

Flutter 怎么画UI（一）

原理：Flutter Widget采用现代响应式框架构建，中心思想就是用widget构建UI,Widget描述他们在视图中当前配置和状态应该看起来像什么，当widget状态发生变化时，widget会重新构建UI,Flutter会对比前后变化的不同，以确定底层渲染树从一个状态转到下一个状态所需的最小更改

Hello World demo

基础Widget

构建布局



Oeschinen Lake Campground

Kandersteg, Switzerland

★ 41



CALL



ROUTE



SHARE

Lake Oeschinen lies at the foot of the Blüemlisalp in the Bernese Alps. Situated 1,578 meters above sea level, it is one of the larger Alpine Lakes. A gondola ride from Kandersteg, followed by a half-hour walk through pastures and pine forest, leads you to the lake, which warms to 20 degrees Celsius in the summer. Activities enjoyed here include rowing, and riding the summer toboggan run.

配置环境配pubspec.yaml‘

图解布局

- 识别出行和列
- 这个布局是否包含一个网格

- 是否有元素重叠
- 界面是否需要标签页
- 注意需要对齐，填充或者边框区域

Column



Oeschinen Lake Campground

Kandersteg, Switzerland

★ 41



CALL



ROUTE



SHARE

Lake Oeschinen lies at the foot of the Blüemlisalp in the Bernese Alps. Situated 1,578 meters above sea level, it is one of the larger Alpine Lakes. A gondola ride from Kandersteg, followed by a half-hour walk through pastures and pine forest, leads you to the lake, which warms to 20 degrees Celsius in the summer. Activities enjoyed here include rowing, and riding the summer toboggan run.

