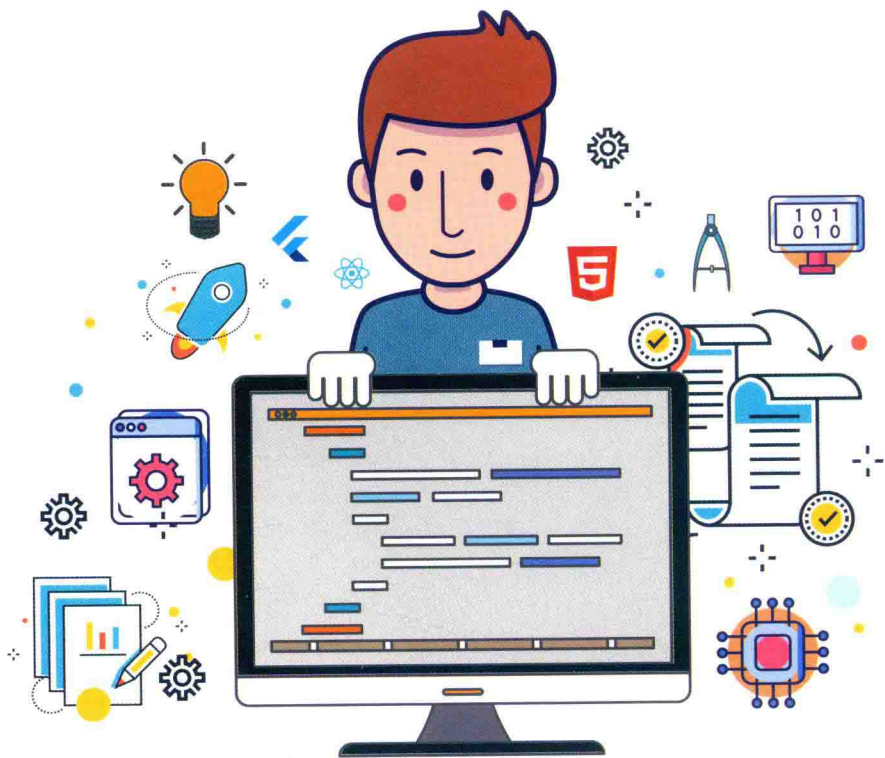




大前端入门指南

任玉刚 曹建波 张玺◎等著

大前端入门指南

任玉刚 曹建波 张玺 李晓萌 吴超 王浩 周及时 史少琰 李京雅◎著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是一本大前端入门书籍，对大前端技术栈进行了全面的讲解，内容涉及 HTML/CSS、JavaScript、React Native、Flutter 和微信小程序等。在内容组织上，采用理论和项目实战相结合的方式，来帮助读者更好地入门大前端。

本书讲解了大前端方向中多个主流的技术栈，针对每个技术栈，本书选取入门必需的内容进行详细讲解。同时，书中还提供了三个难度适中的实战项目，非常适合开发者学习和入门。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目（CIP）数据

大前端入门指南 / 任玉刚等著. —北京：电子工业出版社，2019.6
ISBN 978-7-121-36627-7

I. ①大… II. ①任… III. ①程序设计—指南 IV. ①TP311.1-62

中国版本图书馆 CIP 数据核字（2019）第 100410 号

责任编辑：陈晓猛

印 刷：三河市华成印务有限公司

装 订：三河市华成印务有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱

邮编：100036

开 本：787×980 1/16 印张：26.25

字数：500 千字

版 次：2019 年 6 月第 1 版

印 次：2019 年 6 月第 1 次印刷

定 价：89.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819，faq@phei.com.cn。

前言

从目前的形势来看，大前端的概念越来越火热了，但什么是大前端，如何学习大前端，很多人都不甚了解。

大前端到底是什么呢？直接来说，大前端就是所有前端的统称。在后端眼里，最接近用户的那一层都叫前端，比如 Android、iOS、Web、Watch 等。站在后端的角度，其实并不需要对各个前端都有明确的感知，最好它们能统一起来，这一切就是大前端，除了这些平台，所衍生的跨平台方案及周围生态也是大前端的一部分。

