

Vue.js 编程基础

主讲人：和凌志

主要内容

- 历史由来
- MVVM模式
- 数据驱动和组件式编程
- 从Vue到页面
- Vue常用指令



渐进式 JavaScript 框架



特别赞助



HBuilder X

提示Vue语法更强的编辑器



Vant

超好用的移动端 Vue 组件库

易用

已经会了 HTML、CSS、JavaScript? 即刻阅读指南开始构建应用!

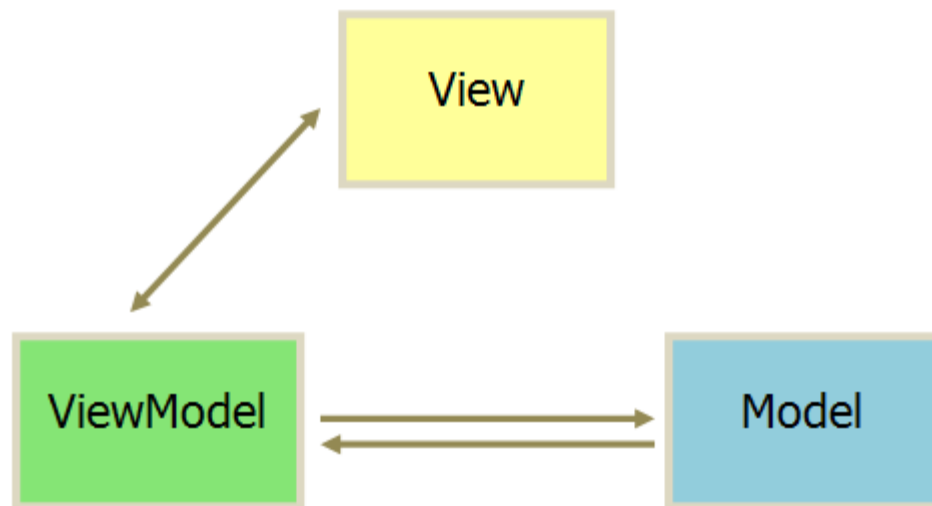
灵活

不断繁荣的生态系统, 可以在一个库和一套完整框架之间自如伸缩。

