

ListView

ListTile

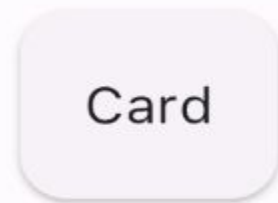
Card



Card Widget

- A material design widget that provides a rectangular container with rounded corners, shadows, and elevation.

```
Card(  
  child: Padding(  
    padding: const EdgeInsets.all(16.0),  
    child: Text('Card'),  
  ),
```



ListTile Widget

- A single, fixed-height row containing text and optional icons or widgets.
- ListTile Components:
 - Title and subtitle.
 - Leading and trailing icons.
 - Customizable interaction onTap function.

ListTile Widget

```
ListTile(  
  leading: Icon(Icons.person),  
  title: Text('Mohammed Ali'),  
  subtitle: Text('STC'),  
  trailing: Checkbox(  
    value: checkValue,  
    onChanged: (value) {  
      setState(() {  
        checkValue = value!  
      });  
    }  
  ),  
  onTap: () {  
    setState(() {  
      checkValue = !checkValue;  
    });  
  },  
);
```



Mohammed Ali
STC



ListView Widget

- A Simple way to display scrollable lists of widgets.
- Can hold any number of child widgets.
- Supports vertical and horizontal scrolling.



ListView Widget

```
ListView(  
  children: [  
    Text('Ali'),  
    Text('Mohammed'),  
    Text('Fahad'),  
  ],  
),
```

Ali
Mohammed
Fahad

ListView Widget

```
ListView(  
  children: [  
    ListTile(title: Text('Ali'), leading: Icon(Icons.person)),  
    ListTile(title: Text('Mohammed'), leading: Icon(Icons.person)),  
    ListTile(title: Text('Fahad'), leading: Icon(Icons.person)),  
  ],  
),
```



Ali



Mohammed




Fahad

Using Divider Widget

```
ListView(  
  children: [  
    ListTile(title: Text('Ali'), leading: Icon(Icons.person)),  
    Divider(),  
    ListTile(title: Text('Mohammed'), leading: Icon(Icons.person)),  
    Divider(),  
    ListTile(title: Text('Fahad'), leading: Icon(Icons.person)),  
    Divider(),  
  ],  
),
```

 Ali

 Mohammed

 Fahad

Customizing Divider Widget

- You can customize the divider by adjusting its properties:

- color
- thickness
- height
- indent
- endIndent

```
Divider(  
    color: Colors.grey,  
    thickness: 2)
```

ListView.builder

- A dynamically generated scrollable list.
- Optimized for large or infinite lists.

```
ListView.builder(  
  itemCount: 10, // or null for infinite scroll  
  itemBuilder: (context, index) {  
    return ListTile(  
      title: Text('Item $index'),  
    );  
  })
```

Item 0

Item 1

Item 2

Item 3

Item 4

Item 5

Item 6

Item 7

Item 8

Item 9

ListView.builder

- Build a ListView from predefined list.

```
List<String> names = ['Ali', 'Ahmed', 'Fahad', 'Saad'];  
ListView.builder(  
    itemCount: names.length,  
    itemBuilder: (context, index) {  
        return ListTile(  
            title: Text(names[index]),  
        );  
    })
```

Ali

Ahmed

Fahad

Saad

ListView.separated

- Build a ListView with a separated divider:

```
ListView.separated(  
    itemCount: names.length,  
    itemBuilder: (context, index) {  
        return ListTile(  
            title: Text(names[index]),  
        );  
    },  
    separatorBuilder: (context, index) {  
        return Divider();  
    },  
),
```

Ali

Ahmed

Fahad

Saad

ListView Properties

- **scrollDirection**: Determines the direction of scrolling (`Axis.vertical` or `Axis.horizontal`).
- **reverse**: Reverses the scrolling direction of the list.
- **shrinkWrap**: control the ListView height or width when it's inside a scrollable parent (e.g., Column)

Practice

- Code this mockup.
- The upper side scrolls horizontally.
- 10 items. Use `ListView.builder`.

