

Flutter 的基础 —— Widget

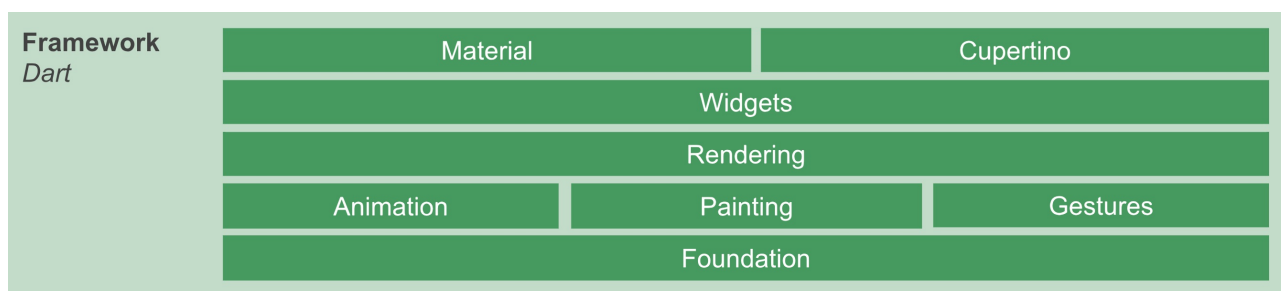
要进行 Flutter 的开发，一定离不开 Widget，Widget 是 Flutter 的基础。

Widget 是什么？

Flutter 中的 Widget 相当于 Android 里的 View，iOS 里的 UIView。

在 Flutter 中要用 Widget 构件 UI。Flutter 的 Widget 渲染采用的是类似 React 的框架：当 Widget 状态发生变化，需要更新界面时，框架会先计算从上一个状态转换到下一个状态所需的最小更改，然后再去刷新界面。

Flutter Framework 里的 Widget



这是 Flutter Framework 层的架构图，可以看到 Framework 里面有一层是 Widgets。

在 Widgets 层下面，有：

- Rendering（渲染层）
- Animation、Painting、Gestures（动画、绘制、手势）
- Foundation（基础库层）

Widgets 下面的层提供的是最基本的功能，但是这些平时很少使用到，因为要使用这些话会比较复杂。我们在开发中使用的都是封装好的东西，也就是 Widgets 上面的那层：

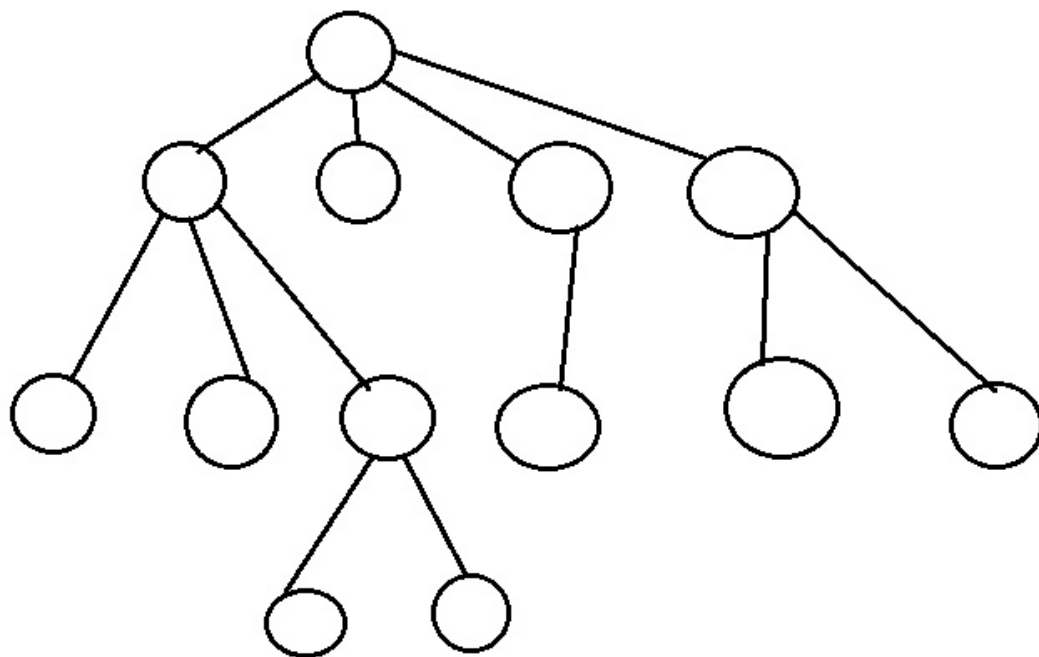
- Material & Cupertino

Material & Cupertino 指的 Widget 的风格是 Material 或 Cupertino。Flutter 为了减轻开发人员的工作量，实现了两种不同风格的组件：Material 和 Cupertino。Material 用于 Android，Cupertino 用于 iOS。有了这些组件，开发人员不需要再做额外的工作，就可以让 Flutter 的 UI 风格适应不同的平台，让 Flutter UI 获得和 Native UI 一样的使用体验。

Widget 的结构：Widget树

Widget 组合的结构是树，所以叫做 Widget树。

Widget树 结构如下图：



父Widget 和 子Widget

在 Widget 树里，Widget 有包含和被包含的关系：

- 父Widget：包含其他 Widget 的就叫 父Widget。
- 子Widget：被 父Widget 包含的 Widget 就叫 子Widget。

根 Widget

根 Widget 也叫 Root Widget。

在前面创建的 Flutter 工程里找到 main.dart,main.dart 是 Flutter 的入口文件。

里面有一个 main() 方法，是 Flutter 的入口方法：

```
void main() => runApp(MyApp());
```

runApp (MyApp()) 里的参数 MyApp() 就是一个 Widget，MyApp 的作用只是封装一下，实际使用的 Widget 是 MaterialApp，这里的 MaterialApp 就是 根(Root)Widget，Flutter会默认把 **根Widget** 充满屏幕。

