

安装RxSwift的三种不同方式

RxSwift - step by step

← [返回视频列表](#)

预计阅读时间: 15分钟

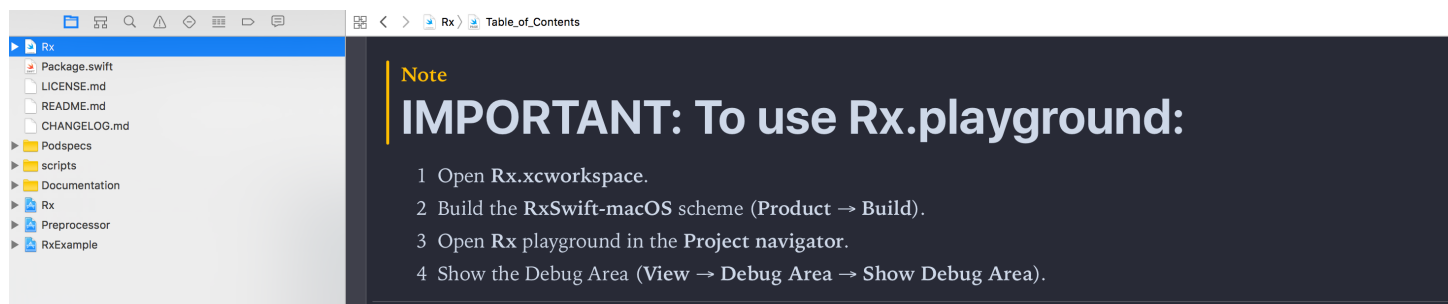
< [PREVIOUS](#)

[NEXT](#) >

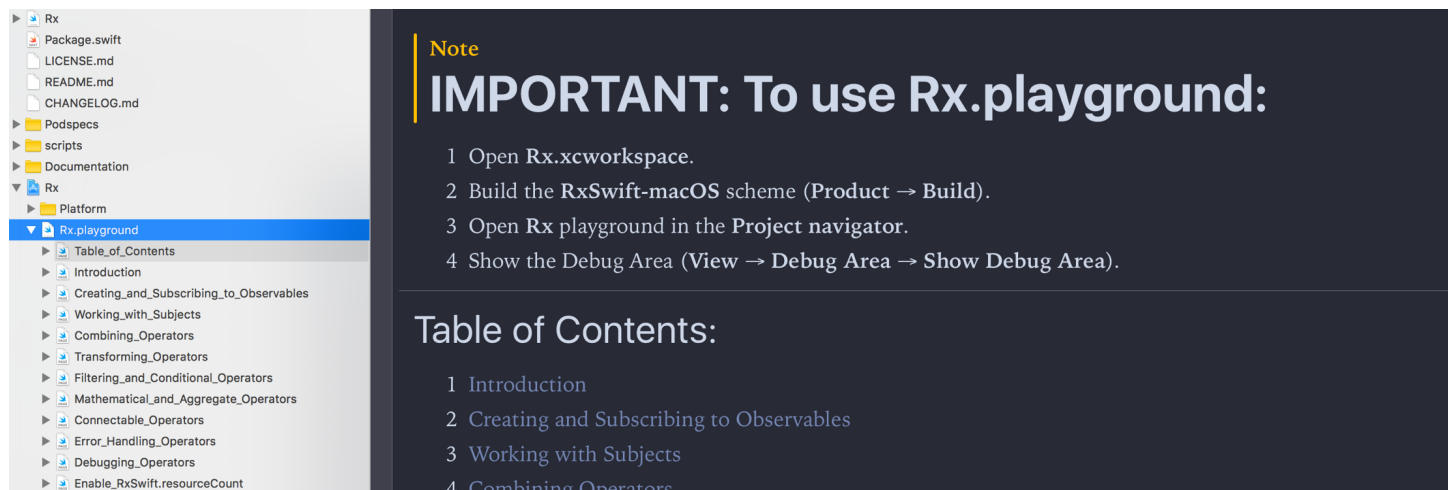
在这一节，我们来看安装RxSwift的几种不同的方法。开始之前，我们先整体了解一下RxSwift这个项目。

