

# Flutter 基础篇

在 Flutter 基础篇中，我们将了解到 Flutter 的发展历史，Flutter 的特性，以及 Flutter 的基础开发知识。

## Flutter 是什么？

Flutter 是 Google 开源的新一代跨平台 UI 框架。不同于其他我们熟知的移动端跨平台方案，Flutter 更像游戏引擎，因为 Flutter 有自己的渲染引擎：我们在 Flutter 上写了界面后，Flutter 会在自己的 canvas 上渲染，移动端不负责绘制。

## Flutter 的历史

- 前身 (2015.4.28)

Flutter 的前身是 2015 年在 Dart Developer Summit (Dart 开发者峰会) 上发布的 Sky，Sky 使用 Dart 开发，可以在 Android 上开发高达 120FPS 的应用。

- 第一个版本 (2017.5.12)

2017年5月12号，Flutter 发布了第一个版本 v0.0.6，在 GitHub 上可以看到发布的记

录：<https://github.com/flutter/flutter/releases/tag/v0.0.6>  
(<https://github.com/flutter/flutter/releases/tag/v0.0.6>)

- alpha 阶段 (2017.5.17)

在 2017 Google I/O 上 Google 对外介绍了 Flutter 的存在，此后 Flutter 一直处在 alpha 阶段。

- 首个 beta 版本 (2018.2.27)

2018 年 2 月 27 号的 MWC (Mobile World Congress, 世界移动通信大会) 上, Flutter 发布首个 beta 版本, Flutter 开始进入 beta 阶段, 发布频率明显加快。

- beta 阶段 (2018.5.8)

2018 Google I/O 大会上, Flutter 发布了 beta3。

- 首个发布预览版 (2018.6.21)

2018 GMTC 大会上, Flutter 发布了首个发布预览版。

- 第二个发布预览版 (2018.9.19)

2018 年上海的 GDD (Google Developer Days) 上, Flutter 发布了第二个发布预览版。

- 首个稳定版 1.0 (2018.12.4)

2018 年的 Flutter Live 上, Flutter 正式发布了 1.0 稳定版, 标志着 Flutter 已经完善, 完全可以投入生产环境, 开始进入新的阶段。

- 第二个稳定版 1.2.1 (2019.2.26)

2019 年 2 月 26 号的 MWC 上, Flutter 发布第二个稳定版本 1.2.1。

- 截止目前最新的稳定版本 1.5.4-hotfix.2 (2019.5.7)

2019 年 5 月 7 号的 Google I/O 上, Flutter 发布了第三个稳定版本 1.5.4-hotfix.2。

## Flutter 的特性

## 1. 跨平台

Flutter 是 Google 新一代的跨平台 UI 框架，除了默认支持的 Android 和 iOS，经过扩展后还支持：

- PC 端：Windows、Linux、MacOS。
- WEB 端：WEB 端已经发布了技术预览版，GitHub 地址：[https://github.com/flutter/flutter\\_web](https://github.com/flutter/flutter_web) ([https://github.com/flutter/flutter\\_web](https://github.com/flutter/flutter_web))。

## 2. 高性能

- 界面可以达到 60 FPS，渲染效率和用户体验堪比 Native。
- Flutter 应用在 Release 下会编译成 Native ARM code，使得跨平台的代码运行效率很高。
- Native 和 Flutter 之间是通过 JNI 桥接的，也使得 Native 和 Flutter 之间交互的损耗很低。

## 3. 丰富的 UI 组件

Flutter 内置众多精美的 Material Design (Android风格) 和 Cupertino (iOS风格) 的小部件，而且 UI 样式高度还原 Native 的样式，可以让我们直接使用，而且这些组件都很方便组合，还有丰富的动画库。

## 4. 学习成本低

Flutter 的学习成本很低，屏蔽了底层的很多实现，可以很快的入手。这也带来一个问题，就是没有办法深入 Flutter 的底层，这也是本教程的目的，除了 Flutter 的使用，也会深入分析 Flutter 的底层实现。

## 5. 开发效率高

Flutter 拥有丰富的 UI 部件，而且支持 Hot Reload，同时 Flutter 使用 Dart 语言开发，Dart 具备很多新的语言特性，可以使得开发速度很快。

## 6. 和任何平台的交互都很方便

由于 Flutter 的架构设计，使得 Flutter 可以在任何平台扩展，也可以和平台很方便的交互，从而可以让 Flutter 使用平台的功能。

## 7. Write once,Run everywhere

只要写一份代码，就可以构建出 Android 和 iOS 两个平台的 APP，体验能达到和原生应用一致，真正做到「Write once , Run everywhere.」

# Flutter 的开发语言

Flutter 使用 Dart 语言开发。

Dart 语言对于大家来说都比较陌生，但是不要担心，Dart 语言功能强大而且很容易学会，本小册也有 Dart 语法的介绍。

# Flutter 的四种版本

Flutter 源代码在 git 上有四个分支，对应着 Flutter 的四种版本，这四种版本，稳定性依次提高，但新特性却依次减少：

## 1. master

master 分支上的代码是最新的，所以包含新的特性，但是 master 分支上的代码没有经过测试，很可能会出现各种各样的 bug。

## 2. dev

Google 工程师会定期将 master 分支的代码滚动到 dev 分支上，在执行滚动的过程中，会对代码进行 Google 的内部测试，所以 dev 渠道是通过测试的最新的构建。但并不意味着 dev 版本不会有 bug，因为这个测试只是最基础的测试，一旦发现有严重的阻塞性的 bug，就会把这个 dev 版本列为 [Bad Builds \(https://github.com/flutter/flutter/wiki/Bad-Builds\)](https://github.com/flutter/flutter/wiki/Bad-Builds)，而且这个版本也不会被选为 beta 版本。

## 3. beta

beta 渠道的更新频率是一个月，每个月初，Flutter 团队会从前一个月左右的 dev 版本中选取最佳版本作为 beta 版发布，这个版本最起码在 dev 分支上使用一周的时间，且没有新的严重的 bug。

## 4. stable

stable 渠道的更新频率是一个季度，每个季度发布一次或几次版本，但 stable 的发布频率是不确定的。stable 版本从 beta 分支中选出，只有当 Flutter 团队确定 beta 分支中有一个非常稳定非常好的构建时，而且通常 stable 版本的发布会在对外的公共活动中公布，例如 Flutter Live。

目前 stable 三个版本：1.0、1.2.1 和 。

当我们用于正式的生产环境时，一定要选择 stable 的版本。

具体版本信息请查看如下的页

面: <https://flutter.dev/docs/development/tools/sdk/archive>  
(<https://flutter.dev/docs/development/tools/sdk/archive>)