VUE

1. VUE.JS 提供了一些高级的开发功能：  
   虚拟dom 可复用组件 前端路由 状态管理 解耦视图和数据

MVVM模式 Model-View-ViewModel 当View（视图层）变化时，会自动更新到DOM上，反之亦然。 View 和ViewModel之间通过双向绑定（data-binding）建立联系。

1. VUE生命周期

Vue 生命周期是指vue实例对象从创建之初到销毁的过程。

Vue生命周期共分为8个阶段： 创建前/后 载入前/后 更新前/后 销毁前/后

1. 创建前/后

BeforeCreate: 在实例初始化之后，数据观测(data observer)和 event/watcher时间配置之前被调用。

（实例被创建之前的函数）阶段，vue实例挂载元素 el 和数据对象data都是undefined,还未初始化。

Created:在实例创建后立即被调用。在这一步，实例已完成以下的配置：数据观测（data observer）,属性和方法的运算， watch/event事件回调。然而挂载阶段还没开始，$el属性目前不可见。

Vue实例的数据对象data被初始化，但是el还没有被初始化。

1. 载入前/后

beforeMount : 在挂载开始之前被调用：相关的render函数首次被调用，该钩子在服务器被渲染器件不被调用。

Vue 实例挂载元素el和数据对象data都被初始化了，但还是挂载之前虚拟的DOM节点。

Mounted: el被新创建的vm.$el替换，并挂载到实例上去之后调用该钩子。如果root实例挂载了一个文档内元素，当mounted被调用时vm.$el也在文档内。

1. 更新前/后

beforeUpdate:数据更新时调用，发生在虚拟DOM打补丁之前。这里适合更新之前访问现有的DOM，比如手动移除已添加的事件监听器。

实时监听data的变化，当data变化时，触发beforeUpdate和updated 方法，会重新渲染DOM,然后对应位置的真实的DOM打补丁。

update:由于数据更改导致的虚拟DOM重新渲染和打补丁，在这之后会调用钩子。

当这个钩子被调用时，组件DOM已经更新，所以你现在可以执行依赖于DOM的操作，然而大多数情况下，你应该避免在此期间更改状态，如果相应状态改变，通常最好使用计算属性 或watcher取代。

1. Activated

Activated: kepp-alive 组件激活时调用。

Deactivated: keep-alive 组件停用时调用。

1. 销毁前/后

BeforeDestroy: 实例销毁之前调用。

Destroyed: Vue实例销毁后调用。调用后，Vue实例指示的所有东西都会解绑定，所有的事件监听器都会被移除，所有的子实例也都会被销毁。

Q1： 第一次页面加载会触发哪几个钩子？

beforeCreate created beforeMound mounted

销毁前/后

1. VUE实例和数据绑定

挂载成功以后，我们可以通过app.$el来访问元素，

生命周期钩子 ： created 实例创建完成后调用，此阶段完成了数据的观测等。但是尚未挂载，初始化数据比较有用。

Monted el挂载在实例上调用，

Beforedestroy 实例销毁之前调用 解绑一些监听事件等。

过滤器 VUE.JS 支持在{{ }} 插值尾部添加一个管道符“（|）”对数据进行过滤，常用于格式化文本。

1. 指令和事件 指令带有前缀 V- ，比如v-if v-html,l v-pre

v-bind 动态更新html元素上的属性 v-on绑定监听事件

语法糖： v-bind ： v-on @

1. 计算属性 遇到复杂的属性应该使用计算属性改写。

每个计算属性都包含一个getter和setter。默认用getter方法来读取一个计算属性。

Methods也可以实现、计算属性所依赖的数据发生变化时，它才会重新取值，所以text只要不改变，计算属性就不更新。

使用计算属性还是methods取决于你是否需要缓存，当遍历大数组和做大量计算时，应当使用计算属性，除非不希望得到缓存。

1. 基本指令

v-cloak 不需要表达式，会在vue实例中结束编译时从html元素上移除。

v-once 不需要表达式的命令，作用是定义它的元素或组件只渲染一次，被视为静态内容。

条件渲染： v-if v-else-if v-else

列表渲染指令 v-for 数组遍历 item in items 还可以迭代整数

方法与事件： vue提供了一个特殊变量$event,用于访问原生DOM事件。

修饰符：在@绑定后的事件加小圆点“.”,再跟一个后缀来使用修饰符。Vue支持一下修饰符 .stop .prevent .capture .self .once

1. 表单与V-model
2. 组件详解

组件（component）是vue.js的核心功能。

9. render函数  
 虽然我们组件模板写在template选项里，但是在Vue.js编译时，但是都会编译为Vitural Dom.

Object -> render ->createElement(h) 🡪diff 🡪patch

Render:生成虚拟节点

createElemnt(h) : 基于虚拟节点创建DOM节点

diff:状态更新后，进行对比，生成补丁对象。

遍历补丁对象：更新DOM节点。

Render函数神奇的地方在于createElement.

10. vuex 状态管理 更好的管理和维护整个项目的状态

一个组件可以分为数据（model）和视图（view）.数据更新时，视图也会自动更新，在视图中又可以绑定一些事件，它们触发methods中指定的方法。从而可以改变数据，更新视图，这是一个组件的基本运行方式。

在实际业务中，经常有跨组件共享数据的需求，因此vuex的设计就是用来统一管理组件状态。

1. 安装 vuex 2.vue.use(vuex)