)
```

```
    Text('item4').backgroundColor('#BFBFBF').fontSize(22)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
.justifyContent(FlexAlign.SpaceBetween)
```

```
// 水平方向，中间间距是两端间距的2倍
```

```
Row() {
```

```
    Text('item1').backgroundColor('#17A8E6').fontSize(22)
```

```
    Text('item2').backgroundColor('#BFBFBF').fontSize(22)
```

```
    Text('item3').backgroundColor('#17A8E6').fontSize(22)
```

```
    Text('item4').backgroundColor('#BFBFBF').fontSize(22)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
.justifyContent(FlexAlign.SpaceAround)
```

```
// 水平方向，间距都相等
```

```
Row() {
```

```
    Text('item1').backgroundColor('#17A8E6').fontSize(22)
```

```
    Text('item2').backgroundColor('#BFBFBF').fontSize(22)
```

```
    Text('item3').backgroundColor('#17A8E6').fontSize(22)
```

```
    Text('item4').backgroundColor('#BFBFBF').fontSize(22)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
.justifyContent(FlexAlign.SpaceEvenly)
```

```
}
```

item1 item2 item3 item4

item1 item2 item3 item4

item1 item2 item3 item4

item1 item2 item3 item4

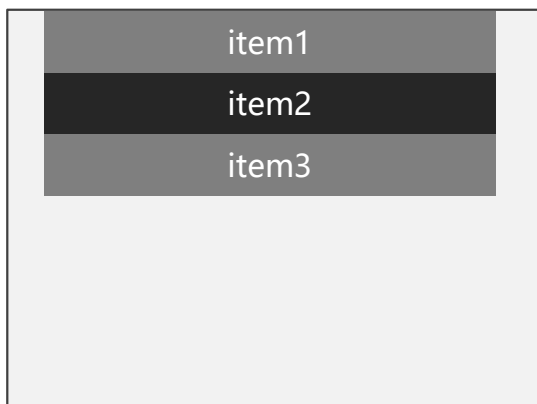
item1 item2 item3 item4

item1 item2 item3 item4

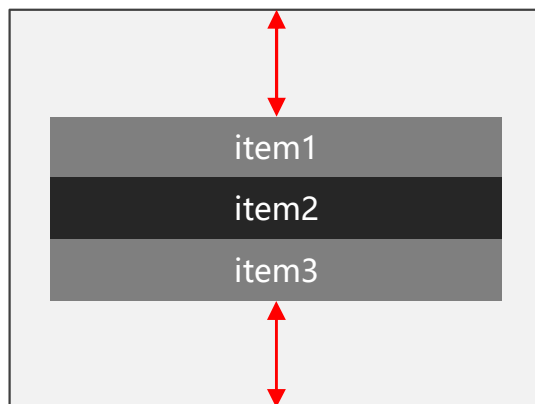
justifyContent子组件在主轴的排列方式

务实创新 极致透明

Column 容器内子组件在主轴上的排列情况如下图所示：



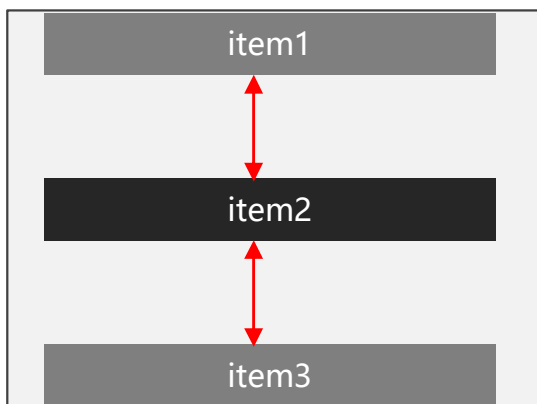
justifyContent(FlexAlign.Start)



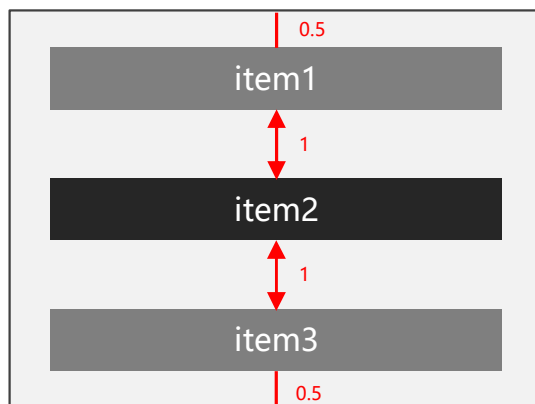
justifyContent(FlexAlign.Center)



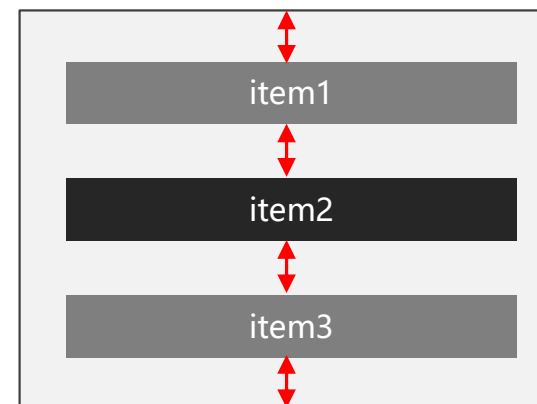
justifyContent(FlexAlign.End)



justifyContent(FlexAlign.Spacebetween)



justifyContent(FlexAlign.SpaceAround)



justifyContent(FlexAlign.SpaceEvenly)

Column容器—主轴为垂直方向

务实创新 极致透明

build() {

Column({ space: 15 }) {

// 垂直方向，顶部对齐

Column() {

Text('item1').backgroundColor('#17A8E6').fontSize(22)

Text('item2').backgroundColor('#BFBFBF').fontSize(22)

Text('item3').backgroundColor('#17A8E6').fontSize(22)

}.width('100%').height('15%').backgroundColor('#f1f1f1')

.justifyContent(FlexAlign.Start)

// 垂直方向，居中对齐

Column() {

Text('item1').backgroundColor('#17A8E6').fontSize(22)

Text('item2').backgroundColor('#BFBFBF').fontSize(22)

Text('item3').backgroundColor('#17A8E6').fontSize(22)

}.width('100%').height('15%').backgroundColor('#f1f1f1')

.justifyContent(FlexAlign.Center)

// 垂直方向，底部对齐

Column() {

Text('item1').backgroundColor('#17A8E6').fontSize(22)

Text('item2').backgroundColor('#BFBFBF').fontSize(22)

Text('item3').backgroundColor('#17A8E6').fontSize(22)

}.width('100%').height('15%').backgroundColor('#f1f1f1')

.justifyContent(FlexAlign.End)

JustifyContentColumn.ets

// 垂直方向，两端对齐

Column() {

Text('item1').backgroundColor('#17A8E6').fontSize(22)

Text('item2').backgroundColor('#BFBFBF').fontSize(22)

Text('item3').backgroundColor('#17A8E6').fontSize(22)

}.width('100%').height('15%').backgroundColor('#f1f1f1')

.justifyContent(FlexAlign.SpaceBetween)

// 垂直方向，中间间距是两端间距的2倍

Column() {

Text('item1').backgroundColor('#17A8E6').fontSize(22)

Text('item2').backgroundColor('#BFBFBF').fontSize(22)

Text('item3').backgroundColor('#17A8E6').fontSize(22)

}.width('100%').height('15%').backgroundColor('#f1f1f1')

.justifyContent(FlexAlign.SpaceAround)

// 垂直方向，间距都相等

Column() {

Text('item1').backgroundColor('#17A8E6').fontSize(22)

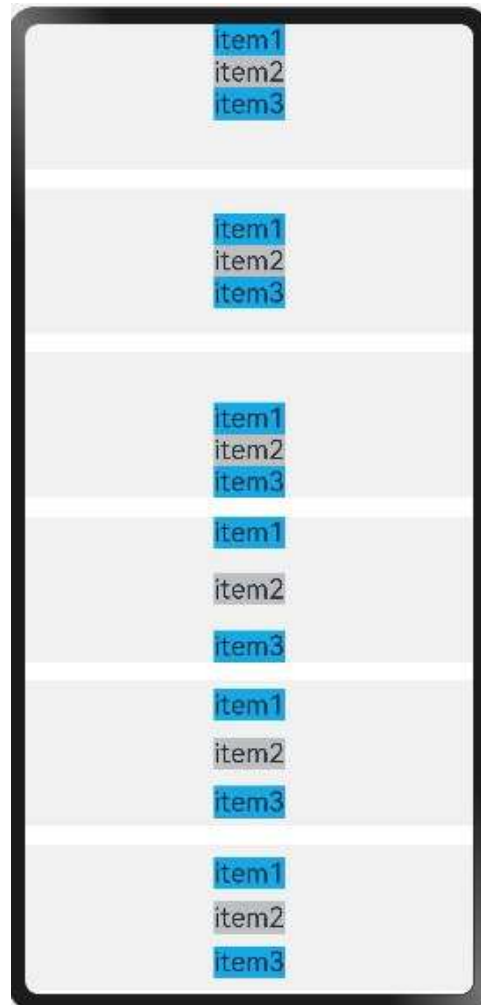
Text('item2').backgroundColor('#BFBFBF').fontSize(22)

Text('item3').backgroundColor('#17A8E6').fontSize(22)

}.width('100%').height('15%').backgroundColor('#f1f1f1')

.justifyContent(FlexAlign.SpaceEvenly)

}



(3) alignItems子组件在交叉轴的排列方式

务实创新 极致透明

alignItems属性控制子组件在交叉轴的排列方式，交叉轴为**垂直方向**取值**VerticalAlign**类型，**水平方向**取值**HorizontalAlign**类型。



Row容器

- alignItems((VerticalAlign.**Top**): 子组件整体与交叉轴**顶部对齐**
- alignItems((VerticalAlign.**Center**): 子组件整体与交叉轴**垂直居中对齐**
- alignItems(VERTICALALIGN.**Bottom**): 子组件整体与交叉轴**底部对齐**。



Column容器

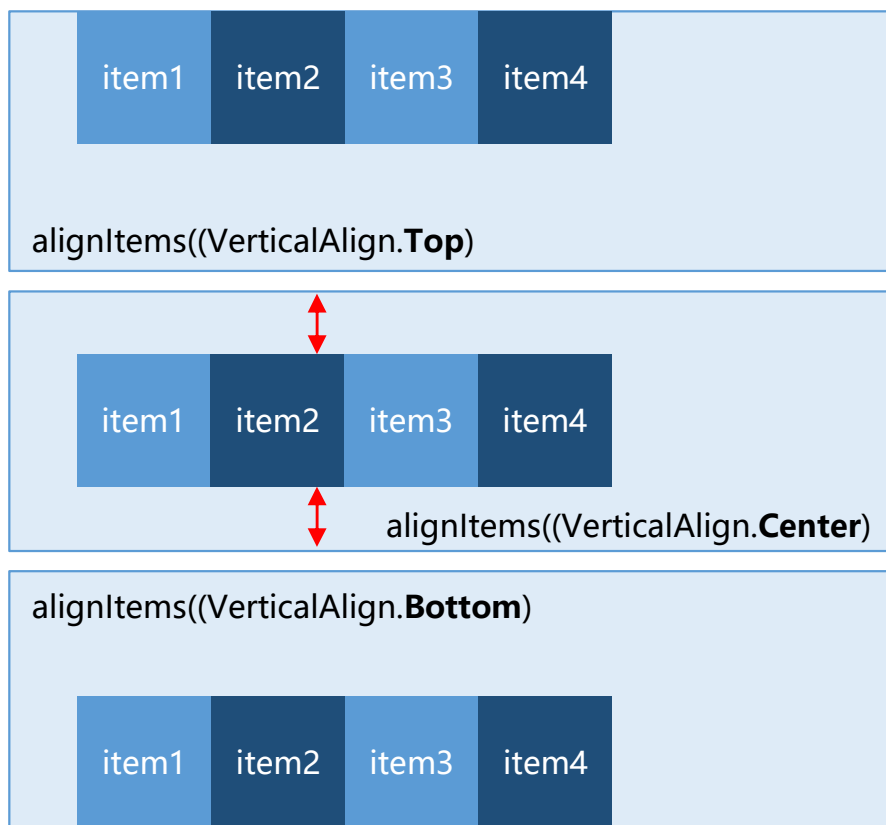
- alignItems(HorizontalAlign.**Start**): 子组件整体与交叉轴**左对齐**。
- alignItems(HorizontalAlign.**Center**): 子组件整体与交叉轴**水平居中对齐**。
- alignItems(HorizontalAlign.**End**): 子组件整体与交叉轴**右对齐**。

alignItems子组件在交叉轴的排列方式

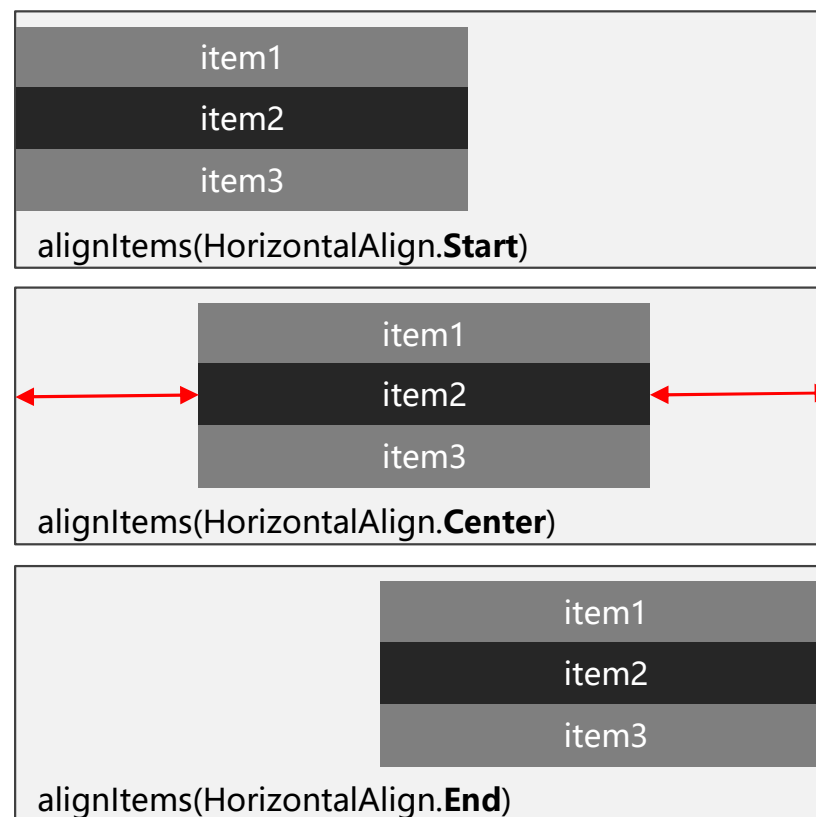
务实创新 极致透明

Row容器和Column容器的alignItems属性对应的子组件的排列情况如下图所示。

Row 容器



Column 容器



alignItems子组件在交叉轴的排列方式

务实创新 极致透明

