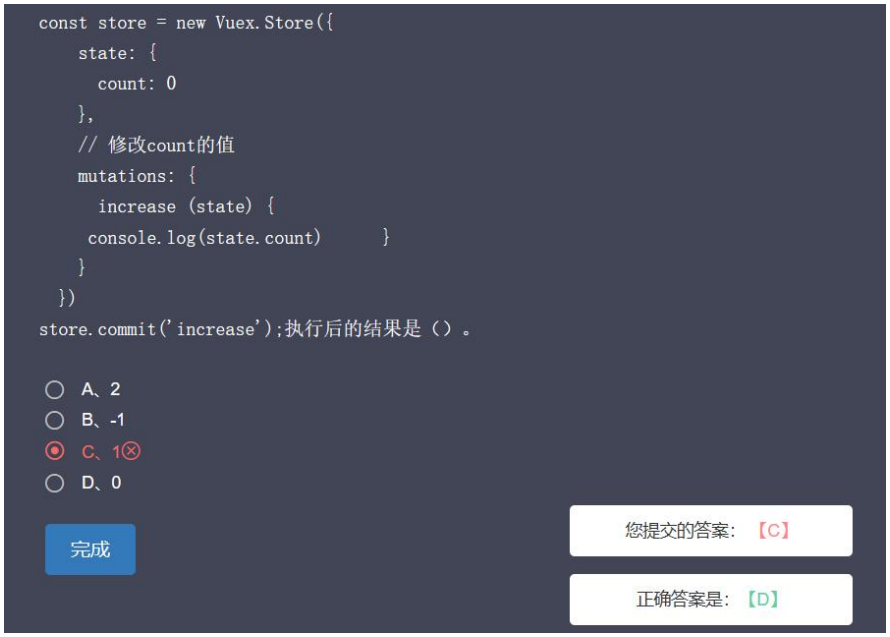
1. 主要用来编写网页结构的是：HTML
2. 代表视图部分的是：View
3. 可以引入 vue.js 文件的是：<script>
4. Windows 系统内置的命令行工具是：cmd
5. 表示一个 Node.js 的包管理工具，用来解决 Node.js 代码部署问题的是：npm
6. 用来创建实例对象的是：new
7. 在 Vue 创建时，用来表示唯一根标签的是：el
8. 可以访问原始数据对象的是：vm.$data
9. 以下代码 执行后的结果是 20

1、以下代码

```
var vm = new Vue({
  el: '#app',
   data: {
     price: 20,
     num: 1
  },
  computed: {
    // 总价格totalPrice
    totalPrice () {
      return this.price * this.num
    }
  }
})
console.log(vm.totalPrice);执行后的结果是（）。
```

10.在插值表达式中可以使用过滤器来对数据进行处理，管道符号是：|
11.可以实现绑定事件的指令是：v-on
12.可以实现获取随机数的是：Math.random()
13.能够实现自定义指令的是：Vue.directive()
14.能够实现在 Vue 中安装插件的是：Vue.use()
15.用于向响应式对象中动态添加一个属性，并确保这个新属性同样是响应式的，且触发视图更新的是:Vue.set
16.用来获取用户自定义选项的是：vm.$options
17.用来访问 vm 实例使用的根 DOM 元素的是：vm.$el
18.用来访问子组件实例的是：vm.$children
19.Vue 全局配置对象 config 中用于控制生产信息的显示和隐藏的是：productionTip
20.Vue 提供了内置的过渡封装组件是：transition
21.transition 类名中表示进入过渡的开始状态的是：v-enter
22.可以通过百分比的形式来规定动画的状态的是:@keyframes
23.可以通过 transition 组件的钩子函数实现动画的是:@before-enter

24.属于 JavaScript 动画库的是:Velocity.js
25.可以实现不同标签名的过渡效果的是:v-if 和 v-else
26.相同标签名元素的过渡需要为元素定义的属性是:key
27.用来实现<transition>组件过渡模式切换的是:mode
28.动态组件需要通过 Vue 中的<component>元素绑定（）属性来实现多组件的过渡 is
29.可以与 v-for 结合使用并实现列表过渡的组件是:<transition-group>
30.可以实现列表循环渲染的是:v-for
31.可以实现列表排序平滑过渡的是:v-move
32.能够判断字符串中是否存在某一字符的是:indexof
33.在 Vue 项目中，可以使用以下（）命令初始化项目 :npm init -y
34.一个单独的组件文件通常应包含哪几部分代码 :<style>样式、<template>模板、<script>逻辑、以上全部
35.下列选项中，不属于 CSS 预处理器的是：scoped
36.下列选项中，关于 router.push()方法说法错误的是：这个方法不会向 history 栈添加记录
37.用来定义 store 实例的初始数据的是：state
38.下面选项中，可以引入 vuex.js 核心文件的标签是：<script>
39.以下代码 执行后的结果是 0



40.下面选项中，可以用来创建 store 实例对象的是：new Vuex.Store({})
41.可以实现 actions 中事件处理函数状态分发的是：dispatch()
42.可以实现 mutations 中事件处理函数状态提交的是:commit()
43.以下代码 </script>中的 this 指的是 Vue 实例

```
以下代码
 <div id="app">
<p>{{this.$store.state.name}}</p>
</div>
<script>
var store = new Vuex.Store({
  state:{name:'store'}
})
var vm  = new Vue({el:'#app',store})
</script>中的this指的是（）。
```

44. devtools 选项用来设置是否在 devtools 调试工具中启用 Vuex，表示启用的值是：true

45.下面选项中，可以用来创建 vue 项目的命令的是：vue init webpack shopcart

46.下列选项中说法正确的是：执行 npm uninstall vue-cli -g 命令可以全局删除 vue-cli 包

47.不属于 package.json 文件中的 script 常用字段的是：text

48.关于 CLI 服务，下列选项说法错误的是：通过 npx vue-cli-service helps 查看所有的可用命令

49.关于单页面应用说法错误的是：整页刷新

50.下列选项中说法正确的是：使用相对路径引入的静态资源文件，会被 webpack 处理解析为模块依赖。错误的是：放在 public 文件夹下的资源将会经过 webpack 的处理、通过绝对路径被引用的资源将会经过 webpack 的处理、URL 以~开始，会被认为是模块请求

51.关于服务器脚本文件说法正确的是：调用 renderer.renderToString()方法来渲染生成 HTML、调用 res.end()方法可以将 HTML 结果发送给浏览器、server.listen()用于启动一个服务器来监听连接

52.V-show 指令可用来切换元素的可见状态

53.不属于 vue-router 的导航钩子的是:页面钩子 属于的是:全局导航钩子、组件内的钩子、单独路由独享组件

54.主要实现数据双向绑定，通常用在表单元素的命令是：v-model

55.不属于 vuex 中的属性的是:init

56.想要获取购物车小球在页面上的位置，以下可以使用的是:getBoundingClientRect()

57.不可以进行路由跳转的是:jump()

58.

一 单选题，每道小题 1 分，共 9 道小题，9 分

1. 下面选项中，可以用来创建 store 实例对象的是（）。

A new Vuex.Store({})

B new Vue({})

C new Vuex({})

D new Store({})

正确答案：A

1. 以下代码

```
<div id="app">

    <p>{{this.$store.state.name}}</p>
</div>
<script>
var store = new Vuex.Store({

   state:{name:'store'}

})
var vm   = new Vue({el:'#app',store})
</script>复制代码
```

中的 this 指的是（）。

A p 标签

B window 对象

C store 实例

D Vue 实例

正确答案：D

1. 以下代码

```
const store = new Vuex.Store({

   state: { name: 'name 初始值' },

  })

store.replaceState({ name: 'b' })
```

console.log(store.state.name)执行后的结果是（）。

A name

B name 初始值

C b

D a

正确答案：C

1. 下面选项中，可以实现 actions 中事件处理函数状态分发的是（）。

A mutations

B actions()

C commit()

D dispatch()

正确答案：D

1. devtools 选项用来设置是否在 devtools 调试工具中启用 Vuex，表示启用的值是（）。

A vuex

B false

C devtools

D true

正确答案：D

1. 下面选项中，可以用来创建 vue 项目的命令的是（）。

A vue init shopcart

B vue init webpack shopcart

C vue shopcart

D vue install shopcart

正确答案：B

答案解析：**store 实例配置中的 devtools 选项用来设置是否在 devtools 调试工具中启用 Vuex，默认值为 true，表示在启用，设为 false 表示停止使用。

devtools 选项经常用在一个页面中存在多个 store 实例的情况。**

1. 以下代码执行后的结果是（）。

```
  state: {
    count: 0
  },
  // 修改 count 的值
  mutations: {
    increase (state) {
     console.log(state.count)       }
  }
})
```

store.commit('increase');复制代码

A 2

B -1

C 1

D 0

正确答案：D

1. 下面选项中，用来定义 store 实例的初始数据的是（）。

A getters

B actions

C mutations

D state

正确答案：D

答案解析：state 配置选项定义组件初始状态，类似于 Vue 实例中的 data 属性。

下面选项中，可以实现 mutations 中事件处理函数状态提交的是（B）。

A actions

B commit()

C mutations()

D dispatch()

正确答案：B

二 多选题，每道小题 2 分，共 7 道小题，14 分

1. 下列选项中，subscribe 函数接收的参数包括（）。

A actions

B getters

C mutation

D state

正确答案：C,D

1. 下列选项中，说法正确的是。（）

A 在 getters 中定义的方法，接收 commit 作为参数

B 只有当 getters 的依赖值发生了改变才会被重新计算

C getters 类似于 Vue 实例的 computed

D store 实例允许在 store 中定义 getters 计算属性

正确答案：B,C,D

1. 下面选项中，说法正确的是。

A mutations 选项可以用 methods 替代使用

B commit()可以接收{ type: 'receive', name: '我是传递的参数' }配置对象作为参数

C 在 Vuex 配置选项中，actions 是同步执行的

D 在 Vuex 配置选项中，mutations 是同步执行的

正确答案：B,D

答案解析：mutations 是同步函数，组件状态发生变化时，触发 mutations 中的事件处理方法来更新页面状态的变化，这是一种同步状态。

定义 mutations 中的事件处理方法，接收受参数为 state 和 param，也可以直接接受{type:'',param:''}配置对象。

methods 是在创建 Vue 实例时定义的，而 mutations 是在 store 实例创建时定义的，所以不能替代使用。

1. 下面选项中，属于模块操作方法的是（）。

A store.unregister('moduleName')

B store.Module('moduleName')

C store.unregisterModule('moduleName')

D store.registerModule('moduleName')

正确答案：C,D

答案解析：暂无解析

1. 下面选项中，关于 Vuex 状态管理模式说法正确的是。（）

A Vuex 是专门为 Vue 设计的状态管理库

B 在 Vue 中，组件的状态变化是通过 Vue 单向数据流的设计理念实现的

C Vue 的单向数据流增强了组件之间的独立性

D Relux 是专门为 Vue 设计的状态管理库

正确答案：A,B,C

1. 下列选项中，属于 modules 中的配置选项的是（）。

A actions

B getters

C state

D key

正确答案：A,B,C

答案解析：modules 是 store 实例对象的选项，其参数构成主要包括:

key: {

state,

mutations,

actions,

getters,

modules

},

其中，state 表示初始数据、mutations 表示状态提交、modules 表示模块注册、getters 表示计算属性和 actions 状态分发。

1. 下面选项中，说法正确的是。（）

A Vuex 的状态存储是响应式的，当 store 中数据状态发生变化，那么页面中的 store 数据也发生相应变化。

B this.$router.state 可以获取到 store 中的 state 数据

C 改变 store 中的状态的唯一途径就是显式地提交 mutation

D 每一个 Vuex 应用的核心就是 store（仓库），即响应式容器

正确答案：A,C,D

三 判断题，每道小题 1 分，共 9 道小题，9 分

1. 通过 store.replaceState({})可以切换 Vue 中 data 的数据状态。（）

正确答案：错

1. store 实例对象中的 plugins 选项的功能是定义初始数据。（）

正确答案：错

答案解析：Vuex 中的插件配置选项为 plugins。

1. 在 Vue 中，组件的状态变化是通过 Vue 单向数据流的设计理念实现的。（）

正确答案：对

1. actions 中的事件方法接收参数只能是 commit 对象，主要是用来完成状态提交。（）

正确答案：错

1. 通过 state 配置选项定义组件初始状态，类似于 Vue 实例中 data 属性。（）

正确答案：对

1. getters 中的事件方法可以接收其它的 getters 作为第二个参数。（）

正确答案：对

答案解析：getters 中的 doneTodosCount 接收其他 getters 作为第 2 个参数；

1. mutations 中的事件方法可以接收参数是 commit 对象，主要是用来完成状态提交。（）

正确答案：错

1. 构造器创建 store 实例后，需要挂载到 vm 实例才能在项目中进行使用。（）

正确答案：对

1. 通过 store.unregisterModule('moduleName')来卸载静态模块。（）

正确答案：错

四 填空题，每道小题 1 分，共 9 道小题，9 分

1. store 实例对象提供了动态创建模块的接口，即_____方法。

正确答案：store.registerModule()

1. store.registerModule()方法之前首先要完成 store 实例对象创建，方法接受_____作为第一个参数。

正确答案：模块名称

答案解析：调用 store.registerModule()方法之前首先要完成 store 实例对象创建，方法接收模块名称"myModule"作为第 1 个参数；

1. mutations 中的事件处理函数可以接受_____作为参数，即初始数据。

正确答案：state

1. actions 中的事件处理函数可以接受{commit}作为第一个参数，用来完成_____状态的提交。

正确答案：mutation

1. state 数据状态操作，可以通过_____方法实现状态替换。

正确答案：store.replaceState()

1. store 实例配置中的_____选项用来设置是否在 vue-devtools 调试工具中启用 Vuex。

正确答案：devtools

1. 通过调用 store.replaceState()方法能够实现状态切换，接收参数为_____新对象。

正确答案：state

1. getters 中定义事件处理函数方法，该方法接收_____参数作为第一个参数。

正确答案：state

1. 以下代码

```
<script>
      const store = new Vuex.Store({        state: {          count :
0        },        // 修改 count 的值        mutations:
{          increase (state)
{            state.count++          }        }     })
```

store.commit('increase')　　console.log(store.state.count)</script>复制代码

执行后的结果是_____。

正确答案：1

一 单选题，每道小题 1 分，共 20 道小题，20 分

1. 下面选项中，可以访问原始数据对象的是（）。

A vm.$methods

B vm.$props

C vm.$el

D vm.$data

正确答案：D

答案解析：Vue 实例创建之后，可以通过 vm.$data 访问原始数据对象。

Vue 实例也代理了 data 对象上所有的属性，因此访问 vm.name 相当于访问 vm.$data.name。

1. 以下代码

```
var vm = new Vue({
  el: '#app',
   data: {
    price: 20,
    num: 1
  },
  computed: {
   // 总价格 totalPrice
   totalPrice () {
    return this.price * this.num
```

```
    }
  }
})
```

console.log(vm.totalPrice);执行后的结果是（）。

A 0

B 2

C 20

D 1

正确答案：C

1. 下列选项中，在插值表达式中可以使用过滤器来对数据进行处理，管道符号是（）。

A /

B >

C |

D \

正确答案：C

1. 下列选项中，可以实现绑定事件的指令是（）。

A v-show

B v-on

C v-html

D v-text

正确答案：B

答案解析：v-show：控制元素显示隐藏；

v-on：监听事件；

v-html：插入包含 HTML 的内容；

v-text：插入文本内容；

1. 下列选项中，代表视图部分的是（）。

A Element

B DOM

C Model

D View

正确答案：D

答案解析：MVVM 主要包含 3 个部分，分别是 Model、View 和 ViewMode。

Model 指的是数据部分，主要负责业务数据；View 指的是视图部分，即 DOM 元素，负责视图的处理。

1. Windows 系统内置的命令行工具是（）。

A npm

B cmder

C git-bash

D cmd

正确答案：D

1. 以下代码

```
{{message | toUpcase}}
```

执行后的结果是（）。

A HELLOworld

B Helloworld

C HELLOWORLD

D HelloWorld

正确答案：C

1. 下列选项中，可以实现阻止事件冒泡行为的是（）。

A .self

B .capture

C .prevent

D .stop

正确答案：D

答案解析：.stop：阻止事件冒泡；

.self：将事件绑定到自身，只有自身才能触发；

.capture：事件捕获；

.prevent：阻止默认事件行为；

1. 下面选项中，在创建 Vue 实例时，用来表示唯一根标签的是（）。

A watch

B components

C data

D el

正确答案：D

答案解析：在创建 Vue 实例时，el 表示唯一根标签，通过 class 或 id 选择器可用来将页面结构与 Vue 实例对象 vm 中的 el 绑定。

1. 下面选项中，是基于 Chrome V8 引擎的 JavaScript 运行环境，它可以让 JavaScript 运行在服务器端的是（）。

A npm.js

B jQuery.js

C vue.js

D Node.js

正确答案：D

1. 下面选项中，主要用来编写网页的结构的是（）。

A php

B HTML

C CSS

D JavaScript

正确答案：B

答案解析：Web 前端使用 HTML、CSS 和 JavaScript 作为基础语言，它们分别用来实现网页的结构、样式和行为。

1. 下面选项中，用于卸载指定包依赖的命令的是（）。

A npm uninstall

B npm install

C npm update

D npm start

正确答案：A

答案解析：? npm install：安装项目所需要的全部包，需要配置 package.json 文件；

? npm uninstall：卸载指定名称的包；

? npm install 包名：安装指定名称的包，后面可以跟参数"-g"表示全局安装，"--save"表示本地安装；

? npm update：更新指定名称的包；

? npm start：项目启动；通过 CDN 方式引入 Vue，可以缓解服务器的压力，加快文件的下载速度。目前，网络上有很多免费的 CDN 服务器可以使用

? npm run build：项目构建。

1. 下列选项中，可以实现获取随机数的是（）。

A Math.max()