高效

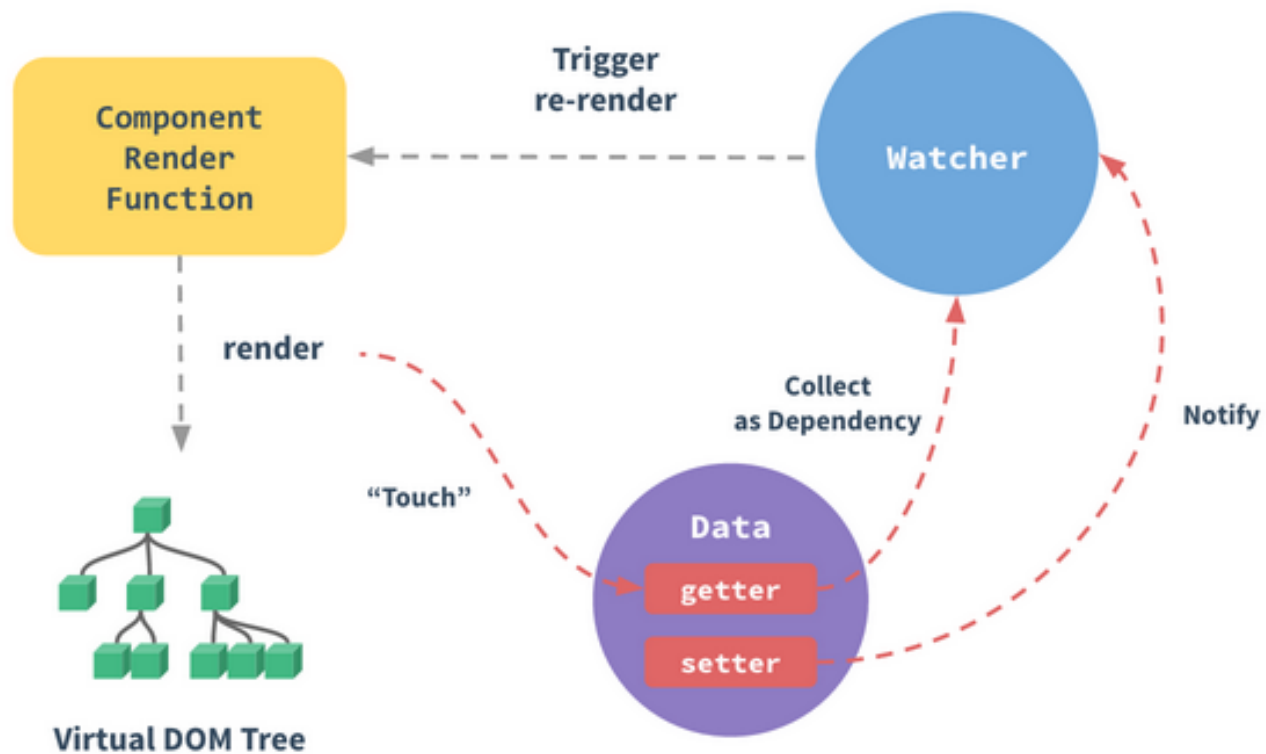
20kB min+gzip 运行大小
超快虚拟 DOM
最省心的优化

MVVM 模式

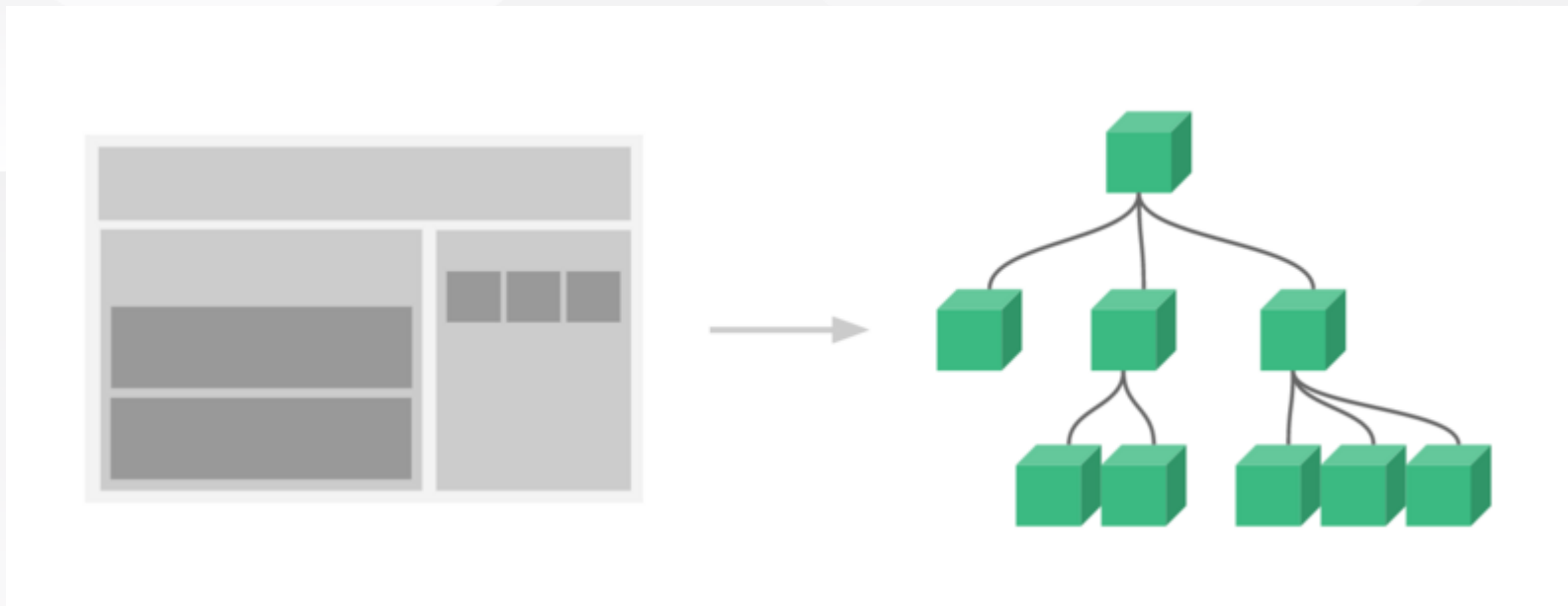


- MVVM模式采用双向绑定（data-binding）：View的变动，自动反映在 ViewModel，反之亦然。
[Vue](#)、[Angular](#) 都采用这种模式，相比于Angular，Vue.js提供了更加简洁、更易于理解的API，使得我们能够快速地上手并使用Vue.js。