```
build() {  
  Column({ space: 15 }) {
```

// 垂直方向, 顶部对齐

```
    Row() {  
      Text('item1').backgroundColor('#17A8E6').fontSize(22)  
      Text('item2').backgroundColor('#BFBFBF').fontSize(22)  
      Text('item3').backgroundColor('#17A8E6').fontSize(22)  
      Text('item4').backgroundColor('#BFBFBF').fontSize(22)  
    }.width('100%').height('15%').backgroundColor('#f1f1f1')  
    .alignItems(VerticalAlign.Top)
```

// 垂直方向, 居中对齐

```
    Row() {  
      Text('item1').backgroundColor('#17A8E6').fontSize(22)  
      Text('item2').backgroundColor('#BFBFBF').fontSize(22)  
      Text('item3').backgroundColor('#17A8E6').fontSize(22)  
      Text('item4').backgroundColor('#BFBFBF').fontSize(22)  
    }.width('100%').height('15%').backgroundColor('#f1f1f1')  
    .alignItems(VerticalAlign.Center)
```

// 垂直方向, 底部对齐

```
    Row() {  
      Text('item1').backgroundColor('#17A8E6').fontSize(22)  
      Text('item2').backgroundColor('#BFBFBF').fontSize(22)  
      Text('item3').backgroundColor('#17A8E6').fontSize(22)  
      Text('item4').backgroundColor('#BFBFBF').fontSize(22)  
    }.width('100%').height('15%').backgroundColor('#f1f1f1')  
    .alignItems(VerticalAlign.Bottom)
```

```
  }
```

AlignItemsRow.ets

item1 item2 item3 item4

item1 item2 item3 item4

item1 item2 item3 item4

```
build() {  
  Column({ space: 15 }) {
```

// 水平方向, 左对齐

```
    Column() {  
      Text('item1').backgroundColor('#17A8E6').fontSize(22)  
      Text('item2').backgroundColor('#BFBFBF').fontSize(22)  
      Text('item3').backgroundColor('#17A8E6').fontSize(22)  
      Text('item4').backgroundColor('#BFBFBF').fontSize(22)  
    }.width('100%').height('15%').backgroundColor('#f1f1f1')  
    .alignItems(HorizontalAlign.Start)
```

// 水平方向, 居中对齐

```
    Column() {  
      Text('item1').backgroundColor('#17A8E6').fontSize(22)  
      Text('item2').backgroundColor('#BFBFBF').fontSize(22)  
      Text('item3').backgroundColor('#17A8E6').fontSize(22)  
      Text('item4').backgroundColor('#BFBFBF').fontSize(22)  
    }.width('100%').height('15%').backgroundColor('#f1f1f1')  
    .alignItems(HorizontalAlign.Center)
```

// 水平方向, 右对齐

```
    Column() {  
      Text('item1').backgroundColor('#17A8E6').fontSize(22)  
      Text('item2').backgroundColor('#BFBFBF').fontSize(22)  
      Text('item3').backgroundColor('#17A8E6').fontSize(22)  
      Text('item4').backgroundColor('#BFBFBF').fontSize(22)  
    }.width('100%').height('15%').backgroundColor('#f1f1f1')  
    .alignItems(HorizontalAlign.End)
```

```
  }
```

AlignItemsColumn.ets

item1
item2
item3
item4

item1
item2
item3
item4

item1
item2
item3
item4

(4) alignSelf单个子组件在交叉轴的排列方式

务实创新 极致透明

alignSelf属性控制单个子组件在交叉轴的排列方式，其优先级高于alignItems属性，如果设置了alignSelf属性，则在单个子元素上会覆盖alignItems属性。alignSelf的取值为ItemAlign类型，其值有以下4种：

- 1 alignSelf(ItemAlign.**Start**): 单个子组件与交叉轴**首端对齐**
- 2 alignItems(ItemAlign.**Center**): 单个子组件与交叉轴**居中对齐**
- 3 alignItems(ItemAlign.**End**): 单个子组件与交叉轴**尾部对齐**。
- 4 alignItems(ItemAlign.**Stretch**): 单个子组件在交叉轴方向上占满空间。

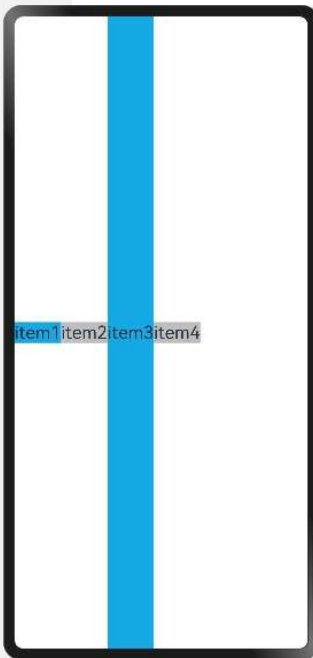
alignSelf单个子组件在交叉轴的排列方式

务实创新 极致透明

alignSelf属性的应用如下

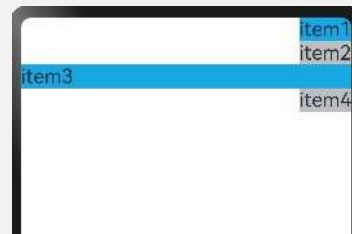
```
build() {  
  Row() {  
    Text('item1').backgroundColor('#17A8E6').fontSize(22)  
    Text('item2').backgroundColor('#BFBFBF').fontSize(22)  
    Text('item3').backgroundColor('#17A8E6').fontSize(22)  
    //设置该子组件在垂直方向上占满容器高度  
    .alignSelf(ItemAlign.Stretch)  
    Text('item4').backgroundColor('#BFBFBF').fontSize(22)  
  }  
  .height('100%')  
  //设置子组件整体与交叉轴垂直居中对齐  
  .alignItems(VerticalAlign.Center)  
}
```

AlignSelfRow.ets



