Vue面试题

1.vue中的MVVM模式

即Model-View-ViewModel。

  Vue是以数据为驱动的，Vue自身将DOM和数据进行绑定，一旦创建绑定，DOM和数据将保持同步，每当数据发生变化，DOM会跟着变化。

ViewModel是Vue的核心，它是Vue的一个实例。Vue实例时作用域某个HTML元素上的，这个HTML元素可以是body，也可以是某个id所指代的元素。

DOM Listeners和Data Bindings是实现双向绑定的关键。DOM Listeners监听页面所有View层DOM元素的变化，当发生变化，Model层的数据随之变化；Data Bindings监听Model层的数据，当数据发生变化，View层的DOM元素随之变化。

2.v-show指令，v-if的区别

条件渲染指令，与v-if不同的是，无论v-show的值为true或false，元素都会存在于HTML代码中；而只有当v-if的值为true，元素才会存在于HTML代码中。v-show指令只是设置了元素CSS的style值。

3.如何让css只在当前组件中起作用

在每一个vue组件中都可以定义各自的css，js，如果希望组件内写的css只对当前组件起作用，只需要在style中写入scoped，即：

<style scoped></style>

4.指令keep-alive

在vue-router写着keep-alive，keep-alive的含义：

如果把切换出去的组件保留在内存中，可以保留它的状态或避免重新渲染。为此可以添加一个keep-alive指令

<component :is='curremtView' keep-alive></component>

5.Vue js组件

  vuejs构建组件使用

  Vue.component('componentName',{ /\*component\*/ })；

这里注意一点，组件要先注册

再使用

Vue.component('mine',{

  template:'#mineTpl',

  props:['name','title','city','content']

});

var v=new Vue({

  el:'#vueInstance',

  data:{

   name:'zhang',

   title:'this is title',

   city:'Beijing',

   content:'these are some desc about Blog'

  }

});

6.路由嵌套

路由嵌套会将其他组件渲染到该组件内，而不是进行整个页面跳转router-view本身就是将组件渲染到该位置，想要进行页面跳转，就要将页面渲染到根组件，在起始配置路由时候写到：

 var App = Vue.extend({ root });

router.start(App,'#app');

这里首先将根组件注册进来，用于将路由中配置好的各个页面渲染出来，然后将根组件挂载

到与#app匹配的元素上。

7.指令v-el的使用

有时候我们想就像使用jquery那样去访问一个元素，此时就可以使用v-el指令，去给这个元素注册一个索引，方便通过所属实例的$el访问这个元素。

注意

HTML不区分大小写，所以v-el:someEl将转换为全小写。可以用v-el:some-el然后设置

this.$el.someEl。

示例

<span v-el:msg>hello</span>

<span v-el:other-msg>world</span>

this.$els.msg.textContent // ->"hello"

this.$els.otherMsg.textContent // ->"world"

this.$els.msg //-><span>hello</span>

8.vuejs中使用事件名

 在vuejs中，我们经常要绑定一些事件，有时候给DOM元素绑定，有时候给组件绑定。绑定事件在HTML中用v-on:click-"event",这时evet的名字不要出现大写，因为在1.x中不区分大小写，所以如果我们在HTML写v-on:click="myEvent"而在js中写myEvent就出错误，所以在vuejs的1.x绑定事件时候，要尽量避免使用大写字母。在2.0中没有该限制！

9.Vue.js是什么

Vue.js（是一套构建用户界面的渐进式框架。与其他重量级框架不同的是，Vue 采用自底向上增量开发的设计。Vue 的核心库只关注视图层，并且非常容易学习，非常容易与其它库或已有项目整合。另一方面，Vue 完全有能力驱动采用单文件组件和Vue生态系统支持的库开发的复杂单页应用。

Vue.js 的目标是通过尽可能简单的 API 实现响应的数据绑定和组合的视图组件

10.VueJS 特性：

I: MVVM

模式（数据变量（model）发生改变视图（view）也改变， 视图（view）改变，数据变量（model）也发生改变）

使用MVVM模式有几大好处：

1. 低耦合。

View可以独立于Model变化和修改，一个ViewModel可以绑定到不同的View

上，当View变化的时候Model可以不变，当Model变化的时候View也可以不变。

2. 可重用性。

可以把一些视图的逻辑放在ViewModel里面，让很多View重用这段视图逻辑。

3. 独立开发。

开发人员可以专注与业务逻辑和数据的开发(ViewModel)。设计人员可以专注于界面

(View)的设计。

4. 可测试性。

可以针对ViewModel来对界面(View)进行测试

II: 组件化

III: 指令系统

IIII: vue2.0开始支持虚拟dom

vue1.0是操作的是真的dom元素而不是虚拟的

 虚拟dom:可以提升页面的刷新速度

虚拟dom有利也有弊。

 A：大小 - 其中之一就是更多的功能意味着代码包中更多行的代码。幸运的是，Vue.js 2.0 依旧比较小（当前版本21.4kb），并且也正在删除很多东西。

B: 内存 -同样，虚拟DOM需要将现有的DOM拷贝后保存在内存中，这是一个在DOM更新速度和内存使用中的权衡。

 C: 并不适用所有情况 -如果虚拟DOM可以一次性进行批量的修改是非常好的。但是如果是单独的、稀少的更新呢？这样的任何DOM更新都将会使虚拟DOM带来无意义的预计算

11.Vue.js特点

 简洁：页面由HTML模板+Json数据+Vue实例组成

 数据驱动：自动计算属性和追踪依赖的模板表达式

 组件化：用可复用、解耦的组件来构造页面

 轻量：代码量小，不依赖其他库

 快速：精确有效批量DOM更新

模板友好：可通过npm，bower等多种方式安装，很容易融入

12.Vue.js 和 React.js 有什么区别?

React.js 和 Vue.js 确实有一些相似——它们都提供数据驱动、可组合搭建的视图组件。

然而，它们的内部实现是完全不同的。React 是基于Virtual DOM——一种在内存中描述DOM 树状态的数据结构。React 中的数据通常被看作是不可变的，而 DOM 操作则是通过

 Virtual DOM 的diff 来计算的。与之相比，Vue.js 中的数据默认是可变的，而数据的变更会直接出发对应的 DOM 更新。相比于Virtual DOM，Vue.js 使用实际的 DOM 作为模板，并且保持对真实节点的引用来进行数据绑定。

  Virtual DOM 提供了一个函数式的描述视图的方法，这很cool。因为它不使用数据观察机制，每次更新都会重新渲染整个应用，因此从定义上保证了视图通与数据的同步。它也开辟了JavaScript 同构应用的可能性。