

flutter 页面布局 Paddiing Row Column Expanded 组件详解

主讲教师：（大地）

合作网站：www.itying.com （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html> （IT 营）

一、Flutter Paddiing 组件.....	1
二、Flutter Row 水平布局组件.....	3
三、Flutter Column 垂直布局组件.....	6
四、Flutter Expanded 类似 Web 中的 Flex 布局.....	8
五、Flutter 实现类似下面的布局.....	11
六、Vscode 调试 Flutter 项目.....	11

一、Flutter Paddiing 组件

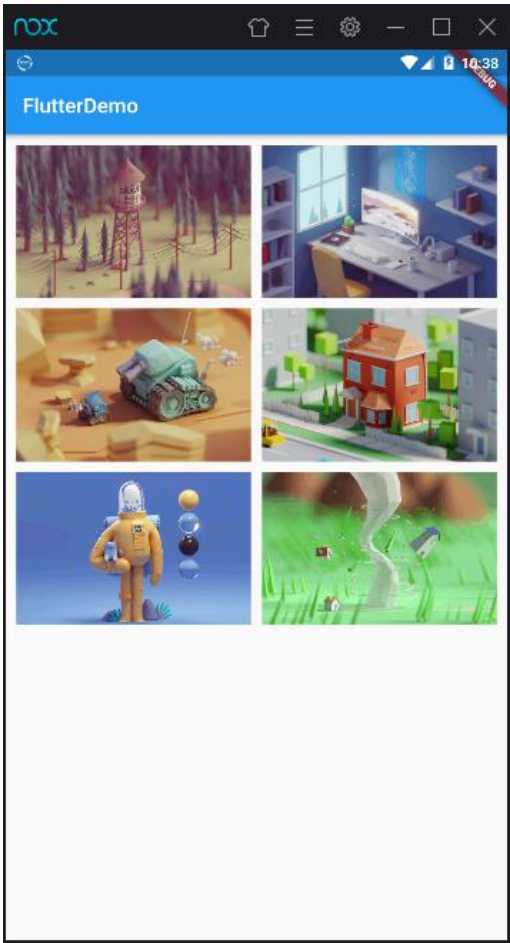
在 html 中常见的布局标签都有 padding 属性，但是 Flutter 中很多 Widget 是没有 padding 属性。这个时候我们可以用 Padding 组件处理容器与子元素直接的间距。

属性	说明
padding	padding 值, EdgeInsetsss 设置填充的值
child	子组件

```
class LayoutDemo extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Padding(  
      padding: EdgeInsetsss.fromLTRB(0, 0, 10, 0),  
      child: GridView.count(  

```

```
crossAxisCount: 2,  
childAspectRatio: 1.5,  
children: <Widget>[  
  Padding(  
    padding: EdgeInsets.fromLTRB(10, 10, 0, 0),  
    child: Image.network('https://www.itying.com/images/flutter/1.png',  
      fit: BoxFit.cover),  
  ),  
  Padding(  
    padding: EdgeInsets.fromLTRB(10, 10, 0, 0),  
    child: Image.network('https://www.itying.com/images/flutter/2.png',  
      fit: BoxFit.cover),  
  ),  
  Padding(  
    padding: EdgeInsets.fromLTRB(10, 10, 0, 0),  
    child: Image.network('https://www.itying.com/images/flutter/3.png',  
      fit: BoxFit.cover),  
  ),  
  Padding(  
    padding: EdgeInsets.fromLTRB(10, 10, 0, 0),  
    child: Image.network('https://www.itying.com/images/flutter/4.png',  
      fit: BoxFit.cover),  
  ),  
  Padding(  
    padding: EdgeInsets.fromLTRB(10, 10, 0, 0),  
    child: Image.network('https://www.itying.com/images/flutter/5.png',  
      fit: BoxFit.cover),  
  ),  
  Padding(  
    padding: EdgeInsets.fromLTRB(10, 10, 0, 0),  
    child: Image.network('https://www.itying.com/images/flutter/6.png',  
      fit: BoxFit.cover),  
  ),  
],  
,  
);  
}
```