B Math.min()

C Math.random()

D Math.floor()

正确答案：C

答案解析：Math.max()获取最大数；

Math.min()获取最小数；

Math.random()获取随机数；

Math.floor()向下取整；

1. 下面选项中，表示一个 Node.js 的包管理工具，用来解决 Node.js 代码部署问题的是（）。

A webpack

B vue

C Node

D npm

正确答案：D

答案解析：npm（Node.js Package Manager）是一个 Node.js 的包管理工具，用来解决 Node.js 代码部署问题。

在安装 Node.js 时会自动安装相应的 npm 版本，不需要单独安装。

1. 下面选项中，可以用来创建实例对象的关键字是（）。

A let

B var

C data

D new

正确答案：D

答案解析：创建 Vue 实例的基本代码如下：

其中，使用 new 关键字创建 Vue 实例对象。

1. 下面选项中，可以通过插值语法将 data 初始数据绑定到页面中的是（）。

A [[]]

B {}

C {{}}

D []

正确答案：C

1. 以下代码

```
var vm = new Vue({
    el: '#app',
    data: {
      cityName: 'shanghai'
    },
    // 使用 watch 监听 cityName 变化
    watch: {
      cityName (newName, oldName) {
        console.log(newName, oldName)
      }
    }
  })
```

vm.$data.cityName = 'beijing'执行后的结果是（）。

A bei shang

B shanghai

C beijing shanghai

D beijing

正确答案：C

1. 下面选项中，那一项是一款基于 Chrome 浏览器的扩展，用于调试 Vue 应用的工具（）。

A VS code

B vue-devtools

C Wechat

D Chrome

正确答案：B

答案解析：vue-devtools 是一款基于 Chrome 浏览器的扩展，用于调试 Vue 应用，只需下载官方压缩包，配置 Chrome 浏览器的扩展程序即可使用。

1. 下列选项中，methods 选项描述错误的是（）。

A methods 选项中可以定义初始数据

B methods 选项中用来定义事件处理函数

C methods 选项中函数可以采用省略 function 的写法来实现

D methods 选项中的函数可以通过 this 指针获取初始数据

正确答案：A

答案解析：methods 属性用来定义方法，通过 Vue 实例可以直接访问这些方法。

在定义的方法中，this 指向 Vue 实例本身。

定义在 methods 属性中的方法可以作为页面中的事件处理方法使用，当事件触发后，执行相应的事件处理方法。

二 多选题，每道小题 2 分，共 13 道小题，26 分

1. 下列选项中，可以实现插入内容的指令是（）。

A v-show

B v-on

C v-html

D v-text

正确答案：C,D

答案解析：v-html：插入包含 HTML 的内容；

v-text：插入文本内容；

1. 下面选项中，关于 Visual Studio Code 编辑器说法正确的是（）。

A Visual Studio Code 编辑器不仅跨平台（支持 Mac Windows 以及 Linux），使用起来也非常简单。

B 不能安装插件使用

C Visual Studio Code 编辑器具备语法高亮显示 智能代码补全 自定义快捷键和代码匹配等功能。

D 轻巧极速，占用系统资源较少。

正确答案：A,C,D

答案解析：VS Code 编辑器特点主要包括：

1.轻巧极速，占用系统资源较少。

2.具备语法高亮显示、智能代码补全、自定义快捷键和代码匹配等功能。

3.跨平台。不同的开发人员为了工作需要，会选择不同平台来进行项目开发工作，这样就一定程度上限制了编辑器的使用范围。VS Code 编辑器不仅跨平台（支持 Mac、Windows 以及 Linux），使用起来也非常简单。

（4）主题界面的设计比较人性化。例如，可以快速查找文件直接进行开发，可以通过分屏显示代码，主题颜色可以进行自定义设置（默认是黑色），也可以快速查看最近打开的项目文件并查看项目文件结构。

（5）提供了丰富的插件。VS Code 提供了插件扩展功能，用户根据需要自行下载安装，只需在安装配置成功之后，重新启动编辑器，就可以使用此插件提供的功能。

1. 下列选项中，可以实现为按钮绑定点击事件的是（）。

A <button $click="showInfo">

B <button #click="showInfo">

C <button @click="showInfo">

D

正确答案：C,D

1. 下面选项中，属于 Visual Studio Code 编辑器优势的是（）。

A 简单

B 繁琐

C 高效

D 便捷

正确答案：A,C,D

答案解析：VS Code 编辑器特点主要包括：

1.轻巧极速，占用系统资源较少。

2.具备语法高亮显示、智能代码补全、自定义快捷键和代码匹配等功能。

3.跨平台。不同的开发人员为了工作需要，会选择不同平台来进行项目开发工作，这样就一定程度上限制了编辑器的使用范围。VS Code 编辑器不仅跨平台（支持 Mac、Windows 以及 Linux），使用起来也非常简单。

（4）主题界面的设计比较人性化。例如，可以快速查找文件直接进行开发，可以通过分屏显示代码，主题颜色可以进行自定义设置（默认是黑色），也可以快速查看最近打开的项目文件并查看项目文件结构。

（5）提供了丰富的插件。VS Code 提供了插件扩展功能，用户根据需要自行下载安装，只需在安装配置成功之后，重新启动编辑器，就可以使用此插件提供的功能。

1. 下面选项中，用来与 Vue 实例对象 vm 中的 el 绑定选择器是（）。

A class

B id

C name

D in

正确答案：A,B

1. Node.js 有两个版本，主要包括（）。

A MIN

B LTS

C Current

D LONG

正确答案：B,C

答案解析：Node.js 有两个版本，LTS（Long Term Support）是提供长期支持的版本，只进行微小的 Bug 修复且版本稳定，因此有很多用户在使用；

Current 是当前发布的最新版本，增加了一些新特性，有利于进行新技术的开发使用。

1. 下列选项中，关于 watch 选项说法正确的是（）。

A watch 选项中可以监听初始数据状态变化

B 当 data 中数据状态发生变化时，会触发 watch 选项中对应的事件处理函数

C watch 选项可以用来替代 computed 选项使用

D watch 选项中函数接收参数用来表示，新值和旧值。

正确答案：A,B,D

1. 下面选项中，关于 Vue 说法正确的是（）。

A Vue 不可以用来构建单页应用

B Vue 通过虚拟 DOM 技术来减少对 DOM 的直接操作

C Vue 支持单向和双向数据绑定

D Vue 组件化的特性提高了开发效率 使代码更容易复用

正确答案：B,C,D

答案解析：单页应用是前端开发的一种形式，在切换页面的时候，不会刷新整个页面，而是通过 Ajax 异步加载新的数据，改变页面的内容。为了更方便地开发这类复杂的应用，市面上出现了 Angular、React、Vue 等框架。Vue 通过虚拟 DOM 技术来减少对 DOM 的直接操作，通过尽可能简单的 API 来实现响应的数据绑定，支持单向和双向数据绑定。组件化的特性提高了开发效率、使代码更容易复用，提高了项目的可维护性，便于团队的协同开发。

1. 下列选项中，Vue.js 的版本主要包括（）。

A vue.press.js

B 开发版本

C 生产版本

D vue.max.js

正确答案：B,C

1. 下面选项中，说法正确的是（）。

A el 用来定义唯一根标签

B data 中定义初始数据

C watch 中定义初始数据

D methods 中可以定义事件处理函数

正确答案：A,B,D

答案解析：Vue 提供了 watch 状态监听功能，只需监听当前 Vue 实例中的数据变化，就会调用当前数据所绑定的事件处理方法。

1. 下列选项中，关于 computed 属性说法正确的是（）。

A 当有一些数据需要随着其他数据变动而变动时，就需要使用 computed 计算属性。

B computed 属性中函数可以通过 this 获取到初始数据

C computed 属性中可以定义页面的事件处理函数

D 实现了一种更通用的方式来观察和响应 Vue 实例上的数据变动

正确答案：A,B,D

答案解析：Vue 提供了一种更通用的方式来观察和响应 Vue 实例上的数据变动，当有一些数据需要随着其他数据变动而变动时，就需要使用 computed 计算属性。

在事件处理方法中，this 指向的 Vue 实例的计算属性结果会被缓存起来，只有依赖的响应式属性变化时，才会重新计算，返回最终结果。

1. 下面选项中，用于开发和调试 Web 项目的工具的是（）。

A 浏览器

B 选择器

C 计时器

D vue-devtools 扩展

正确答案：A,D

1. 下列选项中，关于学生列表案例实现的说法正确的是（）。

A 学生信息列表可以通过数组的 pop()方法实现删除学生信息

B 通过数组定义学生信息列表

C 学生信息列表可以通过数组的 pull()方法实现添加学生信息

D v-for 指令可以实现学生列表渲染

正确答案：A,B,D

三 判断题，每道小题 1 分，共 36 道小题，36 分

1. 指令 npm install webpack@4.27.x -g 表示全局安装 webpack，且版本号是 4.27.x。（）

正确答案：对

1. 生命周期函数的最后阶段是实例的销毁。（）

正确答案：对

答案解析：生命周期函数的最后阶段是实例的销毁，会执行 beforeDestroy 和 destroyed 钩子函数。

1. 生命周期函数的最后阶段是实例的销毁，会执行 beforeDestroy 和 destroyed 钩子函数。（）

正确答案：对

1. Vue 中，可以通过绑定类名实现元素样式。（）

正确答案：对

1. 在 Vue 项目中，每个 Vue 应用都是通过 Vue 构造器创建新的 Vue 实例开始的。（）

正确答案：对

1. Vue 与 Angular 较为相似，但 Angular 在学习上有一定难度。（）

正确答案：对

答案解析：Angular 的学习成本高，使用起来比较复杂，而 Vue 相对简单、直接，所以 Vue 使用起来更加友好，更简单。

1. 通过<a href="#" @click.prevent="">登录实现阻止 a 标签的默认事件行为。（）

正确答案：对

1. 通过"{{data}}"语法，可以将 data 中数据插入到页面中。（）

正确答案：对

1. Vue 配置对象中，使用 watch 监听数据变化，并触发相应事件处理函数。（）

正确答案：对

1. 事件修饰符是自定义事件行为，配合 v-on 指令来使用，写在事件之后，用"."符号连接。（）

正确答案：对

1. 通过使用.prevent 修饰符来阻止<a>标签的默认行为。（） 正确答案：对

答案解析：.prevent 可以实现阻止默认事件行为；

1. methods 属性用来定义方法，通过 Vue 实例对象可以直接访问这些方法。（）

正确答案：对

答案解析：methods 属性用来定义方法，通过 Vue 实例可以直接访问这些方法。

在定义的方法中，this 指向 Vue 实例本身。

定义在 methods 属性中的方法可以作为页面中的事件处理方法使用，当事件触发后，执行相应的事件处理方法。

1. v-text 内部指令表示插入文本内容。（）

正确答案：对

1. JavaScript 主要包括行为，主要是为页面提供交互效果，实现更好的用户体验。（）

正确答案：对

答案解析：Web 前端使用 HTML、CSS 和 JavaScript 作为基础语言，它们分别用来实现网页的结构、样式和行为。

JavaScript 的功能主要包括实现页面逻辑、行为、动作等，用来动态操作元素的属性，主要是为页面提供交互效果，实现更好的用户体验。

1. vue-cli 是快速创建 Vue 项目的脚手架工具。（）

正确答案：对

1. git-bash 命令行工具可以代替 Windows 内置的 cmd 命令行工具来使用。（）

正确答案：对

答案解析：git-bash 是 git（版本管理器）中提供的一个命令行工具，外观类似于 Windows 系统内置的 cmd 命令行工具，但用户体验更友好。

在实际开发中，经常会使用 git-bash 工具代替 cmd。

1. 在构建大型交互式项目时，开发者需要编写大量的 JavaScript 代码来操作 DOM，不需要处理浏览器的兼容问题。（）

正确答案：错

答案解析：在构建大型交互式项目时，开发者需要编写大量的 JavaScript 代码来操作 DOM（文档对象模型），并处理浏览器的兼容问题，代码逻辑越来越烦琐。

1. npm run build 命令表示项目构建。（）

正确答案：对

答案解析：? npm install：安装项目所需要的全部包，需要配置；package.json 文件；

? npm uninstall：卸载指定名称的包；

? npm install 包名：安装指定名称的包，后面可以跟参数"-g"表示全局安装，"--save"表示本地安装；

? npm update：更新指定名称的包；

? npm start：项目启动；

? npm run build：项目构建。

1. 通过 methods 选项，可以定义 Vue 实例中的事件处理函数。（）

正确答案：对

1. 通过 new Vue({})的方式创建 Vue 实例对象。（）

正确答案：对

1. input 中的 v-model 用于在表单控件元素上创建双向数据绑定。（）

正确答案：对

1. Windows 内置的 cmd 命令行工具与 git-bash 完全相同。（）

正确答案：错

答案解析：git-bash 是 git（版本管理器）中提供的一个命令行工具，外观类似于 Windows 系统内置的 cmd 命令行工具，但用户体验更友好。

在实际开发中，经常会使用 git-bash 工具代替 cmd。

1. 管道符之前代码执行的结果会传到后面作为参数进行处理。（）

正确答案：对

答案解析：管道符之前代码执行的结果会传到后面作为参数进行处理。

在多个参数进行传递时，第一个参数就是前一个方法执行的结果。

1. Vue 中的页面结构是由组件构成的，不同组件可以表示不同页面，适合进行单页应用开发。（）

正确答案：对

答案解析：在 Vue 中，组件是构成页面中独立结构单元，能够减少重复代码的编写，提高开发效率，降低代码之间的耦合程度，使项目更易维护和管理。

组件主要以页面结构形式存在，不同组件也具有基本交互功能，根据业务逻辑实现复杂的项目功能。

1. v-show 内部指令表示显示隐藏。（）

正确答案：对

1. Vue 的核心文件有两种版本，其中开发版本和生产版本是完全相同的。（）

正确答案：错

1. VS Code 提供了插件扩展功能，用户根据需要自行下载安装。（）

正确答案：对

1. vue-devtools 可以独立于 Chrome 浏览器单独使用。（）

正确答案：错

答案解析：vue-devtools 是一款基于 Chrome 浏览器的扩展，用于调试 Vue 应用，只需下载官方压缩包，配置 Chrome 浏览器的扩展程序即可使用。