了解RxSwift官方项目

如果我们直接下载RxSwift压缩包，解压缩之后，用Xcode打开RxSwift-master/Rx.xcworkspace，可以看到，一共包含了3个项目：



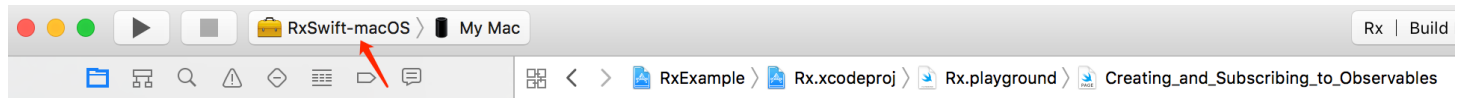
其中，我们感兴趣的，是Rx和RxExample。Rx项目中，包含了完整的RxSwift / RxCocoa / RxBlocking / RxTest实现，以及，一个用于演示RxSwift各种概念的Rx.playground：



Rx.playground

应该说，一旦你理解了reactive programming的思想之后，这份playground文件，是学习RxSwift最直接有效的途径之一了。并且，RxSwift这种通过playground编写文档示例的方式，也值得我们学习和借鉴。

为了可以在Rx.playground中把玩RxSwift，我们需要在Xcode的tool bar上把Target设置成RxSwift-macOS，然后等Xcode自动完成构建（第一次这个过程可能会有点儿长）。



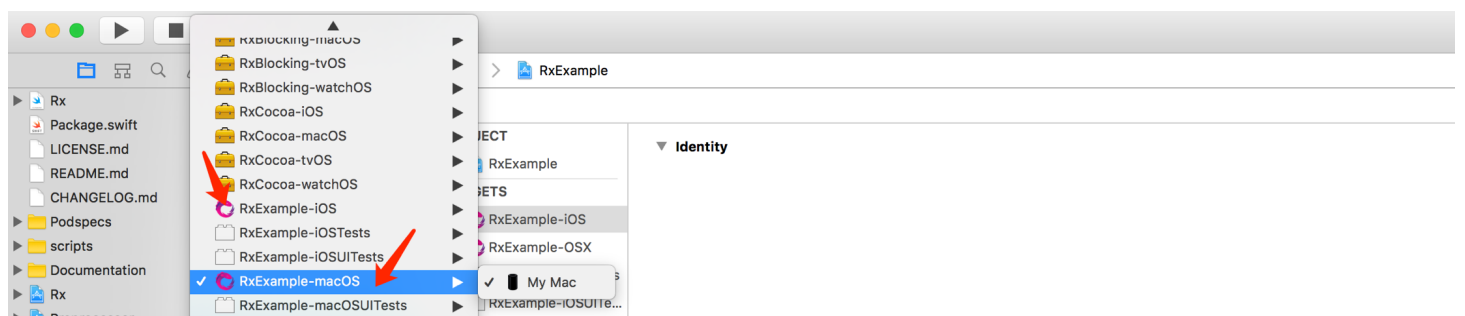
Building完成后，我们就可以选择不同主题的playground文件，在控制台里看到对应示例的执行结果了：