```
build() {  
  Column() {  
    Text('item1').backgroundColor('#17A8E6').fontSize(22)  
    Text('item2').backgroundColor('#BFBFBF').fontSize(22)  
    Text('item3').backgroundColor('#17A8E6').fontSize(22)  
    //设置该子组件在水平方向上占满容器高度  
    .alignSelf(ItemAlign.Stretch)  
    Text('item4').backgroundColor('#BFBFBF').fontSize(22)  
  }  
  .width('100%')  
  //设置子组件整体与交叉轴右对齐  
  .alignItems(HorizontalAlign.End)  
}
```

AlignSelfColumn.ets



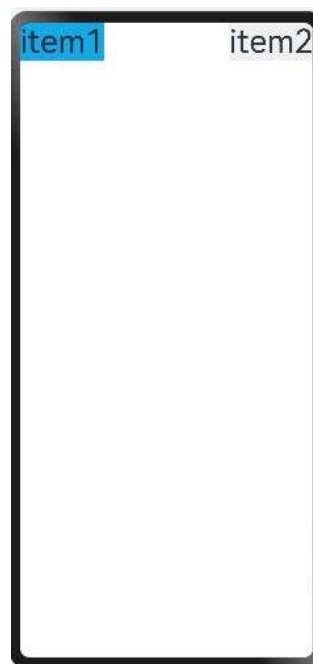
2.2 自适应拉伸

务实创新 极致透明

线性布局通过**空白填充组件Blank**，实现主轴方向上的自适应拉伸效果,具有自动填充容器空余部分的能力。仅当父组件为Row/Column/Flex时生效。

```
@Entry
@Component
struct BlankDemo {
  build() {
    Row() {
      Text('item1').backgroundColor('#17A8E6').fontSize(40)
      Blank()
      Text('item2').backgroundColor('#f1f1f1').fontSize(40)
    }
    .width('100%')
  }
}
```

BlankDemo.ets



竖屏



横屏

2.3 自适应缩放

务实创新 极致透明

在设备大小不同时，子组件按预设比例，随容器尺寸的变化而变化，实现方式如下：

01

宽度使用百分比

当父组件尺寸确定时，子组件的兄弟组件间使用百分比设置宽度，保证各元素在任意尺寸下自适应缩放

02

使用 layoutWeight

设置layoutWeight属性，子组件及兄弟自己按占主轴尺寸的权重分配，实现自适应缩放

LinearLayoutWidthDemo.ets

务实创新 极致透明

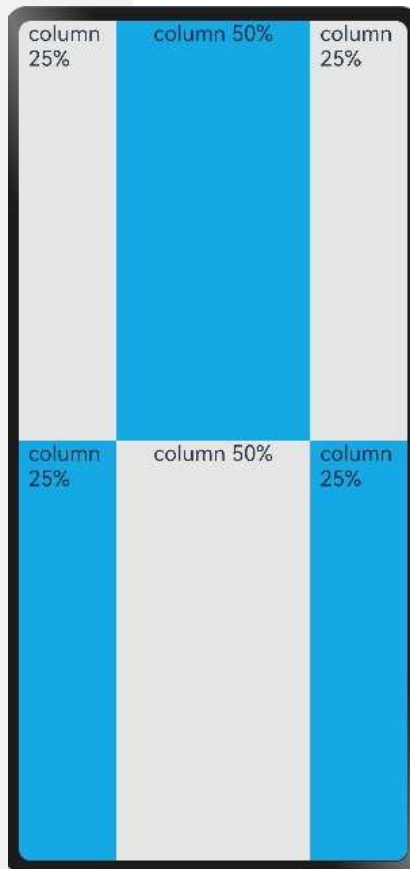
```
@Entry
@Component
struct LayoutWidthDemo {
    build() {
        Column() {
            Row() {
                Column() {
                    Text('column 25%').fontSize(20)
                }.width('25%').backgroundColor('#E6E6E6').height('50%')

                Column() {
                    Text('column 50%').fontSize(20)
                }.width('50%').backgroundColor('#17A8E6').height('50%')

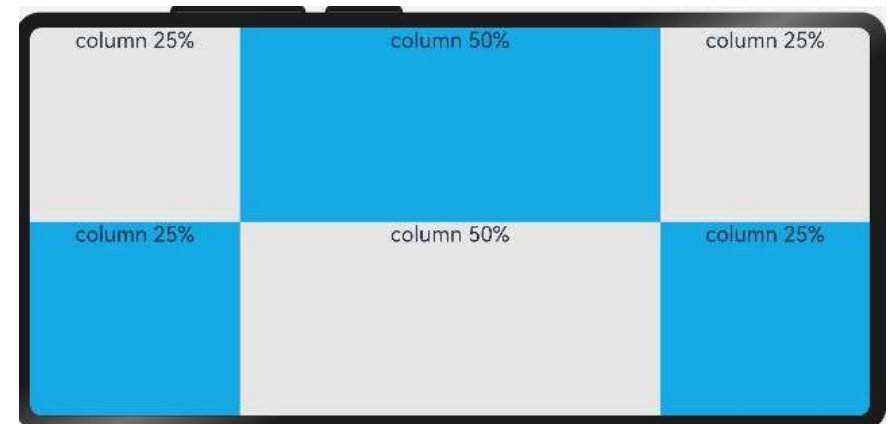
                Column() {
                    Text('column 25%').fontSize(20)
                }.width('25%').backgroundColor('#E6E6E6').height('50%')
            }

            Row() {
                Column() {
                    Text('column 25%').fontSize(20)
                }.layoutWeight(1).backgroundColor('#17A8E6').height('50%')
                Column() {
                    Text('column 50%').fontSize(20)
                }.layoutWeight(2).backgroundColor('#E6E6E6').height('50%')
                Column() {
                    Text('column 25%').fontSize(20)
                }.layoutWeight(1).backgroundColor('#17A8E6').height('50%')}
            }
        }
    }
}
```

LayoutWidthDemo.ets



竖屏

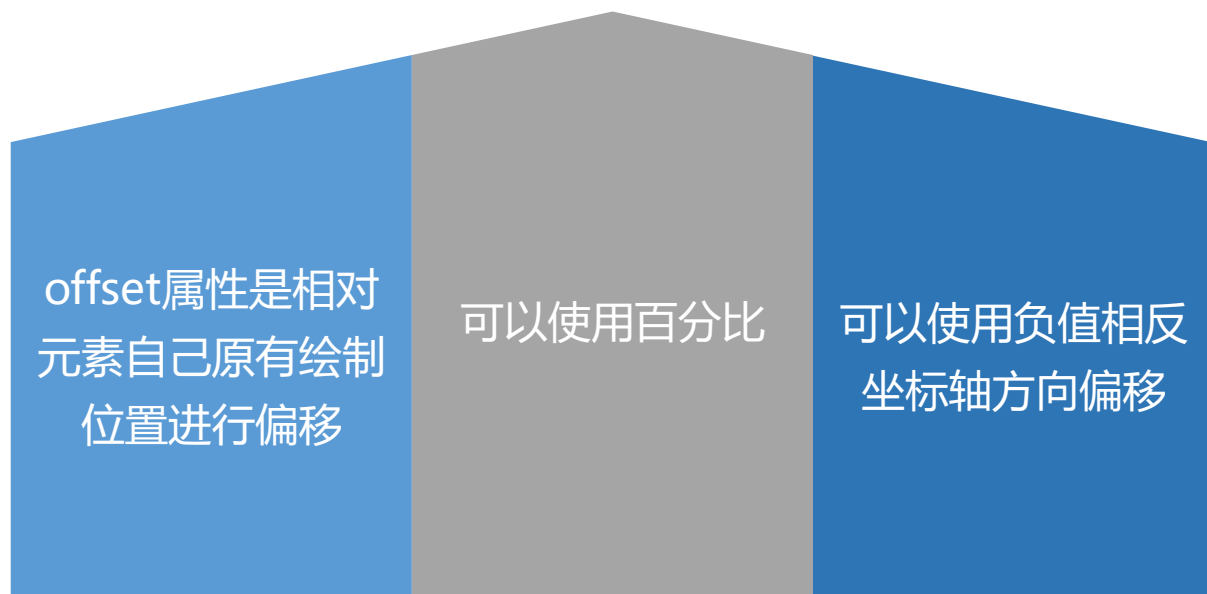


横屏

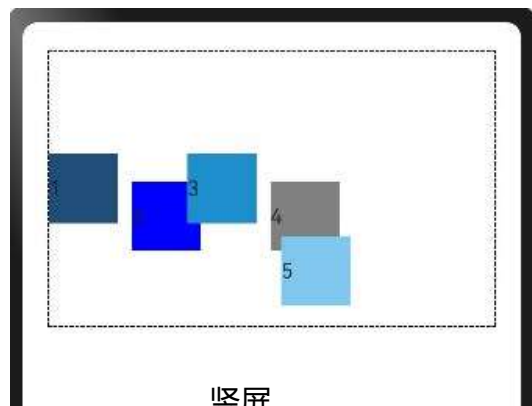
2.4 相对定位

务实创新 极致透明

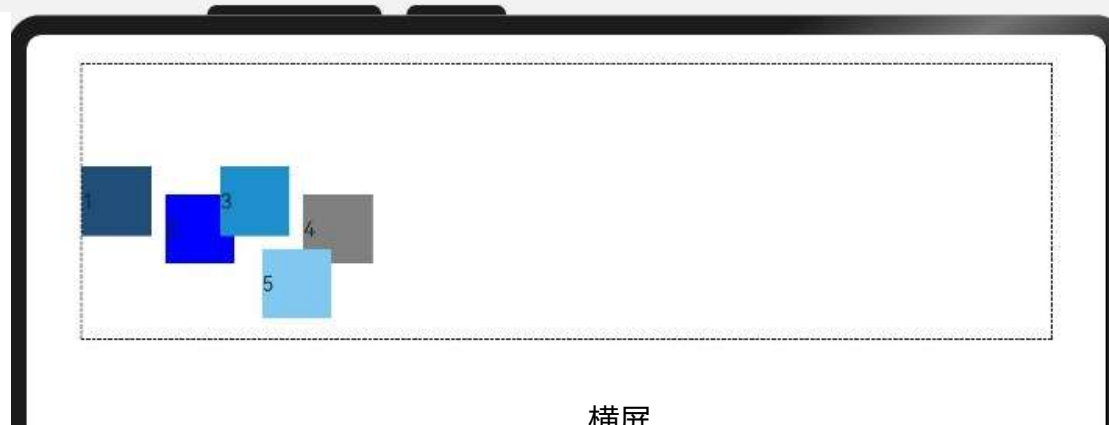
相对定位不影响父容器布局，仅在绘制时进行位置调整，使用组件offset属性实现元素相对于自身的偏移，通过线性布局+offset可以实现大多数布局
相对定位的特点如下：