1. Node.js 还提供了交互式环境 REPL，类似 Chrome 浏览器的控制台。（）

正确答案：对

1. 在使用 vue-devtools 调试工具之前，需要将 vue-devtools 调试工具添加到 Chrome 浏览器的扩展程序中去。（）

正确答案：对

1. Vue 中，可以用来绑定行内样式的是 v-bind。（）

正确答案：对

1. 目前市场三大前端主流框架分别是 Angular、React 和 Vue.js。（）

正确答案：对

答案解析：目前市场三大前端主流框架分别是 Angular、React 和 Vue.js。

Vue 之所以被开发者青睐，主要是 Vue 秉承了 Angular 和 React 框架两者的优势，并且 Vue 的代码简洁、上手容易，在市场上也得到大量应用。

1. 定义在 methods 属性中的方法可以作为页面中的事件处理方法使用，当事件触发后，执行相应的事件处理方法。（）

正确答案：对

答案解析：methods 属性用来定义方法，通过 Vue 实例可以直接访问这些方法。

在定义的方法中，this 指向 Vue 实例本身。

定义在 methods 属性中的方法可以作为页面中的事件处理方法使用，当事件触发后，执行相应的事件处理方法。

1. VS Code 编辑器不可以跨平台（支持 Mac、Windows 以及 Linux），使用起来也非常简单。（）

正确答案：错

答案解析：不同的开发人员为了工作需要，会选择不同平台来进行项目开发工作，这样就一定程度上限制了编辑器的使用范围。

VS Code 编辑器不仅跨平台（支持 Mac、Windows 以及 Linux），使用起来也非常简单。

1. Vue（读音/Vju?/，类似于 View）是一套用于构建用户界面的渐进式框架。（）

正确答案：对

1. Current 是当前发布的最新版本，增加了一些新特性，有利于进行新技术的开发使用。（）

正确答案：对

四 填空题，每道小题 1 分，共 21 道小题，21 分

1. 4.x 版本的 webpack 还需要安装_____脚手架工具才能正常使用。

正确答案：webpack-cli

1. 安装 webpack 完成后，执行_____命令，查看 webpack 版本。

正确答案：webpack -v

1. Vue.component()方法接收的第一个参数是_____。

正确答案：组件名

1. 以下代码

```
<div id=″app″>

 <p>{{msg}}</p>
</div>
<script>

  var vm = new Vue({

       el: '#app',

       data: { msg: '张三' },

       beforeMount () {

           console.log(this.$el.innerHTML)

       },

   })
</script>复制代码
```

执行后结果是_____。

正确答案： {{msg}}

1. _____主要包括内置指令和自定义指令，以"v-"开头，作用于 HTML 元素。

正确答案：指令

答案解析：指令主要包括内置指令和自定义指令，以"v-"开头，作用于 HTML 元素。

指令提供了一些特殊的特性，将指令绑定在元素上时，指令会给绑定的元素添加一些特殊的行为。

例如，v-bind 动态绑定指令、v-if 条件渲染指令、v-for 列表渲染指令等。

1. Vue 的数据驱动是通过_____模式来实现的。

正确答案：MVVM

答案解析：MVVM 主要包含 3 个部分，分别是 Model、View 和 ViewMode。

Model 指的是数据部分，主要负责业务数据；View 指的是视图部分，即 DOM 元素，负责视图的处理。

ViewModel 是连接视图与数据的数据模型，负责监听 Model 或者 View 的修改。

1. _____样式包括颜色、大小、字体等，可以实现漂亮、布局合理的页面效果。

正确答案：CSS

答案解析：Web 前端使用 HTML、CSS 和 JavaScript 作为基础语言，它们分别用来实现网页的结构、样式和行为。

其中，CSS 样式包括颜色、大小、字体等，实现漂亮、布局合理的页面效果。

1. vue-cli 安装完成后，可以执行_____命令，查看安装的版本号。

正确答案：vue -V

1. <template>标签中书写 HTML 代码，然后通过_____值绑定到组件内的 template 属性上。

正确答案：id

1. 局部注册组件的配置对象中，表示组件模板的是_____。

正确答案：template

1. Vue 实例创建后，如果挂载点_____唯一根标签存在，就进行页面挂载。

正确答案：el

1. Vue 生命周期函数中，_____函数会在创建实例对象之前执行。

正确答案：beforeCreate

答案解析：beforeCreate：创建实例对象之前执行；

created：创建实例对象之后执行；

beforeMount：页面挂载成功之前执行；

mounted：页面挂载成功之后执行；beforeUpdate：组件更新之前执行；

beforeUpdate：组件更新之前执行；

updated：组件更新之后执行；

beforeDestroy：实例销毁之前执行；

destroyed：实例销毁之后执行

1. Vue 实例对象中，通过_____选项定义组件。

正确答案：components

1. 当元素绑定的 v-if 指令的值设置为_____时表示显示元素，否则隐藏元素。

正确答案：true

答案解析：v-if 指令值为 true，表示显示当前组件，否则隐藏当前组件；

1. _____即道具，用来接受父组件中定义的数据，其值为数组。

正确答案：props

1. 以下代码

```
var vm = new Vue({
    el: '#app',
    data: { msg: '张三' },
    created () {
```

```
    console.log(this.$data.msg)
  }
})
```

执行后结果是_____。

正确答案：张三

1. Vue 提供方法中，可以用来实现全局注册组件的是_____。

正确答案：Vue.component()

答案解析：Vue.component()方法用于全局注册组件，除了全局注册组件外，还可以局部注册组件，通过 Vue 实例的 components 属性来实现。

1. _____可以触发父组件中定义的事件，子组件的数据信息通过传递参数的方式完成。

正确答案：$emit

1. Vue 实例对象中，通过_____选项表示计算属性。

正确答案：computed

1. 在实现组件切换时，可以通过_____属性匹配组件的名称实现组件切换。

正确答案：is

1. Vue 生命周期函数中，_____函数会在创建实例对象完成之后执行。

正确答案：created

答案解析：beforeCreate：创建实例对象之前执行；

created：创建实例对象之后执行；

beforeMount：页面挂载成功之前执行；

mounted：页面挂载成功之后执行；beforeUpdate：组件更新之前执行；

beforeUpdate：组件更新之前执行；

updated：组件更新之后执行；

beforeDestroy：实例销毁之前执行；

destroyed：实例销毁之后执行

五 程序填空，共 15 个空，每空 2 分，共 30 分

1. 通过在输入框内输入内容在 list 中查找数据并加以显示。

html 内容部分：

```
<div id="app">
    <input type="text" v-model="query" placeholder="请输入要查找
的内容">
    <transition-group [1]_____ ="ul"
@before-enter="beforeEnter" @enter="enter" @leave="leave"
v-bind:css="false">
        <li v-for="(item,index) in resultList" [2]_____
="item.msg" :data-index="index">{{item.msg}}</li>
    </transition-group>
</div>
```

脚本部分：

```
var vm = new Vue({
    el: '#app',
    data: {
        query: '',
        list: [{msg: 'a'}, {msg: 'ab'}, {msg: 'ca'}, {msg: 'd'}]
    },
    methods: {
        beforeEnter(el) {
            el.style.opacity = 0
            el.style.height = 0
        },
        enter(el, done) {
            var delay = el.dataset.index * 150
            setTimeout(function() {
```

```
            Velocity(el, {
                opacity: 1,
                height: '1.6em'
            }, {
                complete: done
            })
        }, delay)
    },
    leave(el, done) {
        var delay = el.dataset.index * 150
        setTimeout(function() {
            Velocity(el, {
                opacity: 0,
                height: 0
            }, {
                complete: done
            })
        }, delay)
    }
},
[3]__________: {
    resultList() {
        var key = [4]_____________
        return this.list.filter(function(item) {
            if (item.msg.indexOf(key) != -1) {
                return [5]____________
            }
        })
    }
}
})
```

1. 通过输入框录入学生信息；通过学号删除学生信息。

html 内容部分：

```
<div id="app">
        <b-component @addstu="addOneStudent"
@deletestu="deleteOneStudent"></b-component>
        <l-component v-bind:stulist="students"></l-component>
    </div>
```

```
    <template id="buttonComponent">
    <div>
      <div>
        班级： <input type="text" v-model="cname">
        学号： <input type="text" v-model="number">
        姓名： <input type="text" v-model="name">
        <button @click="addStudent">添加</button>
      </div>
      <div>
        学号： <input type="text" v-model="deleteNumber"> <button
@click="deleteStudent">删除</button>
      </div>
    </div>
  </template>
    <template id="listComponent">
    <table border="1" width="30%" style="border-collapse:collapse">
      <tr>
        <th>班级</th>
        <th>学号</th>
        <th>姓名</th>
      </tr>
      <tr v-for="item in stulist">
```

```
        <td>{{item.cname}}</td>
        <td>{{item.number}}</td>
        <td>{{item.name}}</td>
      </tr>
    </table>
  </template>复制代码
```

脚本部分：

```
var bc = {
        template: '#buttonComponent',
        data() {
            return {
                cname: '',
                number: '',
                name: '',
                deleteNumber: ''
            }
        },
        methods: {
            addStudent() {
                this.$emit('[1] _____', {
                    cname: this.cname,
                    number: this.number,
                    name: this.name
                })
            },
```

```
            deleteStudent() {
                this.$emit('[2] __________',
this.deleteNumber)
            }
        }
    }
    var lc = {
        template: '#listComponent',
        props: ['[3]________________'],
    }
    var vm = new Vue({
        el: '#app',
        [4] _____________: {
            bComponent: bc,
            lComponent: lc
        },
        data: {
            stu: {},
            students: []
        },
        methods: {
            addOneStudent(one) {
                this.students.push(one)
            },
            deleteOneStudent(dnumber) {
                this.students.splice(this.students.findInde
x(e => [5]_____________ === dnumber), 1)
            }
        }
    })
```

1. 输入框内输入 1，2，3 分别表示石头，剪刀，布；点击 pk 按钮，电脑自动生成石头，剪刀，布。并判断是用户胜利还是电脑胜利。

html 内容部分：

```
<div id="app">
        <div>
            <h3>猜拳游戏</h3>
            数字 1：表示"石头"； 数字 2：表示"剪刀"； 数字 3：表示"布"；<input v-model="userData">
            <div>
                用户：{{user}} 电脑：{{computerData}} 胜负：{{success}}
            </div>
            <div>
                <button @ [1]__________="pk">PK</button>
            </div>
        </div>
</div>
```

脚本部分：

```
var vm = new Vue({
        el: '#app',
        [2] _________ : {
            userData: '',
            computerData: '',
            user: '',
```

```
    success: ''
},
[3] __________: {
    pk() {
        switch (parseInt( [4] __________ )) {
            case 1:
                this.user = '石头'
                break;
            case 2:
                this.user = '剪刀'
                break;
            case 3:
                this.user = '布'
                break;
            default:
                this.user = '无效'
                break;
        }
        var c = Math.floor(Math.random() * 3) + 1;
        switch (c) {
            case 1:
                this.computerData = '石头'
                break;
            case 2:
                this.computerData = '剪刀'
```

```
                    break;
                case 3:
                    this.computerData = '布'
                    break;
                default:
                    this.computerData = '无效'
                    break;
            }
            if (this.user != "无效") {
                if (this.userData == c) {
                    this.success = '平局'
                } else if (this.userData - c ==
[5]_____ || this.userData - c == 2) {
                    this.success = '用户胜利'
                } else {
                    this.success = '电脑胜利'
                }
            } else {
                this.success = '无效'
            }
        }
    }
})
```

学生答案：

第 1 个填空：

正确答案：tag

第 2 个填空：

正确答案：:key

第 3 个填空：

正确答案：computed

第 4 个填空：

正确答案：this.query

第 5 个填空：

正确答案：true

第 6 个填空：

正确答案：addstu

第 7 个填空：

正确答案：deletestu

第 8 个填空：

正确答案：stulist

第 9 个填空：

正确答案：components

第 10 个填空：

正确答案：e.number

第 11 个填空：

正确答案：click

第 12 个填空：

正确答案：data

第 13 个填空：

正确答案：methods

第 14 个填空：

正确答案：this.userData

第 15 个填空：

正确答案：-1

一 单选题，每道小题 1 分，共 7 道小题，7 分

1. 下面选项中，用来访问子组件实例的是（）。

A vm.$children

B vm.$props

C vm.$slots

D vm.$attrs

正确答案：A

答案解析：vm.$children 用来获取当前实例的直接子组件。

需要注意的是，$children 并不保证顺序，也不是响应式的。

1. 下面选项中，用于向响应式对象中动态添加一个属性，并确保这个新属性同样是响应式的，且触发视图更新的是（）。

A Vue.extends

B Vue.use

C Vue.extend

D Vue.set

正确答案：D

1. 下面选项中，能够实现自定义指令的是（）。

A Vue.use()

B Vue.set()

C Vue.component()

D Vue.directive()

正确答案：D

答案解析：Vue 中有很多内置指令，如 v-model、v-for 和 v-bind 等。

除了内部内置指令，开发人员也可以根据需求使用 Vue.directive 注册自定义指令。

1. 下面选项中，用来访问 vm 实例使用的根 DOM 元素的是（）。

A vm.$el

B vm.$props

C vm.$slots

D vm.$attrs

正确答案：A

答案解析：vm.$el 用来访问 vm 实例使用的根 DOM 元素；

1. Vue 全局配置对象 config 中用于控制生产信息的显示和隐藏的是（）。

A devtools

B productionTip

C tip

D silent

正确答案：B

答案解析：通过 Vue 的全局配置 productionTip 可以控制生产信息的显示或隐藏；

1. 下面选项中，能够实现在 Vue 中安装插件的是（）。

A Vue.use()

B Vue.set()

C Vue.component()

D Vue.directive()

正确答案：A

答案解析：Vue.use 主要用于在 Vue 中安装插件，通过插件可以为 Vue 添加全局功能。

1. 下面选项中，用来获取用户自定义选项的是（）。

A vm.$data

B vm.$props

C vm.$attrs

D vm.$options

正确答案：D

二 多选题，每道小题 2 分，共 3 道小题，6 分

1. 下面选项中，vm.$attrs 获取的属性不包括的是（）。

A class

B style

C props

D id

正确答案：A,B,C

答案解析：vm.$attrs 可以获取组件的属性，但其获取的属性中不包含 class、style 以及被声明为 props 的属性。

1. 下面选项中，说法正确的是（）。

A vm.$children 用来获取当前组件的子组件

B vm.$root 用来获取根实例对象

C vm.$slots 用来获取初始数据

D vm.$attrs 用来获取自定义属性

正确答案：A,B,D

1. 下面选项中，关于 vm.$options 说法错误的是（）。

A 自定义选项的值可以是数组 对象 函数等

B data 中的数据也可以通过 vm.options 获取到

C 通过 vm.options 也可以获取到自定义选项

D vm.options 中的"options 中的"options 中的""符号可以使用"@"替换

正确答案：B,C,D

答案解析：Vue 实例初始化时，除了传入指定的选项外，还可以传入自定义选项。

自定义选项的值可以是数组、对象、函数等，通过 vm.$options 来获取。

三 判断题，每道小题 1 分，共 12 道小题，12 分

1. 设置 Vue 的全局配置对象的 productionTip 的值为 false 会隐藏"生产信息"的提示。（）

正确答案：对

1. 通过 Vue 全局配置中的 devtools 可以对该工具启用和停止。（）

正确答案：对

1. vm.$root 获取根 Vue 实例时，如果当前实例没有父实例，则获取到的是该实例本身。（）

正确答案：对

1. 自定义选项与 data 数据一样都具有响应式属性。（）

正确答案：错

答案解析：自定义数据，与 data 不同的是，它不具有响应特性；

1. Vue 的核心具有一套响应式系统，简单来说就是通过监听器监听数据层的数据变化。（）

正确答案：对

1. vm.$attrs 可以获取的属性中也包含 class、style 以及被声明为 props 的属性。（）

正确答案：错

1. mixins 是一种分发 Vue 组件中可复用功能的方式。（）

正确答案：对

答案解析：mixins 是一种分发 Vue 组件中可复用功能的方式。

mixins 对象可以包含任何组件选项，当组件使用 mixins 时，将定义的 mixins 对象引入组件中即可使用，mixins 中的所有选项将会混入到组件自己的选项中。

1. Vue.extend 有一个 options 参数，表示包含组件选项的对象。（）

正确答案：对

1. 插件可以是一个对象或函数，如果是对象，必须提供 install()方法，用来安装插件。（）

正确答案：对

1. vm.$children 并不保证顺序，也不是响应式的。（）

正确答案：对

1. Vue 全局配置对象中，silent 可以取消 Vue 日志和警告，值类型为 boolean。（）

正确答案：对

答案解析：Vue 全局配置对象中，silent 可以取消 Vue 日志和警告，值类型为 boolean；

默认值为 false，设为 true 表示忽略警告和日志，否则不忽略。

1. 通过 vm.$el 获取到的 DOM 对象后，通过 innerHTML 属性可以修改元素内容。（）

正确答案：对

四 填空题，每道小题 1 分，共 11 道小题，11 分

1. 以下代码

执行后结果是_____。

正确答案：vm.$children

1. Vue 全局配置对象是_____。

正确答案：Vue.config

答案解析：Vue 提供了全局配置对象 Vue.config，通过配置可以实现生产信息提示、警告忽略等人性化处理。

1. 设置 Vue 全局配置对象 devtools 值为_____表示允许使用 vue-devtools 进行调试。

正确答案：true

1. 标签通过_____指令实现双向数据绑定。

正确答案：v-model

1. 以下代码

```
var mixin = {
  data () {
    return { message: 'hello' }
  }
}
var vm = new Vue({
  mixins: [mixin],
  data () {
    return { message: 'goodbye' }
  },
  created () {
    console.log(this.$data.message)
  }
});
```