数据驱动和组件式编程



组件式编程



- 这个理念不是来源于vue, 把web组件式开发发扬光大的应该是react了,组件开发是一种朴素的开发思想,分而治之,大型系统拆分成一个个的小模块小组件,分配给不同的人。额外的好处是顺便能复用这个组件。

传统的前端开发模式： 前端程序 = 拼界面+操作 ui+算法

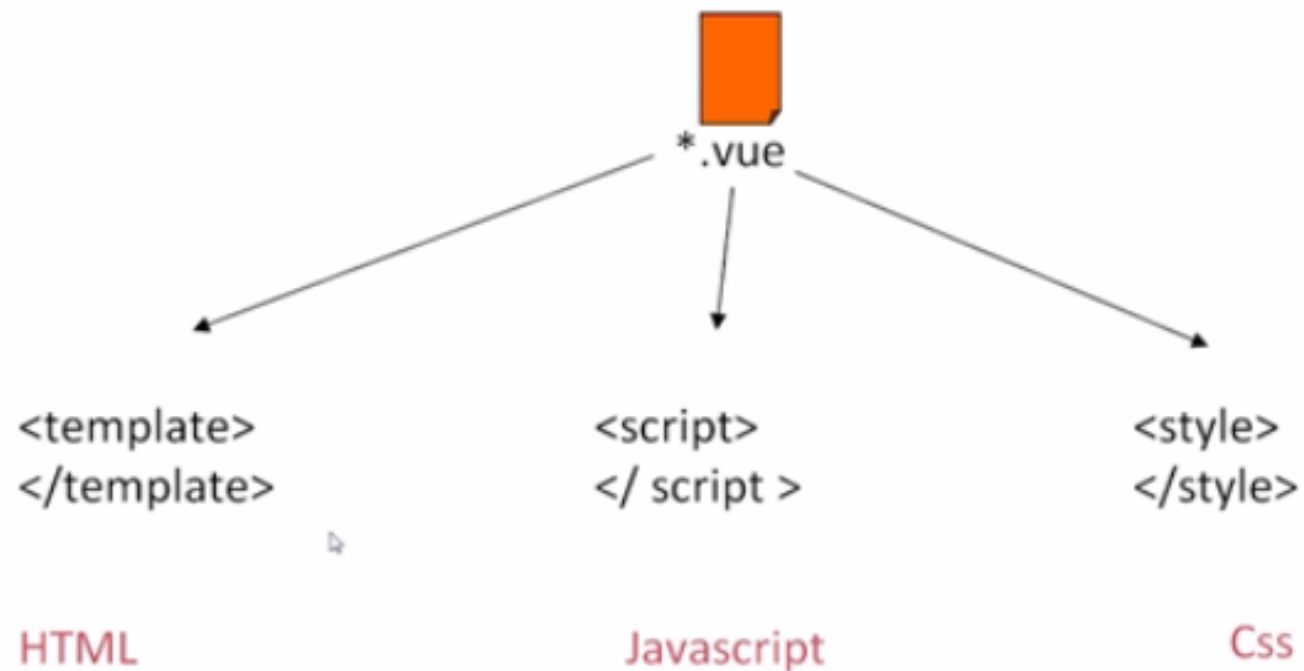
- 前端需要跟界面打交道，html+css 并没被抽象成某种在 js 中使用的数据结构,充当的更多是界面的一种配置，jquery 程序员看待他的方式就一块块的 ui,用到的时候再\$一下,获取之后修改.整个程序写下来是零零散散的节点操作。
- 一个比较实际的情况就是,在 ui 控件有联动的时候,如果没有一种机制来管理这些ui之间的修改,那么依赖程序员自己去手动管理这些 ui 的状态,会让人烦不胜烦,且容易出现bug。
- 基于操作 dom 的前端开发方式： 拼界面->找到 dom 节点->修改属性->检测是否有其他影响的节点->根据刚刚修改的 dom节点更新自己的状态。

