

# 一起来学 Flutter 之图片组件、列表组件（二）

作者: [santa9527](#)

原文链接: <https://ld246.com/article/1568734836684>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



前面学习了Flutter的文本组件和容器组件，今天来熟悉下图片组件和列表组件。

## 1. 图片组件 (Image Widget)

首先，我们应该明确Flutter中图片载入的几种方式：

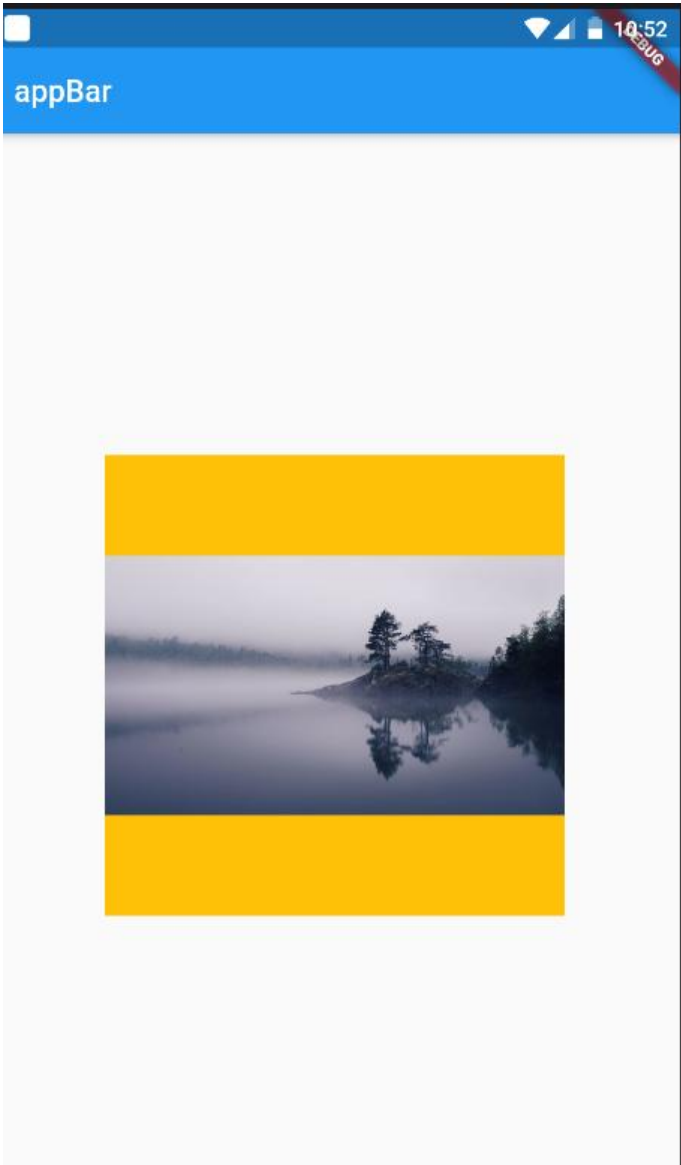
1. Image.network : 网络图片，通常后面跟网络图片地址；
2. Image.file : 加载本地图片，通常后面跟图片的绝对地址；
3. Image.asset : 加载资源图片，通常后面跟图片的相对路径。

下面就以网络图片为例，在容器组件中加入一张图片：

```
import 'package:flutter/material.dart';  
void main() => runApp(MyApp());
```

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    // TODO: implement build  
    return MaterialApp(  
      title: "测试app",  
      home: Scaffold(  
        appBar: AppBar(title: Text("appBar")),  
        body: Center(  
          child: Container(  
            child: Image.network("https://b3logfile.com/bing/20180221.jpg?imageView2/1/w/960  
h/540/interlace/1/q/100",scale: 1.0),  
            alignment: Alignment.center,  
            width: 300.0,  
            height: 300.0,  
            //child: Text("测试"),
```

```
        color: Colors.amber,  
      ),),),);  
    }  
  }
```



接下来就是图片组件的几个常用属性：

## fit属性

- **BoxFit.fill**: 全图显示，图片会被拉伸，并充满父容器。
- **BoxFit.contain**: 全图显示，显示原比例，可能会有空隙。
- **BoxFit.cover**: 显示可能拉伸，可能裁切，充满（图片要充满整个容器，还不变形）。
- **BoxFit.fitWidth**: 宽度充满（横向充满），显示可能拉伸，可能裁切。
- **BoxFit.fitHeight**: 高度充满（竖向充满），显示可能拉伸，可能裁切。
- **BoxFit.scaleDown**: 效果和contain差不多，但是此属性不允许显示超过源图片大小，可小不可大。

## 图片混合模式

- color: 是要混合的颜色。
- colorBlendMode: 是混合模式。

这里的组合比较多，大家可以自行尝试。