执行后结果是_____。

正确答案：goodbye

1. Vue 全局配置对象中的 silent 值设置为_____表示忽略警告和日志。

正确答案：true

答案解析：Vue 全局配置对象中，silent 可以取消 Vue 日志和警告，值类型为 boolean；

默认值为 false，设为 true 表示忽略警告和日志，否则不忽略。

1. 以下代码

```
<div id="app">

  <parent name="vm.$props"/>
</div>
<script>
   Vue.component('parent',{
       template:'<div></div>',
       props:['name'],
       created(){
         console.log(this.$props.name)
     }
   })
   var vm = new Vue({
       el:'#app',
   })
</script>
```

执行后结果是_____。

正确答案：vm.$props

1. 以下代码

```
var vm = new Vue({

    el: '#app',

    data: {
```

```
    obj: {}
  }
})
Vue.set(vm.obj, 'b', '我是 Vue.set 添加的响应式属性 obj.b');
console.log(vm.$data.obj.b);执行后的结果是_____。
```

正确答案：我是 Vue.set 添加的响应式属性 obj.b

1. 以下代码

```
var vm = new Vue({
  el: '#app',
  customOption: '我是自定义数据',
  data: {
    base: '我是基础数据'
  },
  created () {
    console.log(this.$options.customOption)
  }
});
```

执行后的结果是_____。

正确答案：我是自定义数据

1. Vue 插件安装，需要调用_____方法实现。

正确答案：install()

答案解析：Vue.use 主要用于在 Vue 中安装插件，通过插件可以为 Vue 添加全局功能；

插件可以是一个对象或函数，如果是对象，必须提供 install()方法，用来安装插件；

如果是一个函数，则该函数将被当成 install()方法；

1. 以下代码

```
 <div id="app">

    <my-component id="test"></my-component>

  </div>
<script>
   Vue.component('my-component', {
       template: '<button @click="showAttrs">查看属性</button>',
       methods: {
           showAttrs () {
               console.log(this.$attrs.id)
           }
       }
   })
   var vm = new Vue({ el: '#app' });
</script>
```

单击页面"查看属性"按钮，执行后结果是_____。

正确答案：test

一 单选题，每道小题 1 分，共 9 道小题，9 分

1. 下面选项中，可以通过 transition 组件的钩子函数实现动画的是（）。

A v-enter

B v-enter-active

C enter-class

D @before-enter

正确答案：D

1. 下面选项中，可以实现不同标签名的过渡效果的是（）。

A v-show 和 v-hidden

B v-else 和 v-show

C v-show 和 v-if

D v-if 和 v-else

正确答案：D

1. 下面选项中，可以通过百分比的形式来规定动画的状态的是（）。

A v-enter-to

B CSS

C animate

D @keyframes

正确答案：D

1. 下面选项中，属于 JavaScript 动画库的是（）。

A vue.js

B Velocity.js

C animate.css

D react.js

正确答案：B

1. 下面选项中，用来实现组件过渡模式切换的是（）。

A pattern

B mode

C change

D type

正确答案：B

1. 下面选项中，能够判断字符串中是否存在某一字符的是（）。

A inof

B indexOf

C index

D ndexOf

正确答案：B

1. 下面选项中，相同标签名元素的过渡需要为元素定义的属性是（）。

A class

B key

C id

D name

正确答案：B

1. Vue 提供了内置的过渡封装组件是（）。

A transition

B Velocity

C animate

D group

正确答案：A

1. 下面选项中，可以实现列表排序平滑过渡的是（）。

A v-mode

B v-move

C v-enter

D v-leave

正确答案：B

二 多选题，每道小题 2 分，共 2 道小题，4 分

1. 下面选项中，属于组件自定义类名属性的是（）。

A enter-active-class

B enter-to-class

C leave-class

D enter-class

正确答案：A,B,C,D

三 判断题，每道小题 1 分，共 11 道小题，11 分

1. 多个组件之间的过渡，不需要使用 key 特性，只需要使用动态组件即可。（）

正确答案：对

1. 相同标签名元素只有通过 v-if 和 v-else 就可以实现元素的过渡效果。（）

正确答案：错

1. 在进行列表过渡时，过渡模式不可用，因为不再互相切换特有的元素。（）

正确答案：对

1. Velocity.js 支持链式动画，当一个元素连续应用多个 velocity()时，动画以列队的方式执行。（）

正确答案：对

1. Vue 使用了 FLIP 简单动画队列来实现排序过渡，所以即使没有插入或者移除元素，对于元素顺序的变化也支持过渡动画。（）

正确答案：对

1. 使用@keyframes 规则创建动画，就是将一套 CSS 样式逐步演变成另一套样式，在创建动画过程

正确答案：对

1. 在 Vue 中还可以实现列表的交错过渡效果，它是通过 data 属性与 JavaScript 通信来实现的。（）

正确答案：对

1. 不同标签名元素必须绑定 key 值才能通过 v-if 和 v-else 实现元素的过渡效果。（）

正确答案：错

1. 钩子函数可以结合 CSS 过渡（transitions）、动画（animations）使用，还可以单独使用。（）

正确答案：对

1. 自定义类名的优先级高于普通类名。（）

正确答案：对

1. 列表进入和离开的过渡可以通过 name 属性来自定义前缀。（）

正确答案：对

四 填空题，每道小题 1 分，共 10 道小题，10 分

1. 列表结构中的每一项，必须定义_____的值来作为唯一标识，否则将没有过渡效果。

正确答案：key

1. 组件中的 mode 模式值为_____时，表示当前元素先进行过渡，完成之后新元素过渡进入。

正确答案：out-in

1. 元素上使用 v-bind 绑定的 data-index 属性，可以用_____方式来获取 index 值。

正确答案：dataset.index

1. 以下代码

```
<div id="app">
    <transition name="fade">
      <button v-bind:key="isEditing">
        {{isEditing ? '保存' : '编辑'}}
      </button>
    </transition>
</div>
<script>
```

```
  var vm = new Vue({ el: '#app', data: { isEditing: true } })
</script>                                                    ```
```

在浏览器中执行后，页面中展示的结果是_____。

正确答案：保存

5．transition 组件中的钩子函数_____表示在动画入场过程中触发。

正确答案：@enter

6．_____是一种无状态（没有响应式数据）、无实例（没有 this 上下文）的组件。

正确答案：函数式组件

7．在列表过渡中，其<transition-group>标签上的_____属性，可以用来自定义标签名。

正确答案：tag

8．transition 组件的动画类名中表示进入过渡生效时的状态，作用于整个过程的是_____。

正确答案：v-enter-active

9．引入 Velocity.js 动画库之后，调用了 Velocity()函数，该函数的第 1 个参数是_____。

正确答案：DOM 元素

10．Vue 中通过_____指令控制元素显示隐藏。

正确答案：v-if

一 单选题，每道小题 1 分，共 4 道小题，4 分

1．下列选项中，不属于 CSS 预处理器的是（）。

A Less

B Sass/SCSS

C scoped

D Stylus

正确答案：C

2．下列选项中，关于 router.push()方法说法错误的是（）。

A 该方式的参数可以是一个字符串路径

B 这个方法不会向 history 栈添加记录

C 在参数对象中，如果提供了 path，params 会被忽略

D 该方法的参数可以是一个描述路径的对象

正确答案：B

3．一个单独的组件文件通常应包含哪几部分代码（）。

A <style>样式

B <template>模板

C　<script>逻辑

D　以上全部

正确答案：D

4. 在 Vue 项目中，可以使用以下（）命令初始化项目。

A　npm run -y

B　npm install -y

C　npm init run -y

D　npm init -y

正确答案：D

二 多选题，每道小题 2 分，共 5 道小题，10 分

1. 下列选项中，关于单独的组件文件说法正确的是（）。

A　<style>可以省略

B　<script>不能省略

C　<template>不能省略

D　<template>可以省略

正确答案：A,C

2. 下列选项中，关于 CSS 预处理器说法正确的是（）。

A　无需考虑浏览器的兼容性问题

B　可以使 CSS 更加简洁

C　适用性和可读性最佳

D　更易于代码的维护

正确答案：A,B,C,D

3. 下列选项中，关于 router.go()方法说法正确的是（）。

A　表示在 history 历史记录中向前或者后退多少步

B　$router.go()相当于 window.history.go()

C　$router.go(-1)表示后退一步

D　$router.go(1)表示前进一步

正确答案：A,B,C,D

4. 下列选项中，关于子路由 children 属性说法错误的是（）。

A　使用 children 属性实现子路由时，子路由的 path 属性前不要带"/"

B　children 也是一组路由

C　子组件没有路由导航和路由容器

D　每一个子路由里面只能嵌套一个组件

正确答案：C,D

5. 下列选项中，属于路由对象$route 的常用属性的是（）。

A　paths

B　hash

C　params

D　name

正确答案：B,C,D

三 判断题，每道小题 1 分，共 12 道小题，12 分

1. <router-link>标签主要实现跳转链接功能，属性 to='/'即是跳转到 path 为'/'的路径。（）

正确答案：对

2. webpack 打包工具，可以实现自动打包编译功能。（）

正确答案：对

3. params 方式传参，通常会搭配路由的 history 模式，将参数放在路径中或隐藏。（）

正确答案：对

4. 存放在 query 对象中的参数值，在模板中可以使用插值表达式{{this.$route.querys.x}}输出 x 的值。（）

正确答案：错

5. 如果<router-view>没有设置名字，那么默认渲染 default 对应的组件。（）

正确答案：对

6. router.beforeEach()是全局钩子函数。（）

正确答案：对

7. 单击"<route-link :to="...">"等同于调用 router.push(...)方法。（）

正确答案：对

8. 路由配置对象必须包含 path 和 components 属性。（）

正确答案：错

9. 当使用对象作为路由的时候，to 前面要加一个冒号，表示绑定。（）

正确答案：对

10. 后端路由通过用户请求的 URL 分发到具体的处理程序，浏览器每次跳转到不同的 URL，都会重新访问服务器。（）

正确答案：对

11. 命名路由方式，无论 path 多长、多繁琐，都能直接通过 name 来引用，十分方便。（）

正确答案：对

12. 通过 query 方式传递参数，类似于 GET 请求，在页面跳转的时候，可以在地址栏看到请求参数。（）

正确答案：对

四 填空题，每道小题 1 分，共 13 道小题，13 分

1. _____标签给子模板提供插入位置，充当占位符。

正确答案：<router-view>

2. 在<style>标签上设置_____属性，表示 CSS 样式只能作用于当前的组件。

正确答案：scoped

3. 在_____组件上定义 name 属性表示视图的名字，然后就可以根据不同的 name 值展示不同的页面。

正确答案：<router-view>

4. 在路由对象的 routes 配置路由匹配规则时，path 属性中需要使用_____的形式匹配参数。

正确答案：:id

5. _____属性，表示当前路由的名称。

正确答案：$route.name

6. 嵌套子路由的关键属性是_____，该属性也是一组路由，可以配置路由数组。

正确答案：children

7. _____是由 DCloud 推出的一款接近原生 APP 体验的高性能前端框架。

正确答案：MUI

8. 简而言之，_____就是在路由里面嵌套它的子路由。

正确答案：嵌套路由

9. 程序开发中的路由分为后端路由和_____。

正确答案：前端路由

10. 执行_____方法后，导航后不会向 history 栈添加新的记录，而是替换当前的 history 记录。

正确答案：router.replace()

11. 通过_____命令安装路由。

正确答案：npm install vue-router

12. _____表示当前激活的路由的状态信息，包含了当前 URL 解析得到的信息，还有 URL 匹配到的路由记录。

正确答案：路由对象

13. 使用_____方式传参，参数会以查询字符串的形式显示在浏览器地址栏中

正确答案：query

1 不属于 vue-router 的导航钩子的是：页面钩子。 属于的是：全局导航钩子、组件内的钩子、单独路由独享组件

2 主要实现数据双向绑定，通常用在表单元素的命令是：v-model

3 能够基于 Vue 构造器创建一个 Vue 子类的是：Vue-extend

4 用来访问子组件实例的是：vm.$children

5 可以通过百分比的形式来规定动画状态的是：@keyframes

6 属于 JavaScript 动画库的是：Velocity.js

7 可以实现不同标签名的过渡效果的是：v-if 和 v-else

8 关于 router.push()方法说法错误的是：这个方法不会向 history 栈添加记录

9 可以实现 actions 中事件处理函数状态分发的是：dispatch

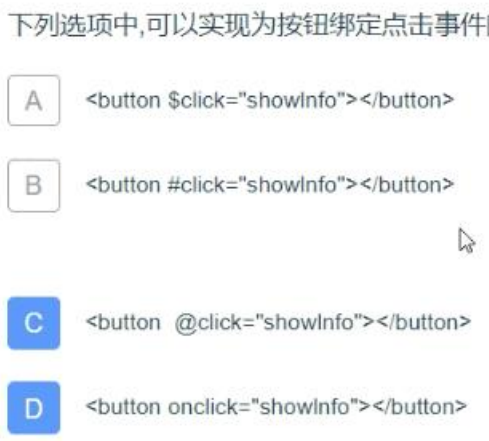
10 属于命令行工具的是：git-bash、cmd

11 用于开发和调试 Web 项目的工具是：浏览器、vue-devtools 扩展

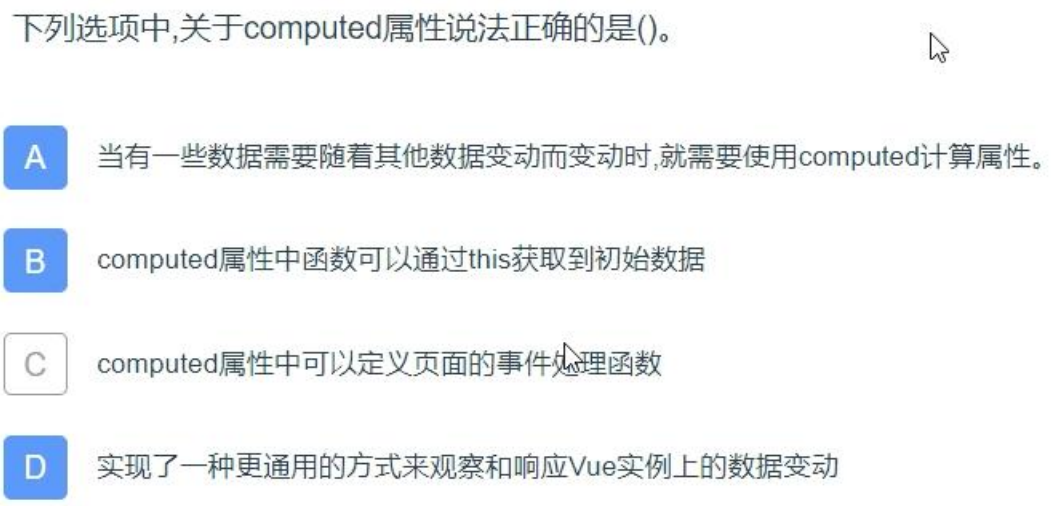
12 说法正确的：el 用来定义唯一根标签、data 中定义初始数据、methods 中可以定义事件处理函数

13 用来与 Vue 实例对象 vm 中的 el 绑定选择器的是：class、id

14 可以为按钮绑定点击事件的是：



15 关于 computed 属性说法正确的是：



16 关于 watch 选项说法正确的是：

下列选项中,关于watch选项说法正确的是()。

A watch选项中可以监听初始数据状态变化

B 当data中数据状态发生变化时,会触发watch选项中对应的事件处理函数

C watch选项可以用来替代computed选项使用

D watch选项中函数接收参数用来表示,新值和旧值。

17 不属于组件默认类名前缀的是：

下面选项中,不属于组件默认类名前缀的是()。

A t-

B float-

C fade-

D v-

18 将路由规则对象注册到 vm 实例上：

下面选项中,关于将路由规则对象注册到vm实例上的说法正确的是()。

A 就会在实例中提供this.$route属性

B 可以在任何组件内通过this.$router访问路由器

C 可以通过this.$route访问当前路由

D 会在实例中提供this.$router方法

19 属于路由对象$router 的常用属性的是：

下列选项中,属于路由对象$route的常用属性的是()。

A paths

B hash

C params

D name

20 关于 Vuex 状态管理模式正确的是：

下面选项中,关于Vuex状态管理模式说法正确的是。()

