

Flutter布局Widget —— 线性布局

线性布局 可以将 子Widget 在同一个方向(水平和垂直)上排列.

Flutter 的线性布局，有两个：

1. [Row \(https://docs.flutter.io/flutter/widgets/Row-class.html\)](https://docs.flutter.io/flutter/widgets/Row-class.html)：水平方向的线性布局
2. [Column \(https://docs.flutter.io/flutter/widgets/Column-class.html\)](https://docs.flutter.io/flutter/widgets/Column-class.html)：垂直方向的线性布局

Row 和 Column 都继承自弹性布局 Flex，其实就是确定了主轴方向的 Flex，其余的用法和 Flex 一致。

Row

Row 可以在水平方向排列其 子widget。

代码所在位置

flutter_widget_demo/lib/linear/RowWidget.dart

Row 的使用

Row 是水平排列，给其 children 参数添加 子Widget 即可，例如：

```
Row(  
  children: <Widget>[  
    Text("Hello Flutter"),  
    Image.asset(  
      "images/flutter.png",  
      width: 200,  
    )  
  ],  
)
```

Row 在一个页面使用的 Demo 代码如下：

```

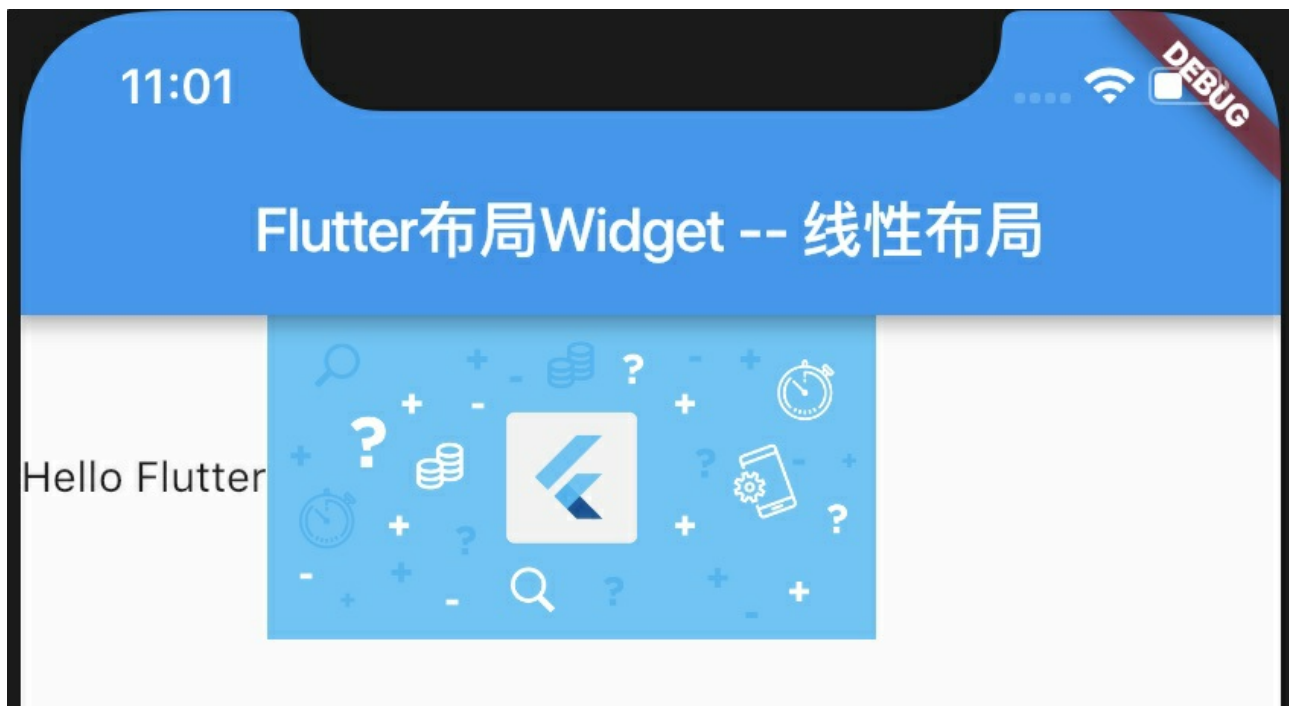
import 'package:flutter/material.dart';

void main() => runApp(RowWidget());

class RowWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: "Flutter Demo",
      theme: ThemeData(
        primaryColor: Colors.blue,
      ),
      home: Scaffold(
        appBar: AppBar(title: Text("Flutter布局
Widget -- 线性布局")),
        body: Row(
          children: <Widget>[
            Text("Hello Flutter"),
            Image.asset(
              "images/flutter.png",
              width: 200,
            ),
          ],
        ),
      ),
    );
  }
}