Row with
3 children

Text

Title section

Icon

Text

Oeschinen Lake Campground

Kandersteg, Switzerland

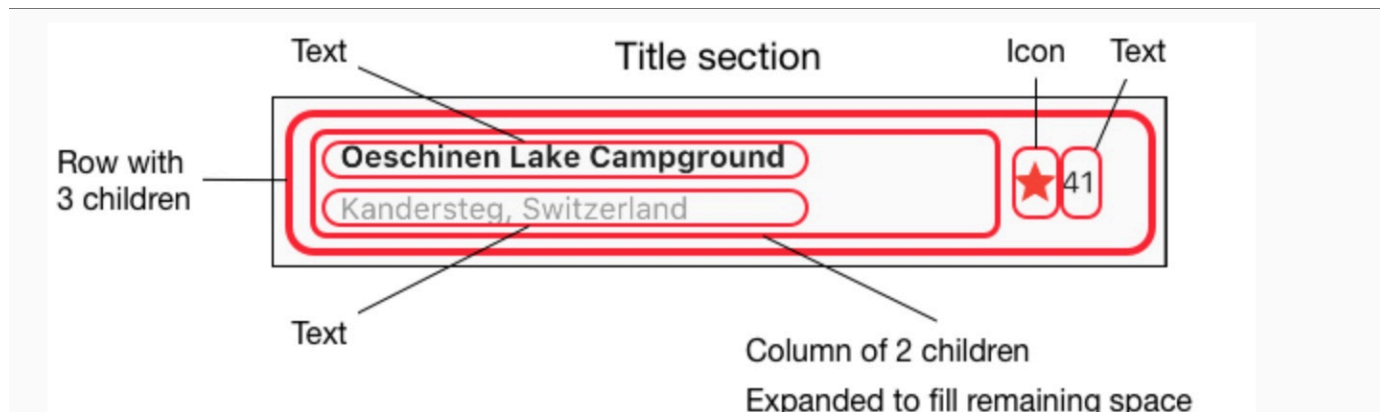


Text

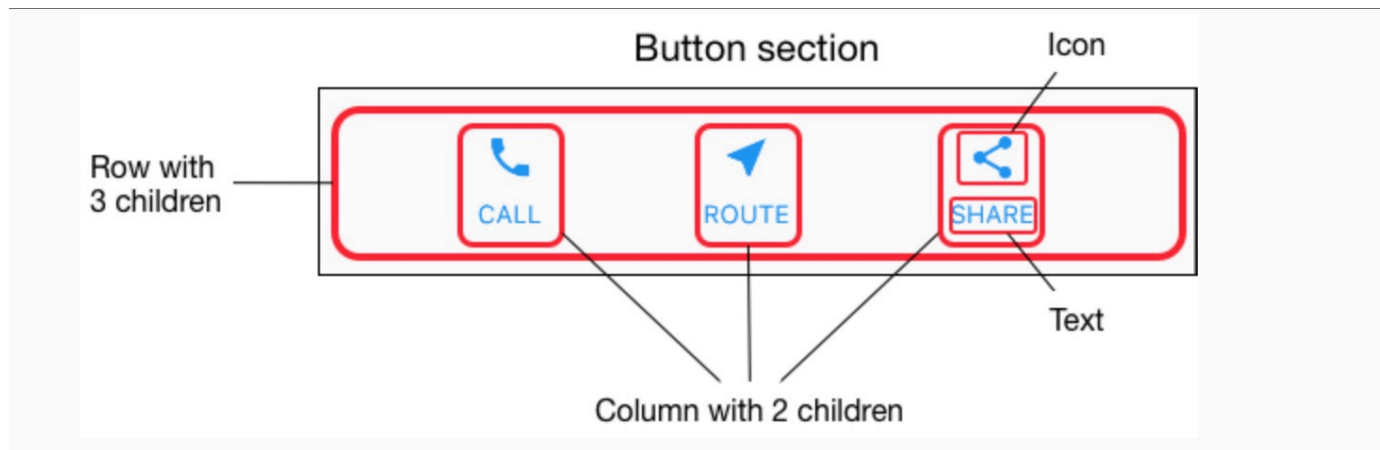
Column of 2 children

Expanded to fill remaining space

1. 找出比较大的元素。四个比较大元素被排列在一个Column控件中：这四个元素是：一张图像，两个Row控件和一个文本块
2. 图解每个Row控件，第一个Row中，称为Title section,他拥有3个子元素：一列文本（一个Column控件包含两个Text控件），一个星星图标和一个数字。