A Vuex是专门为Vue设计的状态管理库

B 在Vue中,组件的状态变化是通过Vue单向数据流的设计理念实现的

C Vue的单向数据流增强了组件之间的独立性

D Relux是专门为Vue设计的状态管理库

21 正确的是 commmit 可以 Vuex 配置

下面选项中,说法正确的是。()

A mutations选项可以用methods替代使用

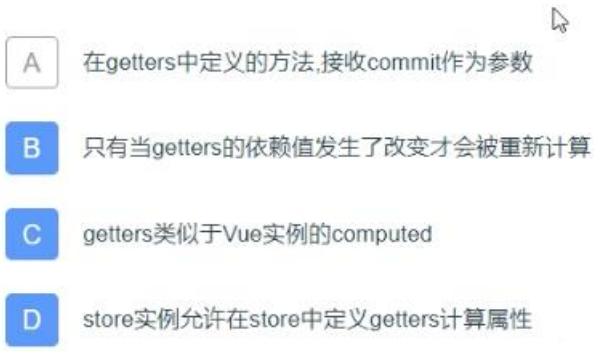
B commit可以接收{ type: 'receive', name: '我是传递的参数' }配置对象作为参数

C 在Vuex配置选项中,actions是同步执行的

D 在Vuex配置选项中,mutations是同步执行的

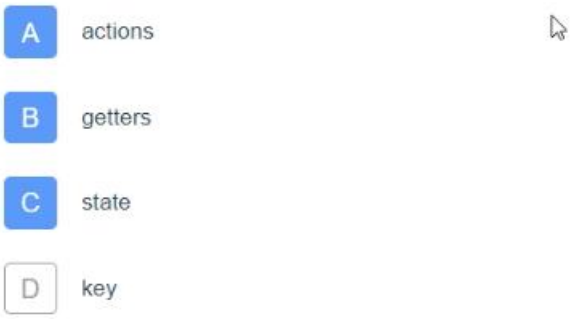
22 正确的是 getters 中 store 实例

下列选项中,说法正确的是。()



23 属于 modules 中的配置选项的是

下列选项中,属于modules中的配置选项的是()。



24router.beforeEach()钩子函数会在路由跳转之前执行 对
25src/router.js 目录用于存放路由文件 对
26Vue.component 方法用于全局注册组件 对
27 这里的/user 是在全局配置对象中配置的 path 值

<router-link to="/user"></router-link>这里的"/user"是在全局配置对象中配置的path值。(
)

28Vuex 实例对象通过 this.$store.state.* 方式来获取
29 程序开发中的路由分为后端路由和前端路由。
30 使用 query 方式传参，参数会以查询字符串的形式显示在浏览器地址栏中
31 在 view 项目中需要使用 npm run serve 指令来启动项目
32 执行后的结果是 1

```
<script>
const store = new Vuex.Store({
    state: {
      count : 0
    },
// 修改 count 的值
    mutations: {
      increase (state) {
        state.count++
      }
    }
})
store.commit('increase')
console.log(store.state.count)
</script>
```

33 执行以下代码，最后的打印结果是 1

```
new Vue({ data: { a: 1, b: 2 }, created: function () { console.log(this.a) },
mounted() { console.log(this.b) } });
```

1. Vue 是一套构建_____的渐进式框架。

```
用户界面
```

1. MVVM 主要包含 3 个部分，分别是 Model、View 和_____。

```
ViewModel
```

1. Vue 中通过_____属性获取相应的 DOM 元素。

```
refs
```

1. 在进行 Vue 调试时，通过使用_____工具来完成项目开发。

```
vue-devtools
```

1. Vue 中页面结构以_____形式存在。

```
组件
```

判断题

1. Vue 与 Angular 和 React 框架不同的是，Vue 设计为自下而上逐层应用。 （对）
2. Vue 完全能够为复杂的单页应用提供驱动。 （对）
3. Vue 是一套构建用户界面的渐进式框架，Vue 的核心只关注视图层。 （对）
4. Vue 中 MVVM 框架主要由 3 部分组成：Model，View 和 ViewModel。 （对）
5. Vue 可以在 Node 环境下进行开发，并借助 npm 包管理器来安装依赖。 （对）

选择题

1. 下面关于 Vue 说法错误的是（D）

```
A. Vue 与 Angular 都可以用来创建复杂的前端项目。

B. Vue 的又是主要包括轻量级、双向数据绑定。

C. Vue 在进行实例化之前，应确保已经引入了核心文件 vue.js。

D. Vue 与 React 语法完全相同。
```

1. 下面关于 Vue 的优势说法错误的是（C）

```
A. 双向数据绑定

B. 轻量级框架

C. 增加代码的耦合度

D. 实现组件化
```

1. 下列不属于 Vue 开发所必须的工具的是（D）

```
A. Chrome 浏览器

B. VSCode 编辑器

C. vue-devtools

D. 微信开发者工具
```

1. npm 包管理器是基于（A）平台使用的。

```
A. Node.js

B. Vue

C. Babel

D. Angular
```

1. 下列选项中，用来安装 vue 模块的正确命令是（A）。

```
A. npm install vue

B. node.js install vue

C. node install vue

D. npm I vue
```

第二章 Vue 开发基础（上）

填空题

1. Vue 实例对象通过_____方式来创建。

```
new Vue（）
```

1. Vue 初始化数据在实例对象的_____参数中进行定义。

```
data
```

1. Vue 实例对象中的 el 参数表示_____。

```
唯一根标签
```

1. Vue 中实现双向数据绑定的指令是 _____。

```
v-model
```

1. Vue 事件绑定指令是_____。

```
v-on
```

判断题

1. 在项目中引入 vue.js 文件，才可以创建 Vue 实例。 （对）
2. Vue 实例对象指令主要包括自定义指令和内置指令，通过指令省去 DOM 操作。 （对）
3. Vue 事件传递方式有冒泡和捕获，默认是冒泡。 （对）
4. Vue 开发提出了组件化开发思想，每个组件都是一个孤立的单元。 （对）
5. 在 Vue 中 beforeDestroy 与 destroyed 钩子函数执行后，都可以获取到 Vue 实例。 （对）

选择题

1. 下列关于 Vue 实例对象说法不正确的是（D）

```
A. Vue 实例对象通过 new Vue({})方式创建的

B. Vue 实例对象只允许有唯一的一个根标签

C. 通过 methods 参数可以定义事件处理函数

D. Vue 实例对象中 data 数据不具有响应特性
```

1. Vue 实例对象中能够监听状态变化的参数是（A）

```
A. watch

B. filters

C. watching

D. components
```

1. Vue 中实现数据双向绑定的是（C）

```
A. v-bind

B. v-for

C. v-model

D. v-if
```

1. 在 Vue 中，能够实现页面单击事件的的代码是（B）

```
A. v-on:enter

B. v-on:click
```

```
C. v-on:mourseenter

D. v-on:doubleclick
```

1. 下面列出的钩子函数在 Vue 实例销毁完成时执行的是（B）

```
A. updated

B. destroyed

C. created

D. mounted
```

第三章 Vue 开发基础（下）

填空题

1. Vue 实例对象通过_____方式来获取。

```
vm.$root
```

1. Vue 初始数据通过_____f 方式获取。

```
vm.$data
```

1. Vue 中通过_____获取当前实例的子组件。

```
vm.$children
```

1. Vue 中创建插件提供的方法是_____。

```
install
```

1. Vue 中通过_____创建自定义指令。

```
Vue.deirective()
```

判断题

1. Vue 提供全局的 API 接口 component() ，不能用来注册组件（错）
2. Vue 中 Vue.config 对象用来实现 Vue 全局配置（对）
3. Vue 中 data 选项中的数据具有响应特性（对）
4. Vue 中通过 vm.$slots 可以获取子组件实例对象（错）
5. Vue 实例对象中通过$options 可以获取到父作用域下的所有属性（错）

选择题

1. 下列关于 Vue 实例对象接口的说法，错误的是（D）

A. Vue 实例对象提供了实例可操作方法

B. Vue 实例对象$data 数据可以由实例 vm 委托代理

C. 通过 Vue 实例对象可以进行 Vue 全局配置

D. Vue 实例对象接口同样可以通过 Vue 方式调用

1. 下面关于 Vue 全局配置的说法，错误的是（CD）

A. Vue.config.devtools 可以设置 devtools 调试工具的启动和关闭

B. Vue.config 是一个对象，包含 Vue 的全局配置

C. Vue.component()可以获取 Vue 全局配置对象

D. Vue.set.config 可以获取到全局配置对象

1. 下列不属于 Vue 实例对象属性的是（B）

A. $data

B. $component

C. $props

D. $root

1. Vue 实例对象获取子组件实例对象的方式（B）

A. $parent

B. $children

C. $child

D. $component

1. 下面关于 Vue.mixin 的说法，错误的是（D）

A. Vue.mixin 是 Vue 提供的全局接口 API

B. Vue.mixin 可以用来注入组件的选项

C. 使用 Vue.mixin 可能会影响所有的 Vue 实例

D. Vue.mixin 不可以用来注入自定义选项的处理逻辑

第四章 Vue 过渡和动画

填空题

1. Vue 提供的内置过渡封装组件是_____。

```
transition
```

1. 在过渡封装组件中使用_____属性可以重置过渡中切换类名的前缀。

```
name
```

1. 通过_____特性设置节点在初始渲染的过渡。

```
appear
```

1. 在离开的过渡中有_____、_____、_____3 个 class 切换。

```
v-leave  v-leave-active    v-leave-to
```

1. _____的类名优先级要高于普通的类名。

```
自定义过渡
```

判断题

1. 函数式组件中的 render()函数用来创建组件的模板（对）
2. 给过渡元素添加 v-bind:class="true"，Vue 会跳过 CSS 的检测（错）
3. 在@before-enter 阶段可以设置元素开始动画之前的起始样式（对）
4. 在使用 animate.css 库时，基本的 class 样式名时 animate （错）
5. enter 和 leave 动画钩子函数，除 el 参数外海湖传入一个 done 作为参数（对）

选择题

1. 下列选项中关于钩子函数说法，正确的时（C）

```
A. @leave-cancelled 函数只能用与 v-if

B. 对于@enter 来说，当与 CSS 结合使用时，回调函数 done 时必选的

C. done 作为参数，作用就是告知 Vue 动画结束

D. 钩子函数需要结合 CSS transitions 或 animations 使用，不能单独使用
```

1. 下列关于 Vue 为标签提供的过度类名的说法，错误的是（D）

```
A. v-enter 在元素被插入之前生效，在元素被插入之后的下一帧移除

B. v-leave 在离开过度被触发时立刻生效，下一帧被移除
```

C. v-enter-active 可以控制进入过渡的不同的缓和曲线

D. 如果 name 属性为 my-name，那么 my-就是过渡中切换的类名前缀

1. 下列选项中关于多个元素过渡的说法，错误的时（D）

A. 当有相同标签名的元素切换时，需要通过 key 特性设置唯一的值来标记以让 Vue 区分它们

B. 不相同元素之间可以通过 v-if 和 v-else 来进行过渡

C. <transition>组件的默认行为指定进入和离开同时发生

D. 不可以给同一个元素的 key 特性设置不同的状态来代替 v-if 和 v-else

第五章 Vue 路由

填空题

1. 在项目中可以通过 npm 命令_____安装路由 vue-router。

```
npm install vue-router
```

1. 使用_____获取当前激活的路由的状态信息。

```
路由对象
```

1. 通过一个名称来标识一个路由的方式叫做_____ 。

```
命名路由
```

1. 在业务逻辑代码中实现导航跳转的方式成为_____。

```
编程式导航
```

1. 单页面应用主要通过 URL 中的_____实现不同页面之间的切换。

```
hash(#号)
```

判断题

1. 后端路由通过用户请求的 URL 导航到具体的 html 页面。（对）
2. 开发环境下使用 import VueRouter from ‘vueRouter’ 来导入路由。（错）
3. 嵌套路由的使用，主要是由页面结构来决定的。（对）
4. params 方式传参类似于 GET 请求。（错）
5. 在单页面应用中更新视图可以不同重新请求页面。（对）

选择题

1. 下列 vue-router 插件的安装命令，正确的是（A）

```
A. npm install vue-router

B. node install vue-router

C. npm install vueRouter

D. npm I vue-router
```

1. 下列关于 query 方式传参的说法，正确的是（C）

```
A. query 方式传递的参数会在地址栏展示

B. 在页面跳转的时候，不能在地址栏看到请求参数

C. 在目标页面中使用“this.route.query.参数名”来获取参数

D. 在目标页面中使用“this.$route.params.参数名”来获取参数
```

1. 下列关于 params 方式传参的说法，错误的是（C）

```
A. 在目标页面中也可以使用“$route.params.参数名”来获取参数

B. 在页面跳转的时候，不能在地址栏看到请求参数

C. 以 params 方式传递的参数会在地址栏展示

D. 在目标页面中使用“this.$route.params.参数名”来获取参数
```

第六章 Vuex 状态管理

填空题

1. Vuex 实例对象通过_____方式获取。

```
vm.$store
```

1. Vuex 实例对象中初始数据状态通过_____方式获取。

```
vm.$store.state
```

1. Vuex 实例对象中组件状态通过_____方式改变。

```
3.vm.$store.commit()
```

1. Vuex 中创建动态模块提供的方法是_____。

```
store.registerModule()
```

1. Vuex 中通过_____实现 actions 状态分发。

```
vm.$store.dispatch()
```

判断题

1. Vuex 实例对象可以调用 Vue 全局接口。（错）
2. Vuex 中的 Vue.config 对象用来实现 Vuex 全局配置。（错）
3. Vuex 的 state 选项中数据是初始数据状态。（对）
4. Vuex 中插槽可以实现组件任意嵌套，且在版本 2.2.6+以后开始支持。（错）
5. 当在 Vuex 实例对象中调用 store 时，一定会获取到 store 实例对象。（对）

选择题

1. 下列关于 Vuex 实例对象接口的说法，错误的是（B）

```
A. Vuex 实例对象共提供了 store 实例对象可操作方法

B. Vuex 实例对象$data 数据可以由实例委托代理

C. 通过 Vuex 实例对象可实现组件状态的管理委托

D. Vuex 实例对象初始数据是 state 数据
```

1. 下面关于 Vuex 核心模块的说法，错误的是（B）

```
A. Vuex 配置对象中，actions 选项是异步的

B. Vuex.config 对象是全局配置对象

C. Vuex 配置对象中，mutations 选项是同步的

D. 通过 commit 完成 mutations 提交
```

1. 下列不属于 Vuex.Store 配置对象接受的是（A）

```
A. data

B. state

C. mutations

D. getters
```

1. Vuex 实例对象中类似于 computed 计算属性功能的选项是（D）

```
A. state

B. mutations
```

```
C. actions

D. getters
```

1. 下面关于 Vuex 中的 actions 的说法，不正确的是（ABD）

```
A. actions 中事件函数通过 commit 完成分支

B. actions 中事件处理函数接受 context 对象

C. actions 与 Vue 实例中的 methods 是类似的

D. 可以用来注入自定义选项的处理逻辑
```

第七章 Vue 开发环境

填空题

1. 对于 CLI 类型的插件，需要以_____为前缀。

```
@vue
```

1. 使用 npm 包通过_____命令全局安装@vue/cli 3.X。

```
npm install -g @vue/cli
```

1. 使用_____来插卡 Vue 的版本号。

```
vue -version(vue -V)
```

1. 使用 yarn 包通过_____命令全局安装@vue/cli 3.X。

```
yarn global add @vue/cli
```

1. 在 Vue CLI 3 中使用_____命令来创建一个 Vue 项目。

```
vue create 项目名
```

判断题

1. 卸载 vue-cli 的命令时 npm uninstall vue-vli -g。（对）
2. 添加 CLI 插件的指令时 vue add vue-eslint。 （错）
3. 插件不能修改 webpack 的内部配置，但是可以向 vue-cli-service 注入命令。（错）
4. Vue CLI 通过 vue ui 命令来创建图形用户界面。（对）
5. 在文件中使用“key=value”(键值对)的方式来设置环境变量。（对）

选择题

1. 下列选项中说法正确的是（B）

```
A. 新版的 Vue VLI 的包名称为 vue-cli

B. 执行 npm uninstall vue-cli -g 命令可以全局删除 vue-cli 包

C. 使用给 yarn install add @vue/cli 命令可以全局安装@vue/cli 工具

D. 通过 vue add ui 命令来创建图形用户界面
```

