



下载APP



## 02 | 预习篇 · Dart语言概览

2019-07-01 陈航

Flutter核心技术与实战

[进入课程 >](#)



**讲述：陈航**

时长 12:13 大小 9.80M



你好，我是陈航。

我们知道，Flutter 开发框架采用的开发语言是 Dart，所以要用好这个框架，我们必须搞清楚 Dart 语言。

关于新技术的学习，一直以来我都非常认同一个观点：千万不要直接陷入细节里，你应该先鸟瞰其全貌，这样才能从高维度理解问题。所以，为了帮助你更高效地掌握 Dart，以最快的速度具备开发一款 Flutter 应用的能力，今天这篇文章，我会先从 Flutter 开发的角度，和你介绍 Dart 语言出现的历史背景、特性以及未来。

然后，我会在本专栏的“Dart 基础”模块，与你详细分享它的特性、基础语法、类型变量、函数等知识，并和你分享一个使用 Dart 的综合案例，帮你学懂、学会这门语言。

如果你已经对 Dart 有一个初步印象了，也可以跳过这篇预习文章，直接学习后面的内容。

## Dart 是什么？

2011 年 10 月，在丹麦召开的 GOTO 大会上，Google 发布了一种新的编程语言 Dart。如同 Kotlin 和 Swift 的出现，分别是为了解决 Java 和 Objective-C 在编写应用程序的一些实际问题一样，Dart 的诞生正是要解决 JavaScript 存在的、在语言本质上无法改进的缺陷。

那么，**JavaScript 到底有哪些问题和缺陷呢？**JavaScript 之父布兰登·艾克（Brendan Eich）曾在一次采访中说，

JavaScript “几天就设计出来了”。

概括来说，他的设计思路是这样的：

借鉴 C 语言的基本语法；

借鉴 Java 语言的数据类型和内存管理机制；

借鉴 Scheme 语言，将函数提升到“第一等公民”（first class）的地位；

借鉴 Self 语言，使用基于原型（prototype）的继承机制。

所以，**JavaScript 实际上是两类编程语言风格的混合产物：（简化的）函数式编程风格，与（简化的）面向对象编程风格。**

由于设计时间太短，一些细节考虑得不够严谨，导致后来很长一段时间，使用 JavaScript 开发的程序混乱不堪。出于对 JavaScript 的不满，Google 的程序员们决定自己写一个新语言来换掉它，所以 Dart 的最初定位也是一种运行在浏览器中的脚本语言。

而为了推广 Dart，Google 甚至将自己的 Chrome 浏览器内置了 Dart VM，可以直接高效地运行 Dart 代码。而对于

普通浏览器来说，Google 也提供了一套能够将 Dart 代码编译成 JavaScript 代码的转换工具。这样一来，开发者们就可以毫无顾虑地使用 Dart 去开发了，而不必担心兼容问题。再加上出身名门，Dart 在一开始就赢得了部分前端开发者的关注。

但，JavaScript 的生命力似乎比预想的更强大。

原本 JavaScript 只能在浏览器中运行，但 Node.js 的出现让它开始有能力运行在服务端，很快手机应用与桌面应用也成为了 JavaScript 的宿主容器，一些明星项目比如 React、React Native、Vue、Electron、NW (node-webkit) 等框架如雨后春笋般崛起，迅速扩展了它的边界。

于是，JavaScript 成为了前后端通吃的全栈语言，前端的开发模式也因此而改变，进入了一个新的世界。就如同 Atwood 定律描述的：凡是能用 JavaScript 写出来的系统，最终都会用 JavaScript 写出来 (Any application that can be written in JavaScript, will eventually be written in JavaScript.) 。

JavaScript 因为 Node.js 焕发了第二春，而 Dart 就没有那么好的运气了。由于缺少顶级项目的使用，Dart 始终不温不火。2015 年，在听取了大量开发者的反馈后，Google