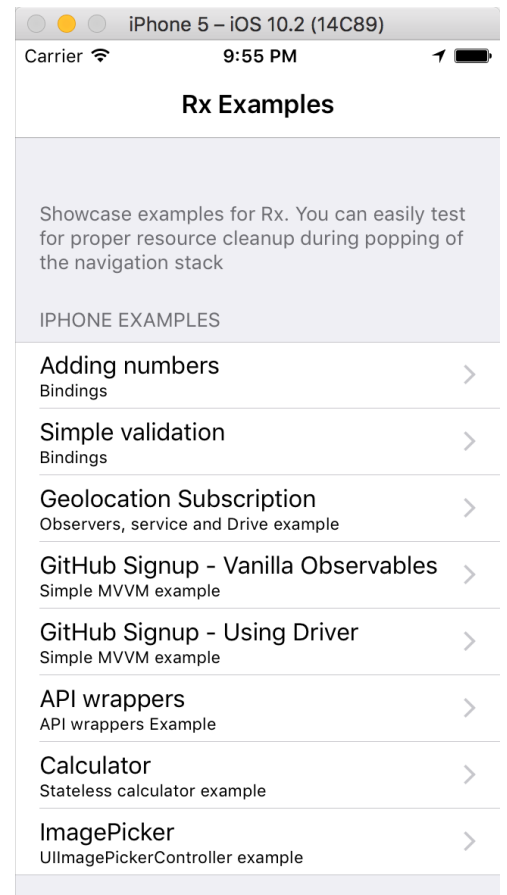
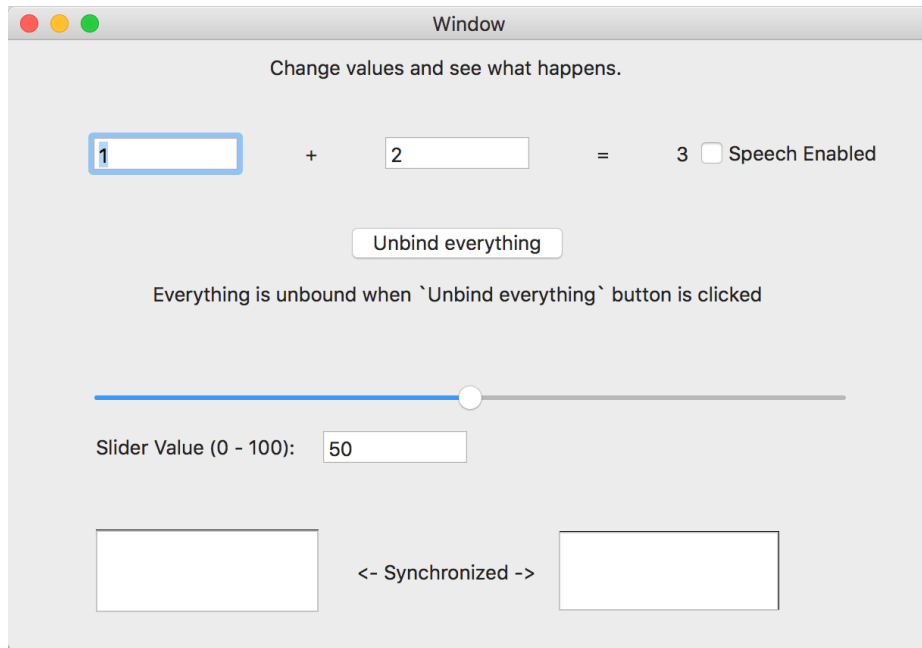
当然，我们现在的重点还不是这些结果的细节。稍后的内容里，我们会详细介绍其中的每个概念。现在，我们只要知道如何使用它就好。

RxExamples

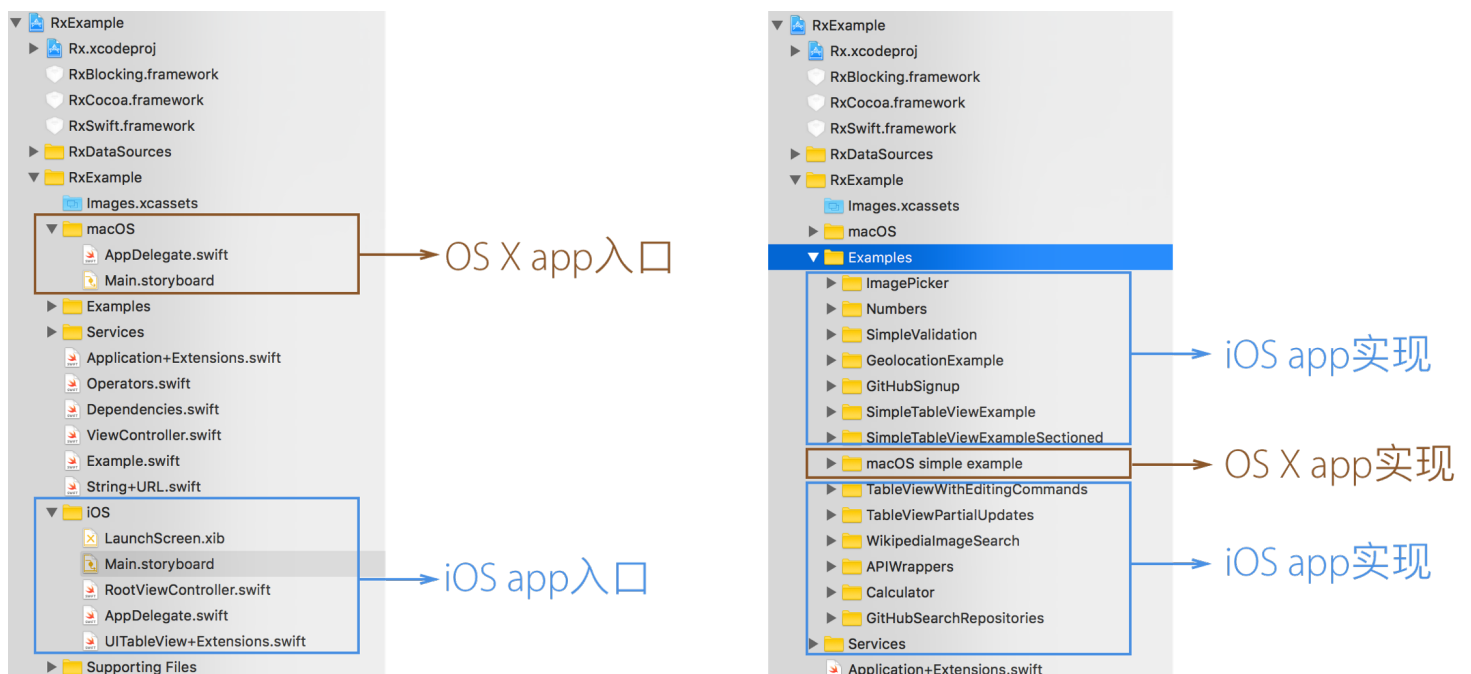
除了一套良好编写的playground之外，RxSwift还提供了一份iOS和macOS app的示例，它们位于RxExample项目中，可以帮助我们进一步了解RxSwift在实际开发中的用法。我们可以通过在Tool bar中如下图所示，切换不同的Target来执行这两个例子：