1. 关于 CLI 服务，下列选项中说法错误的是（C）

```
A. 在@vue/cli-service 中安装了一个名为 vue-cli-service 的命令

B. vue.config.js 是一个可选的配置文件

C. 通过 npx vue-cli-service helps 查看所有的可用命令

D. 通过 vue ui 使用 GUI 图形用户界面来运行更多的特性脚本
```

1. 下列选项中说法正确的是（AD）

```
A. 使用相对路径引入的静态资源文件，会被 webpoack 处理解析为模块依赖

B. 放在 public 文件夹下的资源将会经过 webpack 的处理

C. 通过绝对路径被引用的资源将会经过 webpack 的处理

D. URL 以~开始，会被认为是模块请求
```

第八章 服务器渲染

填空题

1. _____插件可以用来进行页面的热重载。

```
webpack-hot-middleware
```

1. hash 模式路由，地址栏 URL 中会自带_____符号。

```
#
```

1. SSR 的路由需要采用_____ 的方式。

```
history
```

1. _____是利用搜索引擎规则，提高网站在搜索引擎内自然排名的一种技术。

```
SEO（搜索引擎优化）
```

1. Vue 中使用服务器渲染，需要借助 Vue 的扩展工具_____ 。

```
vue-server-renderer
```

判断题

1. 客户端渲染，即传统的单页面应用模式。（对）
2. webpack-dev-middleware 中间件会对更改的文件进行监控。（对）
3. 服务器渲染不利于 SEO。（错）
4. 服务器渲染应用程序，需要处于 Node.js server 运行环境。（对）
5. 使用 git-bash 命令工具，输入指令 mkdirs vue-ssr 来创建项目。（错）

选择题

1. 下列选项说法正确的是（A）

```
A. vue-server-renderer 的版本要与 vue 版本相匹配

B. 客户端渲染，需要使用 entry-server.js 和 entry.client.js 两个入口文件

C. app.js 是应用程序的入口，它对应 vue-cli 创建的项目的 app.js 文件

D. 客户端应用程序既可以运行在浏览器上，又可以运行在服务器上
```

1. 下列关于 SSR 路由的说法，错误的是（B）

```
A. SSR 的路由需要采用 history 的方式

B. history 模式的路由提交不到服务器上

C. history 模式完成 URL 跳转而无重新加载页面

D. hash 模式路由，地址栏 URL 中 hash 改变不会重新加载页面
```

1. 下列关于 Nuxt.js 的说法，错误的是（D）

```
A. 使用“creat-nuxt-app 项目名”命令创建项目

B. 使用 Nuxt.js 搭建的项目中，pages 目录是用来存放应用的路由及视图

C. 在 Nuxt.js 项目中，声明式路由在 html 标签中通过<nuxt-link>完成路由跳转

D. Nuxt.js 项目中需要根据目录结构手动完成对应的路由配置
```

第九章 “微商城”项目

选择题

1. 使用 Mini UI 库的页面，需要通过_____前缀来定义标签名。

```
mt-
```

1. _____是一个基于 Promise 的 HTTP 库，可以用在浏览器和 node.js 中。

```
Axios
```

1. _____是最接近原生 App 体验的高性能前端框架。

```
MUI
```

1. 使用_____，可以给 Vue 函数添加一个原型属性$http,指向 axios。

```
Object.defineProperty
```

1. 使用路由的声明式导航，在 html 标签中使用_____组件来实现路由的跳转。

```
<router-link>
```

判断题

1. MUI 是一套代码片段，提供了配套的样式和 HTML 代码段。（错）
2. 使用 lazy-load 可以实现图片懒加载。（对）
3. 通过 this.$store.state.*可以访问 state 中的数据。（对）
4. 组件想要修改数据，需要调用 mutations 提供的方法，通过语句 this.$store.emit('方法名')实现。（错）
5. better-scroll 是一款支持各种滚动场景需求的插件。（对）

选择题

1. 下列选项中，（A）指令可用来切换元素的可见状态

```
A. v-sjow

B. v-hide

C. v-toggle

D. v-slideHide
```

1. 下列关于 ref 作用的说法，错误的是（D）

```
A. ref 在子组件中使用时，使用 this.$refs.name 获取到组件实例，可以使用组件的所有属性和方法

B. ref 加在普通的元素上，用 this.ref.name 获取到的是 DOM 元素

C. 可以利用 v-for 和 ref 获取一组数组或 DOM 节点

D. ref 在 DOM 渲染完成之前就能使用
```

1. 想要获取购物车小球在页面上的位置，以下可以使用的是（B）

```
A. offset（）

B. getBoundingClientRect()

C. width()

D. height()
```

Vue.js 前端开发实战

简答题

目录

课本

第一章

1. 请简述什么是 Vue。

Vue（读音/Vju:/，类似于 View）是一套用于构建用户界面的渐进式框架，与其他大型框架相比，Vue 被设计为可以自底向上逐层应用。其他大型框架往往一开始就对项目的技术方案进行强制性的要求，而 Vue 更加灵活，开发者既可以选择使用 Vue 来开发一个全新项目，也可以将 Vue 引入到一个现有的项目中。

2. 请简述 Vue 优势有那些。

轻量级：Vue 相对简单、直接，所以 Vue 使用起来更加友好。

数据绑定：Vue 是一个 MVVM 框架，即数据双向绑定。

指令：指令主要包括内置指令和自定义指令，以“v-”开头，作用于 HTML 元素。

插件： 插件用于对 Vue 框架功能进行扩展，通过 MyPlugin.install 完成插件的编写，简单配置后就可以全局使用。

第二章

3. 请简述什么是 Vue 实例对象。

在 Vue 项目中，每个 Vue 应用都是通过 Vue 构造器创建新的 Vue 实例开始的。

通过 new 关键字的方式创建 vm 实例对象。

创建方式：

```
<script>
var vm = new Vue({
  // 选项
})
<script>
```

其中，配置选项的主要内容及含义：

1）.data：Vue 实例数据对象

2）.methods：定义 Vue 实例中方法

3）.components：定义子组件

4）.computed：计算属性

5）.filters：过滤器

6）.el：唯一根元素

7）.watch：监听数据变化

4. 请简述什么是 Vue 组件化开发。

1）.在 Vue 中，组件是构成页面中独立结构单元，能够减少重复代码的编写

2）.提高开发效率，降低代码之间的耦合程度，使项目更易维护和管理

3）.组件主要以页面结构形式存在，不同组件也具有基本交互功能，根据业务逻辑实现复杂的项目功能

5. 请简单介绍 Vue 内置指令主要内容有哪些。

1）.v-model：双向数据绑定

2）.v-on：监听事件

3）.v-bind：单向数据绑定

4）.v-text：插入文本内容

5）.v-html：插入包含 HTML 的内容

6）.v-for：列表渲染

7）.v-if：条件渲染

8）.v-show：显示隐藏

第三章

1. 请简述什么是 Vue 插件。

Vue.use 主要用于在 Vue 中安装插件，通过插件可以为 Vue 添加全局功能。插件可以是一个对象或函数，如果是对象，必须提供 install()方法，用来安装插件；如果插件是一个函数，则该函数将被当成 install()方法。

2. 请简述 Vue 全局 API 接口主要内容。

1）.Vue.directive():Vue 中有很多内置指令，如 v-model、v-for 和 v-bind 等

2）.Vue.use():Vue.use 主要用于在 Vue 中安装插件，通过插件可以为 Vue 添加全局功能

3）.Vue.extend():Vue.extend 用于基于 Vue 构造器创建一个 Vue 子类，可以对 Vue 构造器进行扩展

4）.Vue.set():Vue 的核心具有一套响应式系统，简单来说就是通过监听器监听数据层的数据变化，当数据改变后，通知视图也自动更新

5）.Vue.mixin():Vue.mixin 用于全局注册一个混入，它将影响之后创建的每个 Vue 实例

3. 请简单介绍 Vue 实例对象属性和方法。

1）.vm.$props：使用 vm.$props 属性可以接收上级组件向下传递的数据

2）.vm.$options：Vue 实例初始化时，除了传入指定的选项外，还可以传入自定义选项

3）.vm.$el：vm.$el 用来访问 vm 实例使用的根 DOM 元素

4）.vm.$children：vm.$children 用来获取当前实例的直接子组件

5）.vm.$root：vm.$root 用来获取当前组件树的根 Vue 实例，如果当前实例没有父实例，则获取到的是该实例本身

6）.vm.$slots:插槽就是定义在组件内部的 template 模板，可以通过$slots 动态获取

7）.vm.$attrs：vm.$attrs 可以获取组件的属性，但其获取的属性中不包含 class、style 以及被声明为 props 的属性

第四章

4. 请简述 JavaScript 钩子函数包括哪些。

入场钩子分别是 beforeEnter（入场前）、enter（入场）、afterEnter（入场后）和 enterCancelled（取消入场）

出场钩子分别是 beforeLeave（出场前）、leave（出场）、afterLeave（出场后）和 leaveCancelled（取消出场）

```
<transition
  @before-enter="beforeEnter"
  @enter="enter"
  @after-enter="afterEnter"
  @enter-cancelled="enterCancelled"
  @before-leave="beforeLeave"
  @leave="leave"
  @after-leave="afterLeave"
  @leave-cancelled="leaveCancelled"
  v-bind:css="false">  // Vue 会跳过 CSS 的检测
</transition>
```

5. 请简述 6 个内置的过渡类名有哪些。

进入（enter）：

v-enter：在元素被插入之前生效，在元素被插入之后的下一帧移除

v-enter-active： 在整个进入过渡的阶段中应用，在元素被插入之前生效，在过渡动画完成之后移除

v-enter-to： 在元素被插入之后下一帧生效（与此同时 v-enter 被移除），在过渡动画完成之后移除

离开（leave）：

v-leave：在离开过渡被触发时立刻生效，下一帧被移除

v-leave-active：在整个离开过渡的阶段中应用，在离开过渡被触发时立刻生效，在过渡完成之后移除

v-leave-to：在离开过渡被触发之后下一帧生效（与此同时 v-leave 被移除），在过渡动画完成之后移除

6. 请简述自定义过渡类名的属性有哪些。

```
enter-class
enter-active-class
enter-to-class
leave-class
leave-active-class
leave-to-class
```

注意：自定义类名的优先级高于普通类名

第五章

7. 请简述 npm 方式安装 vue-router 的步骤。

```
1.首先通过 cd 命令进入创建好的项目目录里面
cd 文件名
2.使用以下 npm 命令来安装路由
方式一：npm install vue-router --save（不加版本号）
// --save 会在 package.json 包配置文件中添加对应的配置
```

方式二：npm install vue-router@3.1.x（指定版本号）

安装完成之后可以在 package.json 文件中查看到 vue-router 的相关信息

3.在 main.js 文件中引入路由、安装路由功能等，示例代码如下

```
import Vue from 'vue'
import VueRouter from 'vue-router' // 引入插件
import App from './App'
Vue.use(VueRouter)                    // 注册组件
const router = new VueRouter({        // 创建实例
    routes:[]                         // 配置路由规则
})
const app = new Vue({
    el: '#app',
    router:router,                    // 挂载路由
    render:h=>h(App)
})
```

8. 请简述 vue-router 路由的作用。

根据不同的 url 哈希值，在路由视图中显示不同的页面，实现非跳转式的页面切换

在单页面应用中更新视图可以不用重新请求页面

用户体验好，不需要每次都从服务器全部获取，快速展现给用户

9. 请简单列举并说明路由对象包括哪些属性。

路由对象表示当前激活的路由的状态信息，包含了当前 URL 解析得到的信息，还有 URL 匹配到的路由记录，this.$router 表示全局路由器对象，this.$route 表示当前正在用于跳转的路由器对象，$route 的常用属性信息如下：

$route.path：对应当前路由地址

$route.query：一个{key:value}对象，表示 URL 查询参数

$route.params：一个{key:value}对象，路由转跳携带参数

$route.hash：在 history 模式下获取当前路由的 hash 值（带#），如果没有 hash 值，则为空字符串

$route.fullPath：完成解析后的 URL，包含查询参数和 hash 的完整路径

$route.name：当前路由的名称

$route.matched：路由记录，当前路由下路由声明的所有信息，从父路由（如果有）到当前路由为止

$route.redirectedFrom：如果存在重定向，即为重定向来源的路由

第六章

10. 请简要概述 Vuex 的设计思想。

Vuex 是 Vue 团队提供的一套组件状态管理维护的解决方案。Vuex 作为 Vue 插件来使用，进一步完善了 Vue 基础代码功能，使 Vue 组件状态更加容易维护，为大型项目开发提供了强大的技术支持。

11. 简述 Vuex 配置对象中的主要内容有哪些。

1）.actions：用来定义事件处理方法，用于处理 state 数据

2）.mutations：选项中的事件处理方法接收 state 对象作为参数，即初始数据

3）.getters:store 实例允许在 store 中定义 getters 计算属性,类似于 Vue 实例的 computed

4）.modules：modules 用来在 store 实例中定义模块对象

5）.plugins：Vuex 中的插件配置选项为 plugins，插件本身为函数

6）.devtools：store 实例配置中的 devtools 选项用来设置是否在 devtools 调试工具中启用 Vuex，默认值为 true，表示在启用，设为 false 表示停止使用

12. 简述 Vuex 中的 actions 的含义。

actions 选项用来定义事件处理方法，用于处理 state 数据。actions 类似于 mutations，不同之处在于 actions 是异步执行的，事件处理函数可以接收{commit}对象，完成 mutation 提交，从而方便 devtools 调试工具跟踪状态的 state 变化。

在使用时，需要在 store 仓库中注册 actions 选项，在里面定义事件处理方法。事件处理方法接收 context 作为第 1 个参数，payload 作为第 2 个参数（根据需要进行选择）。

第八章

13. 请简述什么是服务器端渲染。

服务器端渲染（简称 SSR），是将组件或页面通过服务器生成 html 字符串，再发送到浏览器，最后将静态标记"混合"为客户端上完全交互的应用程序，简单理解就是将页面在服务器中完成渲染，然后在客户端直接展示。

14. 请简述服务器端渲染的代码逻辑和处理步骤。

Vue 进行服务器端渲染时，需要利用 Node.js 搭建一个服务器，并添加服务器端渲染的代码逻辑。

使用 webpack-dev-middleware 中间件对更改的文件进行监控，使用 webpack-hot-middleware 中间件进行页面的热更新，使用 vue-server-renderer 插件来渲染服务器端打包的 bundle 文件到客户端。

15. 请简述 Nuxt.js 中，声明式路由和编程式路由的区别

声明式路由：在页面中使用<nuxt-link>完成路由跳转。

编程式路由：在 JavaScript 代码中实现路由的跳转。

传智播客

第一章

1、请介绍什么是 MVVM

MVVM 主要包含 3 个部分，分别是 Model、View 和 ViewModel。
Model 指的是数据部分，主要负责业务数据；
View 指的是视图部分，即 DOM 元素，负责视图的处理；
ViewModel 是连接视图与数据的数据模型，负责监听 Model 或者 View 的修改；

2、请简单描述 Vue 的优势有哪些

①轻量级：Vue 简单、直接，简单易学
②数据绑定：数据驱动视图，视图驱动数据
③指令：指令绑定在元素上，给绑定元素添加行为
④插件：用于对 Vue 框架功能进行扩展

指令：指令主要包括内置指令和自定义指令，以“v-”开头，作用于 HTML 元素；
轻量级：AngularJS 的学习成本高，使用起来比较复杂，而 Vue 相对简单、直接，所以 Vue 使用起来更加友好；
数据绑定：Vue 是一个 MVVM 框架，即数据双向绑定，即当数据发生变化的时候，视图也就发生变化，当视图发生变化的时候，数据也会跟着同步变化，这也算是 Vue 的精髓之处，尤其是在进行表单处理时，Vue 的双向数据绑定非常方便；
插件：插件用于对 Vue 框架功能进行扩展，通过 MyPlugin.install 完成插件的编写，简单配置后就可以全局使用。常用的扩展插件有 vue-router、Vuex 等；

3、请简单描述什么是 npm

npm 提供了快速操作包的命令，只需要简单命令就可以很方便地对第三方包进行管理。

4、请介绍什么是 webpack

webpack 是一个模块打包工具，可以把前端项目中的 js、cs、scss/less、图片等文件都打包在一起，实现自动化构建。

5、Web 前端开发使用的三大语言是什么

1.HTML 页面结构;
2.CSS 页面样式;
3.JavaScript 行为;