决定将内置的 Dart VM 引擎从 Chrome 移除，这对 Dart 的发展来说是重大挫折，替代 JavaScript 就更无从谈起了。

但，Dart 也借此机会开始转型：在 Google 内部孵化了移动开发框架 Flutter，弯道超车进入了移动开发的领域；而在 Google 未来的操作系统 Fuchsia 中，Dart 更是被指定为官方的开发语言。

与此同时，Dart 的老本行，浏览器前端的发展也并未停滞。著名的前端框架 Angular，除了常见的 TS 版本外，也在持续迭代对应的 Dart 版本 [AngularDart](#)。（不过不得不说的是，这个项目的 star 一直以来只有可怜的 1,100 出头）。

也正是因为使用者不多、历史包袱少，所以在经历了这么多的故事后，Dart 可以彻底转变思路，成为专注大前端与跨平台生态的语言。

接下来，我们就从 Flutter 开发的视角，聊聊 Dart 最重要的核心特性吧。

## Dart 的特性

每门语言都有各自的特点，适合自己的才是最好的。

作为移动端开发的后来者，Dart 语言可以说是集百家之长，拥有其他优秀编程语言的诸多特性和影子，所以对于其他语言的开发者而言，学习成本无疑是非常低的。同时，Dart 拥有的特点则恰到好处，在对 Flutter 的支持上做到了独一无二。所以，Dart 成了 Flutter 的选择。

下面，我就和你详细分享下它的核心特性。

## **JIT 与 AOT**

借助于先进的工具链和编译器，Dart 是少数同时支持 JIT (Just In Time, 即时编译) 和 AOT (Ahead of Time, 运行前编译) 的语言之一。那，到底什么是 JIT 和 AOT 呢？

语言在运行之前通常都需要编译，JIT 和 AOT 则是最常见的两种编译模式。

JIT 在运行时即时编译，在开发周期中使用，可以动态下发和执行代码，开发测试效率高，但运行速度和执行性能则会因为运行时即时编译受到影响。

AOT 即提前编译，可以生成被直接执行的二进制代码，运行速度快、执行性能表现好，但每次执行前都需要提前编译，开发测试效率低。

总结来讲，在开发期使用 JIT 编译，可以缩短产品的开发周期。Flutter 最受欢迎的功能之一热重载，正是基于此特性。而在发布期使用 AOT，就不需要像 React Native 那样在跨平台 JavaScript 代码和原生 Android、iOS 代码之间建立低效的方法调用映射关系。所以说，Dart 具有运行速度快、执行性能好的特点。

那么，**如何区分一门语言究竟是 AOT 还是 JIT 呢？**通常来说，看代码在执行前是否需要编译即可。如果需要编译，通常属于 AOT；如果不需要，则属于 JIT。

AOT 的典型代表是 C/C++，它们必须在执行前编译成机器码；而 JIT 的代表，则包括了如 JavaScript、Python 等几乎所有的脚本语言。

## 内存分配与垃圾回收

Dart VM 的内存分配策略比较简单，创建对象时只需要在堆上移动指针，内存增长始终是线性的，省去了查找可用内存的过程。

在 Dart 中，并发是通过 Isolate 实现的。Isolate 是类似于线程但不共享内存，独立运行的 worker。这样的机制，就可以让 Dart 实现无锁的快速分配。

Dart 的垃圾回收，则是采用了多生代算法。新生代在回收内存时采用“半空间”机制，触发垃圾回收时，Dart 会将当前半空间中的“活跃”对象拷贝到备用空间，然后整体释放当前空间的所有内存。回收过程中，Dart 只需要操作少量的“活跃”对象，没有引用的大量“死亡”对象则被忽略，这样的回收机制很适合 Flutter 框架中大量 Widget 销毁重建的场景。