```
build() {  
  Column({ space: 10 }) {  
    Row() {  
      Text('1').width(50).height(50).backgroundColor("#1F4E79").fontSize(15)  
      Text('2').fontSize(15).backgroundColor(Color.Blue).width(50).height(50).offset({ x: 10, y: 20 })  
      Text('3').fontSize(15).backgroundColor("#1D90CB").width(50).height(50)  
      Text('4').fontSize(15).backgroundColor(Color.Grey).width(50).height(50).offset({ x: 10, y: 20 })  
      Text('5').fontSize(15).backgroundColor("#80C8ED").width(50).height(50).offset({ x: '-10%', y: '30%' })  
    }.width('90%').height(200)  
    .border({ width: 1, style: BorderStyle.Dashed })  
  }.width('100%').margin({ top: 20 })  
}
```



竖屏



横屏

绝对定位

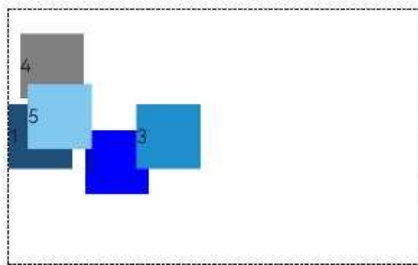
务实创新 极致透明

使用组件position属性实现绝对定位，绝对定位相对于父容器左上角进行定位偏移。对于不同设备绝对定位支撑较差，建议选择相对定位

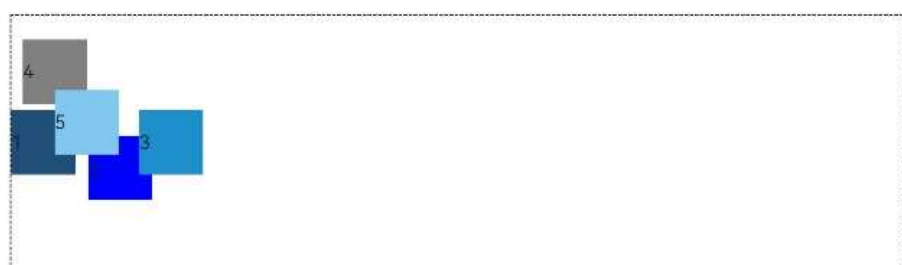
- 文本块1, 3正常位置
- 文本块2, 相对定位
- 文本块4, 5绝对定位

```
build() {  
  Column({ space: 10 }) {  
    Row() {  
      Text('1').width(50).height(50).backgroundColor("#1F4E79").fontSize(15)  
      Text('2').fontSize(15).backgroundColor(Color.Blue).width(50).height(50).offset({ x: 10, y: 20 })  
      Text('3').fontSize(15).backgroundColor("#1D90CB").width(50).height(50)  
      Text('4').fontSize(15).backgroundColor(Color.Grey).width(50).height(50).position({ x: 10, y: 20 })  
      Text('5').fontSize(15).backgroundColor("#80C8ED").width(50).height(50).position({ x: '5%', y: '30%' })  
    }.width('90%').height(200)  
    .border({ width: 1, style: BorderStyle.Dashed })  
  }.width('100%').margin({ top: 20 })  
}
```

LinearlayoutPositionDemo.ets



竖屏



横屏



03 弹性布局Flex

3.1 弹性布局介绍

3.2 自适应拉伸

3.1 弹性布局方向

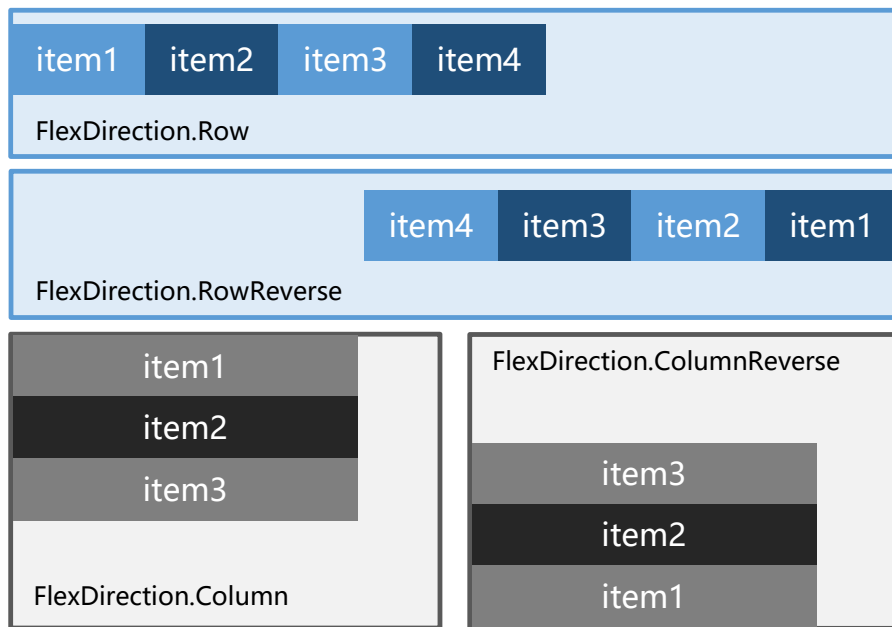
务实创新 极致透明

弹性布局（FlexLayout）：通过Flex组件设置子组件的排列、对齐、空白空间分配。容器默认存在主轴与交叉轴，子元素默认沿主轴排列。Flex接口如下：

```
Flex(options?: { direction?: FlexDirection, wrap?: FlexWrap, justifyContent?: FlexAlign, alignItems?: ItemAlign, alignContent?: FlexAlign })
```

direction控制主轴方向，即子组件排列方向：

- 1 Row（默认值，主轴为水平方向，子组件从左到右排列）
- 2 RowReverse（水平方向，从右到左）
- 3 Column（主轴为垂直方向，子元素从上到下排列）
- 4 ColumnReverse（垂直方向，从下到上）

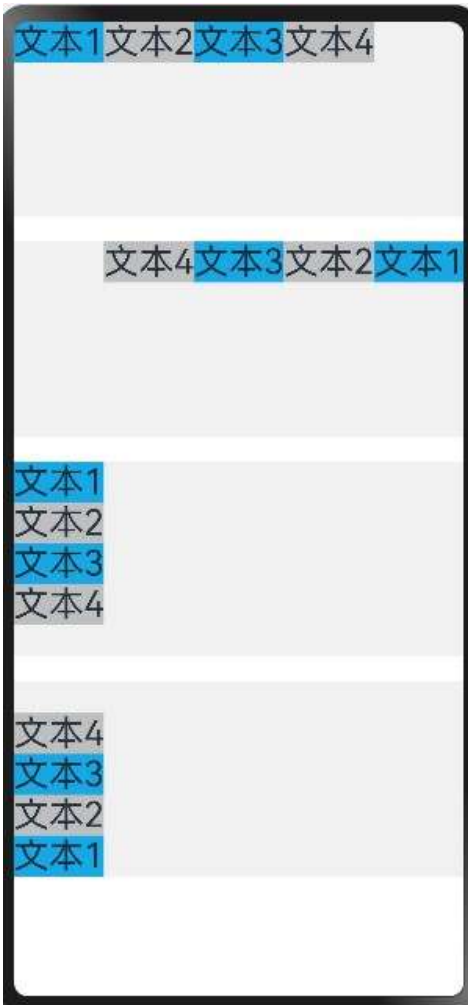


(1) 弹性布局方向

务实创新 极致透明

```
build() {  
  Column({space:20}){  
    // 水平方向, 从左往右  
    Flex({direction:FlexDirection.Row}){  
      Text('文本1').backgroundColor('#17A8E6').fontSize(28)  
      Text('文本2').backgroundColor('#BFBFBF').fontSize(28)  
      Text('文本3').backgroundColor('#17A8E6').fontSize(28)  
      Text('文本4').backgroundColor('#BFBFBF').fontSize(28)  
    }.width('100%').height('20%').backgroundColor('#f1f1f1')  
    // 水平方向, 从右往左  
    Flex({direction:FlexDirection.RowReverse}){  
      Text('文本1').backgroundColor('#17A8E6').fontSize(28)  
      Text('文本2').backgroundColor('#BFBFBF').fontSize(28)  
      Text('文本3').backgroundColor('#17A8E6').fontSize(28)  
      Text('文本4').backgroundColor('#BFBFBF').fontSize(28)  
    }.width('100%').height('20%').backgroundColor('#f1f1f1')  
    // 垂直方向, 从上到下  
    Flex({direction:FlexDirection.Column}){  
      Text('文本1').backgroundColor('#17A8E6').fontSize(28)  
      Text('文本2').backgroundColor('#BFBFBF').fontSize(28)  
      Text('文本3').backgroundColor('#17A8E6').fontSize(28)  
      Text('文本4').backgroundColor('#BFBFBF').fontSize(28)  
    }.width('100%').height('20%').backgroundColor('#f1f1f1')  
    // 垂直方向, 从下到上  
    Flex({direction:FlexDirection.ColumnReverse}){  
      Text('文本1').backgroundColor('#17A8E6').fontSize(28)  
      Text('文本2').backgroundColor('#BFBFBF').fontSize(28)  
      Text('文本3').backgroundColor('#17A8E6').fontSize(28)  
      Text('文本4').backgroundColor('#BFBFBF').fontSize(28)  
    }.width('100%').height('20%').backgroundColor('#f1f1f1')  
  }  
}
```

FlexLayoutDriection.ets



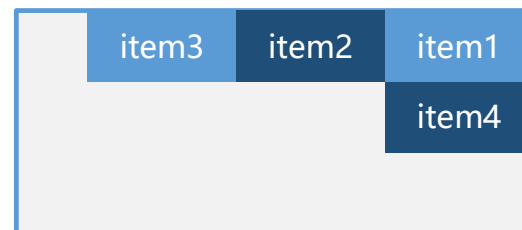
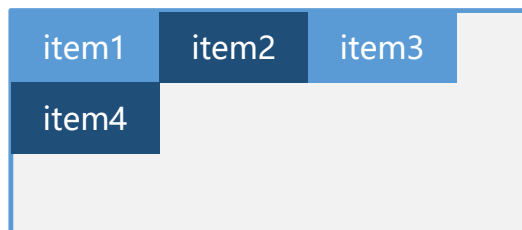
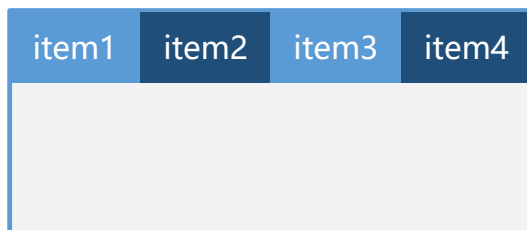
(2) 弹性布局换行

务实创新 极致透明

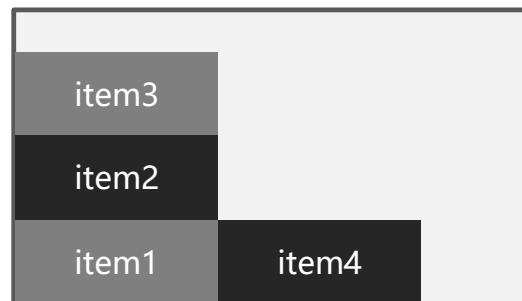
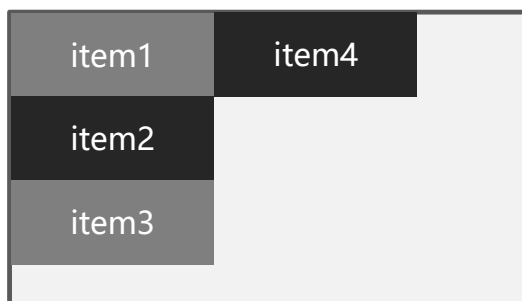
wrap参数控制子组件换行方式

1. FlexWrap.NoWrap: 默认值, 不换行, 子组件总和大于父容器的宽度, 子组件宽度将压缩。
2. FlexWrap.Wrap: 换行, 沿主轴方向进行换行
3. FlexWrap.WrapReverse: 换行, 沿主轴反向排列并换行

主轴为水平方向



主轴为垂直方向



FlexWrap.NoWrap

FlexWrap.Wrap

FlexWrap.WrapReverse

主轴为水平方向

FlexLayoutWrap1.ets