二、Flutter Row 水平布局组件

属性	说明
mainAxisAlignment	主轴的排序方式
crossAxisAlignment	次轴的排序方式
children	组件子元素

```
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    // TODO: implement build
    return MaterialApp(
      home: Scaffold(
```

```
        appBar: AppBar(title: Text('FlutterDemo')),
        body: LayoutDemo(),
    ));
}
}

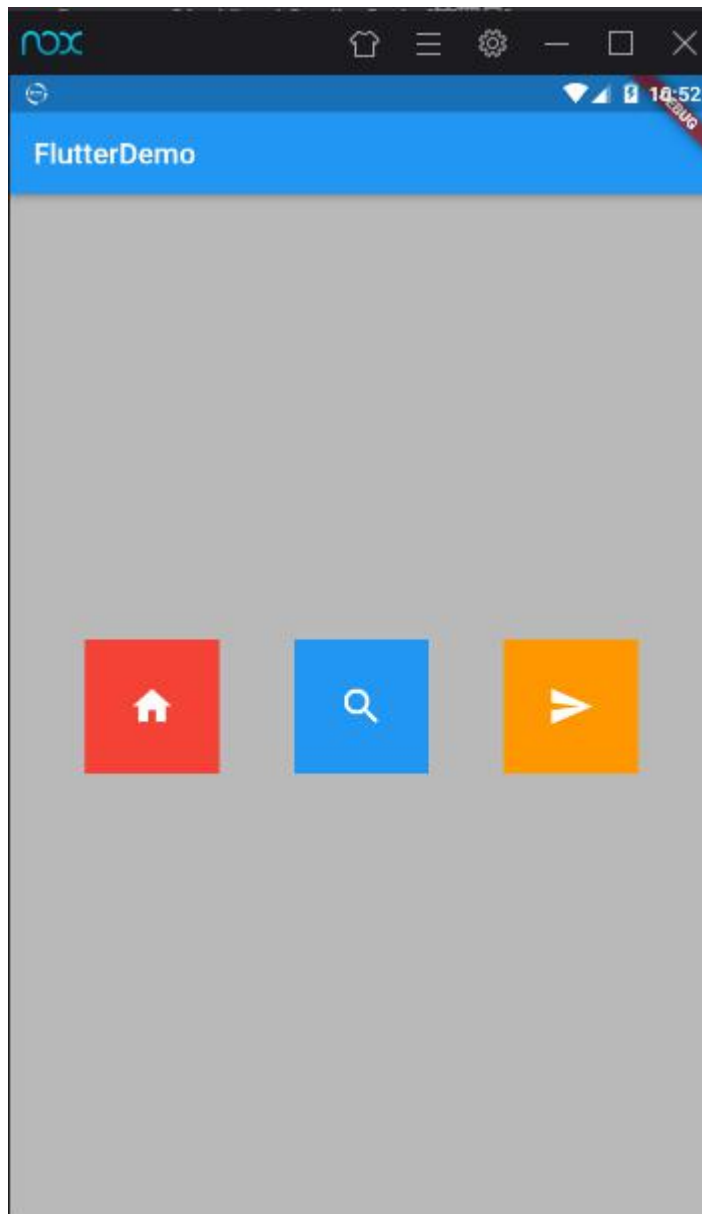
class LayoutDemo extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Container(
      height: 700,
      width: 500,
      color: Colors.black26,
      child: Row(
        crossAxisAlignment: CrossAxisAlignment.center,
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        // crossAxisAlignment: CrossAxisAlignment.center,
        children: <Widget>[
          IconContainer(Icons.home,color:Colors.red),
          IconContainer(Icons.search,color:Colors.blue),
          IconContainer(Icons.send,color:Colors.orange),
        ],
      ),
    );
  }
}

class IconContainer extends StatelessWidget{

  double size;
  IconData icon;
  Color color;

  IconContainer(this.icon,{this.size,this.color=Colors.blue}){
    this.size=32.0;
  }
  @override
  Widget build(BuildContext context) {
    // TODO: implement build
    return Container(
      width: this.size+60,
      height: this.size+60,
```

```
        color:this.color,  
        child: Center(child: Icon(this.icon,color:Colors.white,size:this.size))  
      );  
    }  
  }  
}
```



