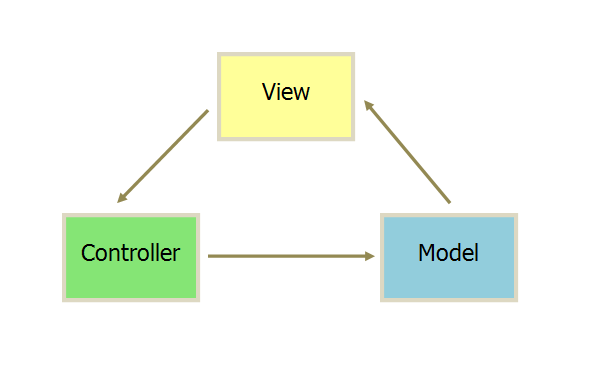
**mvc**

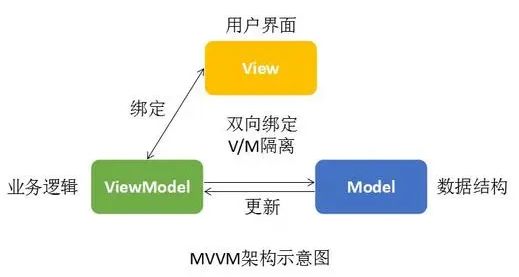


1. View 传送指令到 Controller
2. Controller 完成业务逻辑后，要求 Model 改变状态

3 . Model 将新的数据发送到 View，用户得到反馈

所有通信都是单向的！！！

**MVVM**



MVVM分为三个部分：分别是M（Model，模型层 ），V（View，视图层），VM（ViewModel，V与M连接的桥梁，也可以看作为控制器）

M：模型层，主要负责业务数据相关；

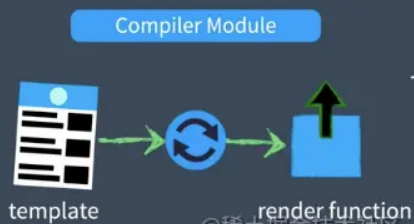
V：视图层，顾名思义，负责视图相关，细分下来就是html+css层；

VM：V与M沟通的桥梁，负责监听M或者V的修改，是实现MVVM双向绑定的要点!!!

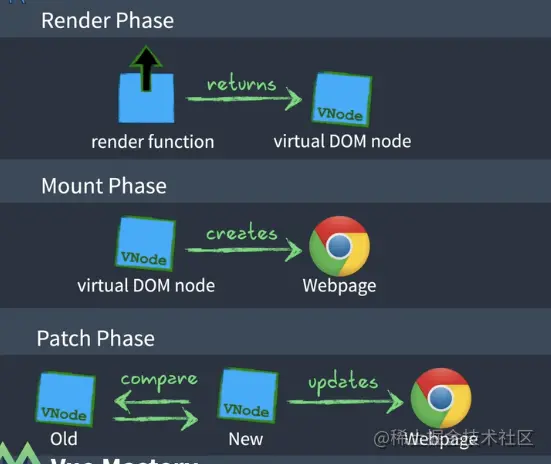
MVVM支持双向绑定，意思就是当M层数据进行修改时，VM层会监测到变化，并且通知V层进行相应的修改，反之修改V层则会通知M层数据进行修改，以此也实现了视图与模型层的相互解耦（独立）



打包时候：



**渲染函数通过以下三个周期将视图渲染到页面上**



· RenderPhase：渲染模块使用渲染函数根据初始化数据生成虚拟Dom

· MountPhase：利用虚拟Dom创建视图页面Html

· PatchPhase：数据模型一旦变化渲染函数将再次被调用生成新的虚拟Dom，然后做Dom Diff更新视图Html

## 数据响应实现

Vue普遍走的就是数据劫持方式。不同的在于使用DefineProperty还是Proxy。也就是一次一个属性劫持还是一次劫持一个对象。当然后者比前者听着就明显有优势。这也就是Vue3的响应式原理

Proxy/Reflect是在ES2015规范中加入的，Proxy可以更好的拦截对象行为，Reflect可以更优雅的操纵对象。 优势在于

* 针对整个对象定制 而不是对象的某个属性，所以也就不需要对keys进行遍历。
* 支持数组,这个DefineProperty不具备。这样就省去了重载数组方法
* Proxy 的第二个参数可以有 13 种拦截方法，这比起 Object.defineProperty() 要更加丰富
* Proxy 作为新标准受到浏览器厂商的重点关注和性能优化，相比之下 Object.defineProperty() 是一个已有的老方法
* 可以通过递归方便的进行对象嵌套。

# **vue和react的区别是什么？**

1、监听数据变化的实现原理不同

1. 数据流的不同

React一直不支持双向绑定，提倡的是单向数据流，称之为onChange/setState()模式

1. 渲染过程不同

Vue可以更快地计算出Virtual DOM的差异，这是由于它在渲染过程中，会跟踪每一个组件的依赖关系，不需要重新渲染整个组件树。

React在应用的状态被改变时，全部子组件都会重新渲染。通过shouldComponentUpdate这个生命周期方法可以进行控制，但Vue将此视为默认的优化。

如果应用中交互复杂，需要处理大量的UI变化，那么使用Virtual DOM是一个好主意。如果更新元素并不频繁，那么Virtual DOM并不一定适用，性能很可能还不如直接操控DOM。