```
build() {  
    Column({space:20}){  
        // 不换行  
        Flex({wrap:FlexWrap.NoWrap}){  
            Text('文本1').backgroundColor('#17A8E6').fontSize(28).width('30%')  
            Text('文本2').backgroundColor('#BFBFBF').fontSize(28).width('30%')  
            Text('文本3').backgroundColor('#17A8E6').fontSize(28).width('30%')  
            Text('文本4').backgroundColor('#BFBFBF').fontSize(28).width('30%')  
        }.width('100%').height('20%').backgroundColor('#f1f1f1')  
  
        // 换行  
        Flex({wrap:FlexWrap.Wrap}){  
            Text('文本1').backgroundColor('#17A8E6').fontSize(28).width('30%')  
            Text('文本2').backgroundColor('#BFBFBF').fontSize(28).width('30%')  
            Text('文本3').backgroundColor('#17A8E6').fontSize(28).width('30%')  
            Text('文本4').backgroundColor('#BFBFBF').fontSize(28).width('30%')  
        }.width('100%').height('20%').backgroundColor('#f1f1f1')  
  
        // 换行并反向排列  
        Flex({wrap:FlexWrap.WrapReverse}){  
            Text('文本1').backgroundColor('#17A8E6').fontSize(28).width('30%')  
            Text('文本2').backgroundColor('#BFBFBF').fontSize(28).width('30%')  
            Text('文本3').backgroundColor('#17A8E6').fontSize(28).width('30%')  
            Text('文本4').backgroundColor('#BFBFBF').fontSize(28).width('30%')  
        }.width('100%').height('20%').backgroundColor('#f1f1f1')  
    }  
}
```

文本1 文本2 文本3 文本4

文本1 文本2 文本3
文本4

文本3 文本2 文本1
文本4

主轴为垂直方向

FlexLayoutWrap2.ets

```
build() {  
    Column({space:20}){  
        // 不换行  
        Flex({direction:FlexDirection.Column,wrap:FlexWrap.NoWrap}){  
            Text('文本1').backgroundColor('#17A8E6').fontSize(28).height('30%')  
            Text('文本2').backgroundColor('#BFBFBF').fontSize(28).height('30%')  
            Text('文本3').backgroundColor('#17A8E6').fontSize(28).height('30%')  
            Text('文本4').backgroundColor('#BFBFBF').fontSize(28).height('30%')  
        }.width('100%').height('20%').backgroundColor('#f1f1f1')  
  
        // 换行  
        Flex({direction:FlexDirection.Column,wrap:FlexWrap.Wrap}){  
            Text('文本1').backgroundColor('#17A8E6').fontSize(28).height('30%')  
            Text('文本2').backgroundColor('#BFBFBF').fontSize(28).height('30%')  
            Text('文本3').backgroundColor('#17A8E6').fontSize(28).height('30%')  
            Text('文本4').backgroundColor('#BFBFBF').fontSize(28).height('30%')  
        }.width('100%').height('20%').backgroundColor('#f1f1f1')  
  
        // 换行并反向排列  
        Flex({direction:FlexDirection.Column,wrap:FlexWrap.WrapReverse}){  
            Text('文本1').backgroundColor('#17A8E6').fontSize(28).height('30%')  
            Text('文本2').backgroundColor('#BFBFBF').fontSize(28).height('30%')  
            Text('文本3').backgroundColor('#17A8E6').fontSize(28).height('30%')  
            Text('文本4').backgroundColor('#BFBFBF').fontSize(28).height('30%')  
        }.width('100%').height('20%').backgroundColor('#f1f1f1')  
    }  
}
```

文本1
文本2
文本3
文本4

文本1 文本4
文本2
文本3

文本3
文本2
文本1 文本4

(3) justifyContent子组件在主轴的排列方式

务实创新 极致透明

justifyContent参数设置主轴对齐方式，有如下6种情况：（同线性布局的justifyContent）

- 1 FlexAlign.**Start**: 子组件整体在主轴方向**首端对齐**
- 2 FlexAlign.**Center**: 子组件整体在主轴方向**居中对齐**
- 3 FlexAlign.**End**: 子组件整体在主轴方向**尾部对齐**。
- 4 FlexAlign.**Spacebetween**: 子组件在主轴方向**两端对齐**，相邻子组件之间距离相同。
- 5 FlexAlign.**SpaceAround**: 子组件在主轴方向排列，**相邻子组件之间距离相同**。第一个子组件到主轴开始位置和最后一个子组件到主轴结束位置的距离是相邻子组件之间距离的一半。
- 6 FlexAlign.**SpaceEvenly**: 子组件在主轴方向均匀分配，相邻元素之间的距离、第一个子组件到主轴开始位置、最后一个子组件到主轴结束位置的**距离都一样**。

justifyContent—主轴为水平方向

务实创新 极致透明

```
build() {  
  Column({space:10}){  
    // 水平方向，左对齐  
    Flex({ justifyContent: FlexAlign.Start }) {  
      Text('文本1').backgroundColor('#17A8E6').fontSize(28)  
      Text('文本2').backgroundColor('#BFBFBF').fontSize(28)  
      Text('文本3').backgroundColor('#17A8E6').fontSize(28)  
    }.width('100%').height('15%').backgroundColor('#f1f1f1')  
    // 水平方向，居中对齐  
    Flex({ justifyContent: FlexAlign.Center }) {  
      Text('文本1').backgroundColor('#17A8E6').fontSize(28)  
      Text('文本2').backgroundColor('#BFBFBF').fontSize(28)  
      Text('文本3').backgroundColor('#17A8E6').fontSize(28)  
    }.width('100%').height('15%').backgroundColor('#f1f1f1')  
    // 水平方向，右对齐  
    Flex({ justifyContent: FlexAlign.End }) {  
      Text('文本1').backgroundColor('#17A8E6').fontSize(28)  
      Text('文本2').backgroundColor('#BFBFBF').fontSize(28)  
      Text('文本3').backgroundColor('#17A8E6').fontSize(28)  
    }.width('100%').height('15%').backgroundColor('#f1f1f1')  
  }  
}
```

JustifyContent1.ets

// 水平方向，两端对齐

```
Flex({ justifyContent: FlexAlign.SpaceBetween }) {  
  Text('文本1').backgroundColor('#17A8E6').fontSize(28)  
  Text('文本2').backgroundColor('#BFBFBF').fontSize(28)  
  Text('文本3').backgroundColor('#17A8E6').fontSize(28)  
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

// 水平方向，中间间距是两端间距的2倍

```
Flex({ justifyContent: FlexAlign.SpaceAround }) {  
  Text('文本1').backgroundColor('#17A8E6').fontSize(28)  
  Text('文本2').backgroundColor('#BFBFBF').fontSize(28)  
  Text('文本3').backgroundColor('#17A8E6').fontSize(28)  
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

// 水平方向，间距都相等

```
Flex({ justifyContent: FlexAlign.SpaceEvenly }) {  
  Text('文本1').backgroundColor('#17A8E6').fontSize(28)  
  Text('文本2').backgroundColor('#BFBFBF').fontSize(28)  
  Text('文本3').backgroundColor('#17A8E6').fontSize(28)  
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

文本1 文本2 文本3

文本1 文本2 文本3

文本1 文本2 文本3

文本1 文本2 文本3

文本1 文本2 文本3

文本1 文本2 文本3

justifyContent—主轴为垂直方向

务实创新 极致透明

JustifyContent2.ets

```
build() {  
    Column({space:10}){  
        // 垂直方向，顶部对齐  
        Flex({direction:FlexDirection.Column,justifyContent: FlexAlign.Start }) {  
            Text('文本1').backgroundColor('#17A8E6').fontSize(20)  
            Text('文本2').backgroundColor('#BFBFBF').fontSize(20)  
            Text('文本3').backgroundColor('#17A8E6').fontSize(20)  
        }.width('100%').height('15%').backgroundColor('#f1f1f1')  
        // 垂直方向，居中对齐  
        Flex({direction:FlexDirection.Column,justifyContent: FlexAlign.Center }) {  
            Text('文本1').backgroundColor('#17A8E6').fontSize(20)  
            Text('文本2').backgroundColor('#BFBFBF').fontSize(20)  
            Text('文本3').backgroundColor('#17A8E6').fontSize(20)  
        }.width('100%').height('15%').backgroundColor('#f1f1f1')  
        // 垂直方向，底部对齐  
        Flex({direction:FlexDirection.Column,justifyContent: FlexAlign.End }) {  
            Text('文本1').backgroundColor('#17A8E6').fontSize(20)  
            Text('文本2').backgroundColor('#BFBFBF').fontSize(20)  
            Text('文本3').backgroundColor('#17A8E6').fontSize(20)  
        }.width('100%').height('15%').backgroundColor('#f1f1f1')  
    }  
}
```

// 垂直方向，两端对齐

```
Flex({direction:FlexDirection.Column,justifyContent: FlexAlign.SpaceBetween }) {  
    Text('文本1').backgroundColor('#17A8E6').fontSize(20)  
    Text('文本2').backgroundColor('#BFBFBF').fontSize(20)  
    Text('文本3').backgroundColor('#17A8E6').fontSize(20)  
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

// 垂直方向，中间间距是两端间距的2倍