3. 第二个Row中，我们称之为Button section,它同样拥有3个子元素：每个子元素是一个Column包裹一个图标和文本



实现标题行

见代码

实现按钮行

见代码

实现文本部分

见代码

实现图像部分

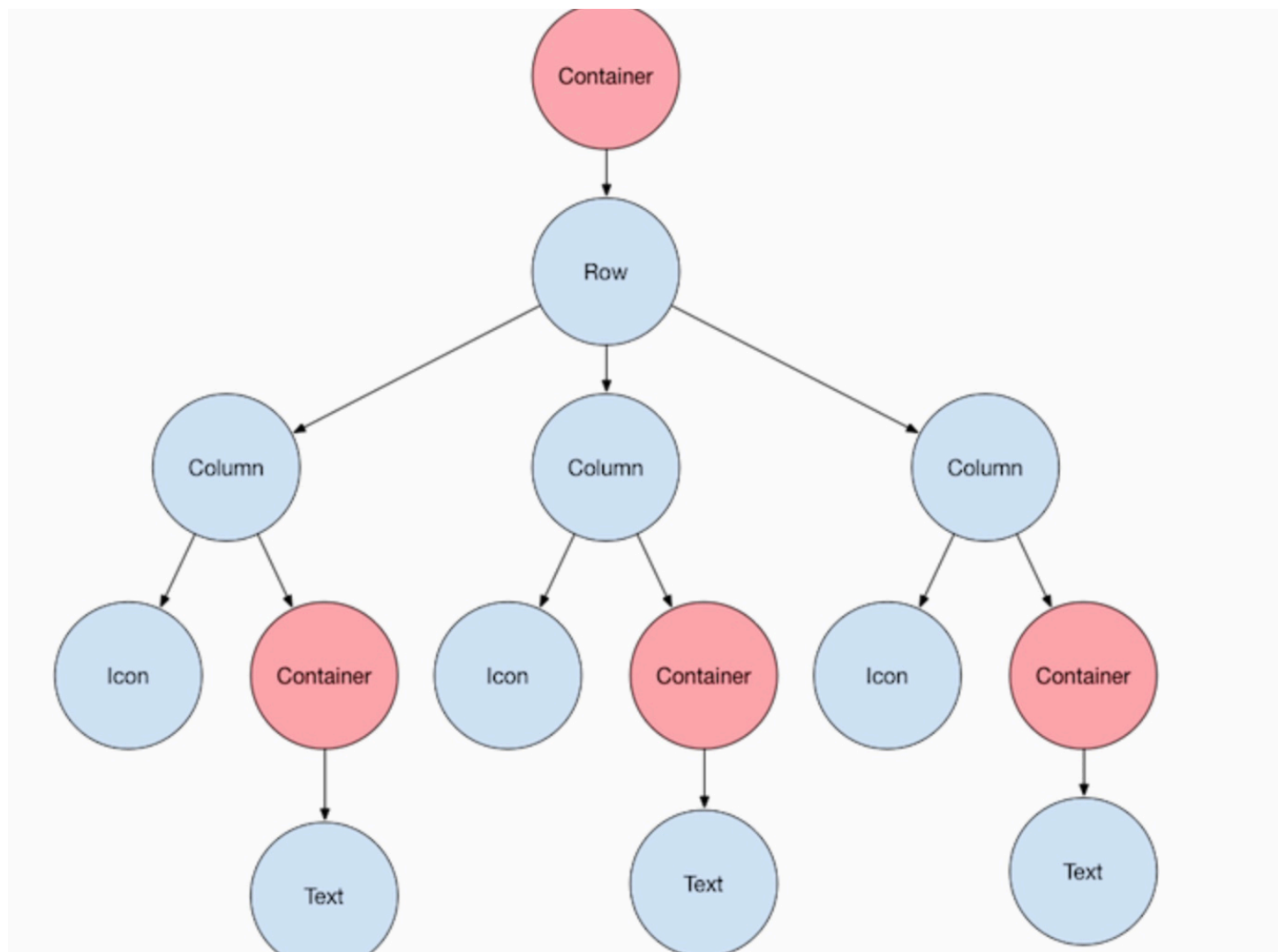
见代码

整合起来

见代码

Flutter 的布局方法

- 控件是用作构建UI的类
- 控件可以同时用作布局和呈现UI元素
- 通过组合简单控件构建复杂控件



