1. 了解Vue3的源码结构以及基本的知识储备

# 前言

Vue简单好上手的特性使得它运用广泛，我们的部门的项目中PC端的前端均是使用vue.js框架来进行开发。学习并且了解vue.js的源码不仅可以培养我们阅读源码的能力，提升编码能力，同时也能更好的掌握框架原理，能有效地解决开发中遇到的一些棘手问题。

由于Vue是一个框架，它的源码实现中存在很多处理针对不同环境和不同使用的判断，我们在学习中暂时先不去管这些与核心逻辑无关的框架健壮性代码。我在github看到有作者实现了一个mini-vue很不错，因此我采用他了的开源代码来进行学习并且输出学习课程与大家分享，鉴于我也是边学习边分享，有问题的地方希望大家指正。

# Vue2与Vue3的对比

1. vue3的源码采用monorepo管理方式，实现了从模块管理到包管理的转变
2. vue2采用的是Flow来做静态类型检测，而vue3使用typescript重构代码，增强类型检测。
3. vue2的方法都放在实例对象上，而vue3中都是函数形式，所以vue3支持tree-shaking，不使用就不会被打包
4. vue2的数据劫持是通过defineProperty，而这也是vue2最大的性能问题，所以vue3中使用Proxy实现数据劫持
5. vue3对模板编译进行了优化，编译时生成Block tree可以收集动态节点，减少比较
6. vue3采用compositionApi进行组织功能，优化复用逻辑，相较于optionApi类型推断更加便捷
7. 增加了 Fragment,Teleport，Suspense组件

# Vue3的源码管理方式：

Monorepo：是管理代码的一种方式，它是指在一个项目仓库（repo）下管理多个包

* 一个仓库中维护多个包
* 便于版本管理、依赖管理，模块间的引用，调用都非常的方便
* 缺点就是仓库的体积会变大

# 四、简述Vue3源码模块组织方式及其结构

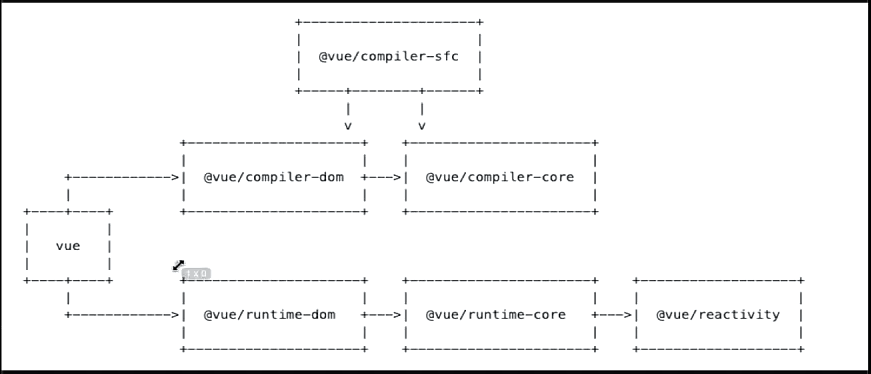


图1 Vue3源码模块组织方式以及结构图

1. 可以看到Vue就是我们平常使用中导入的Vue模块，它的模块主要分为两个点，一个是处理编译的，还有一个是处理运行时的。
2. Sfc模块是解析我们的sfc（单文件组件）的，比如App.vue的这个文件，他就是专门去解析它的。
3. Compiler/dom，它又依赖compiler-core，它们配合起来工作，目的就是将单文件组件中<template>解析成一个Render()函数，这里要注意，将sfc解析成一个js文件和将template解析成一个render函数是两个不同的知识点。
4. Runtime/dom，用来处理我们的Dom对象，runtime/core是我们源码中的核心内容也是我们的课程重点要讲解的部分，最后是Reactivity，它实现了Vue的响应式。
5. 这几个模块都是可以单独拿出来用的，接下来给大家演示一下每个库的功能到底是一个什么的。（演示Demo）
6. 总结

从代码里可以看到，我们通过导入Vue就可以使用那么多的模块，就是通过这种一层层的export来实现的。

从代码中也可以看出，要学习Vue的整个源码核心，我们需要先去了解vue是怎么去做响应式的，如何去收集依赖以及如何去触发依赖是整个理解Vue源码的基础，再下一节的课程里我们将会继续学习vue/reactivity的核心流程，带领你走进Vue实现响应式的原理。

# 五、ES的代理与反射

下一节的响应式实现，运用到了es的代理与反射，这里做简单的介绍，想要深入了解这两个模块，可以找我借资料书查看。

1. 代理
2. 反射