大前端领域有很多技术方案，包括 React Native、Flutter、小程序、PWA，到底谁才是未来胜出的那一个？谁知道呢！

其实事情并没有那么复杂，形势难以捉摸，与其猜测哪个方案会胜出，不如潜下心来，将各个技术方案都学习一下，以不变应万变。互联网上不缺学习资料，但是这些资料往往对初学者不友好，而本书刚好就解决了这个问题，相信读者能从书中收获良多。

本书适合移动开发者和 Web 前端开发者，以及其他对大前端技术感兴趣的读者阅读。

本书内容

本书共 10 章，每章的具体内容如下所述。

第 1 章讲述什么是大前端，并对主流跨平台方案进行简单的介绍。

第 2 章讲述 HTML 和 CSS 的入门知识，详细介绍相关语法及组件用法。

第 3 章讲述 JavaScript 的入门知识，详细介绍了 JavaScript 的语法及应用场景，丰富的小例子更加容易理解和学习。

第 4 章讲述 React Native 的入门知识，详细介绍 React Native 语法和环境搭建，介绍常用组件的用法及用例。

第 5 章针对 React Native 进行项目实战，理论结合实际项目，更快速地学习和理解 React Native。

第 6 章讲述微信小程序的入门知识，详细介绍小程序的环境搭建、架构、组件及常用 API。

第 7 章针对微信小程序进行项目实战，理论结合实际项目，更快速地学习和理解微信小程序。

第 8 章讲述 Flutter 的入门知识，详细介绍 Flutter 的环境搭建、Dart 语法、常用组件，以及如何构建 Flutter 插件。

第 9 章针对 Flutter 进行项目实战，理论结合实际项目，更快速地学习和理解 Flutter。

第 10 章简单地介绍 Weex、PWA 和快应用，帮助读者快速地了解它们。

通过这 10 章的学习，让读者对大前端有一个全面的理解，更快速地入门大前端，少走一些弯路。

致谢

感谢本书的策划编辑陈晓猛，他的高效是本书得以及时出版的一个重要原因。感谢我的公众号读者，他们为本书提了许多宝贵的建议。

由于技术水平有限，书中难免有疏漏，欢迎读者向我反馈：singwhatiwanna@gmail.com。读者也可以关注我的微信公众号，我会定期在上面发布勘误信息。

本书互动地址

微信公众号：玉刚说

QQ 交流群：1026603392

书中源码下载地址：

<https://github.com/android-exchange/cross-platform-guide>

任玉刚 2019 年 5 月于北京

读者服务

轻松注册成为博文视点社区用户（www.broadview.com.cn），扫码直达本书页面。

- **下载资源**：本书如提供示例代码及资源文件，均可在 [下载资源](#) 处下载。
- **提交勘误**：您对书中内容的修改意见可在 [提交勘误](#) 处提交，若被采纳，将获赠博文视点社区积分（在您购买电子书时，积分可用来抵扣相应金额）。
- **交流互动**：在页面下方 [读者评论](#) 处留下您的疑问或观点，与我们和其他读者一同学习交流。

页面入口：<http://www.broadview.com.cn/36627>



目录

第 1 章 大前端概述	1
1.1 什么是大前端	1
1.2 主流跨平台方案简介	2
第 2 章 HTML 和 CSS 入门	6
2.1 HTML 基础	6
2.1.1 HTML 简介	6
2.1.2 基本结构	6
2.1.3 常用标签	8
2.1.4 图片	12
2.1.5 列表	14
2.1.6 超链接	18
2.1.7 表格	23
2.1.8 表单	25
2.2 CSS 基础	31
2.2.1 选择器	31
2.2.2 常用属性	40
2.2.3 盒模型	50
2.2.4 定位	59
2.2.5 浮动	69
2.2.6 FlexBox 布局	77