基于 MVVM 框架的前端开发模式

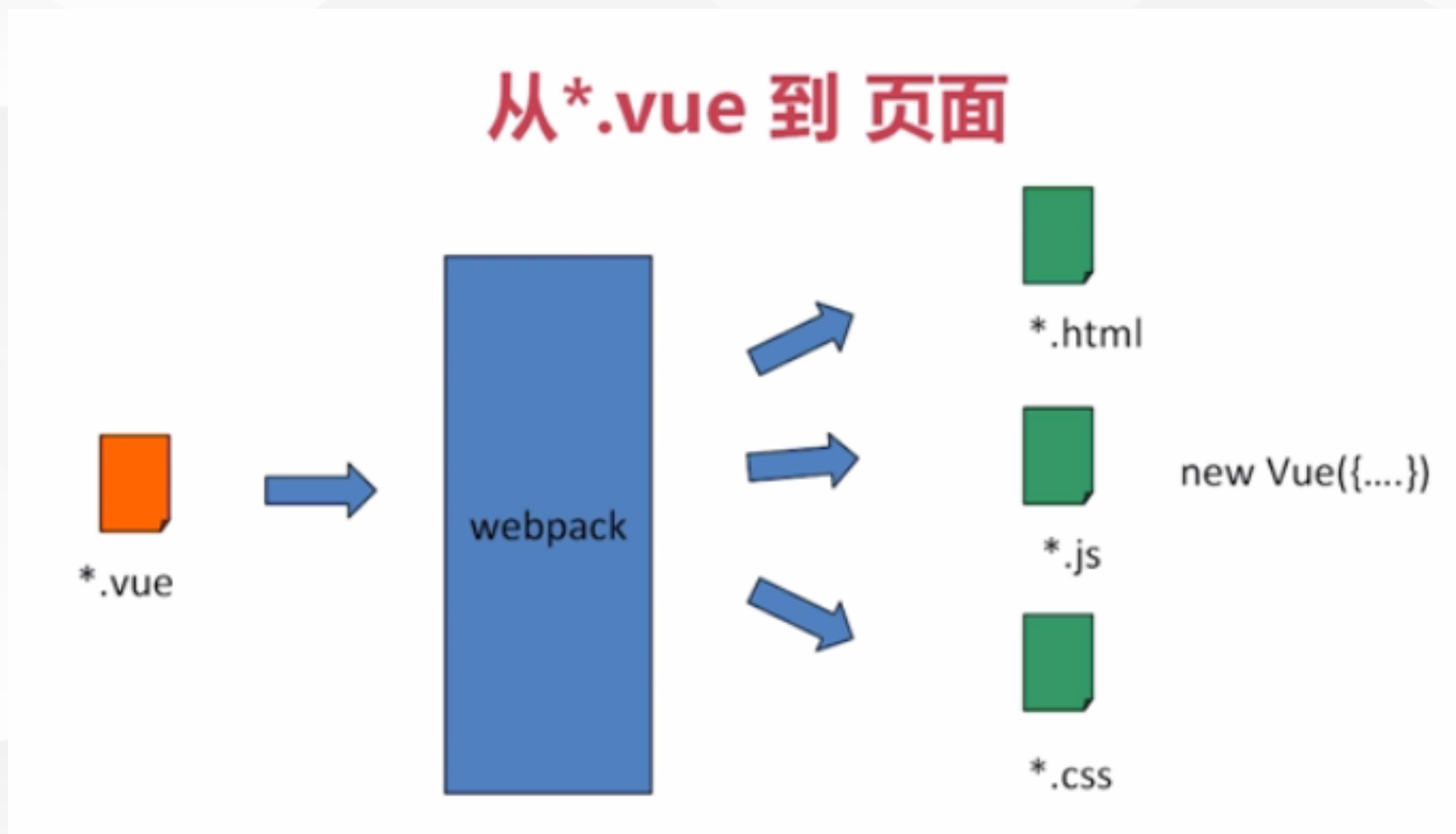
- vue或者angular这些mvvm框架给了前端另一种思路,完全基于数据驱动的编程。
- 如果你之前已经习惯了用jQuery操作DOM, 学习Vue.js时请先抛开手动操作DOM的思维, 因为Vue.js是数据驱动的, 你无需手动操作DOM。
- Vue采用一种数据绑定的方式,自动绑定dom节点的属性.这样就把你从操作dom节点的繁琐过程中解脱出来了,你只要专注于数据的状态,ui更新的事情你不需要去管了,不管是样式还是内容,可见性还是切换class,框架帮你把关注点从传统的dom操作转移到了数据,回归编程的本质:程序=数据结构+算法. 这也是mvvm框架最大的思路上的突破。

从 Vue 到页面

Vue.js的一个组件



从 Vue 到 页面



Vue 常用指令

- v-bind
- v-model
- v-on
- v-if
- v-for
- v-click

动手实验

动手实验

- 引入 Vue.js
- 创建 Vue 组件
- 实现数据绑定

小结

- ✓ 掌握 MVVM 前端框架理念
- ✓ 掌握 Vue.js 实例
- ✓ 掌握 Vue.js 数据绑定与事件绑定