6、请简单描述 vue.js 下载和引入的基本步骤

①下载 vue.js：
通过 vue.js 官网网址来下载 vue.js。进入网页（https://vuejs.org/v2/guide/installation.html）后，点击 Development Version 进行下载。
②引入 vue.js：

```
<script src="vue.js"></script>
```

1.从 Vue 官方网站可以获取下载地址;
2.当在 HTML 网页中使用 Vue 时，使用<script>标签引入 vue.js 即可;

第二章

1、请简单描述 computed 计算属性的基本用法。

Vue 提供了一种更通用的方式来观察和响应 Vue 实例上的数据变动，当有一些数据需要随着其他数据变动而变动时，就需要使用 computed 计算属性。
在事件处理方法中，this 指向的 Vue 实例的计算属性结果会被缓存起来，只有依赖的响应式属性变化时，才会重新计算，返回最终结果。

2、请简述 Vue 钩子函数有哪些以含义

beforeCreate：创建实例对象之前执行；
created：创建实例对象之后执行；
beforeMount：页面挂载成功之前执行；
mounted：页面挂载成功之后执行；
beforeUpdate：组件更新之前执行；
updated：组件更新之后执行；
beforeDestroy：实例销毁之前执行；
destroyed：实例销毁之后执行；

3、请简述什么是组件

在 Vue 中，组件是构成页面中独立结构单元，能够减少重复代码的编写，提高开发效率，降低代码之间的耦合程度，使项目更易维护和管理。
组件主要以页面结构形式存在，不同组件也具有基本交互功能，根据业务逻辑实现复杂的项目功能。

4、请简述 Vue 的内置指令有哪些构成

v-model：双向数据绑定
v-on：监听事件
v-bind：单向数据绑定
v-text：插入文本内容
v-html：插入包含 HTML 的内容
v-for：列表渲染
v-if：条件渲染
v-show：显示隐藏

5、请简单描述 filters 过滤器的功能

在前端页面开发中，通过数据绑定可以将 data 数据绑定到页面中，页面中的数据经过逻辑层处理后展示最终的结果。
数据的变化除了在 Vue 逻辑层进行操作外，还可以通过过滤器来实现。

第三章

1、请简述 Vue.extend()的含义并创建一个 Vue 的子类

Vue.extend 用于基于 Vue 构造器创建一个 Vue 子类，可以对 Vue 构造器进行扩展。它有一个 options 参数，表示包含组件选项的对象。
示例代码：
<script>

```
var Vue2 = Vue.extend({
   data () {
     return { title: 'hello' }
   }
 })
 var vm2 = new Vue2()
console.log(vm2.title) // 输出结果：hello
</script>
```

其中，vm2 是通过实例化 Vue2 类创建的实例对象。

2、请简述 Vue 全局配置对象中的 productionTip 的含义

当在网页中加载了 vue.js（开发版本）文件时，浏览器的控制台会出现英文的提示信息，提醒用户“您正在开发模式下运行 Vue，为生产部署时，请确保打开生产模式”。

3、请简述$options 含义并设置自定义选项 myName 的值为“我是自定义选项”

Vue 实例初始化时，除了传入指定的选项外，还可以传入自定义选项。自定义选项的值可以是数组、对象、函数等，通过 vm.$options 来获取。

示例代码：

```
<script src = "vue.js"></script>
<div id="app"></div>
<script>
var vm = new Vue({
  el:'#app',
  customOption: '我是自定义选项',
})
console.log(vm.$options.customOption); //输出结果为：'我是自定义选项'
</script>
```

4、请简述 vm.$slots 的含义并实现在 my-component 组件中引入一个名为“demo”的插槽

Vue 中的组件中使用 template 模板定义 HTML 结构，为了方便使用 template 公共模板结构，Vue 提出了插槽的概念，插槽就是定义在组件内部的 template 模板，可以通过$slots 动态获取。

```
  <div id="app">
    <my-component>
      <template v-slot:demo>
        <div>
          我是 demo
        </div>
      </template>
    </my-component>
  </div>
  <template id="first">
    <div>
      <slot name="demo"></slot>
    </div>
  </template>
   <script>
```

```
    Vue.component('my-component', { template: '#first' })
    //创建 Vue 实例,得到 ViewModel
    var vm = new Vue({
      el: '#app',
      data: {},
      methods: {}
    })
     </script>
```
执行上述代码后，在页面中展示“我是 demo”。

5、请简述 vm.$attrs 的含义并实现动态的获取组件的 name 属性

vm.$attrs 可以获取组件的属性,但其获取的属性中不包含 class、style 以及被声明为 props 的属性。

示例代码:

```
<div id="app">
    <my-component name="test"></my-component>
  </div>
<script>
Vue.component('my-component', {
  template: '<div></div>',
  created () {        console.log(this.$attrs.name)
 }
  })
 var vm = new Vue({ el: '#app' })
 </script>
```
执行上述代码后，在控制台打印‘test’。

第四章

1、请使用@keyframes 实现元素显示隐藏的动画效果且动画开始是红色，50%变为绿色、100%变为橙色的效果。

```
<style>
  div.circular {
    width: 100px; height: 100px; background: red;
    border-radius: 50%; margin-top: 20px; text-align: center;
    line-height: 100px; color: #fff;
 }
 .bounce-enter-active {
   animation: Ami .5s;
 }
 .bounce-leave-active {
   animation: Ami .5s;
 }
 @keyframes Ami {
```

```
    0% {background-color: red;}
   50% {background-color: orange;}
   100% {background-color:green;}
 }
</style>                                                    <script
src="https://unpkg.com/vue/dist/vue.js"></script>
<body>
  <div id="app">
    <button @click='flag=!flag'>显示/隐藏</button>
    <transition name="bounce">
      <div class="circular" v-if="flag"></div>
    </transition>
  </div>
  <script>
    var vm = new Vue({
      el: '#app',
      data: {
        flag:"true"
      },
    });
  </script>                                    </body>
```

2、请列举 transition 组件的类名有哪些及含义。

v-enter 进入过渡的开始状态，作用于开始的一帧
v-enter-active 进入过渡生效时的状态，作用于整个过程
v-enter-to 进入过渡的结束状态，作用于结束的一帧
v-leave 离开过渡的开始状态，作用于开始的一帧
v-leave-active 离开过渡生效时的状态，作用于整个过程
v-leave-to 离开过渡的结束状态，作用于结束的一帧

3、请简单描述,<transition>组件过渡模式有哪些及含义分别是什么。

在 transition 中加入 mode 属性，它有两个值，分别是 in-out 和 out-in，out-in 表示当前元素先进行过渡，完成之后新元素过渡进入，in-out 表示新元素先进行过渡，完成之后当前元素过渡离开。

4、请简述列表交错过渡的实现过程。

在 Vue 中还可以实现列表的交错过渡效果，它是通过 data 属性与 JavaScript 通信来实现的。

1.引入 js 库

```
<script src="https://unpkg.com/vue/dist/vue.js"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/velocity/1.5.0/velocity
.min.js"></script>
```

2.编写 HTML 结构；

```
 <div id="app">
   <input placeholder="请输入要查找的内容" v-model="query">
    <transition-group      name="item"      tag="ul"      @before-
```

```
enter="beforeEnter"
    @enter="enter" @leave="leave" v-bind:css="false">
      <li v-for="(item, index) in ComputedList" :key="item.msg"
       :data-index="index">
        {{ item.msg }}
      </li>
    </transition-group>
  </div>
```

3.编写 JavaScript 代码；

```
  var vm = new Vue({
    el: '#app',
  data () {
      return {
        query: '',     // v-model 绑定的值
        items: [
         { msg: '张三' }, { msg: '李四' }, { msg: '张芳芳' },
          { msg: '王琳琳' }, { msg: '冯圆' }
        ]
      }
   },
  computed: {  // 计算属性
     ComputedList () {
       var vm = this.query    // 获取到 input 输入框中的内容
        var nameList = this.items  // 数组
        return nameList.filter(function (item) {
         return
item.msg.toLowerCase().indexOf(vm.toLowerCase()) !== -1
        })
      }
  },
   methods: {
     beforeEnter (el) {
       el.style.opacity = 0
      el.style.height = 0
    },
     enter (el, done) {
      var delay = el.dataset.index * 150
        setTimeout(function () {
         Velocity(el, {opacity: 1, height: '1.6em'}, {complete:
done})
       }, delay)
      },
     leave (el, done) {
       var delay = el.dataset.index * 150
```

```
        setTimeout(function () {
          Velocity(el, {opacity: 0, height: 0}, {complete: done})
        }, delay)
    }
  }
})
```

5、请简单描述 filters 过滤器的功能

在前端页面开发中，通过数据绑定可以将 data 数据绑定到页面中，页面中的数据经过逻辑层处理后展示最终的结果。
数据的变化除了在 Vue 逻辑层进行操作外，还可以通过过滤器来实现。

6、请简述怎么使用 Velocity.js 库结合钩子函数实现元素的显示隐藏动画效果。

```
引入 js 库
<script src="https://unpkg.com/vue/dist/vue.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/velocity/1.5.0/velocity.min.js">
</script>
编写 css 样式
<style>
      .circular{
        width: 100px;
        height: 100px;
        border-radius: 50px 50px;
        background-color: red;
      }
</style>
<body>
  <!-- 示例代码 -->
  <div id="app">
    <!-- Velocity 动画使用 -->
    <button @click="show=!show">单击按钮显示隐藏</button>
    <transition     @before-enter="beforeEnter"     @enter="enter"
@leave="leave" v-bind:css="false">
      <div class="circular" v-if="show"></div>
    </transition>
  </div>
  <script>
    //创建 Vue 实例,得到 ViewModel
    var vm = new Vue({
      el: '#app',
      data: {
        show: "true"
      },
      methods: {
        beforeEnter(el) {
```

```
            el.style.opacity = 0     // 透明度为0
            el.style.color = 'red'   // 颜色为红色
          },
          enter(el, done) {          // duration动画执行时间
            Velocity(el, {opacity: 1}, {duration: 300})
          },
          leave(el, done) {
            Velocity(el, {opacity: 0}, {duration: 300})
          }
        }
      });
</script>
</body>
```

7、请简述什么是过渡和动画。

过渡就是从一个状态向另外一个状态插入值，新的状态替换了旧的状态；动画相比过渡来说，可以在一个声明中设置多个状态，当然动画也可以实现过渡的效果，只需要从开始到结束插入状态即可。

第五章

1、简述如何在项目中安装Less预处理器。

第一步： npm install less less-loader -D
第二步：在webpack.config.js文件中添加rules规则 { test: /\.less$/, use: ['style-loader', 'css-loader', 'less-loader'] },
第三步：在页面中使用 less 的地方给<style>添加 lang 属性即可 <style lang="less"></style>

2、简述如何在项目中使用history模式

第一步：修改router.js文件，添加history模式

```
var router = new VueRouter({
  mode: 'history',
}
```

第二步：在module.exports对象中添加devServer的配置

```
devServer: {
historyApiFallback: true // 开启服务器对history模式支持
},
```

3、讲述<template id="contact-tmp"></template>的含义

定义id为contact-tmp的模板组件，渲染contact组件模板的内容

4、简述什么是命名视图

在开发中，有时候想同时或同级展示多个视图，而不是嵌套展示，则可以在页面中定义多个单独命名的视图。例如，创建一个布局，有 header（头部区域）、sidebar（侧导航区域）和 mainBox（主体区域）3个视图，这时候就可以使用命名视图来实现。

5、简述声明式导航和编程式导航的区别

通过<router-link>来完成页面的切换，这种方式属于声明式导航。为了更方便地在项目中开发导航功能，Vue 提供了编程式导航，也就是利用 JavaScript 代码来实现地址的跳转，通过 router 实例方法来实现。

6、简述如何添加自定义 class，实现路由导航的高亮效果。

第一步：在 router.js 路由文件中，找到创建路由实例代码，添加自定义 class

```
var router = new VueRouter({
  linkActiveClass: 'my-active',   linkExactActiveClass: 'my-exact-active',
})
```

第二步：在 App.vue 中添加高亮效果的样式

```
<style lang="scss" scoped>
  .my-active, .my-exact-active {
    background: #007aff;
    font-weight: 800;
    color: #fff;
  }
</script>
```

7、简述如何在项目中安装 Sass/SCSS 预处理器

第一步： npm install sass-loader node-sass -D

第二步：在 webpack.config.js 文件中添加 rules 规则 { test: /\.less$/, use: ['style-loader', 'css-loader', 'sass-loader'] },

第三步：在页面中使用 sass 的地方给<style>添加 lang 属性即可 <style lang="scss"></style>

8、简述如何在项目中安装 Stylus 预处理器

第一步：npm install stylus stylus-loader -D

第二步：在页面中使用 Stylus 的地方给<style>添加 lang 属性即可

```
<style lang="stylus"></style>
```

9、简述什么是命名路由

vue-router 提供了一种隐式的引用路径，可以在创建 Router 实例的时候，在 routes 中给某个路由设置名称 name 值，执行一些跳转的时候，可以通过路由的名称取代路径地址。

第六章

1、请简单描述,Vuex 的下载和安装方式以及步骤。

直接通过<script>标签引入 vuex.js 文件：

1.从 Vue 官方网站可以获取 vuex.js 文件并下载到本地；

2.通过<script>标签引入；

通过 npm 导入 vuex 包；

1.去官网下载 Node 安装包，解压后安装到本地；

2.令行工具执行 npm install vue-cli -g 安装脚手架工具；

3.执行 vue init webpack demo02 命令创建 demo02 项目；

4.打开 demo02 项目，执行 npm install vuex@3.1.1 --save 命令安装 vuex 依赖包；

2、请描述购物车案例的实现过程

主要实现思路是：

案例分析：
购物车案例是在线商城中的基本功能之一，可以实现将顾客想要购买的商品添加到购物车，计算购物车中商品的总价格。
主要由两个页面组成，分别是“商品列表页面”和“购物车”页面。
代码实现：
1.初始化项目；
2.实现底部 Tab 栏切换；
3.获取商品数据 ；
4.实现商品列表页面；
5.实现购物车页面；

3、请使用 actions 实现添加列表功能

```
<!-- 页面结构 -->
  <div id="app">
    <button @click = "addList">添加列表</button>
    <ul>
      <li v-for = "item,key in list" :id = 'key'>{{item}}</li>
    </ul>
  </div>
  <script>
    // 实例化 store
    const store = new Vuex.Store({
      state: {
        list: ['列表 1','列表 1','列表 1']
      },
      mutations:{
        addList(state){
          state.list.push('列表 1')
        }
      },
      actions: {
        addList({commit}){
          commit('addList')
        }
      }
    })
    // 实例化 vm 实例
    var vm = new Vue({
      el: '#app',
      data: {
        id:'',
        list: []
      },
      created(){
        this.list = this.$store.state.list
```

```
    },
    methods: {
      addList() {
        this.$store.dispatch('addList')
      }
    },
    store
  })
</script>
```

4、请使用 getters 对数组中每个元素进行求和计算并返回

```
<!-- 页面结构 -->
<div id="app">
  <button @click="sum">求数组的和</button>
  <div>{{numSum}}</div>
</div>
<script>
  // 实例化 store
  const store = new Vuex.Store({
    state: {
      arr: [1, 2, 3, 4, 5, 6, 3]
    },
    getters: {
      sum: (state) => {
        return state.arr.reduce(function getSum(total, num) {
          return total + num;
        })
      }
    }
  })
  // 实例化 vm 实例
  var vm = new Vue({
    el: '#app',
    data: {
      numSum: 0
    },
    methods: {
      sum() {
        this.numSum = this.$store.getters.sum
      }
    },
    store
  })
</script>
```

5、请介绍 Vuex 单向数据流的主要构成部分以及含义

Vue 的单向数据流增强了组件之间的独立性，但是存在多个组件共享状态的时候，单向数据流状态就会被破坏。
主要构成部分：
State：驱动应用的数据源；
View：以声明方式将 state 映射到视图；
Actions：响应在 View 上的用户输入导致的状态变化；

6、请使用 mutations 实现单击页面的“更新”按钮，实现页面更新

```
<!--页面结构-->
<div id="app">
    <button @click = "update">页面更新</button>
    <div>{{newName}}</div>
  </div>
  <script>
    const store = new Vuex.Store({
      state: {
        name: '初始状态'
      },
      mutations:{
        update(state,payload){
          state.name = payload
        }
      }
    })
    var vm = new Vue({
      el: '#app',
      data: {
        name: 'vue'
      },
      computed:{
        newName(){
          return this.name = this.$store.state.name
        }
      },
      methods: {
        update(){
          this.$store.commit('update','页面已更新')
        }
      },
      store
    })
  </script>
```

