

# 什么是Flutter

---

**Flutter** 是谷歌推出的开发移动UI框架，可以快速的在**IOS**和**Android**上构建高质量的原生用户界面。

**Flutter**可以与现有的代码一起工作，并且**Flutter**是完全免费的，开源的。

总结来说：

Flutter是一款移动应用程序SDK，包含框架、widget和工具，为开发人员提供了一种在Android和iOS上构建和部署精美移动应用程序的简单高效的方式。

## Flutter的优势

---

### 快速开发

毫秒级的热重载，修改后，您的应用界面会立即更新。使用丰富的、完全可定制的widget在几分钟内构建原生界面。

### 富有表现力和灵活的UI

快速发布聚焦于原生体验的功能。分层的架构允许您完全自定义，从而实现难以置信的快速渲染和富有表现力、灵活的设计。

### 原生性能

**Flutter**包含了许多核心的widget，如滚动、导航、图标和字体等，这些都可以在iOS和Android上达到原生应用一样的性能。

## Flutter的开发语言

---

**Flutter**采用的是**Dart**作为开发框架和widget的语言。

### 为什么选择Dart语言？

**Flutter**在四个主要维度进行了评估，并考虑了框架作者、开发人员和最终用户的需求等因素。我们发现不同的语言在不同的层面符合一部分需求，但**Dart**在所有评估维度上得分都很高，并且符合我们的所有要求和标准。

**Dart**运行时和编译器支持**Flutter**的两个关键特性的组合：

**基于JIT的快速开发周期**：允许使用类型的语言进行形状更改和有状态的热重载；

**以及AOT编译器**，可生成高效的ARM代码，可以快速启动并拥有可预测的生产部署性能。

此外，我们有机会与**Dart**社区密切合作，**Dart**社区正在积极投入资源改进**Dart**在**Flutter**中的使用。例如，当我们采用**Dart**时，该语言没有提供生成原生二进制文件的工具链（这对于实现可预测的高性能是很有帮助的），但是现在实现了，因为**Dart**团队为**Flutter**构建了它。同样，**DartVM**之前已经针对吞吐量进行了优化，但团队现在正在优化**VM**的延迟时间，这对于**Flutter**的工作负载更为重要。

**Dart**在以下主要标准上得到高分：

**开发人员的效率** Flutter的主要价值主张之一是通过让开发人员使用相同的代码库为**iOS**和**Android**创建应用程序，从而节省了工程资源。使用高效的语言可以进一步加速开发周期，并使**Flutter**更具吸引力。这对我们的**framework**团队和开发人员都非常重要。大部分**Flutter**功能都是用**Dart**实现，因此我们需要在**10万行代码**时能保持高效的而不会牺牲**framework**和**widget**的可读性。

**面向对象**虽然我们可以使用非面向对象的语言，但这意味着要重新解决几个难题。另外，绝大多数开发人员都具有面向对象开发的经验，因此更容易学习如何使用**Flutter**进行开发。

**可预测，高性**。借助**Flutter**，我们希望使开发人员能够快速创建流畅的用户体验。为了实现这一点，我们需要能够在每个动画帧中运行大量的代码。这意味着我们需要一种既能提供高性能又能提供可预测性能的语言，而不会出现会导致丢帧的周期性暂停。

**快速内存分配** Flutter框架使用函数式流，它很大程度上依赖于底层的内存分配器，从而有效地处理小的、短期的内存分配会非常重要，所以在缺乏此功能的语言中**Flutter**无法有效地工作。

# Flutter应用程序性能如何？

---

**Flutter**应用程序性能非常出色。**Flutter**旨在帮助开发人员轻松实现恒定的60fps。**Flutter**应用程序通过本机编译的代码运行 - 不涉及解释器。这意味着**Flutter**应用程序可以快速启动并执行。**Flutter**开发体验如何？编辑和刷新之间有多长时间？**Flutter**实现了热重载开发循环。您可以在设备或模拟器上实现亚秒级重载。**Flutter**的热重载是有状态的，这意味着应用程序状态在重载后仍然会保留。所以您可以在应用程序中各个页面快速迭代开发，而无需在每次重新加载后都要从主屏幕重新开始。

## Flutter的优势

---

综上所述，总结来说，**Flutter**的优势有如下几个方面

**性能强大，流畅** **Flutter**对比**weex**和**react native**相比，性能的强大是有目共睹的。基于**dom树**渲染原生组件，很难与直接在原生视图上绘图比肩性能，**Google**作为一个轮子大厂，直接在两个平台上重写了各自的**UIKit**，对接到平台底层，减少UI层的多层转换，UI性能可以比肩原生，这个优势在滑动和播放动画时尤为明显。

**路由设计优秀** **Flutter**的路由传值非常方便，**push**一个路由，会返回一个**Future**对象（也就是**Promise**对象），使用**await**或者**.then**就可以在目标路由**pop**，回到当前页面时收到返回值。这个反向传值的设计基本是甩了微信小程序一条街了。弹出**dialog**等一些操作也是使用的路由方法，几乎不用担心出现传值困难

**单例模式** **Flutter**支持单例模式，单例模式的实现也非常简单。单例模式很好的解决了一些问题。相比之下，**js**的单例则并不是一个真正的单例，或者说不是一个简单的单例，这也是受限于**js**所运行的环境。单例模式并不总是合理的，容易被滥用。但是在**App**的初期开发中，往往一个容易实现的单例可以帮助我们快速完成一些逻辑的搭建。

**优秀的动画设计** **Flutter**的动画简单到不可思议，动画对象会根据屏幕刷新率每秒产生很多个（一般是60个）浮点数，只需要将一个组件属性通过补间（**Tween**）关联到动画对象上，**Flutter**会确保在每一帧渲染正确的组件，从而形成连贯的动画。这种十分暴力的操作在**Flutter**上却看不到明显的卡顿，这也是**Flutter**的一个魔力所在。相比之下其他跨平台框架几乎不能设计动画.....往往会遭遇非常严重的性能问题。

**UI跨平台稳定** **Google**直接在两个平台上在底层重写了**UIKit**，不依赖于**Css**等外部解释器，几乎不存在UI表达不理想，渲染不正常的情况，可以获得非常稳定的UI表达效果。**Css**换个浏览器就有不同的表现，基于**Css**的跨平台框架很难获得稳定的UI表现。

**可选静态的语言，语言特性优秀** **Dart**是一个静态语言，这也是相对于**js**的一个优势。**Dart**可以被编译成**js**，但是看起来更像**java**。静态语言可以避免错误，获得更多的编辑器提示词，极大的增加可维护性。很多**js**库也已经用**ts**重写了，**Vue3.0**的底层也将全部使用**ts**编写，静态语言的优势不言而喻。

## 最后

---

当然，**Flutter**也有一些不足，还有和**React Native**的比较我会在下一篇文章当中详细介绍介绍。

**Flutter**已经是**Top20**的软件库，通过接下来的一系列的文章，希望我和大家一起来学习**Flutter**，一起进步，一起有所收获，掌握未来技术主流的主动权！

有什么好的建议，意见，想法欢迎给我留言！

### 参考

<https://www.jianshu.com/p/c51fc925bfd1> <https://flutterchina.club/>