把它们分别执行起来，看上去是这样的：



而这两个项目的源代码目录是这样的：



在图中，我们分别标记了macOS和iOS对应的入口文件以及代码实现，当大家对RxSwift有了一定的了解之后，就可以自己回过头来翻翻这些官方的例子，会有不少收获和启发。

以上，就是对于RxSwift整个项目文件的一个大体介绍。当然，我们使用RxSwift并不仅仅是为了学习的，最终还是要把它集成到自己的项目中完成开发任务。接下来，我们就来了解安装RxSwift的三种不同的方式。

通过CocoaPods安装

最简单的，当然就是用CocoaPods。如果你还没安装过cocoapods，直接在终端执行`sudo gem install cocoapods`安装一下就好了。

然后，在我们之前创建的FilterNum目录，执行`pod init`，cocoapods会在当前目录新建一个叫做Podfile的文件，我们在其中添加下面的内容：

```
# Uncomment the next line to define a global platform for your project
# platform :ios, '9.0'

target 'FilterNumber' do
  # Comment the next line if you're not using Swift and don't want to use dynamic frameworks
  use_frameworks!

  # Pods for FilterNumber
  pod 'RxSwift',      '~> 3.0'
  pod 'RxCocoa',      '~> 3.0'
end
```

保存退出后，执行`pod install`，等待cocoapods安装完成就好了。完成之后，我们可以在控制台看到类似下面这样的结果：

```
→ FilterNumber git:(master) ✗ pod install
Analyzing dependencies
Downloading dependencies
Installing RxCocoa (3.2.0)
Installing RxSwift (3.2.0)
Generating Pods project
Integrating client project

[!] Please close any current Xcode sessions and use `FilterNumber.xcworkspace` for this project from now on.
Sending stats
Pod installation complete! There are 2 dependencies from the Podfile and 2 total pods installed.
```

Cocoapods会在当前目录下为我们创建一个叫做FilterNumber.xcworkspace的文件，一旦集成了RxSwift之后，我们就要打开这个文件继续我们的开发任务了。要说明的一点是，如果是第一次打开workspace文件，我们直接`import RxSwift` Xcode会提示我们*No such module RxSwift*。这时，只要按`Cmd + B` build一次整个工程，就好了。

通过SPM安装

如果要在非Mac环境下使用RxSwift，SPM应该是你首选的方法。首先，新建一个项目目录，例如：`RxSwift_SPM`；其次，在`RxSwift_SPM`目录中，执行`swift package init --type=executable`。这样，SPM就会在当前目录中创建一个Package.swift文件，在这个文件里，我们添加下面的内容：

```
import PackageDescription

let package = Package(
    name: "RxSwift_SPM",
    targets: [],
    dependencies: [
        .Package(url: "https://github.com/ReactiveX/RxSwift.git", majorVersion: 3)
    ]
)
```