## 单线程模型

支持并发执行线程的高级语言（比如，C++、Java、Objective-C），大都以抢占式的方式切换线程，即：每个线程都会被分配一个固定的时间片来执行，超过了时间片后线程上下文将被抢占后切换。如果这时正在更新线程间的共享资源，抢占后就可能导致数据不同步的问题。

解决这一问题的典型方法是，使用锁来保护共享资源，但锁本身又可能会带来性能损耗，甚至出现死锁等更严重的问题。



这时，Dart 是单线程模型的优势就体现出来了，因为它天然不存在资源竞争和状态同步的问题。这就意味着，一旦某个函数开始执行，就将执行到这个函数结束，而不会被其他 Dart 代码打断。

所以，**Dart 中并没有线程，只有 Isolate（隔离区）**。Isolates 之间不会共享内存，就像几个运行在不同进程中的 worker，通过事件循环（Event Looper）在事件队列（Event Queue）上传递消息通信。

## 无需单独的声明式布局语言

在 Flutter 中，界面布局直接通过 Dart 编码来定义。

Dart 声明式编程布局易于阅读和可视化，使得 Flutter 并不需要类似 JSX 或 XML 的声明式布局语言。所有的布局都使用同一种格式，也使得 Flutter 很容易提供高级工具使布局更简单。

开发过程也不需要可视化界面构建器，因为热重载可以让我们立即在手机上看到运行效果。

## Dart 的未来

那么，在这样的背景下诞生的 Dart，今后发展会怎样呢？

Dart 是一个优秀而年轻的现代语言，但一种编程语言并不是搞定了引擎和开发者接口就算完成了，而是必须在这个语言得以立足的库、框架、应用程序等“生态”都成熟起来之后，其价值才会真正开始体现。而要走到这一步，通常需要花上数年的时间。

目前，基于 Dart 语言的第三方库还很少，并且质量一般，不过值得庆幸的是，因为 Flutter 和 Fuchsia 的推动，Dart SDK 更新迭代的速度快了很多，开发者的热情也急剧增长，Dart 生态增速很快。

毕竟，在 Dart 社区目前最顶级的产品就是 Flutter 和 Fuchsia 了，因此 Dart 开发者主要以 Flutter 开发者居多，当然了也有用 Dart 开发浏览器前端的开发者，但人数并不多。所以，**我觉得 Dart 是否能够成功，目前来看主要取决于 Flutter 和 Fuchsia 能否成功。而，Flutter 是构建 Fuchsia 的 UI 开发框架，因此这个问题也变成了 Flutter 能否成功。**

正如我在开篇词中提到的，Flutter 正式版发布也就半年多的时间，在 GitHub 上 Star 就已经超过了 68,000，仅落后 React Native 10,000 左右，可见热度之高。

现在，我们一起回到 Flutter 自身来看，它的出现提供了一套彻底的跨平台方案，也确实弥补了当今跨平台开发框架的短板，解决了业界痛点，极有可能成为跨平台开发领域的终极解决方案，前途光明，未来非常值得期待。

至此，我们已经可以清晰地看到，Google 在遭受与 Oracle 的 Java 侵权案后，痛定思痛后下定决心要发展自己的语言生态的布局愿景：Dart 凭借 Flutter 与 Fuchsia 的生态主攻前端和移动端，而服务端，则有借助于 Docker 的火热势头增长迅猛的 Go 语言。

所以说，Google 的布局不仅全面，应用和影响也非常广泛，前后端均有杀手级产品用来构建语言生态。相信随着 Google 新系统 Fuchsia 的发布，Flutter 和 Dart 会以更迅猛的速度释放它们的力量，而 Google 统一前后端开发技能栈的愿望也会在一定程度上得以实现。

## 总结

今天，我带你了解了 Dart 出现的历史背景，从 Flutter 开发者的视角详细介绍了 Dart 语言的各种特性，并分析了 Dart 的未来发展。

Dart 是一门现代语言，集合了各种优秀语言的优点。如果你不了解 Dart 也无需担心，只要有其他编程语言，尤其是 Java、JavaScript、Swift 或 Objective-C 编程经验的话，可以很容易地在 Dart 身上找它们的影子，以极低的成本快速上手。