Container是一个控件，并且允许自定义子控件。如果要添加内边距，外边距，边框或者背景色，可以在创建容器时设置对应属性值。

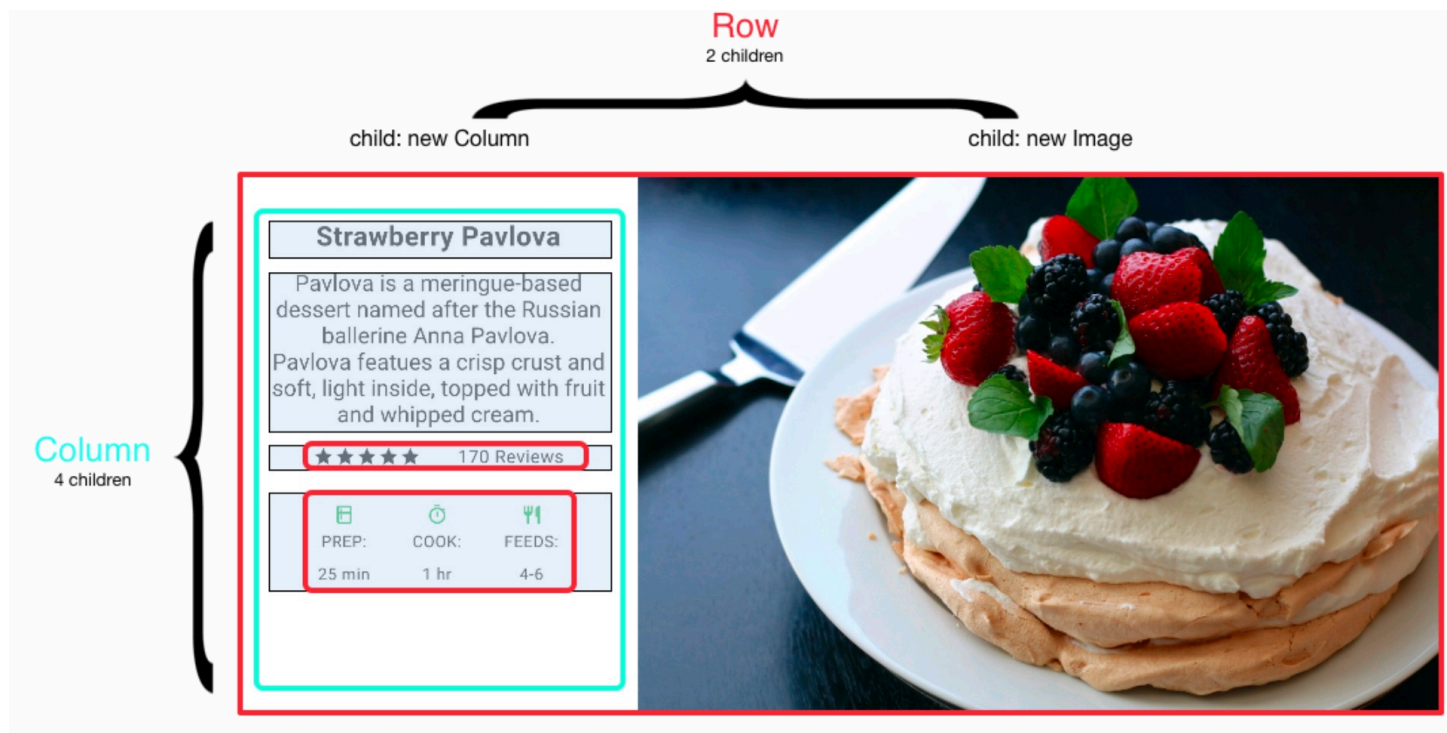
布局一个控件

- 应用本身就是一个控件
- 在Flutter中很容易创建一个控件然后添加到应用中

- 在Flutter中最方便使用的是Scaffold，提供标题栏，背景颜色，并且具有添加drawer,snack bar和bottom sheets的API
- 练习例子，见demo

