

01 | 学习 React Native，你需要掌握哪些知识？

2022-03-28 蒋宏伟

《React Native 新架构实战课》

[课程介绍 >](#)



讲述：蒋宏伟

时长 17:57 大小 16.44M



你好，我是蒋宏伟。

在专栏的第一讲，我们先打个“基础”，让你对即将要学习的 **React Native** 核心基础知识有个心理准备。

这些年，我经常活跃于公司内外的交流群中，和大家讨论一些问题，比如：状态管理怎么选，长列表的性能问题该怎么解决，遇到紧急的线上 **BUG** 该怎么处理，等等。

这些我们都是日常开发时经常会遇到的问题，出现这些问题的其中一个原因，就是对 **React Native** 的基础知识掌握得不够牢固。

当然，这些问题在我们的核心基础篇中都会有答案。不过在第一讲中，我不打算和你说这些细节，而是带你看看 **React Native** 基础知识的地图，让你知道自己该学什么、要学什么，让你对基础知识有一个整体的认知。

基础知识

那所谓的基础知识是什么呢？**我认为能够满足开发业务的最小知识集合都是基础**。比如，语言、框架和一些编程的必备工具，还有 **React Native** 框架和社区提供的核心组件、核心 **API**，再有就是熟悉 workflow 包括搭建环境、开发流程、上线流程，这些都是 **React Native** 的基础知识。

不过，每个人的技术背景不一样、工作年限也不一样，新手可能对大部分的基础知识都不太熟悉。而已经有过多年 **React Native** 开发经验的老手，对有些基础知识的使用方法都已经掌握得差不多了，千人千面。

但我认为，无论是新手还是老手，你都应该系统地、有深度地再学习一遍，把基础打扎实了。我相信，核心基础篇能给你带来更深刻、更系统的认知。

那些需要深层理解的知识，我会把它给你讲清楚、讲透彻，让你弄明白它是怎么来的，又有哪些使用技巧，底层原理又是什么。

在广度上，为了能让你对基础知识有个系统的了解，我为你准备了一张知识地图。即便有些知识，我们没有进行深层的讲解，你也可以根据知识地图和补充材料中的提示，自己搜索。



其中，蓝色背景和蓝色文字的内容就是我们专栏核心基础篇中的重点内容，那些灰色背景、灰色文字的内容就是你需要自己学习和掌握的基础知识。如果灰色部分中有不是很了解的知识，你可以根据知识地图自学一下，遇到了任何问题，都欢迎你给我留言。

你可以看到，在这张知识地图中，我把知识分为了三类。

第一类是开发语言、**React** 框架、开发必备工具这些预备知识。虽然这些预备知识，并不是 **React Native** 本身的重点内容，但这些都是你在开发之前需要掌握的。如果你对 **JavaScript** 不了解，你可以参考 [MDN JS 教程](#) 自行学一下，其中 **npm** 等工具使用起来会比较简单，你可以参考 [npm 中文文档](#) 自行摸索学习。但考虑到有些小伙伴对 **React** 可能没有那么熟悉，所以这方面的预备知识，我还会用三讲的内容进行讲解。

第二类知识是 **React Native** 本身的知识。组件是这类知识中的重点，包括框架提供的组件和社区提供的组件，这方面我会重点讲解。但 **API** 类的知识，本身比较简单，所以我只挑选了样式内容和你进行讲解。至于其他 **API** 知识，你在用的时候，查查 [🔗 React Native 官方文档](#) 就能很快学会了。

第三类知识是工作流中的实操知识。学习实操类知识最好的方法，其实是根据手册一步一步操作，所以这一块我不会细讲。但其中有些能帮你节约自己独立探索时间的经验类的知识，比如 **UI 开发技巧和逻辑调试思路**，我也会和你分享。

