**1.v-if（虚拟dom元素）和v-show有什么区别？**

答案：

#### **1).手段：**

* v-if是动态的向DOM树内添加或者删除DOM元素；
* v-show是通过设置DOM元素的display样式属性控制显隐；

#### **2).编译过程：**

* v-if切换有一个局部编译/卸载的过程，切换过程中合适地销毁和重建内部的事件监听和子组件；
* v-show只是简单的基于css切换

#### **3).编译条件**

* v-if是惰性的，如果初始条件为假，则什么也不做；只有在条件第一次变为真时才开始局部编译（编译被缓存？编译被缓存后，然后再切换的时候进行局部卸载）；
* v-show是在任何条件下（首次条件是否为真）都被编译，然后被缓存，而且DOM元素保留；

#### **4).性能消耗**

* v-if有更高的切换消耗；
* v-show有更高的初始渲染消耗；

#### **50.使用场景**

* v-if适合运营条件不大可能改变；
* v-show适合频繁切换；

**2.vue-router使用步骤，请用代码作答。**

答案：

<div class="container"> <router-link to="/user?name=张三&age=22">用户中心</router-link> <router-view></router-view> </div>

步骤：

### **1）.定义不同的路由切换时需要显示的内容**

var User=Vue.component("User",{

template:"<div>{{$route.query.name}}</div>"

})

#### **2）.配置路由规则**

var routes=[{

path:"/user",

component:User

}]

#### **3）.创建vue-router实例**

var router=new VueRouter({

routes:routes

})

#### **4）.将vue-router的实例挂载到vue的实例**

new Vue({

el:".container",

router:router

})

**3.请说下封装vue组件的过程**

答案:

首先，组件可以提升整个项目的开发效率。能够把页面抽象成多个相对独立的模块，解决了我们传统项目开发：效率低，难维护，复用性等问题

然后，使用Vue.extend方法创建一个组件，然后使用Vue.component方法注册组件。子组件需要数据，可以在props中接受定义，而子组件修改好数据后想把数据传递给父组件。可以采用emit方法。

**4.vue.cli项目中目录中src目录每个文件夹和文件的用法？**

答案：

* assets文件夹是放静态资源；
* components是放组件；
* router是定义路由相关的配置；
* view视图；
* app.vue是一个应用组件；
* main.js是入口文件；

**5.构建的 vue-cli 工程都到了哪些技术，它们的作用分别是什么？**

答案：

1、vue.js：vue-cli工程的核心，主要特点是 双向数据绑定 和 组件系统。  
2、vue-router：vue官方推荐使用的路由框架。  
3、vuex：专为 Vue.js 应用项目开发的状态管理器，主要用于维护vue组件间共用的一些 变量 和 方法。  
4、axios（ 或者 fetch 、ajax ）：用于发起 GET 、或 POST 等 http请求，基于 Promise 设计。  
5、vux等：一个专为vue设计的移动端UI组件库。  
6、创建一个emit.js文件，用于vue事件机制的管理。  
7、webpack：模块加载和vue-cli工程打包器。

**6.vue-cli 工程常用的 npm 命令有哪些？**

答案：

下载 node\_modules 资源包的命令：

npm install

启动 vue-cli 开发环境的 npm命令：

npm run dev

vue-cli 生成 生产环境部署资源 的 npm命令：

npm run build

用于查看 vue-cli 生产环境部署资源文件大小的 npm命令：

npm run build --report

**7.vue中做数据渲染的时候如何保证将数据原样输出？**

答案：

1. . v-text：将数据输出到元素内部，如果输出的数据有HTML代码，会作为普通文本输出

2). v-html：将数据输出到元素内部，如果输出的数据有HTML代码，会被渲染

3). {{}}：插值表达式，可以直接获取Vue实例中定义的数据或函数，使用插值表达式的时候，值可能闪烁；而使用v-html、v-text不会闪烁，

有值就显示，没值就隐藏

**8.vue中methods和computed有什么区别？应用场景如何？**

答案：

1).methods：方法，函数，绑定事件调用；不会使用缓存 2).computed： 计算属性 本质是方法，使用时可以像属性一样使用， 当操作的值没有发生改变时，会使用缓存，当值发生改变，才会改变；提高速度

**9.Vue中如何处理原生的DOM事件？**

### 答案：

1).DOM：HTML DOM定义了访问和操作HTML文档的标准方法。DOM将 HTML文档表达为树结构。我们可以通过DOM灵活的获取树节点，并且操作节点（重新设置值、重新设置属性、重新设置样式）

2). DOM事件: onclick、onmouseover、onmouseout、onchange、onkeyup、onkeydown

3).Vue是如何处理的：通过v-on，简写@的方式定义事件

@click、@mouseover、@mouseout、@change、@keyup、@keydown...并绑定函数

**10.什么是js的冒泡？Vue中如何阻止冒泡事件？**

答案：

js冒泡概念：当父元素内多级子元素绑定了同一个事件，js会依次从内往外或者从外往内（？）执行每个元素的该事件，从而引发冒泡

js解决冒泡：event.stopPropagation()

vue中：@click.stop="" [,@mouseover.stop=""](mailto:,@mouseover.stop=\"\")

**11.写一个父子组件之间的传值？**

答案：

1. .父组件与子组件传值：

//父组件通过标签上面定义传值

<template>

<Main :obj="data"></Main>

</template>

<script>

//引入子组件

import Main form "./main"

exprot default{

name:"parent",

**data**(){

return {

data:"我要向子组件传递数据"

}

},

//初始化组件

components:{

Main

}

}

</script>

//子组件通过props方法接受数据

<template>

<div>{{data}}</div>

</template>

<script>

exprot default{

name:"son",

//接受父组件传值

props:["data"]

}

</script>

1. .子组件与父组件传值：

//子组件通过$emit方法传递参数

<template>

<div v-on:click="events"></div>

</template>

<script>

//引入子组件

import Main form "./main"

exprot default{

methods:{

events:**function**(){

。。。。。。。

}

}

}

</script>

<template>

<div>{{data}}</div>

</template>

<script>

exprot default{

name:"son",

//接受父组件传值

props:["data"]

}

</script>