希望通过这篇文章，你可以先对 Dart 语言有个初步了解，为我们接下来的学习打好基础。在本专栏的“Dart 基础”模块中，我会对照着其他编程语言的特性，和你讲述 Dart 与它们相似的设计理念，帮助你快速建立起构建 Flutter 程序的所需要的 Dart 知识体系。

## 思考题

对于学习 Dart 或是其他编程语言，你有什么困扰或者心得吗？

欢迎你在评论区给我留言分享你的经历和观点，我会在下一篇文章中等你！感谢你的收听，也欢迎你把这篇文章分享给更多的朋友一起阅读。

---

# Flutter 核心技术与实战

来自 Google 的高性能跨平台开发框架

陈航

美团点评高级技术专家



新版升级：点击「 请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 01 | 预习篇 · 从0开始搭建Flutter工程环境

下一篇 03 | 深入理解跨平台方案的历史发展逻辑

## 精选留言 (24)

 写留言



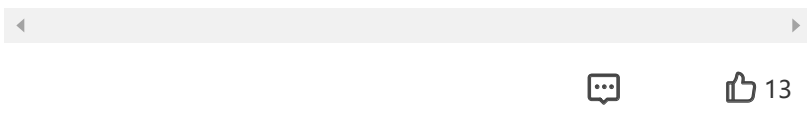
荣

2019-07-01

入门简单，精通难. 如何掌握一门语言的精髓呢？

展开 ∨

作者回复: 首先明确, 我们学习一门编程语言并不是为了成为语言专家, 而是为了解决实际问题。我认为通过1-2天的学习, 在了解一门语言是如何表达信息和处理信息之后, 语言的基本套路就搞懂了, 然后可以根据实际问题上手去实践, 找相应成熟的框架和库, 边做边学, 加深理解。



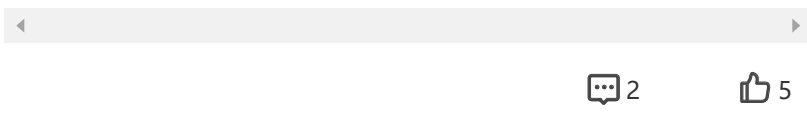
**创建昵称**

2019-07-02

flutter现在计量单位是什么dp还是px, 屏幕适配怎么解决!

展开 ∨

作者回复: 通常是dp



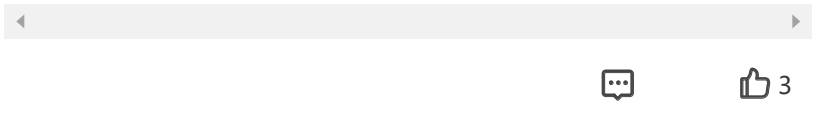
**海斌**

2019-07-02

1. 控件布局, 写法和可读性给我感觉还是不怎么习惯。
2. 没有可视化的布局工具, 手写布局给人感觉还是比较痛苦。
3. 不同android设备的兼容性, 这个也是困扰我的问题之