在 Flutter 中，根Widget 只能是以下三个：

- WidgetsApp

WidgetsApp 是可以自定义风格的 根Widget。

- MaterialApp

MaterialApp 是在 WidgetsApp 上添加了很多 material-design 的功能，是 Material Design 风格的 根Widget。

- CupertinoApp

CupertinoApp 也是基于 WidgetsApp 实现的 iOS 风格的 根 Widget。

这三个中最常用的是 MaterialApp，因为 MaterialApp 的功能最完善。MaterialApp 经常与 Scaffold 一起使用。

Widget 的标识符：Key

因为 Flutter 采用的是 react-style 的框架，每次刷新 UI 的时候，都会重新构建新的 Widget 树，然后和之前的 Widget 树进行对比，计算出变化的部分，这个计算过程叫做 diff，在 diff 过程中，如果能提前知道哪些 Widget 没有变化，无疑会提高 diff 的性能，这时候就需要使用到标识符。

在 diff 过程中，如何知道哪些 Widget 没有变化呢？

为了在 diff 过程中，知道 Widget 有没有变化，就需要给 Widget 添加一个唯一的标识符，然后在 Widget 树的 diff 过程中，查看刷新前后的 Widget 树有没有相同标识符的 Widget，如果标识符相同，则说明 Widget 没有变化，否则说明 Widget 有变化。

假设 UI 刷新前，Widget 树是 A，在 A 里有一个标识符为 a 的 Widget，在 UI 刷新后，重建的 Widget 树是 B，如果 B 里还有标识符为 a 的 Widget，则说明这个 Widget 没变，但是如果 B 里没有标识符为 a 的 Widget，那么说明这个 Widget 发生了变化。

这个标识符在 Flutter 中就是 Key，所有 Widget 都有 Key 这一个属性。

Flutter 中如何在 diff 过程中判断哪些 Widget 没有变化？

Flutter 在 diff 过程中判断哪些 Widget 没有变化，稍微有些复杂，有两种情况：

- 默认情况下（Widget 没有设置 Key）

当没有给 Widget 设置 Key 时，Flutter 会根据 Widget 的 runtimeType 和显示顺序是否相同来判断 Widget 是否有变化。

runtimeType 是 Widget 的类型，例如 Text 和 RaisedButton 就是不同的类型。

- Widget 有 Key

当给 Widget 设置了 Key 时，Flutter 是根据 Key 和 runtimeType 是否相同来判断 Widget 是否有变化。

Key 的分类

Key 总共分为两类：

1. Local Key（局部Key）
2. Global Key（全局Key）

1. Local Key（局部Key）

在有相同父级的 Widget 中，Key 必须是唯一的，这样的 Key 叫做局部Key。

局部Key 在 Flutter 中对应的抽象类是 GlobalKey。GlobalKey 有不同的实现，主要的区别就是使用什么值来作为 Key 的值：

- GlobalKey

将对象作为 Key 的值。

- ValueKey

使用特定类型的值来作为 Key 的值。

- UniqueKey

使用 UniqueKey 自己的对象作为 Key 的值，所以只与自身相等，称为 唯一Key。

2. Global Key（全局Key）

全局Key 是在整个APP中唯一的 Key。

全局Key 在 Flutter 中对应的抽象类是 GlobalKey。GlobalKey 有不同的实现，主要区别是使用的场景不同：

- LabeledGlobalKey

LabeledGlobalKey 用于调试，不会用来比较 Widget 是否有变化。

- GlobalKeyObjectKey

将对象作为 Global Key 的值。

Key 的使用

一般情况下我们不需要使用 Key，但是当页面比较复杂时，就需要使用 Key 去提升渲染性能。

Widget 的分类：StatelessWidget 和 StatefulWidget

因为渲染是很耗性能的，为了提高 Flutter 的帧率，就要尽量减少不必要的 UI 渲染，所以 Flutter 根据 UI 是否有变化，将 Widget 分为：

- StatefulWidget

StatefulWidget 是 UI 可以变化的 Widget，创建完后 UI 还可以在更改。

- StatelessWidget

StatelessWidget 是 UI 不可以变化的 Widget，创建完后 UI 就不可以在更改。

Widget 大全

[Flutter官网](https://flutter.dev/docs/development/ui/widgets)

(<https://flutter.dev/docs/development/ui/widgets>) 上将 Widget 分为14类，总共有上百个 Widget：

1. Accessibility
2. Animation and Motion
3. Assets, Images, and Icons
4. Async
5. Basics
6. Cupertino (iOS-style widgets)
7. Input
8. Interaction Models
9. Layout
10. Material Components
11. Painting and effects
12. Scrolling
13. Styling
14. Text

可以看到，Widget 几乎实现了所有的功能，除了 UI、布局之外，还有交互、动画等，也可见 Widget 在 Flutter 中的地位。

可以在 [官网](https://flutter.dev/docs/development/ui/widgets)

(<https://flutter.dev/docs/development/ui/widgets>) 查看所有
的 Widget。

Flutter 中 Widget 的使用

我们在 Flutter 中使用 Widget 的时候，有以下两点：

1. StatefulWidget 与 StatelessWidget

Flutter 的大部分 Widget 都可以分为 StatefulWidget 和 StatelessWidget 这两类，所以要弄懂 StatefulWidget 与 StatelessWidget 的区别和使用范围。

2. MaterialApp 与 Scaffold

MaterialApp 大部分情况下要作为 Flutter 的根Widget，并且 MaterialApp 经常和 Scaffold 搭配一起使用。