```
Flex({direction:FlexDirection.Column,justifyContent: FlexAlign.SpaceAround }) {  
    Text('文本1').backgroundColor('#17A8E6').fontSize(20)  
    Text('文本2').backgroundColor('#BFBFBF').fontSize(20)  
    Text('文本3').backgroundColor('#17A8E6').fontSize(20)  
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

// 垂直方向，间距都相等

```
Flex({direction:FlexDirection.Column,justifyContent: FlexAlign.SpaceEvenly }) {  
    Text('文本1').backgroundColor('#17A8E6').fontSize(20)  
    Text('文本2').backgroundColor('#BFBFBF').fontSize(20)  
    Text('文本3').backgroundColor('#17A8E6').fontSize(20)  
}.width('100%').height('15%').backgroundColor('#f1f1f1')  
}
```



(4) alignItems子组件在交叉轴的排列方式

务实创新 极致透明

容器组件可以设置交叉轴对齐，同时子组件自身也可以设置自己相对交叉轴的位置，而且优先级较高。容器组件通过Flex的alignItems参数设置子组件交叉轴对齐方式。子组件可以通过自己的alignSelf属性设置自己相对于交叉轴的对齐方式。

- 1 ItemAlign.Auto：使用Flex容器默认配置，Flex容器默认配置为ItemAlign.Start，即与交叉轴首部对齐
- 2 ItemAlign.Start：交叉轴首部对齐，Flex布局默认
- 3 ItemAlign.Center：交叉轴居中对齐
- 4 ItemAlign.End：交叉轴底部对齐
- 5 ItemAlign.Stretch：交叉轴拉伸填充，拉伸到与最大尺寸组件对齐
- 6 ItemAlign.Baseline：交叉轴方向，**文本**基线对齐

alignItems—交叉轴为垂直方向

务实创新 极致透明

```
build() {
```

AlignItems1.ets

```
Column({space:20}){
```

```
// 垂直方向，默认顶部对齐
```

```
Flex({alignItems:ItemAlign.Auto}){
```

```
Text('文本1').backgroundColor('#17A8E6').fontSize(28).height(40)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(28).height(50)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28).height(70)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
// 垂直方向，顶部对齐
```

```
Flex({alignItems:ItemAlign.Start}){
```

```
Text('文本1').backgroundColor('#17A8E6').fontSize(28).height(40)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(28).height(50)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28).height(70)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
// 垂直方向，居中对齐
```

```
Flex({alignItems:ItemAlign.Center}){
```

```
Text('文本1').backgroundColor('#17A8E6').fontSize(28).height(40)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(28).height(50)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28).height(70)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
// 垂直方向，底部对齐
```

```
Flex({alignItems:ItemAlign.End}){
```

```
Text('文本1').backgroundColor('#17A8E6').fontSize(28).height(40)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(28).height(50)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28).height(70)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
// 垂直方向，拉伸填充
```

```
Flex({alignItems:ItemAlign.Stretch}){
```

```
Text('文本1').backgroundColor('#17A8E6').fontSize(28).height(30)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(28).height(40)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28).height(50)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
// 文本基线对齐
```

```
Flex({alignItems:ItemAlign.Baseline}){
```

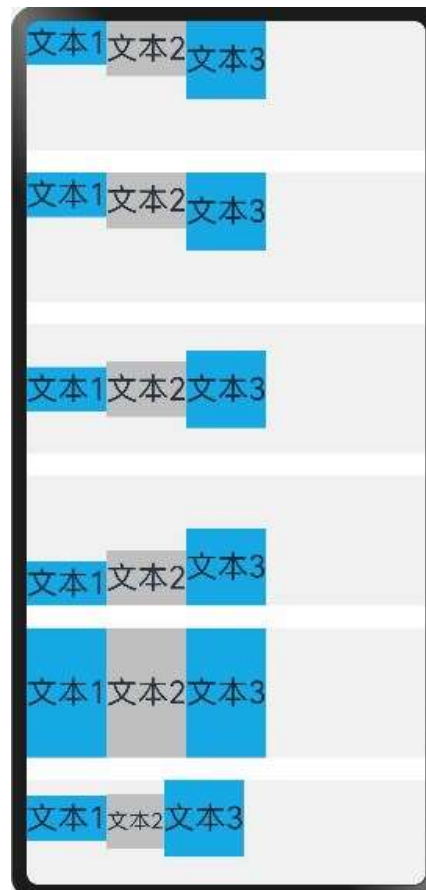
```
Text('文本1').backgroundColor('#17A8E6').fontSize(28).height(40)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(20).height(50)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28).height(70)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
}
```



alignItems—交叉轴为水平方向

务实创新 极致透明

```
build() {
```

```
Column({space:20}){
```

```
// 水平方向，默认左对齐
```

```
Flex({direction:FlexDirection.Column,alignItems:ItemAlign.Auto}){
```

```
Text('文本1').backgroundColor('#17A8E6').fontSize(28)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(28)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
// 水平方向，左对齐
```

```
Flex({direction:FlexDirection.Column,alignItems:ItemAlign.Start}){
```

```
Text('文本1').backgroundColor('#17A8E6').fontSize(28)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(28)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
// 水平方向，居中对齐
```

```
Flex({direction:FlexDirection.Column,alignItems:ItemAlign.Center}){
```

```
Text('文本1').backgroundColor('#17A8E6').fontSize(28)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(28)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

AlignItems2.ets

```
// 水平方向，右对齐
```

```
Flex({direction:FlexDirection.Column,alignItems:ItemAlign.End}){
```

```
Text('文本1').backgroundColor('#17A8E6').fontSize(28)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(28)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
// 水平方向，拉伸填充
```

```
Flex({direction:FlexDirection.Column,alignItems:ItemAlign.Stretch}){
```

```
Text('文本1').backgroundColor('#17A8E6').fontSize(28)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(28)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
// 文本基线对齐
```

```
Flex({direction:FlexDirection.Column,alignItems:ItemAlign.Baseline}){
```

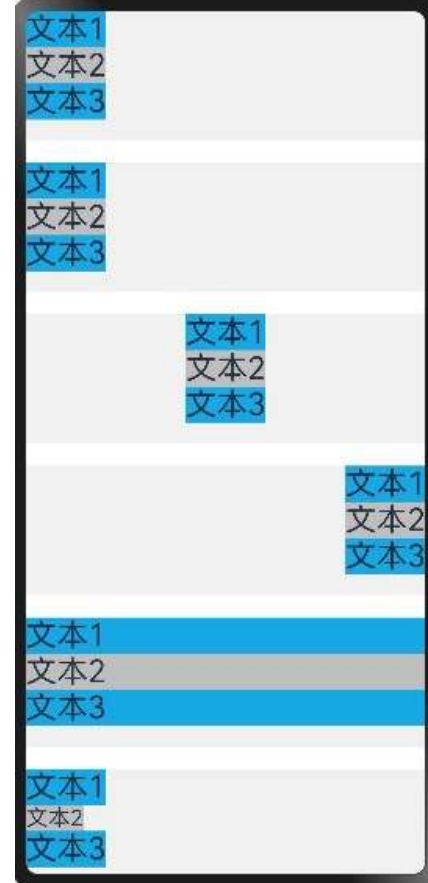
```
Text('文本1').backgroundColor('#17A8E6').fontSize(28)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(20)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
}
```



子组件自身设置交叉轴对齐: alignSelf属性

务实创新 极致透明

子组件alignSelf属性设置子组件在父容器交叉轴上的对齐方式，优先级高于Flex容器的alignItems配置

```
@Entry
@Component
struct AlignSelf {
  build() {
    Column() {
      Flex({ alignItems: ItemAlign.Center }) {
        Text('文本1').backgroundColor('#17A8E6').fontSize(28).alignSelf(ItemAlign.Start)
        Text('文本2').backgroundColor('#BFBFBF').fontSize(28)
        Text('文本3').backgroundColor('#17A8E6').fontSize(28).alignSelf(ItemAlign.Stretch)
      }.width('100%').height('15%').backgroundColor('#f1f1f1')
    }
  }
}
```

AlignSelf.ets



(5) alignContent多行文本的对齐方式

务实创新 极致透明

Flex组件的alignContent参数用于处理**多行内容**时，子组件在交叉轴方向上的对齐方式（仅在Flex容器中存在多行时有效）alignContent参数值：

- 1 FlexAlign.**Start**：子组件整体在交叉轴方向**首端对齐**
- 2 FlexAlign.**Center**：子组件整体在交叉轴方向**居中对齐**
- 3 FlexAlign.**End**：子组件整体在交叉轴方向**尾部对齐**。
- 4 FlexAlign.**Spacebetween**：子组件在交叉轴方向**两端对齐**，相邻子组件之间距离相同。
- 5 FlexAlign.**SpaceAround**：子组件在交叉轴方向排列，**相邻子组件之间距离相同**。第一个子组件到交叉轴开始位置和最后一个子组件到交叉轴结束位置的距离是相邻子组件之间距离的一半。
- 6 FlexAlign.**SpaceEvenly**：子组件在交叉轴方向均匀分配，相邻元素之间的距离、第一个子组件到交叉轴开始位置、最后一个子组件到交叉轴结束位置的**距离都一样**。

alignContent—交叉轴为垂直方向

务实创新 极致透明

```
build() {
```

```
Column({space:10}){
```

```
// 垂直方向，顶部对齐
```

```
Flex({wrap:FlexWrap.Wrap,alignContent:FlexAlign.Start}) {
```

```
Text('文本1').backgroundColor('#17A8E6').fontSize(28).width(80)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(28).width(150)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28).width(120)
```

```
Text('文本4').backgroundColor('#BFBFBF').fontSize(28).width(90)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
// 垂直方向，居中对齐
```

```
Flex({wrap:FlexWrap.Wrap,alignContent:FlexAlign.Center}) {
```

```
Text('文本1').backgroundColor('#17A8E6').fontSize(28).width(80)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(28).width(150)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28).width(120)
```

```
Text('文本4').backgroundColor('#BFBFBF').fontSize(28).width(90)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
// 垂直方向，底部对齐
```

```
Flex({wrap:FlexWrap.Wrap,alignContent:FlexAlign.End}) {
```

```
Text('文本1').backgroundColor('#17A8E6').fontSize(28).width(80)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(28).width(150)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28).width(120)
```

```
Text('文本4').backgroundColor('#BFBFBF').fontSize(28).width(90)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

