

(1) Flutter 模块样式

decoration: BoxDecoration(样式属性)

1. 颜色】 color

```
1 color: const Color(0xFF0099ff), //颜色
```

2. 插入间隔】 SizedBox()

```
1 const SizedBox(height: 5), //插入间隔
```

3. 内外边距】 padding

```
1 padding: const EdgeInsets.only(top: 10, left: 10), //内边距
2 margin: const EdgeInsets.all(20), //外边距
```

4. 圆角边框】 borderRadius

```
1 borderRadius: BorderRadius.circular(20), //圆角边框大小
```

5. 边框设计】 border

```
1 // 边框设计
2 border: Border.all(
3   color: const Color.fromARGB(255, 209, 38, 55), //颜色
4   width: 3, //边框大小
5 ),
```

6. 边框阴影】 boxShadow

```

1 // 边框阴影
2 boxShadow: const [
3     BoxShadow(
4         color: Color.fromARGB(255, 217, 255, 0), //颜色
5         blurRadius: 10, //阴影宽度
6     )
7 ],

```

7. 背景渐变】 gradient

```

1 //背景线性渐变 LinearGradient //径向渐变 RadialGradient
2 gradient: const LinearGradient(
3     begin: Alignment.centerLeft,
4     end: Alignment.centerRight,
5     colors: [
6         Color.fromARGB(255, 255, 77, 0), //颜色
7         Color.fromARGB(255, 30, 26, 143), //颜色
8     ],
9 ),

```

8. 上下居中】 alignment

```

1 alignment: const Alignment(0, 0), //上下居中】 上下值
2 mainAxisAlignment: MainAxisAlignment.center, //居中

```

9. 水平/垂直 间距】 spacing

```

1 spacing: 10, //水平间距
2 runSpacing: 15, //垂直间距

```

10. 位移】 alignment

```
1 alignment: Alignment(1.2, -1), //位移
2 alignment: Alignment.topLeft, //居左
```

11. 外边距】 margin

```
1 margin: const EdgeInsets.fromLTRB(0, 10, 0, 20), //外边距】 四边值
```

12. 内边距】 padding

```
1 padding: const EdgeInsets.fromLTRB(0, 10, 0, 10), //内边距】 四边值
```

13. 旋转/位移/倾斜】 transform

```
1 transform: Matrix4.translationValues(1, 2, 3), //位移】 左上右
2 transform: Matrix4.rotationY(1), //旋转】 X向上 Y向后 Z向左
3 transform: Matrix4.skew(0, 1), //倾斜】 X向右 Y向下
```

14. 背景图片】 image

```
1 /*设计背景图片 */
2 image: const DecorationImage(
3   image: NetworkImage("https"),
4   fit: BoxFit.cover, //图片显示模式 铺满显示
5 ),
```

15. 置顶导航定位】 Positioned

```
1 /* 置顶定位 */
2 Positioned(
3   left: 0, //定位坐标
```

```
4   top: 0, //定位坐标
5   // 标签
6   child: Container(
7     child: const Text("二级导航标题"),
8   ),
9 )
```

16.

(2) Flutter 标签样式

1. 文本】 Text

```
1 child: const Text(
2     "你好", //内容
3     textAlign: TextAlign.right, //右居中
4     maxLines: 1, //显示行数
5     overflow: TextOverflow.ellipsis, //溢出显示省略号
6
7     style: TextStyle(
8       color: Color(0xffffffff), //颜色
9       fontSize: 20, //字体大小
10      textAlign: TextAlign.right, //右居中
11      fontWeight: FontWeight.bold, //字体变粗
12      fontStyle: FontStyle.italic, //字体倾斜
13      backgroundColor: Color.fromARGB(255, 203, 43, 43), //背景颜色
14      letterSpacing: 2, //字体间距
15
16      /* 字体底部横行 */
17      decoration: TextDecoration.underline, //字体底部横行
18      decorationColor: Color.fromARGB(255, 231, 239, 6), //字体底部横行颜色
19    ),
20 ),
```

2. 图片】 Image

```

1  /* 图片标签 */
2  child: Image.network(
3    "img",
4    scale: 2, //图片缩放
5    alignment: Alignment.center, //居中显示
6    width: double.infinity,
7    fit: BoxFit.contain, //图片显示模式
8    repeat: ImageRepeat.repeat, //图片平铺
9  ),

```

3. 本地图片】Image.asset()

```

1  //加载本地图片
2  return Container(
3    width: 120,
4    height: 120,
5
6    //在 项目根目录下 pubspec.yaml 文件 flutter: => assets: 中配置本地图片
7    child: Image.asset(
8      "images/flutter-logo.png", //本地图片地址
9      fit: BoxFit.cover, //图片显示模式
10  ),
11 );