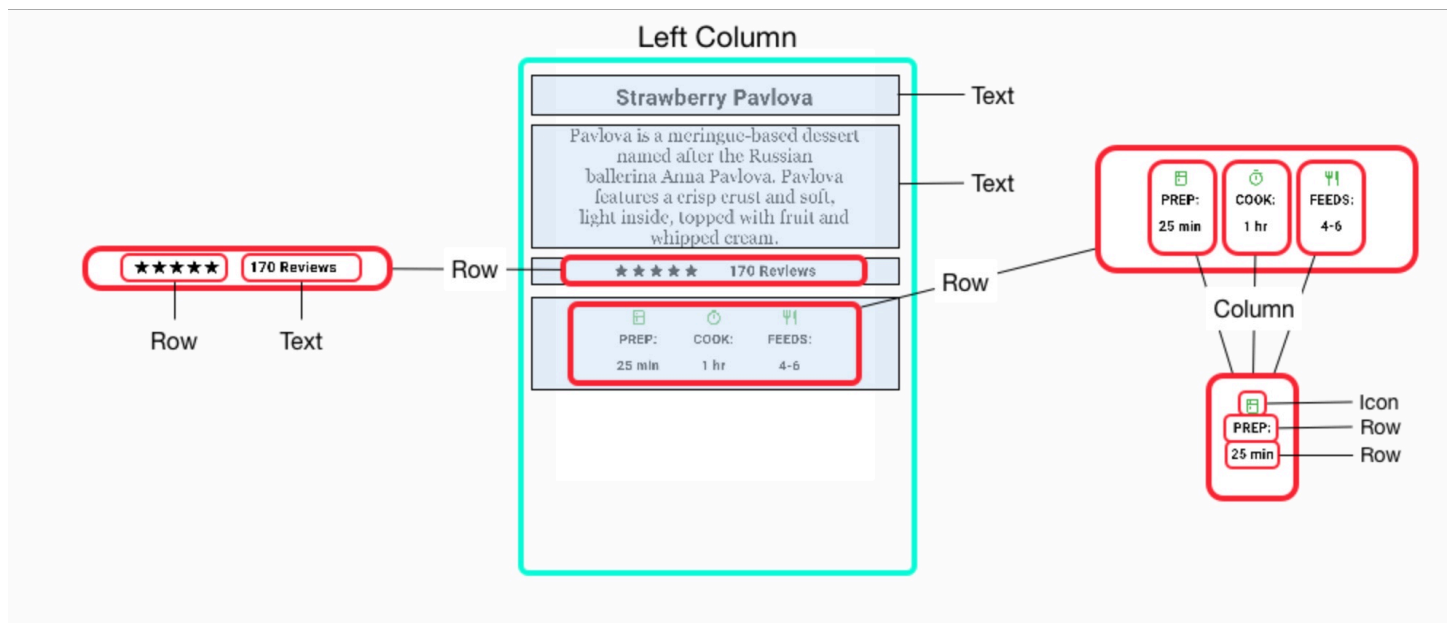
垂直和水平放置多个控件

- Row和Column是两种常用的布局模式
- Row和Column都拥有一个控件列表
- 子控件本身可以是Row，Column或者其他复杂的控件
- 可以指定Row和Column如何在垂直和水平方向上对齐子控件
- 可以指定拉伸或者约束特定子控件
- 可以指定子控件如何使用Row或者Column的可用空间



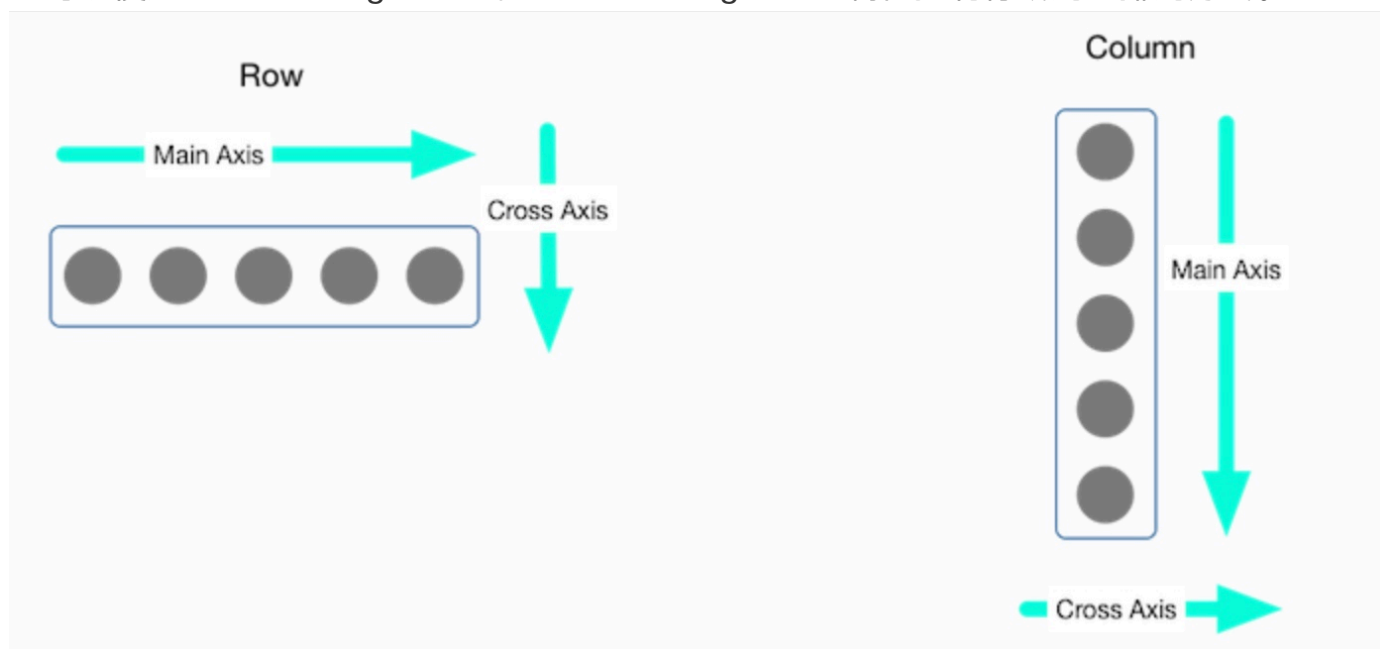
以上布局按行组织，该行包括两个子控件，左侧的Column控件和右侧的Image控件