这样，SPM就会知道，我们的项目依赖主版本号为3的RxSwift项目了。保存退出后，执行`swift build`，SPM就会把RxSwift clone下来并自动构建。

然后，我们就可以在`Sources/main.swift`中，`import RxSwift`了。

手动安装

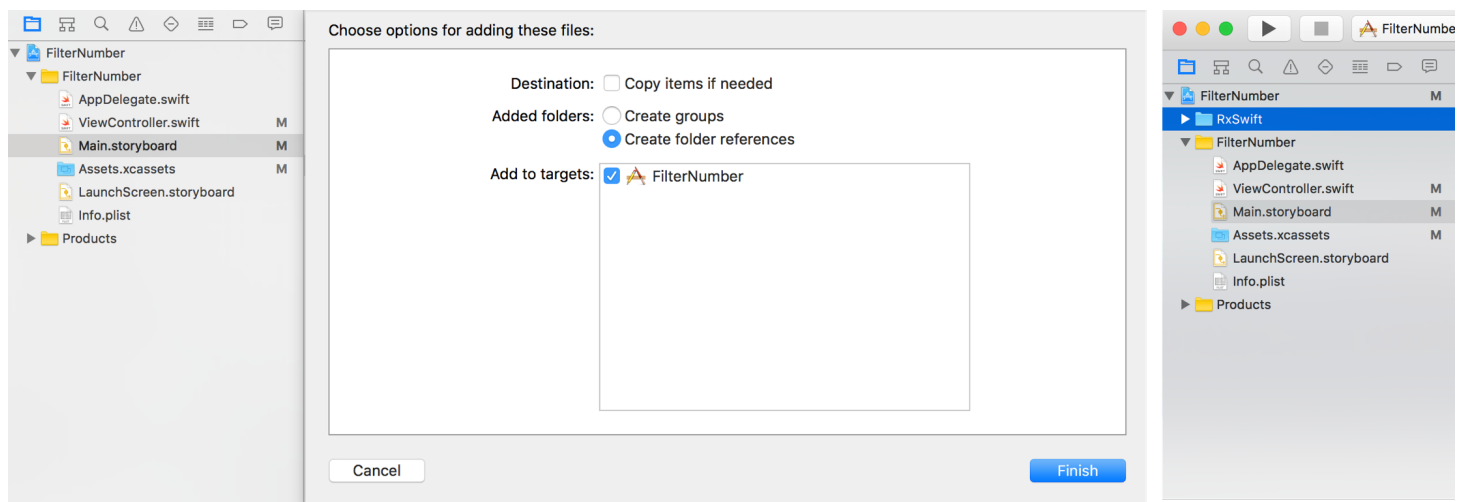
当然，如果你的项目中，不方便使用上面我们提及的两个工具。也可以手动把RxSwift添加到自己的Xcode工程，别担心，这个过程并没有听起来这么麻烦。

首先，在你的项目根目录，执行`git submodule add git@github.com:ReactiveX/RxSwift.git`，让RxSwift以submodule的方式集成到项目目录：