第 3 章 JavaScript 入门	84
3.1 JavaScript 初探.....	84
3.1.1 搭建开发环境	84
3.1.2 第一个程序	85
3.2 数据类型	87
3.2.1 变量、常量和字面量	87
3.2.2 基本类型和对象类型	88
3.2.3 内置类型	89
3.2.4 类型转换	90
3.2.5 标识符命名	91
3.3 运算符和表达式	92
3.3.1 运算符	92
3.3.2 运算符优先级	98
3.4 控制流	98
3.4.1 逻辑判断	99
3.4.2 循环控制流	100
3.5 函数和闭包	104
3.5.1 函数	105
3.5.2 闭包	106
3.6 程序异常	108
3.6.1 常见异常	108
3.6.2 异常捕获	108
3.6.3 异常抛出	109
3.7 ES6.....	110
3.8 Node.js.....	110
3.8.1 安装 Node.js.....	110
3.8.2 NPM 的使用.....	111
第 4 章 React Native 入门	115
4.1 React 语法基础	115
4.1.1 React 简介	115
4.1.2 搭建 React 开发环境	116

4.1.3	JSX 语法	117
4.1.4	组件	118
4.1.5	组件的生命周期	122
4.2	环境搭建	124
4.2.1	React Native 开发环境搭建	124
4.2.2	WebStorm 代码编辑器环境搭建	127
4.2.3	Visual Studio Code 代码编辑器环境搭建	127
4.2.4	运行 React Native 项目	128
4.3	常用 UI 组件	128
4.3.1	View 组件	128
4.3.2	Image 组件	130
4.3.3	Text 组件	135
4.3.4	TextInput 组件	140
4.3.5	ScrollView 组件	144
4.3.6	ListView 组件	147
4.3.7	FlatList 组件	151
4.3.8	SwipeableFlatList 组件	155
4.3.9	SectionList 组件	158
4.4	网络	161
4.5	导航器 React Navigation	168
4.6	数据存储	183
4.7	原生模块开发	187
4.7.1	Android 原生模块的封装	187
4.7.2	iOS 原生模块的封装	191
第 5 章	React Native 实战	196
5.1	项目创建	196
5.1.1	创建 React Native 项目	196
5.1.2	项目结构介绍	197
5.2	完善功能页面	199
5.2.1	登录注册	199
5.2.2	首页	210

5.2.3 个人中心页面	215
5.2.4 书单详情	218
5.2.5 侧滑页面	222
5.3 打包	225
5.3.1 Android 打包	225
5.3.2 iOS 打包	227
第 6 章 微信小程序入门	229
6.1 认识小程序	229
6.1.1 小程序简介	229
6.1.2 开发前的准备	230
6.1.3 创建小程序	232
6.1.4 代码构成	233
6.1.5 小程序的能力	234
6.2 小程序框架	235
6.2.1 小程序配置	235
6.2.2 小程序的生命周期	236
6.2.3 路由	238
6.2.4 视图层	239
6.2.5 动画	243
6.3 常用组件	243
6.3.1 视图容器	243
6.3.2 基础内容	246
6.3.3 表单组件	247
6.3.4 媒体组件	251
6.3.5 地图	254
6.3.6 web-view	255
6.4 常用 API	255
6.4.1 网络	255
6.4.2 数据缓存	257
6.4.3 位置	257
6.4.4 设备	258

