

**Java Web讨论课报告**

**题 目： Vue框架**

**学院（系）： 信息科学与工程学院**

**专业班级： 计科21-4班**

**姓 名： 刘运硕**

目录

[1什么是Vue 2](#_Toc132789857)

[2 Vue核心概念 2](#_Toc132789858)

[2．1组件化 2](#_Toc132789859)

[2.2 MVVM数据双向绑定 3](#_Toc132789860)

[2.3 响应式 虚拟DOM 4](#_Toc132789861)

[2.4生命周期 4](#_Toc132789862)

[3 Vue的优缺点 6](#_Toc132789863)

[3.1优点： 6](#_Toc132789864)

[3.2缺点 6](#_Toc132789865)

[4 Vue的挑战与未来 7](#_Toc132789866)

[4.1 Vue的面临的挑战 7](#_Toc132789867)

[4.2 Vue的发展前景 7](#_Toc132789868)

## 1什么是Vue

Vue是一套用于构建用户界面的**渐进式**JavaScript框架。采用**自底向上增量开发**的设计。

Vue就是一个JS库，并且无依赖别的JS库，直接引入一个JS文件就可以使用，与传统JS和JQuery框架不同，Vue的渐进式框架表示开发者可以由简单组件写起，渐渐搭建出一个复杂的前端平台。

渐进式理解：1、需要什么用什么 2、从简单到困难3、组件化管理4、模块化编程。



## 2 Vue核心概念

形成Vue渐进式框架的核心概念为：**组件化，MVVM，响应式，和生命周期**。

### 2．1组件化

Vue将组成一个页面的HTML，CSS和JS合并到一个组件中，可以被其他组件或页面引入而重复利用。通常每个.Vue文件作为一个组件导出，组件可以作为基础组件（如按钮）或一个页面（如登录页面）。组件化很好的将一个庞大复杂的前端工程拆分为一个个组件，重复利用的性质也大大提高了开发的效率。

### 2.2 MVVM数据双向绑定

MVVM模式（全称为Model-View-ViewModel）为Vue实现数据双向绑定。

在MVVM中，View为视图层，ViewModel为业务逻辑层，Model为数层。

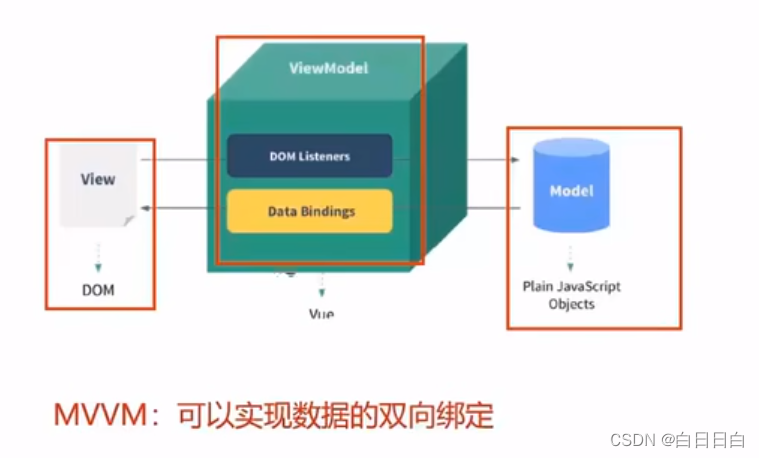
什么是数据双向绑定呢？

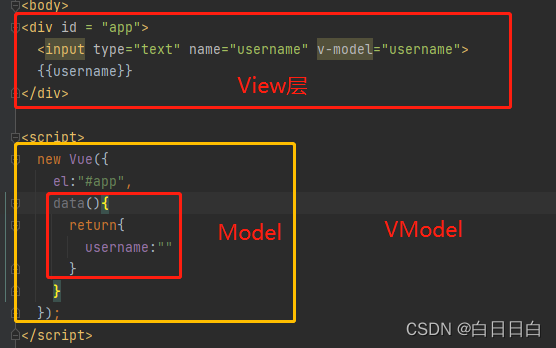
当用户使View变化时（如填写表单），变化会自动同步到ViewModel处理相应逻辑，并将变化更新到Model数据库。

反之，若服务端数据变化（如股价波动），变化会自动同步到ViewModel处理相应逻辑，并将变化同步到View展现给用户。

在用Vue之前，完成HTML和JS之间的交互需要使用大量的DOM操作来实现动态加载。

MVVM的数据双向绑定减少了DOM操作，更高效地实现了视图和数据的交互。同时，MVVM使界面、交互和数据层分离，便于设计人员负责设计界面，后端开发人员提供数据接口，而前端开发人员专注于业务交互逻辑的实现。





### 2.3 响应式 虚拟DOM

对于DOM来说，当HTML的一个元素（如div）需要响应数据更改时，会刷新整个页面，导致效率堪忧。

对于虚拟DOM，浏览器会将HTML文件转换为JS文件并复制一个额外使用（虚拟）。对于任何更改，虚拟DOM都将复制的JS与原始JS进行比较，只重新加载更改的部分，局部修改到真实DOM上。