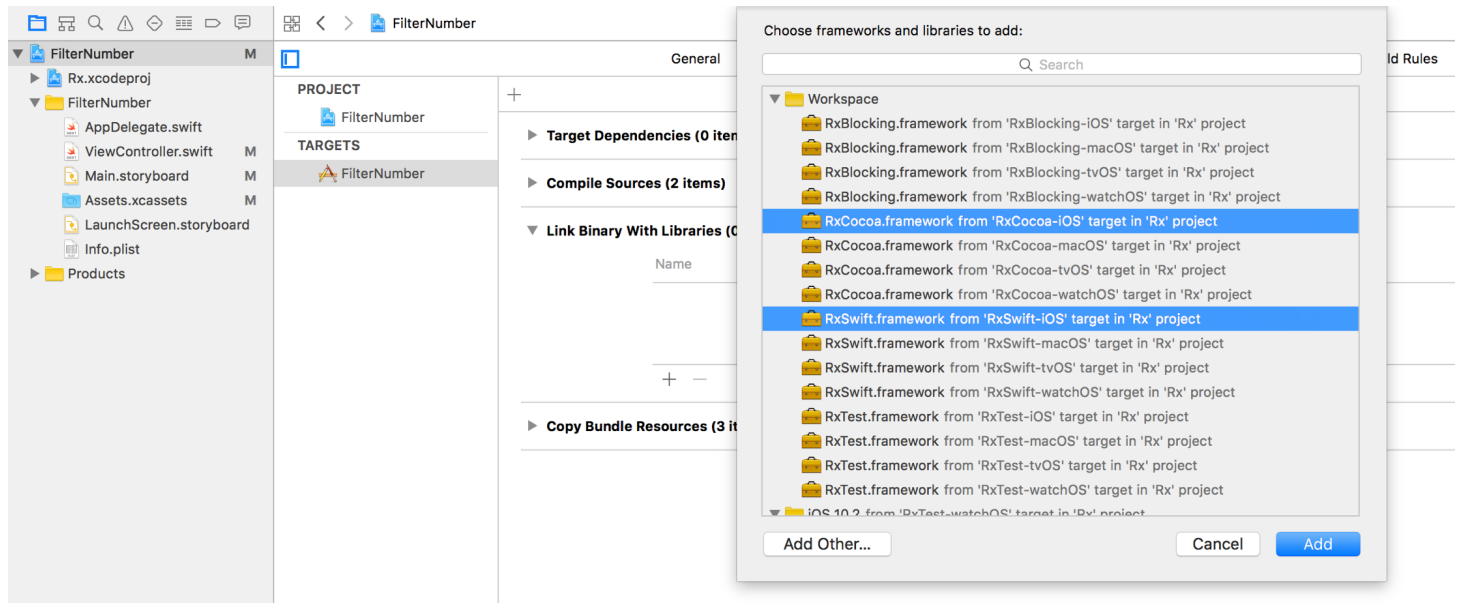
```
→ FilterNumber1 git:(master) X git submodule add git@github.com:ReactiveX/RxSwift.git
Cloning into '/Users/puretears/Desktop/tmp/FilterNumber1/RxSwift'...
remote: Counting objects: 26440, done.
remote: Total 26440 (delta 0), reused 0 (delta 0), pack-reused 26440
Receiving objects: 100% (26440/26440), 8.82 MiB | 1.01 MiB/s, done.
Resolving deltas: 100% (18573/18573), done.
```

完成之后，项目中就会多出一个RxSwift文件夹。

其次，用Xcode打开之前项目文件，例如：`FilterNumber.xcodeproj`，把`RxSwift/Rx.xcodeproj`拖到项目中，Xcode会提示我们处理添加进来的文件，直接点击Finish按钮。这样，在项目中，就会多了一个RxSwift目录：



第三，在TARGETS中，选中FilterNumber，点击Build phases，在Link Binary With Libraries中，点击左下角的+，Xcode就会提示我们要添加进来的程序库。可以看到，在workspace里，一共有16个程序库，分别对应了RxBlocking / RxCocoa / RxSwift / RxTest各自在iOS / macOS / watchOS / tvOS上的版本。这里，我们应该选择iOS对应的版本：



第四，添加完成之后，按Cmd + B重新构建一次，我们就可以在项目中import RxSwift了。

What's next?

以上，就是几种常用的安装RxSwift的方式，完成之后，我们就可以开始逐步了解reactive programming了。下一节里，我们就用若干切实可行的例子，来了解reactive programming中最重要的的一些基本概念。

< **Prev:** 异步事件是以时间为索引的
常量队列

≡ **安装RxSwift的三种不同方式**

Next: RxSwift中的那些“术语”到底在
说什么? >

关于我们

想循序渐进的跟上最新的技术趋势？想不为了学点东西到处搜索？想找个伙伴一起啃原版技术经典书？技术之外，还想了解高效的工作流技巧？甚至，工作之余，想找点儿东西放松心情？没问题，我们用4K开发视频，配以详尽的技术文档，以及精心准备的广播节目，让你渴望成长的技术需求，也是一种享受。

Email Address

10@boxue.io

客户服务

📞 2085489246

相关链接

- › 版权声明
- › 用户隐私及服务条款
- › 京ICP备15057653号-1
- › 京公网安备 11010802020752号

关注我们

在任何你常用的社交平台上关注我们，并告诉我们你的任何想法和建议！



邮件列表

订阅泊学邮件列表以了解泊学视频更新以及最新活动，我们不会向任何第三方公开你的邮箱！

Email address

立即订阅

2019 © All Rights Reserved. Boxue is created by 10 ❤️ 11.