左侧Column的控件树包括行和列



- 对齐控件

您可以使用 `mainAxisAlignment` 和 `crossAxisAlignment` 属性控制行或列的排列方式。



`MainAxisAlignment` 和 `CrossAxisAlignment` 提供了用于控制对齐的各种常量

- 调整控件大小

使用 `Expanded` 控件，可以调整空间大小。`Expanded` 控件具有 `flex` 属性，它是一个整数，用于确定控件的弹性因子。`Expanded` 控件默认弹性因子是 1

- 挤压控件

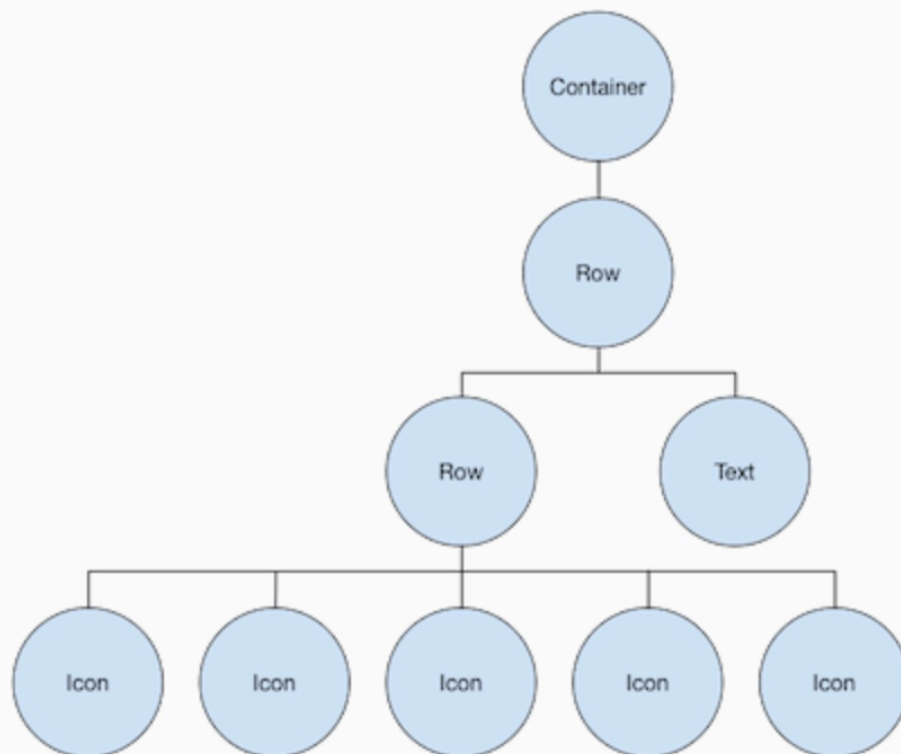
默认情况下，行或列沿着其主轴占据尽可能多的空间，但如果要将子控件紧密挤压一起，请将 `mainAxisSize` 设置为 `MainAxisSize.min`。以下示例使用此属性将星形图标挤压在一起。



