

# Scrolling Widget

2021年7月1日 9:15

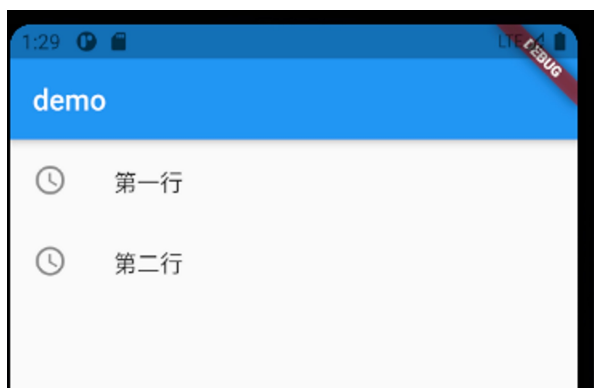
## 一、ListView

ListView有四种构造函数:

1. 默认构造函数
2. ListView.builder, 适用于有大量列表项
3. ListView.seperated, 可以配置分割线, 适用于具有固定数量列表项的ListView
4. ListView.custom, 提供了自定义子Widget的能力

### 1.1 默认构造函数

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Flutter Demo',  
      theme: ThemeData(  
        primarySwatch: Colors.blue,  
      ),  
      //home: MyHomePage(title: 'Flutter Demo Home Page'),  
      home: Scaffold(  
        appBar: AppBar(title: Text('demo')),  
        body: ListView(  
          children: <Widget>[  
            ListTile(leading: Icon(Icons.access_time), title: Text("第一行")),  
            ListTile(leading: Icon(Icons.access_time), title: Text("第二行")),  
          ],  
        ),  
      );  
    }  
  }  
}
```



其中, ListTile是Material组件中的Widget, 用于填充ListView, 图片+文字。

### 1.2 ListView.builder创建

用于展示大量列表项

```
void main() {
```

```

runApp(MyApp(items:newList<String>.generate(300,(index)=>'第$index行')));
}

class MyApp extends StatelessWidget{
  final List<String> items;

  MyApp({required this.items});

  @override
  Widget build(BuildContext context){
    return MaterialApp(
      title: 'FlutterDemo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      //home: MyHomePage(title: 'FlutterDemoHomePage'),
      home: Scaffold(
        appBar: AppBar(
          title: Text('demo'),
        ),
        body: ListView.builder(
          itemCount: items.length,
          itemBuilder: (context, index){
            return ListTile(
              leading: Icon(Icons.access_time),
              title: Text('${items[index]}'),
            );
          }
        )
      )
    );
  }
}

```



### 1.3 ListView.seperated 创建

```

void main(){
  runApp(MyApp(items:newList<String>.generate(300,(index)=>'第$index行')));
}

```

```

class MyApp extends StatelessWidget {
  final List<String> items;

  MyApp({required this.items});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      //home: MyHomePage(title: 'Flutter Demo Home Page'),
      home: Scaffold(
        appBar: AppBar(
          title: Text('demo'),
        ),
        body: ListView.separated(
          itemCount: items.length,
          itemBuilder: (context, index) {
            return ListTile(
              leading: Icon(Icons.access_time),
              title: Text('${items[index]}'),
            );
          },
          separatorBuilder: (context, index) {
            return Container(
              constraints: BoxConstraints.tightFor(height: 1),
              color: Colors.black45,
            );
          },
        ));
  }
}

```



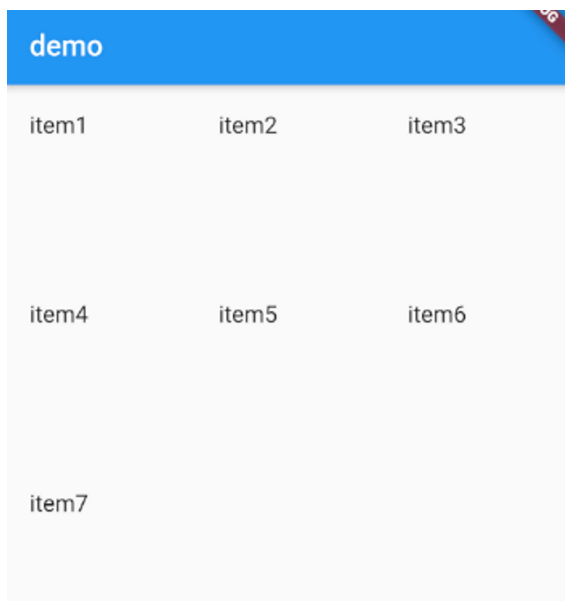
## 二、GridView

用法与ListView相似

1. 默认构造函数
2. GridView.count: 在横轴上具有固定数量的GridView
3. GridView.extent: 在横轴方向上具有最大范围的GridView

4. GridView.builder: 适用于具有大量的列表项
5. GridView.costom: 提供了自定义子Widget的能力

```
body:GridView.count(  
  crossAxisCount:3,  
  children:<Widget>[  
    ListTile(title:Text('item1'),),  
    ListTile(title:Text('item2'),),  
    ListTile(title:Text('item3'),),  
    ListTile(title:Text('item4'),),  
    ListTile(title:Text('item5'),),  
    ListTile(title:Text('item6'),),  
    ListTile(title:Text('item7'),),  
  ],  
)
```



### 三、 PageView

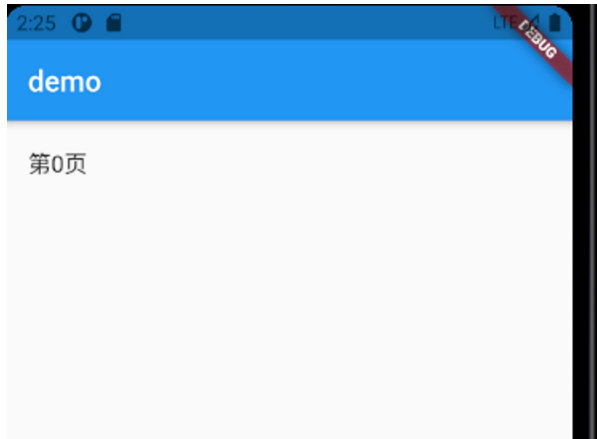
PageView是一个可逐页滚动的列表，主要有以下三个构造函数

1. 默认构造函数
2. PageView.builder: 适用于具有大量列表项
3. PageView.custom: 提供了自定义子Widget能力

以默认构造函数为例：

```
home:Scaffold(  
  appBar:AppBar(  
    title:Text('demo'),  
  ),  
  body:PageView(  
    onPageChanged:(index){  
      print('当前为第$index页');  
    },  
    children:<Widget>[  
      ListTile(title:Text('第0页'),),  
      ListTile(title:Text('第1页'),),  
    ],  
  ),  
)
```

```
ListTile(title:Text('第2页'),),  
],  
)  
)
```



onPageChanged可以知道当前在第几页