三、Flutter Column 垂直布局组件

属性	说明
mainAxisAlignment	主轴的排序方式
crossAxisAlignment	次轴的排序方式
children	组件子元素

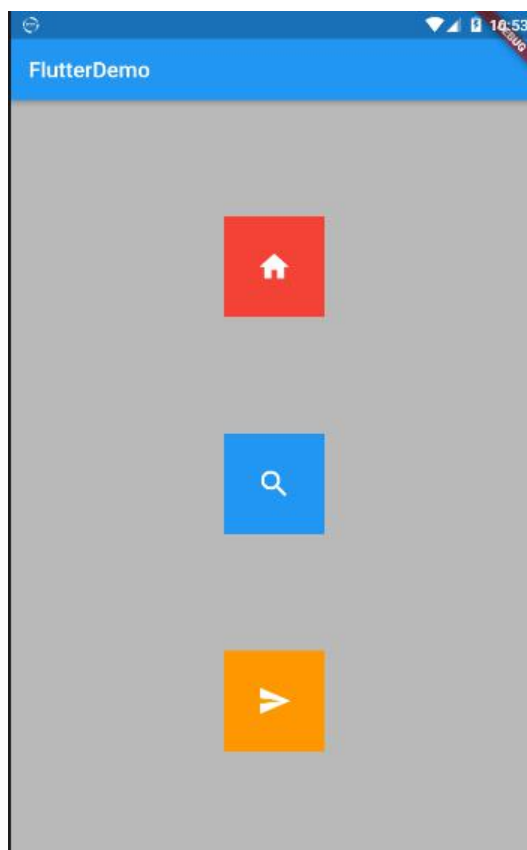
```
import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    // TODO: implement build
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text('FlutterDemo')),
        body: LayoutDemo(),
      ));
  }
}

class LayoutDemo extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Container(
      height: 700,
      width: 500,
      color: Colors.black26,
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.center,
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        // crossAxisAlignment: CrossAxisAlignment.center,
        children: <Widget>[
          IconContainer(Icons.home,color:Colors.red),
          IconContainer(Icons.search,color:Colors.blue),
          IconContainer(Icons.send,color:Colors.orange),
        ],
      ),
    );
  }
}
```

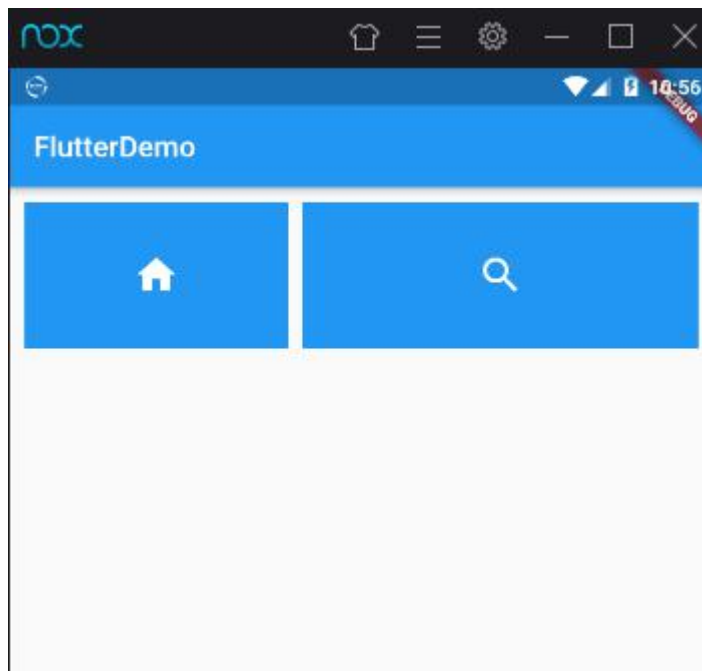
```
    ),  
    );  
  
    }  
}  
  
class IconContainer extends StatelessWidget{  
  
    double size;  
    IconData icon;  
    Color color;  
  
    IconContainer(this.icon,{this.size,this.color=Colors.blue}){  
        this.size=32.0;  
    }  
    @override  
    Widget build(BuildContext context) {  
        // TODO: implement build  
        return Container(  
            width: this.size+60,  
            height: this.size+60,  
            color:this.color,  
            child: Center(child: Icon(this.icon,color:Colors.white,size:this.size))  
        );  
    }  
}
```