6.4.5	开放接口	259
6.4.6	更新	261
第 7 章	微信小程序实战	262
7.1	项目结构	262
7.2	项目实战	263
7.2.1	数据请求	264
7.2.2	登录与注册页面	265
7.2.3	首页	269
7.2.4	个人中心页面	276
7.2.5	图书详情页面	281
7.2.6	收藏页面	292
7.3	打包上线	295
7.3.1	上传代码	295
7.3.2	提交审核	296
第 8 章	Flutter 入门	299
8.1	前期准备	299
8.1.1	Flutter 简介	300
8.1.2	安装和配置编辑器	300
8.1.3	体验 Flutter	304
8.1.4	Dart 语法	306
8.2	构建用户界面	312
8.2.1	如何布局? 布局文件跑哪去了	312
8.2.2	Widget 组件介绍	313
8.2.3	添加交互	316
8.2.4	手势监测和事件处理	318
8.2.5	在 Flutter 中添加资源和图片	320
8.3	使用设备和 SDK API 相关	321
8.3.1	异步 UI	321
8.3.2	页面跳转和生命周期事件	323
8.3.3	文件读写	324

8.3.4	网络和 HTTP	325
8.3.5	JSON 和序列化	327
8.3.6	数据库和本地存储	327
8.3.7	Flutter 插件	330
8.3.8	封装新 API	331
8.3.9	更多资料	336
第 9 章	Flutter 实战	337
9.1	项目结构	337
9.1.1	结构目录	337
9.1.2	项目概述	338
9.2	项目代码	339
9.2.1	登录、注册页面	339
9.2.2	首页	350
9.2.3	个人中心页面	365
9.2.4	图书详情页面	372
9.2.5	侧滑页面	377
9.3	多平台打包	381
9.3.1	Android 打包	381
9.3.2	iOS 打包	386
第 10 章	Weex、PWA 和快应用	390
10.1	Weex	390
10.1.1	Weex 简介	390
10.1.2	Weex 基础知识	391
10.1.3	Weex 项目之 Hello World	392
10.2	PWA	395
10.2.1	PWA 简介	395
10.2.2	PWA 基础知识	395
10.2.3	PWA 项目之 Hello World	398
10.3	快应用	402
10.3.1	快应用简介	402

10.3.2 快应用基础知识	402
10.3.3 快应用项目之 Hello World.....	406
10.4 小结	408



第 1 章

大前端概述

本章主要介绍大前端的基本概念及主流的几款跨平台方案。

1.1 什么是大前端

大前端技术的发展已经有一段时间了，但这个概念正式映入大家的眼帘是在 2017 年，当时以饿了么为代表的一些企业开始提出大前端的概念。在 2018 年年中，InfoQ 举办了首届全球大前端技术大会（GMTC），在大会中为前后端分离、跨平台和 PWA 等技术设立了专场，这次大会具有重要的意义，它预示了大前端时代的正式到来。

大前端到底是什么呢？直接来说，大前端就是所有前端的统称。在后端眼里，最接近用户的那一层都叫前端，比如 Android、iOS、Web、Watch 等。站在后端的角度，其实并不需要对各个前端都有明确的感知，最好它们能统一起来，这一切就是大前端。除了这些平台，所衍生的跨平台方案及周围生态也是大前端的一部分。

最早期，在 Web 开发过程中，前后端是统一的。后端团队不但要处理业务逻辑，还需要考虑 UI 和特效。在这种情况下，后端工程师和 Web 工程师的协作就变得很重要，协作的好坏直接影响了开发进度和质量。为了解决这一问题，业界出现了前后端分离的概念，分离后，前后端的职责更加明确：前端只负责 UI 展示和特效，而后端则处理业务逻辑并对外提供 JSON 格式的 Model，如图 1-1 所示。