一，在oppo,vivo上还好好的，在华为手机上就显示不出来。

作者回复: 这种情况一般是布局约束写的有问题，建议多试试几种不同的分辨率，也关注下运行时异常



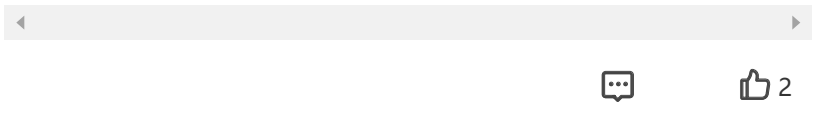
**Aron**

2019-07-03

原生与flutter混合开发会介绍吗？毕竟大多数项目都是原生的。

展开 ∨

作者回复: 会的

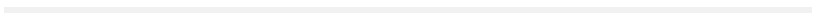


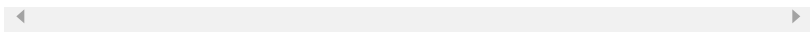
**公众号：业余草**

2019-07-02

Flutter 是不是代表着未来，这是一个后起之秀啊。看的我都想写一些入门教程了！

作者回复: 加油





👍 2

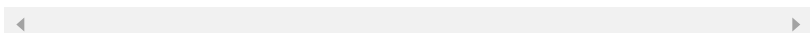


**红穹**

2019-07-02

函数式编程可以多介绍一下吗，fish redux就是函数式设计，代码读起来很困难，毕竟不是Java的面向对象

作者回复: 会介绍一些，但不会专门讲。状态管理那节会主要讲Provider



👍 2



**yasuoyuhao**

2019-07-02

有了原生開發的經驗與概念，使用 flutter 真是有如神助，概念都是互通的。



👍 2



**不拘小节**

2019-07-03

大神请教一个问题，一个ListView，里面套了一个Row，Row有两个Container，第一个Container设置宽高，第

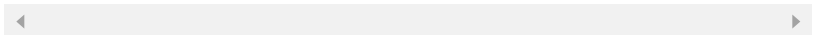


二个设置高度，想第二个宽度自适应撑满ListView，有什么方案吗

展开 ▾

作者回复: 试试IntrinsicHeight，这个组件会根据实际情况把子widget的高度调整它本身实际的高度。

```
IntrinsicHeight(  
  child: Row(children: [  
    Container(width: 100,color: Colors.yellow),  
    Expanded(flex:1,child: Container(height:300,  
color: Colors.amber)),  
  ]),  
)  
);
```

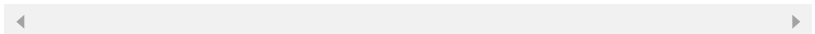


**SHERRAN要好好学习**

2019-07-02

请问网页开发者入手学习flutter有阻碍吗？我是学生，表示context啥的都不太理解。😞

作者回复: 懂的基本的前端概念（视图、组件、状态、生命周期、路由）就可以上手了





1



zr

2019-07-02

快点更新哦

展开 ∨



1



Trust me

2019-07-05

iOSer爱写单例 然而在Flutter里面 isolate 嘿嘿

展开 ∨

作者回复: 单例和Isolate是两回事啊



朱雯

2019-07-05

这一篇其实我没怎么读懂，说说我的问题吧。

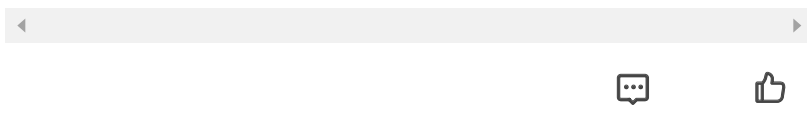
关于单线程问题:一般来说，多线程用于解决并发问题，Isolate这种不共享内存却类似与线程的东西我怎么看都像进程，多线程对比多线程的优势在于它可以节约资源，那么isolate是否是没有多线程优点呢，如果是为什么要放...

展开 ∨

作者回复: 我们需要建立起这么一个概念, 单线程也是可以异步执行的:

- 1.App绝大多数时间都在等待 (等待用户交互、等待网络请求返回、等待I/O结果)
- 2.等待行为并不是阻塞的, 以I/O和网络访问为例, 操作系统均提供了非阻塞型API, 事件回调告知状态

如果对这一块内容不清楚的话, 可以看下之前我在GMTC 2019上的分享: 《Event Loop、Future与Isolate - 单线程模型下的Dart异步编程模式》  
后续分享我也会专门介绍Dart的异步和并发。



