

1. 开关框】 Switch()



```
1 Transform.scale(  
2     scale: 1.5, //放大  
3     child: Switch(  
4         value: true, //选择状态  
5         onChanged: (value) {  
6             //被选择回调  
7         },  
8         materialTapTargetSize:  
9             MaterialTapTargetSize.shrinkWrap, //去掉边距  
10        activeColor: Colors.red, //被选择圆圈颜色  
11        activeTrackColor: Colors.black, //被选择颜色  
12        inactiveThumbColor: Colors.green, //未选择圆圈颜色  
13        inactiveTrackColor: Colors.blue, //未选择颜色  
14        // activeThumbImage: AssetImage(  
15        //     'images/1.png',  
16        // ),  
17    ),  
18 ),
```

2. 多选框】 Checkbox()



```
1 Transform.scale(  
2     scale: 1.5, //放大  
3     child: Checkbox(  
4         value: true, //选择状态  
5         onChanged: (value) {  
6             //选择回调  
7         },
```

```

8         activeColor: Colors.red, //被选择圆圈颜色
9         checkColor: Colors.amber, //被选择打勾颜色
10        materialTapTargetSize:
11            MaterialTapTargetSize.shrinkWrap, //去掉边距
12    ),
13 ),

```

3. 单选框】 Radio()



```

1 Transform.scale(
2     scale: 1.5, //放大
3     child: Radio(
4         value: 1, //选择序号
5         groupValue: 1, //被选择序号
6         onChanged: (value) {
7             //选择回调
8         },
9         materialTapTargetSize: MaterialTapTargetSize.shrinkWrap, //左右收缩
10        visualDensity: VisualDensity.compact, //上下收缩
11    ),
12 ),

```

4. 横线滚动条】 Slider()

```

1 Slider(
2     value: 20.0, //当前值
3     min: 1.0, //最小值
4     max: 30.0, //最大值
5     label: "被按住时提示", //divisions设置显示在节点上的label
6     divisions: 10, //分成几个模块
7     activeColor: Colors.black, //已滚动后的线条颜色
8     inactiveColor: Colors.cyan, //未滚动的线条颜色
9     onChanged: (double newValue) {
10        //状态回调

```

```

11         },
12         onChangeStart: (value) {
13             //开始状态回调一次
14         },
15         onChangeEnd: (value) {
16             //结束状态回调一次
17         },
18         semanticFormatterCallback: (double newValue) {
19             return "AAA";
20         },
21     ),

```

5. 圆形进度条】CircularProgressIndicator()

```

1  CircularProgressIndicator(
2      backgroundColor: Colors.grey, //指示器背景色
3      valueColor: AlwaysStoppedAnimation(Colors.blue), //进度背景色
4      strokeWidth: 8, //宽度
5      value: 0.8, //当前进度 满值为1
6  ),

```

6. 线条进度条】LinearProgressIndicator()

```

1  LinearProgressIndicator(
2      backgroundColor: Colors.grey, //指示器背景色
3      valueColor: AlwaysStoppedAnimation(Colors.blue), //进度背景色
4      color: Colors.black,
5      value: 0.8, //进度 满值为1 //在没有给value赋值的情况下会执行循环动画
6      minHeight: 22, //线条高度
7  ), //不能放在 Row模块中

```

7. 下拉选择框】DropDownButtonFormField()

The image shows a Flutter widget, specifically a `DropDownButtonFormField`. It features a title '性别选择标题' (Gender Selection Title) and a dropdown menu with the selected item '女' (Female). The widget is styled with a light blue border and a white background.

```

1 DropdownButtonFormField(
2     isExpanded: true,
3     decoration: const InputDecoration(
4         border: OutlineInputBorder(), //设置带边框
5         labelText: "性别选择标题",
6     ),
7     value: ArrSEX[0], // 设置默认值
8     onChanged: (String? newPosition) {
9         // 选择回调
10    },
11    // 弹出传入可选的数组
12    items: ArrSEX.map((String sex) {
13        return DropdownMenuItem(value: sex, child: Text(sex));
14    }).toList(),
15 ),

```

8. 方法涵 底部弹出层】 showModalBottomSheet()

```

1 //自定义 底部弹出层
2 void funBottomSheet() {
3     //内置属性 底部打开弹框
4     showModalBottomSheet(
5         builder: (BuildContext context) {
6             //构建弹框中的内容
7             return StatefulBuilder(
8                 builder: (BuildContext context, StateSetter setState) {
9                     return Container(
10                        padding: const EdgeInsetsDirectional.all(22),
11                        child: const Text("data"),
12                    ),);
13        },
14        shape: BeveledRectangleBorder(side: BorderSide.none), //切角
15        elevation: 222,
16        backgroundColor: Colors.amber, //背景颜色
17        barrierColor: Colors.black12, //设置遮挡底部的半透明颜色,
18        isScrollControlled: false, //滚动控制 是否时全屏还是半屏