AlignContent1.ets

```
// 垂直方向，两端对齐
```

```
Flex({wrap:FlexWrap.Wrap,alignContent:FlexAlign.SpaceBetween}) {
```

```
Text('文本1').backgroundColor('#17A8E6').fontSize(28).width(80)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(28).width(150)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28).width(120)
```

```
Text('文本4').backgroundColor('#BFBFBF').fontSize(28).width(90)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
// 垂直方向，中间间距是两端的2倍
```

```
Flex({wrap:FlexWrap.Wrap,alignContent:FlexAlign.SpaceAround}) {
```

```
Text('文本1').backgroundColor('#17A8E6').fontSize(28).width(80)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(28).width(150)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28).width(120)
```

```
Text('文本4').backgroundColor('#BFBFBF').fontSize(28).width(90)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
// 垂直方向，间距都相等
```

```
Flex({wrap:FlexWrap.Wrap,alignContent:FlexAlign.SpaceEvenly}) {
```

```
Text('文本1').backgroundColor('#17A8E6').fontSize(28).width(80)
```

```
Text('文本2').backgroundColor('#BFBFBF').fontSize(28).width(150)
```

```
Text('文本3').backgroundColor('#17A8E6').fontSize(28).width(120)
```

```
Text('文本4').backgroundColor('#BFBFBF').fontSize(28).width(90)
```

```
}.width('100%').height('15%').backgroundColor('#f1f1f1')
```

```
}
```

文本1 文本2 文本3
文本4

文本1 文本2 文本3
文本4

文本1 文本2 文本3
文本4

文本1 文本2 文本3

文本4

文本1 文本2 文本3

文本4

文本1 文本2 文本3

文本4

alignContent—交叉轴为水平方向

务实创新 极致透明

build() {

AlignContent2.ets

Column({space:10}){

// 水平方向，左对齐

Flex({direction:FlexDirection.Column,wrap:FlexWrap.Wrap,alignContent:FlexAlign.Start}) {

Text('文本1').backgroundColor('#17A8E6').fontSize(28)

Text('文本2').backgroundColor('#BFBFBF').fontSize(28)

Text('文本3').backgroundColor('#17A8E6').fontSize(28)

Text('文本4').backgroundColor('#BFBFBF').fontSize(28)

}.width('100%').height('15%').backgroundColor('#f1f1f1')

// 水平方向，居中对齐

Flex({direction:FlexDirection.Column,wrap:FlexWrap.Wrap,alignContent:FlexAlign.Center}) {

Text('文本1').backgroundColor('#17A8E6').fontSize(28)

Text('文本2').backgroundColor('#BFBFBF').fontSize(28)

Text('文本3').backgroundColor('#17A8E6').fontSize(28)

Text('文本4').backgroundColor('#BFBFBF').fontSize(28)

}.width('100%').height('15%').backgroundColor('#f1f1f1')

// 水平方向，右对齐

Flex({direction:FlexDirection.Column,wrap:FlexWrap.Wrap,alignContent:FlexAlign.End}) {

Text('文本1').backgroundColor('#17A8E6').fontSize(28)

Text('文本2').backgroundColor('#BFBFBF').fontSize(28)

Text('文本3').backgroundColor('#17A8E6').fontSize(28)

Text('文本4').backgroundColor('#BFBFBF').fontSize(28)

}.width('100%').height('15%').backgroundColor('#f1f1f1')

// 水平方向，两端对齐

Flex({direction:FlexDirection.Column,wrap:FlexWrap.Wrap,alignContent:FlexAlign.SpaceBetween}) {

Text('文本1').backgroundColor('#17A8E6').fontSize(28)

Text('文本2').backgroundColor('#BFBFBF').fontSize(28)

Text('文本3').backgroundColor('#17A8E6').fontSize(28)

Text('文本4').backgroundColor('#BFBFBF').fontSize(28)

}.width('100%').height('15%').backgroundColor('#f1f1f1')

// 水平方向，中间间距是两端间距的2倍

Flex({direction:FlexDirection.Column,wrap:FlexWrap.Wrap,alignContent:FlexAlign.SpaceAround}) {

Text('文本1').backgroundColor('#17A8E6').fontSize(28)

Text('文本2').backgroundColor('#BFBFBF').fontSize(28)

Text('文本3').backgroundColor('#17A8E6').fontSize(28)

Text('文本4').backgroundColor('#BFBFBF').fontSize(28)

}.width('100%').height('15%').backgroundColor('#f1f1f1')

// 水平方向，间距都相等

Flex({direction:FlexDirection.Column,wrap:FlexWrap.Wrap,alignContent:FlexAlign.SpaceEvenly}) {

Text('文本1').backgroundColor('#17A8E6').fontSize(28)

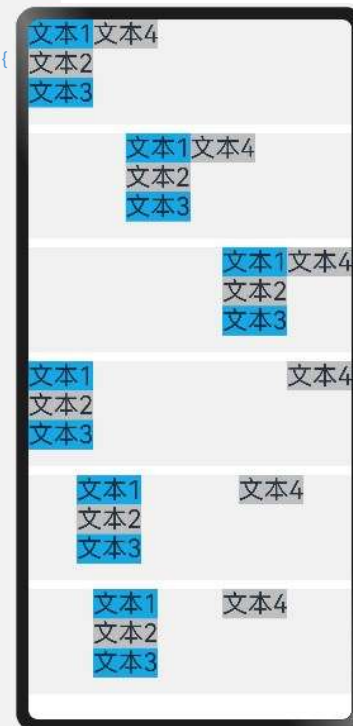
Text('文本2').backgroundColor('#BFBFBF').fontSize(28)

Text('文本3').backgroundColor('#17A8E6').fontSize(28)

Text('文本4').backgroundColor('#BFBFBF').fontSize(28)

}.width('100%').height('15%').backgroundColor('#f1f1f1')

}



3.2 自适应拉伸

务实创新 极致透明

在弹性布局父组件尺寸不够大的时候，通过子组件的下面几个属性设置其在父容器的占比，达到自适应布局能力。

01

flexBasis

设置子组件在父容器主轴方向上的基准尺寸。如果设置了该值，则子项占用的空间为设置的值，权重高于width/height的值；

02

flexGrow

定义子组件在有剩余空间的情况下是否放大，默认为0，不放大。不能为负值

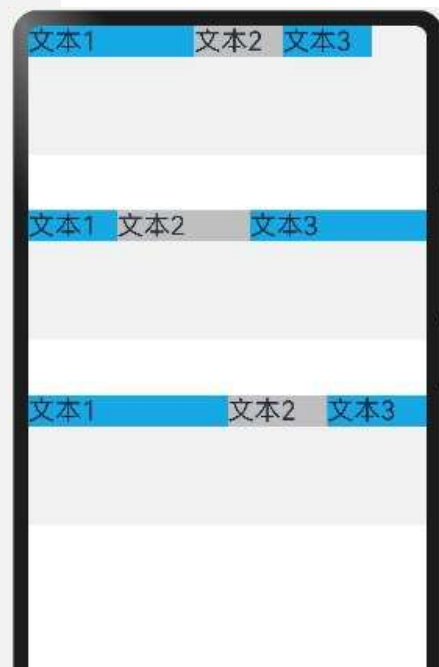
03

flexShrink

定义子组件在空间不足时是否缩小，默认为1，即空间不足时所有子组件等比缩小。当某个子组件的flex-shrink值为0时，即使空间不足也不缩小。

FlexDemo.ets

```
build() {  
  Column({space:50}) {  
    Flex() {  
      // 文本1的宽度应用flexBasis的值150  
      Text('文本1').backgroundColor('#17A8E6').fontSize(24).flexBasis(150).width(80)  
      Text('文本2').backgroundColor('#BFBFBF').fontSize(24).width(80)  
      Text('文本3').backgroundColor('#17A8E6').fontSize(24).width(80)  
    }.width('100%').height('15%').backgroundColor('#f1f1f1')  
    Flex() {  
      // 文本1不放大，宽仍为80;  
      // 文本2的总宽度为：(剩余空间÷3) × 1 + 80，(3为平分的份数1+2，该组件占3份中的1份，80为该组件原来的宽度)  
      // 文本3的总宽度为：(剩余空间÷3) × 2 + 80，(该组件占3份中的2份，80为该组件原来的宽度)  
      Text('文本1').backgroundColor('#17A8E6').fontSize(24).width(80).flexGrow(0)  
      Text('文本2').backgroundColor('#BFBFBF').fontSize(24).width(80).flexGrow(1)  
      Text('文本3').backgroundColor('#17A8E6').fontSize(24).width(80).flexGrow(2)  
    }.width('100%').height('15%').backgroundColor('#f1f1f1')  
    Flex() {  
      // 文本1不缩小，宽度仍为180  
      // 文本2和文本3等比缩小  
      Text('文本1').backgroundColor('#17A8E6').fontSize(24).width(180).flexShrink(0)  
      Text('文本2').backgroundColor('#BFBFBF').fontSize(24).width(180).flexShrink(1)  
      Text('文本3').backgroundColor('#17A8E6').fontSize(24).width(180).flexShrink(1)  
    }.width('100%').height('15%').backgroundColor('#f1f1f1')  
  }  
}
```





04 列表布局List



列表是一种复杂的容器，当列表项达到一定数量，内容超过屏幕大小时，可以**自动提供滚动功能**。列表布局常见于通讯录、音乐列表、购物清单等。



列表布局通过List容器组件**ListItemGroup**或**ListItem**子组件实现。其中ListItemGroup用于列表数据的**分组展示**，其子组件也是ListItem。ListItem表示**单个列表项**。在List组件中可以使用space参数在列表项之间添加间距。



List组件也存在两根轴，主轴和交叉轴。默认情况下，**主轴为垂直方向**。可通过listDirection属性改变主轴的方向，其值为：

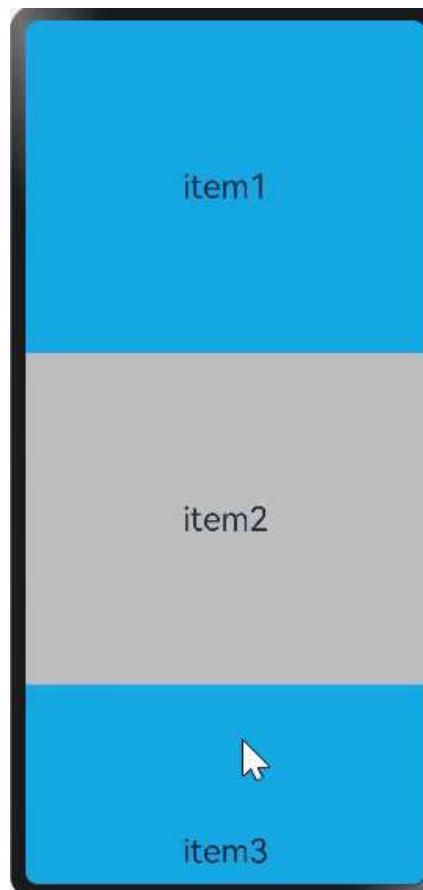
- Axis.Vertical：默认值，主轴为垂直方向
- Axis.Horizontal：主轴为水平方向