专栏的核心基础篇分为 **12** 讲，它是一个总分总的结构，今天这一讲就是对基础知识的整体认知，第 **11** 讲、第 **12** 讲就是对基础知识的项目实战，那中间的几讲就是我为你挑选的几个最重要的基础知识点。我希望通过知识地图和实战案例，把这基础知识点给你串起来，让这些知识能够真正为你所用。接下来，我就给你简要介绍下核心基础篇的这几类知识点。

第一部分：React 框架

我们刚才说，学习 **React Native** 要掌握的第一类知识，包含了 **React** 框架。众所周知，**React Native** 是一个基于 **React** 的原生应用框架。那学习 **React** 最关键的就是，理解 **React 是一个基于组件的、声明式的 UI 框架**，我把这句话拆开和你解释。

先来看 **React** 的第一个特点：**基于组件（Component-Based）**。

最开始 **React** 只能用来写 **Web** 应用，它是在 **2013** 年正式开始开源的。在此之前，业内的大多数 **Web** 框架都是 **MVC** 框架。**MVC** 框架解决了代码大杂烩的问题，它把代码分成了职责分明的三层，**M** 指的是数据模型 **Modal**，**V** 指的视图模板 **View**，**C** 指的是控制器 **Controller**。**MVC** 强调的是数据、视图和逻辑之间松耦合，其文件结构也是按这三类分门别类地组织起来的。

但 **React** 不一样，**React** 强调的是组件的可组合型。在 **React** 中，颗粒度最小的是浏览器提供的 **HTML** 标签，或者是 **React Native** 框架提供的组件。这些颗粒度最小的基础标签、基础组件可以相互嵌套、拼装成一个颗粒度稍大一些的自定义组件。小的自定义组件可以拼装成大的自定义组件，大的自定义组件又可以拼装成页面和应用。

从代码大杂烩，到 **MCV** 三层架构，再到 **React** 的自定义组件，代码组织形式经历很大的变化。**React** 基于组件的代码组织形式，也经历住了时间的考验，证明了它的优秀。

React Native 应用也继承了“基于组件的”特性，它可以让我们将一个大的 UI 页面，拆分成若干个颗粒度更小的自定义组件，这能让我们的应用更容易扩展和维护。

接着我们再来看 **React** 的第二个特点：**声明式（Declarative）**。

声明式是什么呢？所谓的“声明式的”就是，你只需要告诉程序“做什么”（**What**），程序会自动帮你解决“怎么做”（**How**）的问题，这让我们创建复杂的交互应用变得轻而易举。

你使用过的 **HTML** 也算一种“声明式”的语言。不同的是，**HTML** 功能太简单了，而且它也不是图灵完备的语言。但 **React** 不一样，它使用的是 **JavaScript** 来描述 UI 页面的结构，并且在 **JavaScript** 中创建了一种类似 **HTML** 的方言，也就是 **JSX** 语法扩展。

JSX 语法既保留了 **JavaScript** 的灵活、强大和图灵完备特性，也保留了 **HTML** 这类“声明式”语言易写、易读的特点。

还值得多说一句的是，“声明式”是一种编程范式，它与“命令式”编程范式相对立的，它们之间是“是什么”（**What**）和“怎么做”（**How**）的对立关系。“命令式”就好像你自己开车，你得自己控制方向盘。而“声明式”就好像完全自动驾驶的汽车，你告诉它要去哪儿，它就会自己开到哪。

声明式编程是一个大的概念，除了我们前面提到了 **HTML** 所属的领域专属语言（**DSL**）之外，还有函数式编程等子编程范式。熟悉 **React** 的同学可能就知道了，**React** 借鉴了大量的函数式编程的思想。核心基础篇中的函数组件、**hook**，还有我们后面要讲的状态管理工具 **Redux** 和新架构原理，都大量使用了函数式编程的思想。

好了，现在你知道了，**React Native** 的基础是 **React**，而 **React** 又是一个基于组件的、声明式的 UI 框架，因此后面我会花两讲的内容和你介绍，这两个特点背后的基础知识：组件 **Component** 和状态 **State**。