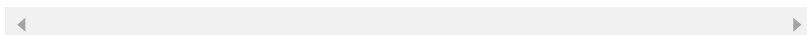
**liyuan**

2019-07-05

要是能用kotlin写flutter就好了。习惯了kotlin, 感觉dart很不好用。

展开 ∨

作者回复: Dart和Kotlin不冲突呀。前端语言的语法和特性都是类似的, 大家都是越来越像





**Spring**

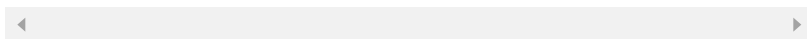
2019-07-04

您好，对于H5+原生以及RN这种框架来说通过webview他们的优势是可以热更新，大量的H5页面都可以动态改变。然而像Flutter这种框架一套代码打包成Android和IOS，但是却失去了热更新这个大的优点。但是往往热更新是很多公司的诉求，我想知道大佬怎么看？

作者回复: Flutter确实不支持动态化，而且目前看从官方层面短期内也不会支持的。理由是发布期AOT是Flutter代码执行效率高的保障，不可能改回JIT；另外苹果也不允许除了JSCore之外还有别的VM可以动态执行逻辑。

这个事情有三个解法：

- 1.Flutter可以内嵌webview支持原有的H5
- 2.Flutter提高了迭代效率节省了研发资源，对于一些需要快速上线的功能，用Flutter开发完再发一个新版本就行了
- 3.业界已经有团队使用JSCore实现动态布局了，预计下半年会有一些较为成熟的方案出现



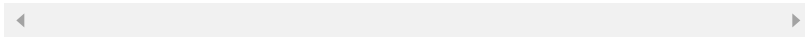
**kaixiao7**

2019-07-03

flutter的嵌套写法让我觉得难受，看起来不爽😞

展开 ∨

作者回复: 最新的IDE插件可以支持" Editor UI Guides "了, 一定程度上解决了可读性问题。

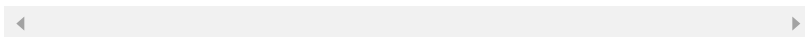


**少年阿埜**

2019-07-03

作为一个后端, 我居然订阅了这个课程。是不是有点飘了  
展开 ∨

作者回复: 加油



**Raptor**

2019-07-03

学习了

展开 ∨

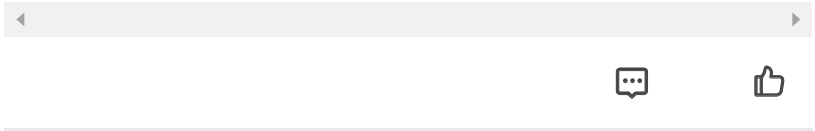


**anynew**

2019-07-03

有没有key的介绍，特别是globalkey，网上介绍的都是抄的，还是没看懂怎么理解与使用

作者回复: 会在性能优化中捎带讲一些，但不会重点讲

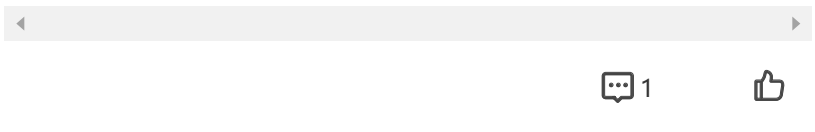


**dragon**

2019-07-03

第三方库对于flutter的支持感觉很少，这会不会影响其发展。目前的解决方案是通过调用原生实现，在后面的课程中会有这方面的介绍吗？

作者回复: 有的



**YOUNG**

2019-07-02

RN最大的优势就是热更新，Flutter会有支持的可能吗？

展开 ∨

作者回复: Flutter高性能的其中一个重要因素就是因为Dart在发布期是AOT，所以Google是不可能改成动态编译的，

另外苹果也不允许除了JSCore之外的其他动态编译/解释执行环境。

不过如果你的诉求是希望能够实现动态界面，目前业界已经有在上层封装DSL实现Flutter动态布局的实践了，可以看看GMTC大会上美团和微信的分享

