# 4.1 布局类组件简介

布局类组件都会包含一个或多个子组件，不同的布局类组件对子组件排版(layout)方式不同。我们在前面说过Element树才是最终的绘制树，Element树是通过Widget树来创建的（通过Widget.createElement()），Widget其实就是Element的配置数据。在Flutter中，根据Widget是否需要包含子节点将Widget分为了三类，分别对应三种Element，如下表：

|  |  |  |
| --- | --- | --- |
| Widget | 对应的Element | 用途 |
| LeafRenderObjectWidget | LeafRenderObjectElement | Widget树的叶子节点，用于没有子节点的widget，通常基础组件都属于这一类，如Image。 |
| SingleChildRenderObjectWidget | SingleChildRenderObjectElement | 包含一个子Widget，如：ConstrainedBox、DecoratedBox等 |
| MultiChildRenderObjectWidget | MultiChildRenderObjectElement | 包含多个子Widget，一般都有一个children参数，接受一个Widget数组。如Row、Column、Stack等 |

注意，Flutter中的很多Widget是直接继承自StatelessWidget或StatefulWidget，然后在build()方法中构建真正的RenderObjectWidget，如Text，它其实是继承自StatelessWidget，然后在build()方法中通过RichText来构建其子树，而RichText才是继承自MultiChildRenderObjectWidget。所以为了方便叙述，我们也可以直接说Text属于MultiChildRenderObjectWidget（其它widget也可以这么描述），这才是本质。读到这里我们也会发现，其实**StatelessWidget和StatefulWidget就是两个用于组合Widget的基类，它们本身并不关联最终的渲染对象（RenderObjectWidget）**。

布局类组件就是指直接或间接继承(包含)MultiChildRenderObjectWidget的Widget，它们一般都会有一个children属性用于接收子Widget。我们看一下继承关系 Widget > RenderObjectWidget > (Leaf/SingleChild/MultiChild)RenderObjectWidget 。

RenderObjectWidget类中定义了创建、更新RenderObject的方法，子类必须实现他们，关于RenderObject我们现在只需要知道它是最终布局、渲染UI界面的对象即可，也就是说，对于布局类组件来说，其布局算法都是通过对应的RenderObject对象来实现的，所以读者如果对接下来介绍的某个布局类组件的原理感兴趣，可以查看其对应的RenderObject的实现，比如Stack（层叠布局）对应的RenderObject对象就是RenderStack，而层叠布局的实现就在RenderStack中。

在本章中，为了让读者对布局类Widget有个快速的认识，所以我们并不会深入到RenderObject的细节中去。在学习本章时，读者的重点是掌握不同布局组件的布局特点，具体原理和细节等我们对Flutter整体入门后，感兴趣的话再去研究。