# 5.5 Container

我们在前面的章节示例中多次用到过Container组件，本节我们就详细介绍一下Container组件。Container是一个组合类容器，它本身不对应具体的RenderObject，它是DecoratedBox、ConstrainedBox、Transform、Padding、Align等组件组合的一个多功能容器，所以我们只需通过一个Container组件可以实现同时需要装饰、变换、限制的场景。下面是Container的定义：

Container({  
 this.alignment,  
 this.padding, //容器内补白，属于decoration的装饰范围  
 Color color, // 背景色  
 Decoration decoration, // 背景装饰  
 Decoration foregroundDecoration, //前景装饰  
 double width,//容器的宽度  
 double height, //容器的高度  
 BoxConstraints constraints, //容器大小的限制条件  
 this.margin,//容器外补白，不属于decoration的装饰范围  
 this.transform, //变换  
 this.child,  
})

Container的大多数属性在介绍其它容器时都已经介绍过了，不再赘述，但有两点需要说明：

* 容器的大小可以通过width、height属性来指定，也可以通过constraints来指定；如果它们同时存在时，width、height优先。实际上Container内部会根据width、height来生成一个constraints。
* color和decoration是互斥的，如果同时设置它们则会报错！实际上，当指定color时，Container内会自动创建一个decoration。

### 实例

我们通过Container来实现如图5-16所示的卡片：

图5-16

实现代码如下：

Container(  
 margin: EdgeInsets.only(top: 50.0, left: 120.0), //容器外填充  
 constraints: BoxConstraints.tightFor(width: 200.0, height: 150.0), //卡片大小  
 decoration: BoxDecoration(//背景装饰  
 gradient: RadialGradient( //背景径向渐变  
 colors: [Colors.red, Colors.orange],  
 center: Alignment.topLeft,  
 radius: .98  
 ),  
 boxShadow: [ //卡片阴影  
 BoxShadow(  
 color: Colors.black54,  
 offset: Offset(2.0, 2.0),  
 blurRadius: 4.0  
 )  
 ]  
 ),  
 transform: Matrix4.rotationZ(.2), //卡片倾斜变换  
 alignment: Alignment.center, //卡片内文字居中  
 child: Text( //卡片文字  
 "5.20", style: TextStyle(color: Colors.white, fontSize: 40.0),  
 ),  
);

可以看到Container具备多种组件的功能，通过查看Container源码，我们会很容易发现它正是前面我们介绍过的多种组件组合而成。在Flutter中，Container组件也正是组合优先于继承的实例。

### Padding和Margin

接下来我们来研究一下Container组件margin和padding属性的区别:

...  
Container(  
 margin: EdgeInsets.all(20.0), //容器外补白  
 color: Colors.orange,  
 child: Text("Hello world!"),  
),  
Container(  
 padding: EdgeInsets.all(20.0), //容器内补白  
 color: Colors.orange,  
 child: Text("Hello world!"),  
),  
...

图5-17

可以发现，直观的感觉就是margin的留白是在容器外部，而padding的留白是在容器内部，读者需要记住这个差异。事实上，Container内margin和padding都是通过Padding 组件来实现的，上面的示例代码实际上等价于：

...  
Padding(  
 padding: EdgeInsets.all(20.0),  
 child: DecoratedBox(  
 decoration: BoxDecoration(color: Colors.orange),  
 child: Text("Hello world!"),  
 ),  
),  
DecoratedBox(  
 decoration: BoxDecoration(color: Colors.orange),  
 child: Padding(  
 padding: const EdgeInsets.all(20.0),  
 child: Text("Hello world!"),  
 ),  
),  
...