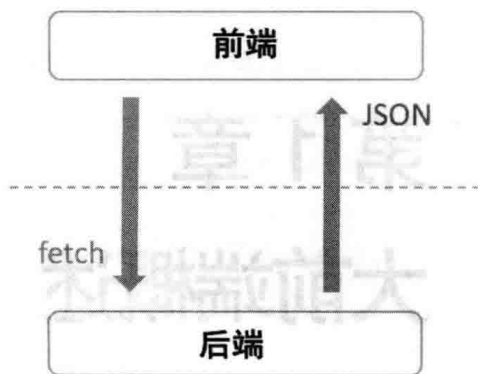


图 1-1

除了前后端分离，在企业开发中还会遇到这样一个头疼的问题。当企业需要上线一个新产品时，服务端只需要开发一次，但是面向用户的客户端却需要开发许多个版本，比如 Android 版、iOS 版、H5 版，甚至还需要开发微信小程序版。每一个版本的开发工作量都是无法复用的，这意味着企业需要付出更多冗余的人力成本。这个时候问题就来了：有没有一种技术可以同时适用于多个平台呢？这种技术就是跨平台技术。有了跨平台技术，各个平台的差异性就被“抹平”了，开发者只需要一套技术栈就可以开发出适用于多个平台的客户端，这就是大前端生态中最重要的一环。

可以看到，大前端的核心概念就是跨平台。虽然大前端的概念已经“炒”了好几年，但相关技术的发展却磕磕绊绊，好在后面 JavaScript 大有“一统江湖”的趋势，目前来看 JavaScript 和 Dart 在跨平台中是首选的语言。JavaScript 是 React Native、Weex、PWA 的语言，而 Dart 则是 Flutter 的语言，下面将对这些技术方案做初步的介绍，更详细的介绍请阅读后续内容。

1.2 主流跨平台方案简介

目前主流的跨平台方案有 React Native、Weex、微信小程序、PWA 和 Flutter，下面一一介绍。

React Native

React Native 是 Facebook 推出的跨平台方案，它使用 JavaScript 作为开发语言。使用 React Native 编写的 App 可以同时运行在 Android 和 iOS 上，极大地节约了开发成本。它在设计原理上和 React 一致，通过声明式的组件机制来搭建丰富多彩的用户界面。令人惊喜的是，React Native 产出的并不是“网页应用”或“HTML5 应用”，也不是“混合应用”，它最终产出的是一

个真正的 App，从使用体验上来说，它和原生应用几乎是没有任何差异的。

React Native 所使用的基础 UI 组件和原生应用完全一致，开发人员要做的就是把这些基础组件使用 JavaScript 和 React 的方式组合起来。React Native 可以快速迭代，比起原生应用漫长的编译过程，开发人员可以在瞬间刷新应用。在开启 Hot Reloading 的情况下，React Native 能在保持应用运行状态的情况下实现代码的热更新。目前 Facebook、Instagram、Uber 等 App 都使用了 React Native。

Weex

Weex 是阿里巴巴推出的解决方案，在各方面都和 React Native 很类似。不同的是 React Native 使用 react.js 作为开发框架，而 Weex 使用 vue.js 作为开发框架。其实，react.js 和 vue.js 堪称目前前端领域最火的两款 JavaScript 框架，它们在易用性和功能性上来说都异常强大。Weex 目前在阿里巴巴的产品（比如淘宝）中得到了广泛的应用。

微信小程序

严格来说，微信小程序算不上跨平台方案，因为它不是运行在手机系统上的，而是运行在微信里，并不是所有的国家和地区都用微信。顾名思义，微信小程序是微信推出的一种技术方案，只要手机可以使用微信 App，就可以使用微信小程序。开发者使用 JavaScript 来开发小程序，通过微信审核后即可上线。用户必须安装微信才能使用小程序，目前微信小程序主要集中在小游戏、工具和生活助手等领域，大型企业也会上线微信小程序，但是一般作为原生 App 的辅助。

PWA

PWA (Progressive Web App) 是一种理念，使用多种技术来增强 Web App 的功能，可以让网站的体验变得更好，能够模拟一些原生功能，比如离线能力、本地缓存和通知推送。在移动端利用标准化框架，让网页应用呈现和原生应用相似的体验。PWA 完全使用前端技术栈，它需要手机系统和浏览器的支持。目前支持 Service Worker 和 Google Play Service 的 Android 手机，以及搭载 iOS 11.3 以上版本的手机可以使用 PWA，很显然 PWA 更适合国外，由于国内手机厂商和浏览器厂商的统一性问题，其在国内的发展遇到了很大的阻力。