在Vue中，每个绑定data属性的组件都有一个Watcher检测data属性的变化。一旦检测到改变，则重新渲染该组件，这就是响应式。

Vue的响应式大概会经过下面几个阶段

1. 使用 Object.defineProperty 把属性全部转为getter/setter

2. 属性变更时通知观察者(watcher)变更

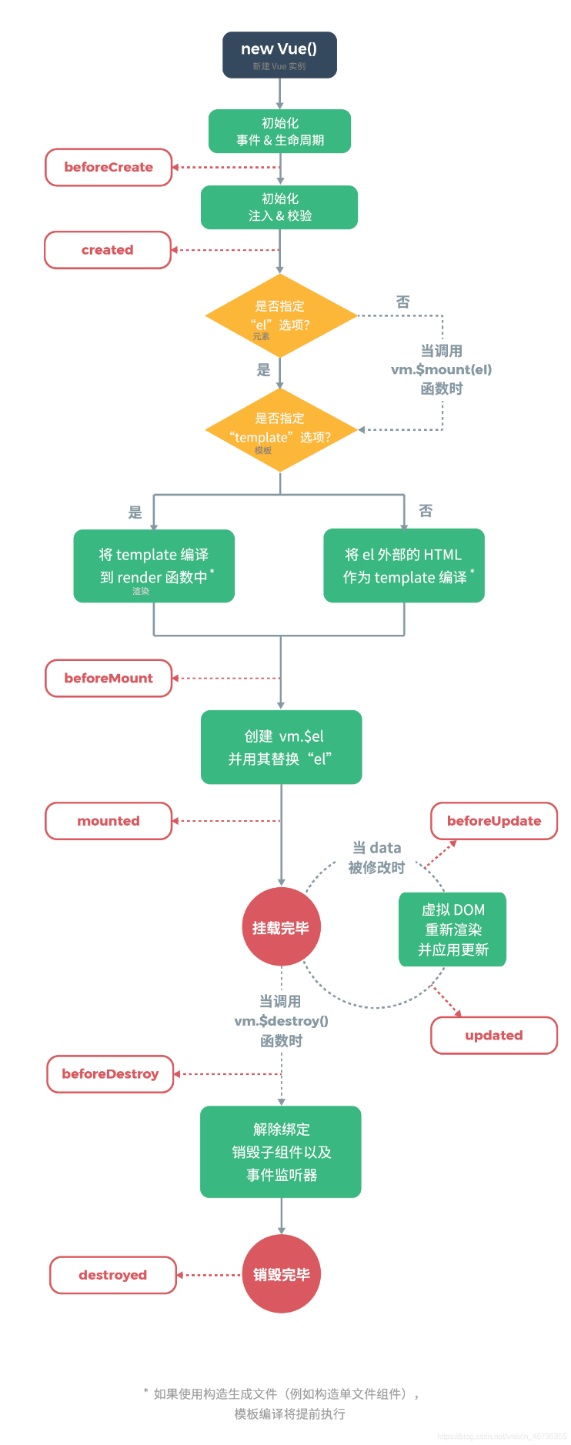
3. watcher触发重新渲染生成虚拟 DOM

4. Vue框架遍历计算新旧虚拟 DOM差异，由于 JavaScript 的限制，Vue 不能检测数组和对象的变化

5. 加载操作，将差异局部修改到真实 DOM

### 2.4生命周期

每个Vue组件都有生命周期，过程为**创建 -> 挂载 -> 更新 -> 销毁**。开发者可以通过**钩子函数**（如mounted）在组件生命周期中的不同时刻进行操作。下面是一张Vue生命周期的完整图解。



## 3 Vue的优缺点

### 3.1优点：

1.简单易学: Vue的API简单易懂，学习曲线较低。与其他前端框架相比，Vue的上手难度较低，使得初学者能够更快地开始开发应用程序。

2.高效性: Vue使用虚拟DOM，可以避免频繁更新DOM所带来的性能问题。它也采用了异步渲染机制，减少了服务器响应时间。

3.组件化: Vue采用了组件化的开发方式，使得开发人员可以对UI组件进行重复利用、维护和开发。组件化也促进了代码的可读性和可维护性。

4.响应式编程: Vue使用了双向数据绑定和响应式编程的思想，监听数据变化并自动更新UI。这使得开发者可以更轻松地实现动态UI和交互效果。

5.生态环境丰富: Vue拥有丰富的生态环境，有大量的第三方工具和库，包括路由、状态管理、构建工具等等。

### 3.2缺点

大型应用程序管理问题: 对于大型、复杂的应用程序，Vue的组件化能力可能会导致在管理方面出现问题。这是因为应用程序中包含大量的组件，并且组件之间的关系变得非常复杂。为了解决这个问题，开发人员需要使用一些高级工具和技巧。

过度灵活性: Vue的灵活性也可能导致代码结构变得混乱和难以维护。因为Vue允许开发者以多种方式实现相同的任务，这可能导致代码库中出现大量的重复代码或不必要的复杂性。

可扩展性: 当应用程序需要大规模扩展时，Vue框架的可扩展性存在限制。如果应用程序需要更高的可扩展性，开发人员需要使用其他工具或框架。

## 4 Vue的挑战与未来

### 4.1 Vue的面临的挑战

1. 市场竞争：与其他前端框架如React和Angular等的竞争对手相比，Vue在用户规模和生态系统等方面还有一定差距，需要不断改进和完善。
2. 技术更新：随着前端技术的不断发展，Vue需要不断更新和优化自己的特性，以满足用户需求和应对新的技术挑战。
3. 跨平台支持：随着移动互联网和各种智能设备的普及，Vue需要进一步提高跨平台支持能力，满足不同设备和场景下的应用需求。

### 4.2 Vue的发展前景

Vue作为一种优秀的前端框架，将继续在技术创新、功能丰富、性能优化和生态系统建设等方面不断发展壮大。值得注意的是，随着移动互联网和各种智能设备的普及，Vue将逐渐提高跨平台支持能力，满足不同设备和场景下的应用需求。同时，Vue也将进一步提高服务端渲染支持，提供更好的SEO和首屏加载效果。总之，随着互联网技术的不断发展和Vue生态系统的不断完善，Vue的未来发展前景十分广阔。