```

```

19     useRootNavigator: false, //根导航器
20     isDismissible: true, //点击遮挡层关闭
21     enableDrag: false, // 是否可以向下拖动关闭
22     useSafeArea: false, //刘海安全区
23     context: context,
24 );
25 }

```

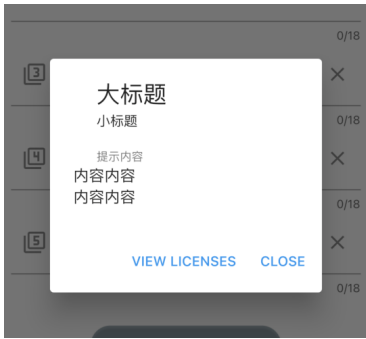
9. 方法涵 全屏弹出层】 showDialog()

```

1 //自定义 置顶全屏弹出层
2 void funShowDialog() {
3     //内置属性 置顶全屏弹框
4     showDialog(
5         builder: (BuildContext context) {
6             //构建弹框中的内容
7             return Container(
8                 width: 320,
9                 padding: const EdgeInsetsDirectional.all(12),
10                child: const Text("置顶弹框 置顶弹框 置顶弹框 置顶弹框 置顶弹框 置顶弹框"),
11            );
12        },
13        barrierColor: Colors.black54, //设置遮挡底部颜色,
14        useRootNavigator: false, //根导航器
15        useSafeArea: true, //刘海安全区
16        barrierDismissible: true, //点击边缘关闭
17        context: context,
18    );
19 }

```

10. 方法涵 描述App信息描述弹框】 showAboutDialog()



```
1 //自定义方法涵
2 void funShowDialog() {
3     //内置属性 描述App信息描述弹框
4     showAboutDialog(
5         applicationName: "大标题", //应用程序名称
6         applicationVersion: '小标题', //应用程序版本
7         applicationLegalese: "提示内容", //著作权
8         // applicationIcon:// 应用程序的图标
9         children: [Text("内容内容"), Text("内容内容")], //内容
10        useRootNavigator: false, //根导航器
11        anchorPoint: Offset(1,2) //锚点 可以使整个弹出框位置发生偏移
12        context: context,
13    );
14 }
```

11. 窗口框组件】 AlertDialog()

```
1 //自定义方法涵
2 void funShowDialog() {
3     //内置属性 弹框
4     showDialog(
5         useRootNavigator: false, //根导航器
6         anchorPoint: Offset(2, 2), //锚点 可以使整个弹出框位置发生偏移
7         context: context,
8         builder: (context) {
9             return AlertDialog(
10                titlePadding: EdgeInsets.all(10),
11                elevation: 10,
12                backgroundColor: Colors.pink, //背景颜色
```

```

13     shape: RoundedRectangleBorder(
14         borderRadius: BorderRadius.circular(20)), //设置形状
15
16     title: const Text('我是标题'),
17     // icon: Icon(Icons.work_rounded),
18     content: const Padding(
19         padding: EdgeInsets.all(8.0),
20         child: Text('我可以是文本内容我可以是文本内容我可以是文本内容我可以是文本内
    容'),
21     ),
22     contentTextStyle: const TextStyle(color: Colors.black), //文本内容的
    text样式
23     actions: [
24         Padding(
25             padding: const EdgeInsets.all(8.0),
26             child: ElevatedButton(
27                 onPressed: () {
28                     Navigator.of(context).pop(true);
29                 },
30                 child: const Text('确定')),
31         ),
32         Padding(
33             padding: const EdgeInsets.all(8.0),
34             child: ElevatedButton(
35                 onPressed: () {
36                     Navigator.of(context).pop();
37                 },
38                 child: const Text('取消')),
39         ),
40     ],
41 );
42 },
43 );
44 }

```

12. 确认提示弹框】 showDialog()



```
1 //自定义方法涵
2 void funShowDialog(bool tfTap) {
3     //内置属性 确认弹框
4     showDialog(
5         context: context,
6         builder: (context) {
7             return AlertDialog(
8                 title: const Text(
9                     'AlertDialog 标题',
10                    style: TextStyle(color: Colors.blueAccent),
11                ),
12                content: const Text('我是 AlertDialog 对话框!'),
13                actions: [
14                    ElevatedButton(
15                        child: const Text("确定"),
16                        onPressed: () {
17                            setState(() {
18                                tfTap = true;
19                            });
20                            Navigator.of(context).pop(); //关闭窗口
21                    },
22                ),
23                    ElevatedButton(
24                        child: const Text("取消"),
25                        onPressed: () {
26                            setState(() {
27                                tfTap = false;
28                            });
29                            Navigator.of(context).pop(); //关闭窗口
30                    },
31                )
32            );
33        }
34    );
35 }
```



```
32         ],
33     );
34     },
35 );
36 }
```

13.