```
child: Container(  
  child: Image.network("https://b3logfile.com/bing/20180221.jpg?imageView2/1/w/960  
h/540/interlace/1/q/100",scale: 1.0,  
  fit: BoxFit.cover,  
  color: Colors.blue,  
  colorBlendMode: BlendMode.darken,),  
  alignment: Alignment.center,  
  width: 300.0,  
  height: 300.0,  
  color: Colors.amber,  
),
```



## repeat图片重复属性

- **ImageRepeat.repeat**: 横向和纵向都进行重复，直到铺满整个画布。
- **ImageRepeat.repeatX**: 横向重复，纵向不重复。
- **ImageRepeat.repeatY**: 纵向重复，横向不重复。

```
child: Container(  
  child: Image.network("http://bgweb.yundatainfo.com/uploadFiles/uploadImgs/20190  
10/f81cb770dd004658bddbf57c7d2ed965.jpg",scale: 1.0,  
  repeat: ImageRepeat.repeatY,),  
  alignment: Alignment.center,  
  width: 500.0,  
  height: 500.0,  
  color: Colors.amber,
```

以上就是图片组件及其属性。

## 2.列表组件

首先，我们申明一个简单的列表组件：

```
class MyApp extends StatelessWidget{
  @override
  Widget build(BuildContext context) {
    // TODO: implement build
    return MaterialApp(
      title: "测试app",
      home: Scaffold(
        appBar: AppBar(title: Text("appBar")),
        body: ListView(
          children: <Widget>[
            ListTile(
              leading: Icon(Icons.access_time),
              title: Text("时间"),
            ),
            ListTile(
              leading: Icon(Icons.access_alarms),
              title: Text("提醒"),
            ),
            ListTile(
              leading: Icon(Icons.search),
              title: Text("搜索"),
            ),
          ],), );
  }
}
```



如代码，静态列表的使用非常简单，而且Flutter内置了相当多的Icon，十分精美，同样的，大家也可把列表中的ListTile元素换成上面刚提到的图片组件，有兴趣的可以自己尝试。

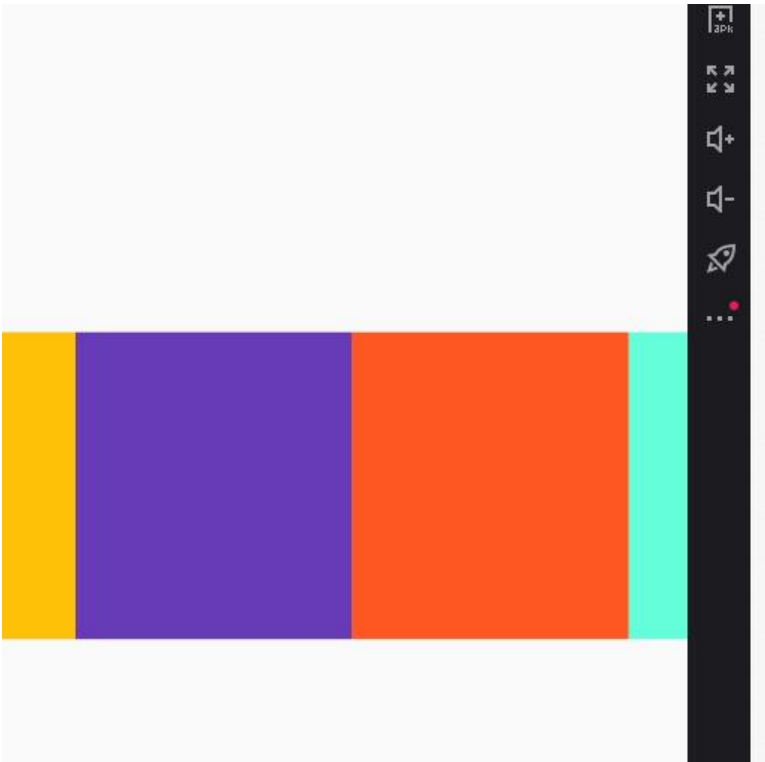
## 横向滚动属性

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

```

return MaterialApp(
  title: "测试app",
  home: Scaffold(
    appBar: AppBar(title: Text("appBar")),
    body: Center(
      child: Container(
        height: 200,
        child: ListView(
          scrollDirection: Axis.horizontal,
          children: <Widget>[
            Container(
              width:180.0,
              color: Colors.amber,
            ),
            Container(
              width:180.0,
              color: Colors.deepPurple,
            ),
            Container(
              width:180.0,
              color: Colors.deepOrange,
            ),
            Container(
              width:180.0,
              color: Colors.tealAccent,
            ),
          ],), ), );
}
}

```



ListView组件的`scrollDirection`属性只有两个值，一个是横向滚动，一个是纵向滚动。默认的就是垂直滚动，所以如果是垂直滚动，我们一般都不进行设置。

- `Axis.horizontal`:横向滚动或者叫水平方向滚动。

- `Axis.vertical`:纵向滚动或者叫垂直方向滚动。

以上就是静态列表的基本用法和属性。