四、Flutter Expanded 类似 Web 中的 Flex 布局

Expanded 可以用在 Row 和 Column 布局中

属性	说明
flex	元素占整个父 Row /Column 的比例
child	子元素



```
import 'package:flutter/material.dart';

import 'res/listData.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    // TODO: implement build
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(title: Text('FlutterDemo')),
        body: LayoutDemo(),
      ));
  }
}

class LayoutDemo extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: EdgeInsets.all(10),
      child: Row(
        // crossAxisAlignment: CrossAxisAlignment.start,
        mainAxisAlignment: MainAxisAlignment.center,
```

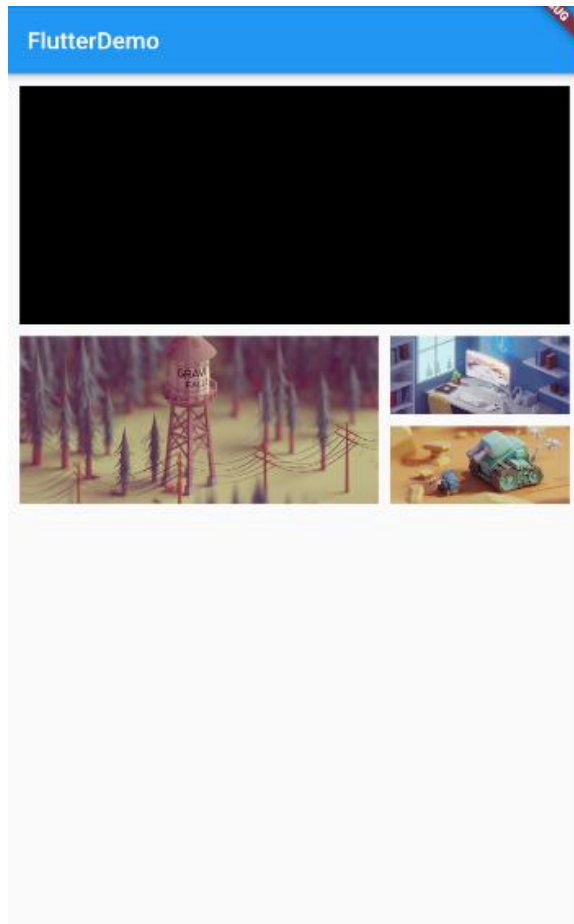
```
// crossAxisAlignment: CrossAxisAlignment.center,
children: <Widget>[
  Expanded(flex:2,child: IconContainer(Icons.home)),
  SizedBox(width: 10),
  Expanded(flex: 3,child: IconContainer(Icons.search)),
  // SizedBox(width: 10),
  // Expanded(child: IconContainer(Icons.send))
],
),
)

;
}
}

class IconContainer extends StatelessWidget{

  double size;
  IconData icon;
  IconContainer(this.icon,{this.size}){
    this.size=32.0;
  }
  @override
  Widget build(BuildContext context) {
    // TODO: implement build
    return Container(
      width: 100.0,
      height: 100.0,
      color: Colors.blue,
      child: Center(child: Icon(this.icon,color:Colors.white,size:this.size))
    );
  }
}
```

五、Flutter 实现类似下面的布局



六、Vscode 调试 Flutter 项目

1、Vscode 中打开 flutter 项目进行开发

2、运行 Flutter 项目

```
flutter run
```

r 键: 点击后热加载，也就算是重新加载吧。

p 键: 显示网格，这个可以很好的掌握布局情况，工作中很有用。

- o 键: 切换 android 和 ios 的预览模式。
- q 键: 退出调试预览模式。

3、Vscode 默认连不上第三方模拟器解决方案

cd 到对应夜神模拟器 D:\Program Files\Nox\bin 目录 然后运行

```
nox_adb.exe connect 127.0.0.1:62001
```