设置主轴方向

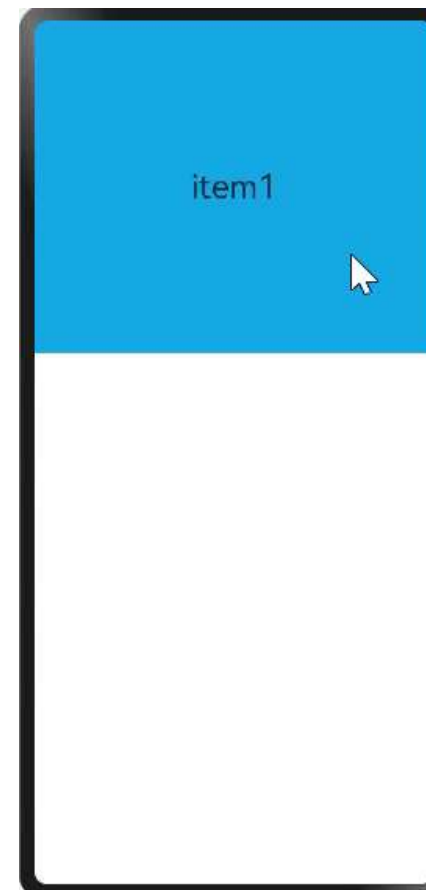
务实创新 极致透明

```
build() {  
  List(){  
    ListItem(){  
      Text('item1').fontSize(30)  
    }.backgroundColor('#17A8E6').width('100%').height(300)  
    ListItem(){  
      Text('item2').fontSize(30)  
    }.backgroundColor('#BFBFBF').width('100%').height(300)  
    ListItem(){  
      Text('item3').fontSize(30)  
    }.backgroundColor('#17A8E6').width('100%').height(300)  
    ListItem(){  
      Text('item4').fontSize(30)  
    }.backgroundColor('#BFBFBF').width('100%').height(300)  
    ListItem(){  
      Text('item5').fontSize(30)  
    }.backgroundColor('#17A8E6').width('100%').height(300)  
  }  
  // .listDirection(Axis.Horizontal) 设置主轴为水平方向  
}
```

ListLayoutDemo1.ets



主轴为垂直方向



主轴为水平方向

列表布局属性介绍

务实创新 极致透明

List组件不仅可以设置主轴的方向，还可以设置交叉轴的布局方式、添加分割线、滚动条等，这些效果通过以下属性来实现：

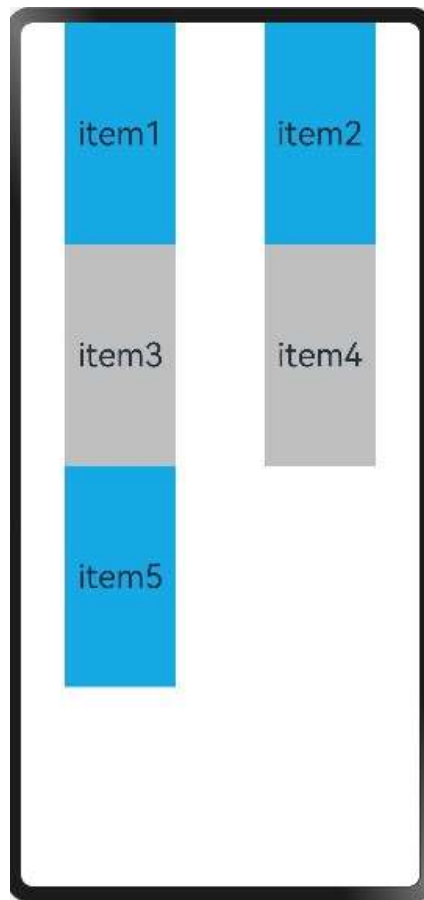
属性	描述
lanes	用于确定交叉轴排列的列表项数量，默认为1，即只有一列
alignListItem	alignListItem用于设置子组件在交叉轴方向的对齐方式。其值有： <ul style="list-style-type: none">ListItemAlign.Start：子组件与交叉轴首端对齐；ListItemAlign.Center：子组件与交叉轴居中对齐；ListItemAlign.End：子组件与交叉轴末端对齐。
divider	用于给列表项之间添加分隔线。在设置divider属性时，还可以通过以下参数来设置分割线的样式： <ul style="list-style-type: none">strokeWidth和color属性设置分隔线的粗细和颜色。startMargin属性设置分隔线距离列表侧边起始端的距离。endMargin属性设置分隔线距离列表侧边结束端的距离。
scrollBar	控制列表滚动条的显示。scrollBar的取值类型为BarState，取值有： <ul style="list-style-type: none">BarState.Auto表示按需显示滚动条。此时，当触摸到滚动条区域时显示控件，可上下拖拽滚动条快速浏览内容，拖拽时会变粗。若不进行任何操作，2秒后滚动条自动消失。BarState.On表示显示滚动条，滚动条一直存在BarState.Off表示不显示滚动条

(1) 设置交叉轴方向

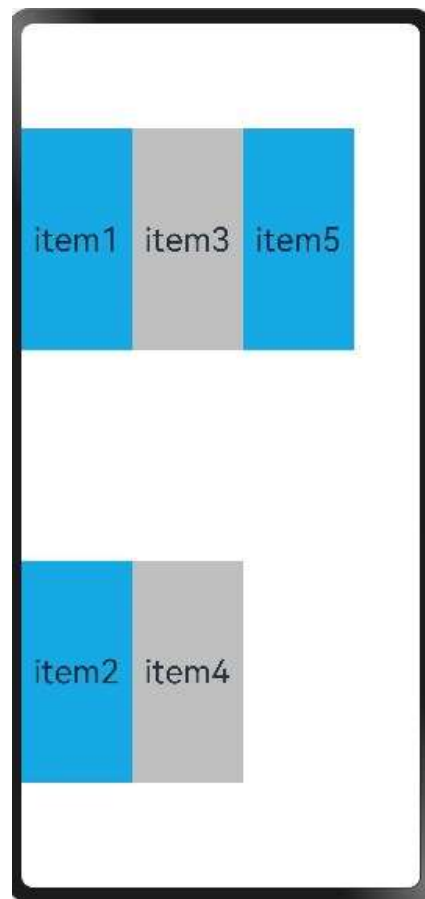
务实创新 极致透明

```
build() {  
  List(){  
    ListItem(){  
      Text('item1').fontSize(30)  
    }.backgroundColor('#17A8E6').width(100).height(200)  
    ListItem(){  
      Text('item2').fontSize(30)  
    }.backgroundColor('#17A8E6').width(100).height(200)  
    ListItem(){  
      Text('item3').fontSize(30)  
    }.backgroundColor('#BFBFBF').width(100).height(200)  
    ListItem(){  
      Text('item4').fontSize(30)  
    }.backgroundColor('#BFBFBF').width(100).height(200)  
    ListItem(){  
      Text('item5').fontSize(30)  
    }.backgroundColor('#17A8E6').width(100).height(200)  
  }  
  // .listDirection(Axis.Horizontal) 设置主轴为水平方向  
  .lanes(2) //设置在交叉轴方向有2列  
  .alignListItem(ListItemAlign.Center) //设置在交叉轴方向居中对齐  
}
```

ListLayoutDemo2.ets



交叉轴为水平方向



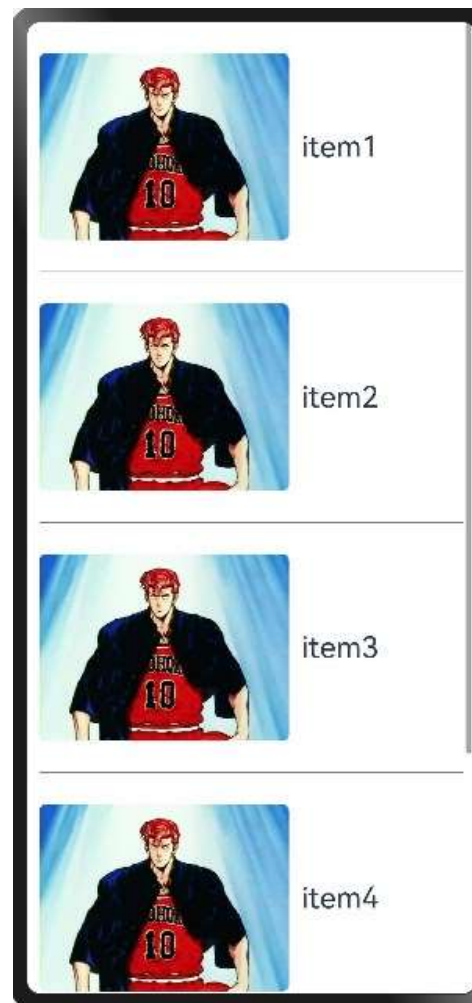
交叉轴为垂直方向

(2) 设置分割线和滚动条

务实创新 极致透明

```
build() {  
  List(){  
    ListItem(){  
      Row(){  
        Image($r('app.media.yingmu')).width(200).borderRadius(5).margin({left:10})  
        Text('item1').fontSize(24).margin({left:10})  
      }.justifyContent(FlexAlign.Start).width('100%').height(200)  
    }  
    ListItem(){  
      Row(){  
        Image($r('app.media.yingmu')).width(200).borderRadius(5).margin({left:10})  
        Text('item2').fontSize(24).margin({left:10})  
      }.justifyContent(FlexAlign.Start).width('100%').height(200)  
    }  
    .....  
  }  
  .divider({strokeWidth:1,color:Color.Gray,startMargin:10,endMargin:10})  
  .scrollBar(BarState.On)  
}
```

ListLayoutDemo3.ets



- ◆ Column布局的主轴在垂直方向，Row布局的主轴在水平方向；属性**justifyContent**控制子组件在主轴上的排列方式，为FlexAlign类型；属性**alignItems**控制子组件在交叉轴上的排列方式，在Column布局中该属性为HorizontalAlign类型，Row布局中该属性为VerticalAlign类型；
- ◆ Flex布局的参数有**direction**（控制主轴方向）、**wrap**（控制子组件换行方式）、**justifyContent**（子组件在主轴的排列方式）、**alignItems**（子组件在交叉轴的排列方式）、**alignContent**（多行文本的对齐方式）。子组件属性**flexBasis**（设置子组件在父容器主轴方向上的基准尺寸）、**flexGrow**（定义子组件在有剩余空间的情况下是否放大）、**flexShrink**（定义子组件在空间不足时是否缩小）
- ◆ List列表组件，当内容超过屏幕大小时，可以**自动提供滚动功能**。列表布局通过List容器组件**ListItemGroup**（列表数据的**分组展示**）或**ListItem**（**单个列表项**）子组件实现。