第二部分：核心组件和样式

第二类你需要掌握的基础知识是 **React Native** 的核心组件和样式。

先说样式。**样式决定了页面的“颜值”**，它可以控制组件的颜色、字形、排列、大小，等等。有意思的是，**Web** 中的 **CSS** 是一门博大精深的学问，要学很久，但 **React Native** 中的样式内

容却很少，它核心声明文件也就 600 多行代码。而且根据我的调研，大家也很少使用 React Native 样式工具、样式库来辅助开发，大部分时候使用默认的样式表 StyleSheet 的 “CSS In JS”写法就够用了。

那为什么 React Native 能用这么少的样式，来满足复杂的 UI 开发呢？

关键原因就是，React Native 的组件非常丰富，组件提供的属性满足了那些复杂“颜值”需求。比如，Web CSS 中的粘性定位 `position:sticky`，就可以用 React Native 中滚动组件 ScrollView 的粘性头部 StickyHeaderComponent 属性来满足，而且 React Native 的功能还更强大一些。

因此，我会花很大的篇幅和你重点讲讲 React Native 中的组件，包括图片组件 Image、点按组件 Pressable、输入组件 TextInput、列表组件 RecyclerView。选择这 4 个组件来讲，不仅仅是因为它们用的频率高，我们要学习它的使用方法，**我们也要去研究它的原理，希望它们背后的设计思想能够为我所用。**

学习这 4 种组件，其实各有侧重。学习图片组件时，我们最应该关注的是**加载性能和开发的便捷性应该如何取舍**。比如，内置图片性能是好，但会增加包体积，而且加上容易删掉难；远程图片加载慢了点，但只需要管理远程地址，更新也是非常方便；又比如 Base64 会导致图片体积增加 1/4，但是它在热更新的情况下，能让图片第一时间展示出来。这些方案应该怎么选？又有没有自动化的、工程化的方法帮忙我们管理图片，进一步降低我们选择成本呢？

学习点按组件、输入组件时，我们最应该关注的是**交互体验**。点按按钮很简单，但做好交互体验这件事不简单。一个 App 的体验好不好，PM、UI、UX 设计得好与不好是一方面，但最终还是得靠工程师来实现。比如微信右上角的 + 号按钮，在屏幕上面不是那么好点，微信工程师就把它的可点击范围扩大了一些，让大家容易点中，这些都是值得我们学习的。

学习列表组件时，我们最应该关注的是**FPS 流畅度**，也就是滚动性能。React Native 老版本提供的 FlatList 的滚动性能不是很好，在低端机器上会有点卡，但是社区中提供了性能更好的 RecyclerView，我们得学会用起来。

第三部分：实践经验和实战练习

然后，在项目实战练习之前，我还会和你讲讲有哪些高效开发 UI 的技巧，和调试疑难杂症的实践经验，帮你少走弯路。

核心基础篇的最后，就是项目的实战练习了。整个核心基础篇的目的，是希望能让你搭建一个简易的电商首页。我始终相信，只有实战才能将学到的知识变现为自己能力。

搭建一个电商首页，我们要解决两个问题。第一个问题是，现在电商首页大都是瀑布流形式的，在 **React Native** 如何实现一个高性能的瀑布流组件呢？社区中并没有现成答案，但是我们可以基于社区组件自己动手进行改造，我会带你一步一步实现一个高性能的瀑布流组件。

要解决的第二个问题是，搭建项目的最佳实践是什么？如果你是项目的负责人，你会怎么思考，来保障项目的可扩展性和可维护性？根据我的个人经验和业内的最佳实践，我总结出了一套适合我自己的方案，在《页面实战》这一讲中，我也会把它分享给你。