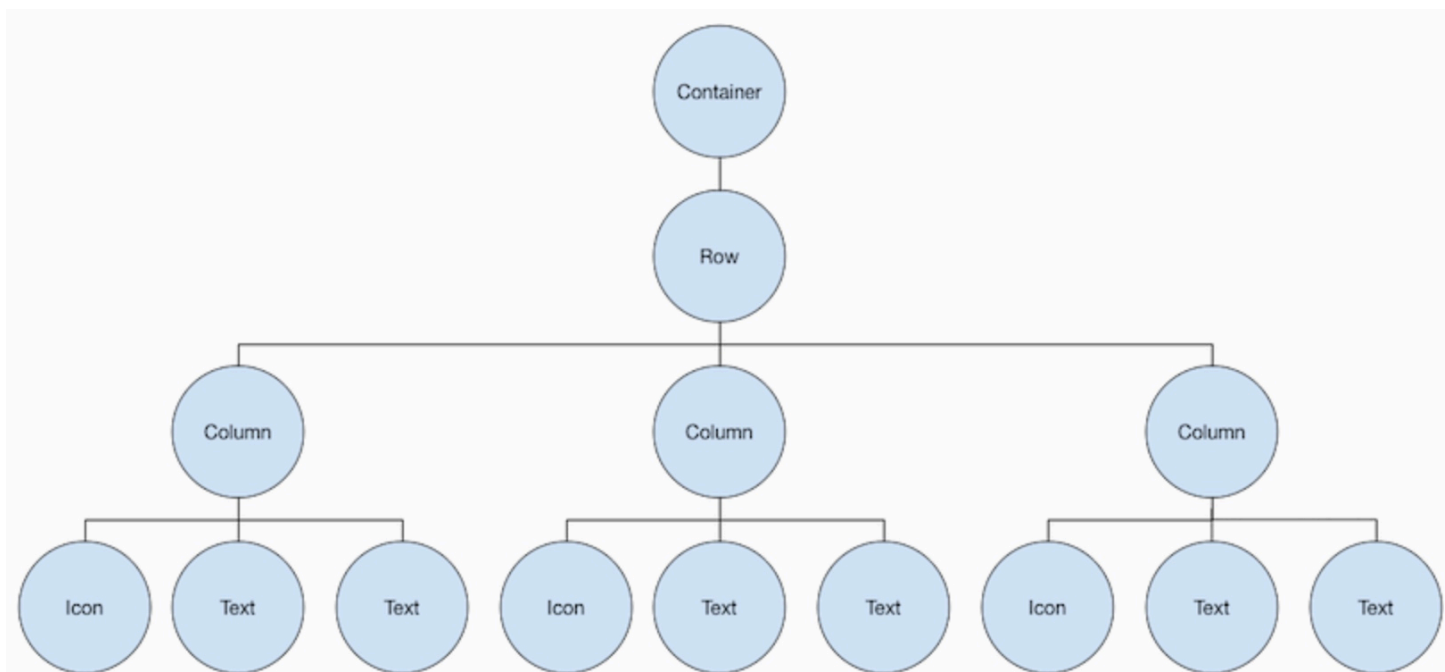
- 嵌套行和列 布局框架允许根据需要在行和列内部嵌套任意行和列



主题分为两行: 评级行和图标行 评级行控件树



图标行:

