

课程介绍

珠峰前端架构师技术分享课 <<https://ke.qq.com/course/272058>>

1. 第一个问题，如何安排课程

在写这个课题的时候，我最头疼的不是如何能读懂源码，也不是能讲清楚源码，而是如何能让课程整体连贯，并且能让读者有所收获。

最理想的情况是把按照作者编写的顺序重新实现类似的功能的简版。这需要讲解的人有很强的还原能力或者逻辑思维。

2. 第二个问题，如何处理理论和实践的关系

如果先讲理论，没有例子，我们往往不知道理论如何使用。

如果先讲例子，后讲理论，那我们奇怪例子是如何实现的。在实际的框架中，理论和实践被称之为两个常用的名词：

- 功能：也叫做机制，模块，类库或工具。比如**事件、事物、缓冲池、mixin**等等。
- 层次：也叫做层级，是串联理论的实现。比如**视图层，虚拟DOM**。

2.1 功能：

通俗点讲，功能可以理解成公共代码，通常来说是个独立的可移植模块，可以被多处复用，是框架中的组成部分。也有的大的功能会涉及到其他功能，并且渗透到各个层次之间，比如组件，会涉及**事物、生命周期、mixin、事件**等功能，也会跟各个层级有关系，但这种功能少之又少，大多数功能都能独立使用。

2.2 层次：

而层级则指的是框架的全部数据结构和算法，也会聚合功能，通常不能独立使用，它有以下几个特点：

- 单向调用：也就是说调用方向不能反向，比如业务逻辑层不能调用视图层的代码。
- 不能跨层级
- 底层屏蔽实现细节

react框架并没有遵循传统的三层架构，在它的层次里，只有视图层和控制层。

视图层直接与浏览器打交道，即dom层。

数据（state,props）全部集中在控制层之内，绝大多数逻辑都封装在控制层，也就是virtual dom层，包含我们熟知的事件挂载，setState，生命周期等等逻辑。

3. 为什么学习源码，你能获得什么？

很多学了源码的同学在阅读一段时间源码后，会有一个疑问：我能学到什么？

源码是作者在众多版本改进后的呈现，如果仅是看懂，不加以提炼或思考，很容易在阅读后忘记。

我曾经读过一些框架或类库的源码，不自主的在难以理解的代码上标记中文注释，有可能会借助框架的思想写一些伪码框架。

可到如今，整体流程忘得一干二净，记忆最多的，是源码的设计模式以及思想。

读react的目的：有的人是为了面试，有的人是为了增长经验，也有人是为了跟风或者赶潮流。就我个人而言，有几个目的：

- **读源码是为了读懂源码。**

通过读大型框架的源码，总结出一套适合自己的通用方法。

- **通过开发者的思想，能举一反三。**

比如读到继承，除了我们曾经熟知的原型继承，react还提供了mixins功能，其实它本身也是原型拷贝继承。此外，我们写react时，还用到了hoc、decorator实现类似继承的功能。他们到底有什么优缺点和使用场景，作者为什么这么设计，是我们读源码时应该考虑的事情。

- **解决自身的一些疑惑(setState，事件代理，事物机制，dom-diff等如何实现)。**

使用框架时，我们肯定都有一些疑惑。比如dom-diff对比的到底是什么，props检测是如何实现的，setState为什么是异步的？递归解析组件，是在渲染入口做好，渲染mount每个虚拟节点时做的呢？为什么事件回调的this会丢失？

带着问题去阅读，每当读完一个模块还会有一些新的疑惑。比如读到mount的过程时，我们会发现调用，transition功能，它到底用来做什么用的？然后就解锁了事物的大门，你会发现一个新的概念与思路来实现需求。像这样的疑惑阅读源码时会时常涌现。

- **想重新写一个react-like框架。**

当然，也有非常少数的读者，想重新实现一遍react类似的简版，比如preact，inferno，nerv等。

4. 课前准备

- 事先预习每节课的代码，对整体调用流程有大体了解。
- 思考课前提出的问题。

课程目录

1. **开端我们会讲解思维模式的培养，如何思考以及如何设计框架的基本思路。**
 2. **第二章会分享读框架源码的几种通用方法。**
 3. **介绍框架的设计思路、模块关系以及运行流程，你将会看到react框架蓝本，初步了解作者的设计思想。**
 4. **通过react入口，来逐渐深入react源码，各位同学可以借此熟悉课程的基本形式。**
 5. **讲解react最重要的部分：虚拟dom，无论是界面、数据、diff等都和此层次有密切关系，react的主要逻辑流程以及代码均在此层实现。**
 6. **借助react设计思路，实现一个react-like的简版框架了解整体流程。**
 7. **讲解react的具体功能模块实现，比如mixin继承机制，缓冲池机制，回调队列以及事件、事物等等。**
 8. **react整体设计思想理念。**
-