```

运行效果：



Row 的 children 有两个，分别是 Text 和 Image，这两个按照水平方向排列。

Row 的构造函数及参数说明

Row 的构造函数为：

```

class Row extends Flex {
  Row({
    Key key,
    MainAxisAlignment mainAxisAlignment =
MainAxisAlignment.start,
    MainAxisSize mainAxisSize = MainAxisSize.max,
    CrossAxisAlignment crossAxisAlignment =
CrossAxisAlignment.center,
    TextDirection textDirection,
    VerticalDirection verticalDirection =
VerticalDirection.down,
    TextBaseline textBaseline,
    List<Widget> children = const <Widget>[],
  }) : super(
    children: children,
    key: key,
    direction: Axis.horizontal,
    mainAxisAlignment: mainAxisAlignment,
    mainAxisSize: mainAxisSize,
    crossAxisAlignment: crossAxisAlignment,
    textDirection: textDirection,
    verticalDirection: verticalDirection,
    textBaseline: textBaseline,
  );
}

```

参数名字	参数类型	意义	必选 or 可选
key	Key	Widget 的标识 表示 子Widget	可选
mainAxisAlignment	MainAxisAlignment	在主轴的对齐方 式	可选

mainAxisSize	MainAxisSize	表示主轴应该占用多大的空间	可选
crossAxisAlignment	CrossAxisAlignment	表示 子Widget 在交叉轴的对齐方式	可选
textDirection	TextDirection	表示 子Widget 在主轴方向上的布局顺序	可选
verticalDirection	VerticalDirection	表示 子Widget 在交叉轴方向上的布局顺序	可选
textBaseline	TextBaseline	排列 子Widget 时使用哪个基线	可选
children	List< Widget>	Flex布局 里排列的内容	可选

Column

Column 可以在垂直方向上排列其 子widget。

代码所在位置

flutter_widget_demo/lib/linear/ColumnWidget.dart

Column 的快速上手

Column 是垂直方向上排列，给其 children 参数添加 子Widget 即可，例如：

```
Column(  
  children: <Widget>[  
    Text("Hello Flutter"),  
    Image.asset(  
      "images/flutter.png",  
      width: 200,  
    )  
  ],  
)
```