Twitter 和 Flipboard 都推出了 PWA，你可以将它放在 Android 或 iOS 的桌面上，使用起来和原生 App 类似，不但可以离线使用，还可以收到后台的推送，如图 1-2 所示。

但是，PWA 不能包含原生 OS 相关代码，PWA 仍然是网站，只是在缓存、通知、后台功能等方面表现更好，使其看起来像是一个原生 App 而已。

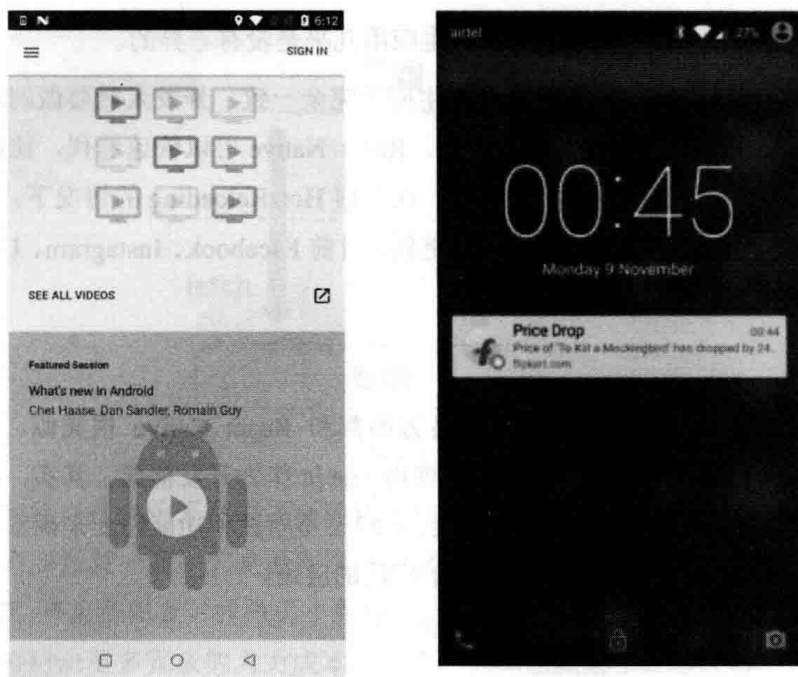


图 1-2

Flutter

Flutter 是 Google 的高性能 UI 框架，它适用于 Android 和 iOS 平台。和上面几个方案都不同，Flutter 另辟蹊径，它没有使用 JavaScript 作为开发语言。Flutter 使用 Dart 语言来开发应用程序，但是它依然允许开发人员使用平台 API、第三方框架及原生代码（Java、Swift 和 Object-C）。

Flutter 可以让开发者快速地开发出漂亮灵活的 UI 界面，用一套代码库就能开发 iOS 和 Android 应用，最关键的是，Flutter 程序的运行效率和原生应用一模一样。Flutter 支持程序的热更新，可以帮助我们快速地进行实验、开发 UI、给程序添加新功能、修复 bug——不管是 Android，还是 iOS 平台。

Flutter 提供了丰富的 UI 组件库，还提供了各种 API，比如手势检测、平滑滚动等，除此之外，Flutter 还预置了 Android 平台的 Material Design 风格，以及 iOS 平台的 Cupertino（iOS-flavor）。通过 Flutter 的响应式框架和许多平台、布局及基础组件，开发者可以快速地构建 UI 界面。而且，开发者还可以使用更多强大的 API（2D、动画、手势和特性等）来完成复杂棘手的 UI 展示，图 1-3 是 Flutter 开发的 App 示例。