```

4. 列表】ListTile

```

1  /* 列表 */
2  ListTile(
3    // 左侧
4    leading: Icon(
5      Icons.home,
6      color: Color(0xFF8977ff),
7    ),
8
9    title: const Text("列表标题"),
10   subtitle: const Text("内容内容内容"),
11

```

```

12 //右侧
13 trailing: Image.network(
14     "https://img-s-msn-com.akamaized.net/tenant/amp/entityid/AA1aCc4z.img"),
15 ),
16 const Divider(), //列表下横线

```

5. 按钮】ElevatedButton

```

1 //普通按钮
2     ElevatedButton(
3         //按钮样式
4         style: ButtonStyle(
5             backgroundColor: MaterialStateProperty.all(Colors.red), //背景颜色
6             foregroundColor: MaterialStateProperty.all(Colors.amber), //字体颜色
7         ),
8         onPressed: () {}, //按下触发的事件
9         child: const Text("凸起按钮"), //按钮名称
10    ),

```

```

1 //圆角按钮
2     ElevatedButton.icon(
3         // 样式设计
4         style: ButtonStyle(
5             // 形状设计
6             shape: MaterialStateProperty.all(
7                 // 圆形/边框设计
8                 RoundedRectangleBorder(
9                     borderRadius: BorderRadius.circular(20), //圆角
10                ),
11            ),
12        ),
13        onPressed: () {}, //触发事件
14        icon: const Icon(Icons.send),
15        label: const Text("发送3"),
16    ),

```

```

1 //圆按钮
2     Container(
3         height: 60,
4         width: 60,
5         child: ElevatedButton(
6             // 样式设计
7             style: ButtonStyle(
8                 // 形状设计
9                 shape: MaterialStateProperty.all(
10                     // 圆框设计
11                     const CircleBorder(
12                         //边框
13                         side: BorderSide(
14                             color: Color(0xffffffff),
15                         ),
16                     ),
17                 ),
18             ),
19             onPressed: () {
20                 print("ABCE");
21             }, //触发事件
22             child: const Text("圆按钮"),
23         ),
24     ),
25 ],

```

输入框】 TextField()

```

1 //输入框
2     TextField(
3         //接收输入
4         onChanged: (value) {
5             print(value);
6         },
7         // obscureText: true, //密码隐藏
8         maxLength: 10, //最大长度
9         //样式

```

```
10         decoration: const InputDecoration(  
11             border: OutlineInputBorder(), //边框  
12             labelText: '提示内容',  
13         ),  
14     )
```

(3) Flutter 类型模块

1. 圆形模块】 ClipOval()

```
1 //设计圆形图片  
2 return ClipOval(  
3     child: Image.network(  
4         "jpg",  
5         width: 200,  
6         height: 200,  
7         fit: BoxFit.cover,  
8     ),  
9 );
```

2. 居中模块】 Center()

```
1 //Center使模块居中  
2 return Center(  
3     child: Container(  
4         width: 200, //宽度  
5         height: 200, //高度  
6     ),  
7 );
```

3. 滚动长列表】 ListView()

```
1 //ListView 长滚动列表
```

```

2  // 横向显示外部必须套一层Container 并设置height高度
3  return ListView(
4      shrinkWrap:true, //范围内进行包裹 (内容多高ListView就多高)
5      physics:NeverScrollableScrollPhysics(), //禁止滚动
6      children: [
7          /* 列表 */
8          ListTile(
9              // 左侧
10             leading: Icon(
11                 Icons.home,
12                 color: Color(0xFF8977ff),
13             ),
14             scrollDirection: Axis.horizontal, //决定纵向横向
15             title: const Text("列表标题"),
16             subtitle: const Text("内容内容内容"),
17
18             //右侧
19             trailing: Image.network(
20                 "https://img-s-msn-com.akamaized.net/tenant/amp/entityid/AA1aCc4z.img"),
21         ),
22         const Divider(), //列表下横线
23     ],
24 );
25 //循环模式
26 ListView.builder(
27     itemCount: ArrData.length,
28     itemBuilder: (context, index) {
29         return Container();
30     },

```

4. 模块】 Container()

```

1  //模块
2  Container(
3      alignment: Alignment.center, //模块居中
4      //模块样式
5      decoration: const BoxDecoration(

```

```

6      color: Color(0xffff234ff),
7      //设置四周圆角
8      borderRadius: BorderRadius.only(
9        topLeft: Radius.circular(30),
10       topRight: Radius.circular(30),
11     ),
12   ),
13   clipBehavior: Clip.hardEdge, //超出部分, 可裁剪
14   //标签
15   child: Column(
16     children: [
17       Image.asset("./images/flutter-logo.png"), //本地图片
18       const SizedBox(height: 5), //插入间隔
19       const Text("带样式", style: TextStyle(fontSize: 20)), //文本标签
20     ],
21   ),
22 ),