但我明白，你要把这些知识都学好，还得靠自己在实战中动手、摸索，因此我还会给你留作业，特别是 **React Native** 新手，请你一定要重视作业的重要性。

小结

好了，介绍到这里相信你 already 知道学习 **React Native** 要掌握那些基础知识了，这些知识分为三类：

1. **React** 类知识。**React** 是一个基于组件的、声明式的 UI 框架，而用好 **React** 的关键是用好自定义组件和状态，这也是你学好这门专栏必要的前提条件；
2. **React Native** 本身的知识。这是核心基础篇的重点内容，它不仅包括 UI 怎么写、组件怎么用，你还去了解它们背后的原理，去提高页面的加载性能、交互体验、FPS 流畅度；
3. 实践类知识。这里主要是开发 UI 和调试代码的一些经验技巧，并且我也给你留了一个搭建简易电商网页的任务，我希望你以此实战为你学习核心基础篇的目标，同时也通过这个项目实战检验自己的学习成果，将知识内化为自己的能力。

相信你通过核心基础篇 12 讲的学习，咱们一步一个脚印前进，一定能把基础打扎实了，轻松应对业务开发中的各种挑战。

作业

1. 如果你是新手，你可以参考 [🔗 React Native 官网](#) 和 [🔗 React Native 中文网](#) 搭建一套原生环境。搭建环境是学习的第一步，iOS 或 Android 环境你可以任选其一，先把 **React Native** 在你的电脑里跑起来。

2. 如果你是有经验的 **React Native** 开发者，你可以看一下🔗《大家开发 RN 都用什么？》这份调查表单，看看其他开发者都在用什么，也欢迎你给我留言告诉我，你开发 **React Native** 都用的是是什么。

欢迎在留言区留言，分享你和 **React Native** 的故事。我是蒋宏伟，咱们下节课见。

分享给需要的人，Ta 订阅超级会员，你最高得 **50** 元

Ta 单独购买本课程，你将得 **20** 元

生成海报并分享

👍 赞 4 🗨 提建议

© 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 开篇词 | 为什么要选择 **React Native** 做你的跨端方案？

下一篇 02 | **Component**：搭建静态页面的正确思路是什么？

精选留言 (10)

🗨 写留言



姑苏小沈

2022-03-29

react native在国内搭建开发环境和项目，一坑又一坑，墙不胜墙

共 2 条评论 >

👍 3



大土豆

2022-03-28

抛开主观看法，新项目Flutter的使用率，确实是高于rn的。

共 4 条评论 >

👍 3



天择

2022-03-29

对于国内RN环境搭建，以iOS为例，可以参考这篇博文：<https://www.jianshu.com/p/412d760bcacd>

对于cocoapods的依赖，更新了源依然可能有问题，但可以多试几次pod install，能把所有依赖安装成功。有些是依赖包不是总下不下来，而是偶尔会超时（比如GitHub的资源）。更新源的时候，需要下载新的pod repo，可能会花上相当一段时间。供参考。



2



Forest

2022-03-29

老师 你好 在开发RN项目的过程中 应该如何选择RN生态的第三方库呢 有很多库维护更新不是很积极 甚至issue都得靠使用者（踩坑者）自己修改源码来维持项目的正常运行

共 1 条评论 >



2



焦糖大瓜子

2022-03-29

如果没有科学上网,估计很多人停在了android开发环境搭建🙄

共 1 条评论 >



1



刘琦-Ava Ba

2022-04-07

无论是新手还是老手，你都应该系统地、有深度地再学习一遍，把基础打扎实了；

非常赞同



Shopman

2022-03-30

听起来很像flutter



凌宇之蓝

2022-03-29

干了3年多的ReactNative，现在干Android和Flutter, 哈哈

共 1 条评论 >



李慧文

2022-03-29

本固枝荣，没有基础，何谈优化



woJA1wCgAArt3u3Beza...

2022-03-29

对于现在很火的flutter，老师怎么看待RN和flutter之间的关系的

共 1 条评论 >