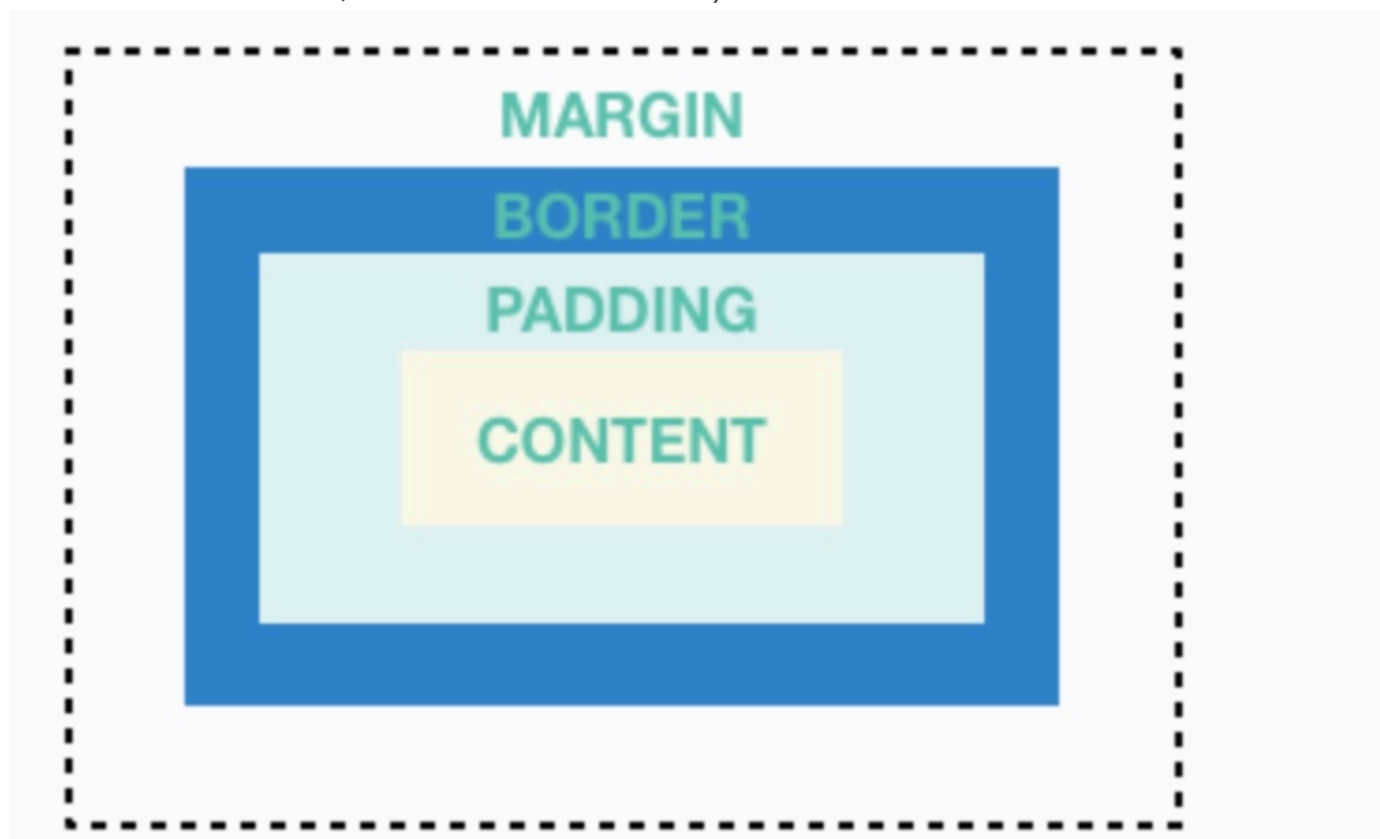


常用布局控件

标准控件

- Container

(添加内边距, 外边距和边框,改变背景颜色和背景图像,可以容纳一个子控件, 这个控件也可以是 Row 或 Column , 甚至控件树的根结点。)



- GridView

可以将控件布局为可滚动的表格

- ListView

可以将控件布局为可滚动的列表

- Stack

可以将控件覆盖到另一个控件上

标准组件

- Card

可以在带圆角和投影的盒子中进行布局。

- ListTile

包括三行文本，可选图标和标题

引用

[Material Components Widget](#)