```

5. 卡片模式】 Card()

```

1  //卡片模式
2      Card(
3        //Card的阴影效果
4        shape: RoundedRectangleBorder(
5          borderRadius: BorderRadius.circular(10), //圆形
6        ),
7        elevation: 10, //阴影深度
8        margin: const EdgeInsets.all(10), //外边距
9
10       child: Column(
11         children: const [
12           ListTile(
13             title: Text("zzz"),
14             subtitle: Text("12345678765"),
15           ),
16           Divider(), //下横行
17           ListTile(

```

```

18         title: Text("zzz"),
19         subtitle: Text("12345678765"),
20     ),
21 ],
22 ),
23 ),

```

6. 点击属性】 InkWell()

```

1 //点击属性
2 InkWell(
3   onTap: () {
4     print("onTap点击");
5   },
6   //水纹
7   //外层包裹Container 需再加Material包裹InkWell
8   splashColor: Color(0xD2912020), //点击时水纹颜色
9   highlightColor: Color(0xDC3823B1), //点下后水纹颜色
10  //圆角设计
11  borderRadius: const BorderRadius.only(
12    topLeft: Radius.circular(30),
13    bottomRight: Radius.circular(30),
14  ),
15  child: Container(),
16 )

```

7. 水平模块】 Row()

```

1 Row(//行显示
2   // mainAxisAlignment: MainAxisAlignment.spaceBetween, //两端对齐
3   mainAxisAlignment: MainAxisAlignment.spaceEvenly, //均分排列
4   //mainAxisAlignment: MainAxisAlignment.center,
5   children: const [
6   ]
7 Column(//列显示
8   children: const [
9   ]

```

8. 并排显示】 Wrap()

```
1 Wrap( //并排显示
2     alignment: WrapAlignment.center,
3     children: [
4         NumberL(Xx: 22),
5         NumberL(Xx: 23),
6         NumberL(Xx: 24),
7         NumberL(Xx: 25),
8         NumberL(Xx: 26),
9         NumberL(Xx: 27),
10        NumberL(Xx: 29),
11        NumberL(Xx: 30),
12    ],
13 ),
```

9. 右上角带点】 Badge()

```
1 Badge(
2     child: ,
3 ),
```

10. 轮播页面】 PageView()

```
1 Container(
2     height: 342, //必须设置高度
3     child: PageView.builder(
4         onPageChanged: (index) {
5             setState(() {
6                 intView = index;
7             });
8         },
9         itemCount: 3,
```

```

10         controller: PageController(initialPage: 2, keepPage: true), //指
    定滚动位置
11         itemBuilder: (context, index) {
12             return Text("AAAAA"); //数据块
13         },
14     ),
15 ),
16 ),
17
18 /* 首先将 PageView.builder controller: pageController 抽离 */
19         //pageController.jumpToPage(2); //无动画切换到指定的页面
20         // pageController.jumpTo(3); //无动画 切换到指定的位置
21         //动画的方式滚动到指定的页面
22         pageController.animateToPage(
23             2, //子Widget的索引
24             curve: Curves.ease, //动画曲线
25             duration: const Duration(milliseconds: 200), //滚动时间
26         );

```

11. 刘海安全区】 SafeArea()

```

1 SafeArea(
2     child:
3 )

```

12. 堆叠布局】 Stack() 用于设计Positioned()

```

1 Stack(
2     children: Positioned()
3 )

```

(4) Flutter 内置API

1. 取屏幕参数】 MediaQuery

```
1 final size = MediaQuery.of(context).size; //获取屏幕 (使用方式 size.xxx)
```

2. 随机生成key

```
1 key = UniqueKey(); //随机生成key ,
2 key2 = ObjectKey( Box(Color:Colors.amber)); //
3 key3 = ValueKey("1");
```

3. 删除数组

```
1 Arrlist.removeAt(index); //删除列表数组
```

4. 加载状态

```
1 child: CircularProgressIndicator(), //加载状态
```

5. Json转对象

```
1 objxxx.fromJson(jsonDecode(jsonxxx));
```

6. 序列化/反序列化

```
1 utf8.encode(json.encode(arrData)) //序列化
2 json.decode(jsonData) ///反序列化
```

(5) 计算类

1. 创建一个循环 for

```
1 List<Widget>_backDataList() {  
2     // 声明了一个数组，里面的数据类型是Widget  
3     List<Widget> list = [];  
4     for (var i = 0; i < 10; i++) {  
5         list.add(ListTile(  
6             title: Text('我是标题$i'),  
7         ));  
8     }  
9     return list;  
10 }
```