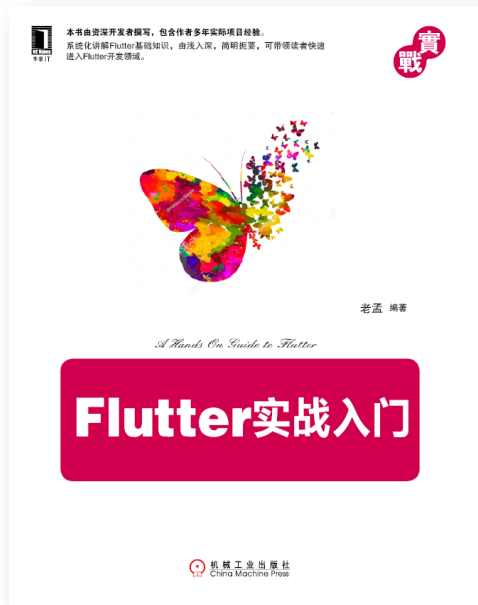


微信交流群



点击购买

ListView

ListView是我们最常用的组件之一，用于展示大量数据的列表。

构建方式

数据较少时，可以直接使用如下方式：

```
1  ListView(  
2    children: <Widget>[  
3      item,item1,item2,  
4    ],  
5  )
```

这种方式一次加载所有的组件，没有“懒加载”，因此当有大量数据时，使用动态创建列表的方式：

```
1  ListView.builder(  
2    itemBuilder: (BuildContext context, int index) {  
3      return Text('Item$index');  
4    },  
5    itemExtent: 50,  
6  )
```

如果想在每一项中间增加分割线可以使用如下方式：

```
1  ListView.separated(  
2    itemBuilder: (BuildContext context, int index) {  
3      return Text('Item$index');  
4    },  
5    separatorBuilder: (BuildContext context, int index){  
6      return Divider();  
7    },  
8    itemCount: 50,  
9  )
```

一般上面的方式就够用了，如果以上都不能满足你的要求，还可以使用 `ListView.custom`，自定义 `SliverChildDelegate` 构建自己的 `ListView`。

基础属性

通过 `scrollDirection` 参数控制滚动方向，默认是垂直方向，向上滚动，设置为水平方向：

```
1  ListView.builder(  
2    scrollDirection: Axis.horizontal,  
3    ...  
4  )
```

滚动方向如果是垂直方向，默认是向上滚动，通过 `reverse` 参数设置其向下滚动，代码如下：

```
1  ListView.builder(  
2    reverse: true,  
3    ...  
4  )
```

controller 可以控制ListView的滚动，比如获取当前滚动的位置，或者代码直接滚动到指定位置，用法如下：

```
1  ScrollController controller;  
2  @override  
3  void initState() {  
4    super.initState();  
5    controller = ScrollController()  
6    ..addListener((){  
7      print('${controller.position}');  
8    });  
9  
10 }  
11  
12 @override  
13 Widget build(BuildContext context) {  
14   return ListView.builder(  
15     controller:controller,  
16     itemBuilder: (context, index) {  
17       return Text('Item $index');  
18     });  
19 }
```

physics 参数控制滚动到物理特性，比如设置为不可滚动：

```
1  ListView.builder(  
2    physics: NeverScrollableScrollPhysics(),  
3    ...  
4  )
```

系统提供的ScrollPhysics有：

- AlwaysScrollableScrollPhysics：总是可以滑动
- NeverScrollableScrollPhysics：禁止滚动
- BouncingScrollPhysics：内容超过一屏 上拉有回弹效果
- ClampingScrollPhysics：包裹内容 不会有回弹

addAutomaticKeepAlives 参数表示当关闭屏幕时Item是否进行垃圾回收，默认为true。

喜欢作者



版权所有，禁止私自转发、克隆网站。

[< ListTileTheme](#)

[ListViewWheelScrollView >](#)