Column 在一个页面使用的 Demo 代码如下：

```

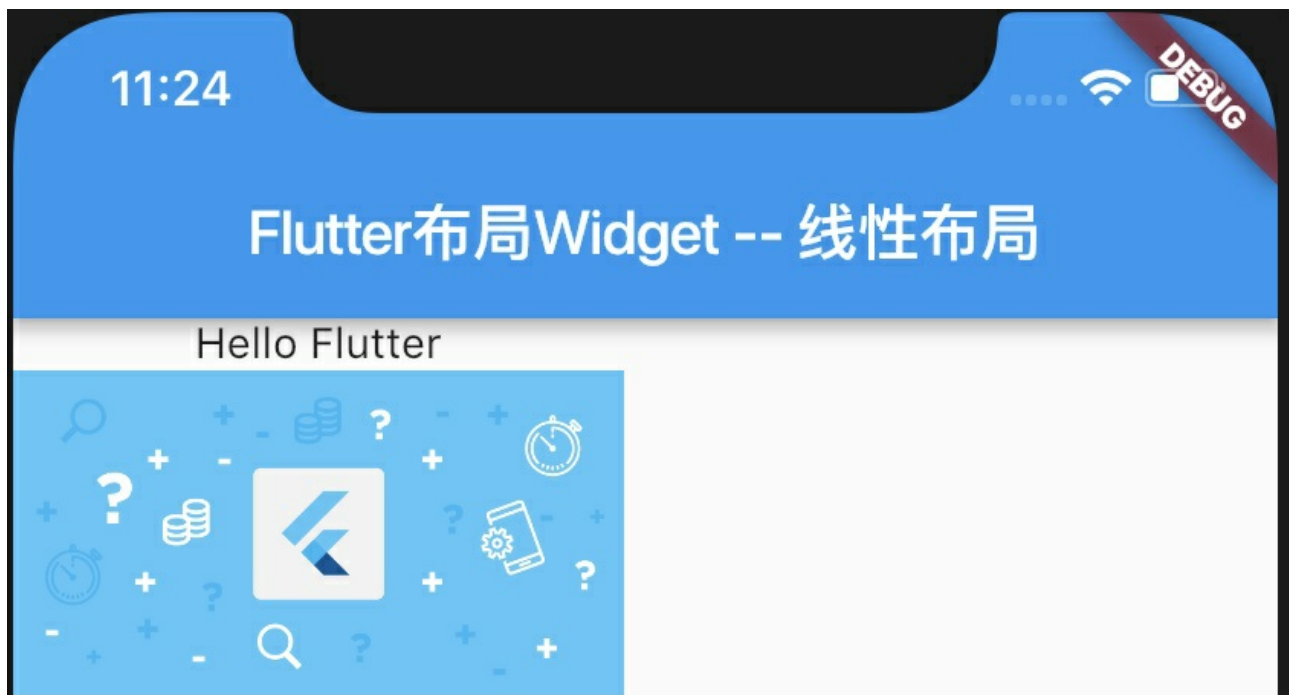
import 'package:flutter/material.dart';

void main() => runApp(ColumnWidget());

class ColumnWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: "Flutter Demo",
      theme: ThemeData(
        primaryColor: Colors.blue,
      ),
      home: Scaffold(
        appBar: AppBar(title: Text("Flutter布局Widget -- 线性布局")),
        body: Column(
          children: <Widget>[
            Text("Hello Flutter"),
            Image.asset(
              "images/flutter.png",
              width: 200,
            ),
          ],
        ),
      ),
    );
  }
}

```

运行结果：



Column 的构造函数及参数说明

Column 的构造函数为：

```

class Column extends Flex {
  Column({
    Key key,
    MainAxisAlignment mainAxisAlignment =
MainAxisAlignment.start,
    MainAxisSize mainAxisSize = MainAxisSize.max,
    CrossAxisAlignment crossAxisAlignment =
CrossAxisAlignment.center,
    TextDirection textDirection,
    VerticalDirection verticalDirection =
VerticalDirection.down,
    TextBaseline textBaseline,
    List<Widget> children = const <Widget>[],
  }) : super(
    children: children,
    key: key,
    direction: Axis.vertical,
    mainAxisAlignment: mainAxisAlignment,
    mainAxisSize: mainAxisSize,
    crossAxisAlignment: crossAxisAlignment,
    textDirection: textDirection,
    verticalDirection: verticalDirection,
    textBaseline: textBaseline,
  );
}

```

参数名字	参数类型	意义	必选 or 可选
key	Key	Widget 的标识 表示 子Widget	可选
mainAxisAlignment	MainAxisAlignment	在主轴的对齐方 式	可选

mainAxisSize	MainAxisSize	表示主轴应该占用多大的空间	可选
crossAxisAlignment	CrossAxisAlignment	表示 子Widget 在交叉轴的对齐方式	可选
textDirection	TextDirection	表示 子Widget 在主轴方向上的布局顺序	可选
verticalDirection	VerticalDirection	表示 子Widget 在交叉轴方向上的布局顺序	可选
textBaseline	TextBaseline	排列 子Widget 时使用哪个基线	可选
children	List< Widget>	Flex布局 里排列的内容	可选

特殊情况

如果 Row 里面嵌套 Row，或者 Column 里面再嵌套 Column，那么只有最外面的 Row 或 Column 会占用尽可能大的空间，里面 Row 或 Column 所占用的空间为实际大小。