7、请简述什么是 Vuex

Vuex 是 Vue 团队提供的一套组件状态管理维护的解决方案。
Vuex 作为 Vue 插件来使用，进一步完善了 Vue 基础代码功能，使 Vue 组件状态更加容易

维护，为大型项目开发提供了强大的技术支持。
通过 new Vuex.Store({})实例化创建 store 实例对象。

第七—九章

1、简述如何实现动态设置页面头部标题效果

在 router.js 文件中，为每个路由添加 meta 属性的页面 title，示例代码如下：

```
    routes: [
     { path: '/', redirect: '/home', name: 'home', meta: { title: '首页' } },
     { path: '/home', component: HomeCon, meta: { title: '首页' } },
     ...
],
```

在 App.vue 组件中使用如下代码获取 meta 数据，示例代码如下

```
<mt-header fixed :title="$route.meta.title"></mt-header>
```

在 main.js 文件中的路由钩子函数里获取 meta 数据，示例代码如下

```
router.beforeEach((to, from, next) => {
  // 路由发生改变修改页面 title
  if (to.meta.title) {
    document.title = to.meta.title
  }
  next() // 确保要调用 next 方法
})
```

2、简述<template>标签的作用

Vue 提供了<template>标签来定义结构的模板，可以在该标签中书写 HTML 代码，然后通过 id 值绑定到组件内的 template 属性上。

3、简述服务器端渲染的不足

第一点：服务器端压力增加　　第二点：涉及构建设置和部署的要求

4、简述什么是 vue-server-renderer 模块及其作用

vue-server-renderer 是 Vue 中处理服务器加载的一个模块，给 Vue 提供在 Node.js 服务器端渲染的功能。vue-server-renderer 依赖一些 Node.js 原生模块，所以目前只能在 Node.js 中使用。

5、简述 npm 和 cnpm 的区别。

npm 即 node.js 包管理工具的全称为 node.js package manager，cnpm 为淘宝镜像，一般同步频率为 10 分钟一次。cnpm 与 npm 使用语法相同，区别在于服务器不同。

6、简述一个项目或者产品的开发流程

1、产品创意
2、产品原型
3、美工设计
4、前端实现
5、后端实现
6、测试、试运行、上线
前端工程师主要专注第 4 步的前端代码实现，其他步骤了解即可。

7、简述如何使用 GUI 创建 vue 项目

第一步：使用 mkdir vue-ui（项目名）命令，创建一个名称为 vue-ui 的项目目录
第二步：执行 cd vue-ui 进入项目中，
第三步：执行 vue ui 命令，执行完毕后，默认会启动一个本地服务，在浏览器打开 localhost:8000 网址，会出现一个 Vue 项目管理器（中文），说明搭建成功，在该界面中需要用户根据项目需求去手动创建并选择配置。

8、简述什么是生产 / 开发依赖。

生产依赖：项目运行时需要的依赖包
开发依赖：项目构建打包时需要的依赖包

9、简述什么是 Nuxt.js 框架。

Nuxt.js 是一个基于 Vue.js 的轻量级应用框架，可用来创建服务端渲染应用，也可充当静态站点引擎生成静态站点应用，具有优雅的代码结构分层和热加载等特性。

10、简述使用 Vue CLI 3 创建项目的方法步骤。

步骤如下：
打开命令行工具，切换到项目根目录，执行以下指令来创建项目：
vue create hello-vue（项目名）
在交互界面中，选择手动配置项，进行配置
项目创建完成后，执行以下命令进去项目目录：
cd hello-vue
执行命令，启动项目
npm run server

11、简述什么是服务器端渲染。

服务器端渲染，顾名思义就是将页面或者组件通过服务器生成 HTML 字符串，将它们直接发送到浏览器，最后将静态标记“混合”为客户端上完全交互的应用程序。

12、讲述 delete 和 Vue.delete 删除数组的区别。

delete 只是被删除的元素变成了 empty/undefined 其他的元素的键值还是不变。
Vue.delete 直接删除了数组，改变了数组的键值。

13、简述如何实现新闻资讯详情页面的数据获取和展示。

请求接口，获取页面数据。
1、在 data 函数中定义空数组，用来存放接口返回数据

```
import { Toast } from 'mint-ui'
export default {
  data () {
    return { newslist: [] }  // 存放列表数据
  },
}
```

2、在 methods 函数中定义获取列表数据的方法 getNewsList()

```
methods: {
  getNewsList () {
    this.$http.get('接口 API').then(result => {
      // 请求成功处理的代码块
    })
    .catch(function (error) {
      // 请求失败处理的代码块
```

```
    })
  }
}
```

3、在 created()钩子函数中调用 getNewsList()方法

```
created () {
  this.getNewsList()
},
```

4、在页面中使用 v-for 进行数据的循环遍历，注意在跳转到详情页时，需要把 id 传递过去，用来区分页面内容

```
<li v-for="item in newslist" :key="item.id">
  <router-link :to="'/home/newsinfo/'+item.id"></router-link>
</li>
```

14、简述使用代码演示父组件向子组件传值。

```
子组件:
<template>
 <h2>{{msg}}</h2> // msg 必须是父组件传递的
</template>
<script>
  export default (){
     props:["msg"]   // 可以是数组，也可以是对象
  }
</script>
 父组件:  // 动态传值,titleVar 是变量
<template>
  <child   :msg = "titleVar"></child>
</template>
<script>
 import Child from '../components/child.vue' // 引入子组件
 export default (){
    components: {Child}, // 注册子组件
    data(){ titleVar :'你好' }
  }
</script>
```

答案说明:

父组件向子组件传值，使用 props 属性

15、简单描述 Vue CLI 3 安装的过程。

步骤如下：以 npm 包管理器为例

推荐使用 Node 8.11.0+和 NPM 3+

安装版本要求:

Node.js 8.11.0+

NPM 3+

如果之前已经全局安装了旧版的 vue-cli（1.x 或 2.x），需要先进行卸载，指令如下:

npm uninstall vue-cli -g

如果之前没有全局安装旧版，则直接全局安装@vue/cli 脚手架，指令如下:

npm install @vue/cli –g
vue –V 查看版本号

16、简述如何安装 vuetify 第三方 UI 插件。

第一步：在项目中，执行 vue add vuetify 命令进行安装。 执行上述命令之后，程序会提示安装选项，使用默认值即可 。
第二步：安装完成后，会在 src 目录里创建一个 plugins 目录，里面会自动生成关于插件的配置文件。

17、简述如何使用 Nuxt.js 脚手架创建项目。

方式一：npx　　npx create-nuxt-app（项目名）
方式二：yarn　　yarn create-nuxt-app（项目名）

18、简述什么是生产 / 开发环境。

生产环境：项目运行时需要的环境
开发环境：项目构建打包时需要的环境

19、简述什么是 Koa 框架。

Koa 是一个基于 Node.js 平台的 Web 开发框架，致力于成为 Web 应用和 API 开发领域更富有表现力的技术框架。

20、简述什么是客户端渲染。

客户端渲染，即传统的单页面应用（SPA）模式，Vue.js 构建的应用程序默认情况下是一个 HTML 模板页面，只有一个 id 为 app 的 根容器，然后通过 webpack 打包生成 css、js 等资源文件，浏览器加载、解析来渲染 HTML

21、简述什么是 Mint UI 框架。

Mint UI 是基于 Vue.js 的移动端组件库，使用 Vue 技术封装出来了成套的组件，可以无缝的和 Vue 项目进行集成开发。

22、简述常用的实现服务器端渲染的方式有哪些。

第 1 种：手动进行项目的简单搭建，
第 2 种：使用 Vue CLI 3 脚手架进行搭建，
第 3 种：是利用一些成熟框架来搭建（如 Nuxt.js）。

23、简述如何实现 SPA 类型的项目。

SPA 就是单页面应用程序，主要依靠路由来实现，路由根据不同的值来展示不同的组件。

24、简述服务器端渲染对 Vue 相关插件版本要求有哪些。

需要的最低 Vue 相关插件版本如下:
vue & vue-server-renderer 2.3.0+
vue-router 2.5.0+
vue-loader 12.0.0+ & vue-style-loader 3.0.0+

25、简述 webpack 服务器端渲染的基本流程。

webpack 将这 entry-server.js 和 entry-client.js 两个入口文件分别打包成给服务器端用的 Server Bundle 和给客户端用的 Client Bundle。当服务器接收到了来自客户端的请求之后，会创建一个 Bundle Renderer 渲染器，这个渲染器会读取 Server Bundle 文件，并且执行它的代码，然后发送一个生成好的 HTML 到浏览器。

26、简述单独路由的配置。

1、使用 npm 方式为项目安装 vue-router
　npm install vue-router --save
2、在 src 目录下，创建单独的路由文件 router.js。

```
import Vue from 'vue'
import VueRouter from 'vue-router'
Vue.use(VueRouter)
var router = new VueRouter({ })  // 创建路由实例对象 router
export default router      // 暴露路由对象属性
3、在 src/main.js 入口文件中引入 router.js 文件。
import router from './router.js'
new Vue({
  router,
  render: h => h(App)
}).$mount('#app')
```

27、简述图片预览插件 vue-preview 的安装与导入。

```
1、安装 vue-preview 插件
npm install vue-preview --save
2、在 main.js 文件导入插件
import VuePreview from 'vue-preview'
Vue.use(VuePreview)
```

28、简述使用代码演示子组件向父组件传值。

```
父组件示例代码:
<template>
 <div>
  <h1>{{title}}</h1>
  <child @getMessage="showMsg"></child>
 </div>
</template>
<script>
 import Child from '../components/child.vue'
 export default {
  components: {Child},
  data(){
   return{
    title:''
   }
  },
  methods:{
   showMsg(title){
    this.title=title;
   }
 }
 }
</script>
 子组件示例代码:
<template>
 <h3>我是子组件</h3>
```

```
</template>
<script>
 export default {
   mounted: function () {
     this.$emit('getMessage','我是父组件！')// 触发当前实例上的事件 getMessage，并把“我是父组件”传递给父组件中。
   }
 }
</script>
```

答案说明:

子组件向父组件传值，使用$emit 触发父组件的自定义事件

29、简述 Nuxt.js 中，声明式路由和编程式路由的区别。

声明式路由：在页面中使用<nuxt-link>完成路由跳转。
编程式路由：在 JavaScript 代码中实现路由的跳转。

30、代码实现在服务器脚本文件 test.js 中将 Vue 实例的渲染结果输出到控制台。

```
// ① 创建一个 Vue 实例
const Vue = require('vue')
const app = new Vue({
  template: '<div>SSR 的简单使用</div>'
})
// ② 创建一个 renderer 实例
const renderer = require('vue-server-renderer').createRenderer()
// ③ 将 Vue 实例渲染为 HTML
renderer.renderToString(app, (err, html) => {
  if (err) {
    throw err
  }
  console.log(html)
})
```

31、简述如何解决在 Windows 上通过 MinTTY 使用 git-bash，交互提示符不起作用的问题。

方式一：使用 winpty 来执行 vue 命令，如 winpty vue.cmd create hello（项目名）
方式二：在 git-bash 安装目录下找到 etc\bash.bashrc 文件，添加“alias vue='winpty vue.cmd'”行为为命令添加别名，重新启动 Git Bash 终端会话，这样更新后的 bashrc 文件才会生效。

32、简述什么是 GUI。

Vue CLI 引入了图形用户界面（GUI）来创建和管理项目，功能十分强大，给初学者提供了便利，可以快速搭建一个 Vue 项目。

2.请简述 Vue 全局 API 接口的主要内容。

1）. Vue.directive():Vue 中有很多内置指令，如 v-model、v-for 和 v-bind 等

2）. Vue.use():Vue.use 主要用于在 Vue 中安装插件，通过插件可以为 Vue 添加全局功能

3）. Vue.extend():Vue.extend 用于基于 Vue 构造器创建一个 Vue 子类，可以对 Vue 构造器进行扩展

4）. Vue.set():Vue 的核心具有一套响应式系统，简单来说就是通过监听器监听数据层的数据变化，当数据改变后，通知视图也自动更新

5）. Vue.mixin():Vue.mixin 用于全局注册一个混入，它将影响之后创建的每个 Vue 实例

3.请简单介绍 Vue 实例对象的属性和方法。

1.请简述 JavaScript 钩子函数包括哪些。

入场钩子分别是 beforeEnter（入场前）、enter（入场）、afterEnter（入场后）和 enterCancelled（取消入场）

出场钩子分别是 beforeLeave（出场前）、leave（出场）、afterLeave（出场后）和 leaveCancelled（取消出场）

< transition> @before-enter="beforeEnter" @enter="enter"

@after-enter="afterEnter" @enter-cancelled="enterCancelled"

@before-leave="beforeLeave" @leave="leave"

@after-leave="afterLeave" @leave-cancelled="leaveCancelled"

v-bind:css="false"> // Vue 会跳过 CSS 的检测 < /transition>

2.请简述 6 个内置的过渡类名。

进入（enter）： v-enter: 在元素被插入之前生效，在元素被插入之后的下一帧移除 v-enter-active:

在整个进入过渡的阶段中应用，在元素被插入之前生效，在过渡动画完成之后移除 v-enter-to:

在元素被插入之后下一帧生效（与此同时 v-enter 被移除），在过渡动画完成之后移除 离开（leave）:

v-leave：在离开过渡被触发时立刻生效， 下一帧被移除

v-leave-active：在整个离开过渡的阶段中应用，在离开过渡被触发时立刻生效，在过渡完成之后移除

v-leave-to：在离开过渡被触发之后下一帧生效（与此同时 v-leave 被移除），在过渡动画完成之后移除

3.请简述自定义过渡类名的属性有哪些。

enter-class

enter-active-class

enter-to-class

leave-class

leave-active-class

leave-to-class

注意：自定义类名的优先级高于普通类名

1. 简述以 npm 方式安装 vue-router 的步骤

① 在指定目录下 npm install vue-router@3.1.x

② 创建 main.js 逻辑入口

```
import Vue from 'vue'
import App from './App.vue'
import VueRouter from 'vue-router'
Vue.use(VueRouter)
import router from './router.js'
new Vue({
    el: '#app',
    render: c => c(app),
    router
})
```

Router.js 文件

```
import VueRouter from 'vue-router'
 var router =new VueRouter({
 routes:[
     {path:'/',redirect:'/login'},
     { path:'/',component:Login}
 ]
})
 export default router
```

2. **简述 vue-roter 路由的作用**

根据不同的 url 哈希值，在路由视图中显示不同的页面，实现非跳转式的页面切换在单页面应用中**更新视图**可以**不用重新请求页面**

用户体验好，不需要每次都从服务器全部获取，快速展现给用户

3. **请简单列举并说明路由对象包括哪些属性**

$route.path　对应当前路由的名字

$route.query：{key : value} URL 查询参数

$route.params：{key : value} 路由转跳携带参数

$route.hash：在 history 模式下获取当前路由的 hash 值（有 # 的那种）

$route.fullPath：完成解析后的 URL，包含查询参数和 hash 的完整路径

$route.name　当前路由的名称

$route.matched：路由记录，当前路由下路由声明的所有信息，从父路由（如果有）到当前路由为止

$route.redirectedFrom：若有重定向，即为重定向来源的路由

如何安装 Vue CLI 3.x 版本的脚手架

npm install @vue/cli –g　　# -g 表示全局安装

如何在现有项目中安装 CLI 插件和第三方插件

在项目目录下，使用 vue add 指令可以安装插件

Eg：vue add router

第三方插件：

如果**不带 @vue/ 前缀**，将会安装第三方包

例如 vue add vuetify

简单介绍 CLI 服务 vue-cli-service<command>中的 command 命令包含哪些？

serve 启动服务

build 生成用于生产环境的包

inspect 审查 webpack 配置

lint 并修复源文件

简要分析 Vuex 的设计思想

用来管理状态，Vuex 作为 Vue 插件来使用，使 Vue 组件状态更加容易维护

简述 vuex 配置对象中的主要内容

1）. actions：用来定义事件处理方法，用于处理 state 数据

2）. mutations：选项中的事件处理方法接收 state 对象作为参数，即初始数据

3）. getters：store 实例允许在 store 中定义 getters 计算属性，类似于 Vue 实例的 computed

4）. modules：modules 用来在 store 实例中定义模块对象

5）. plugins：Vuex 中的插件配置选项为 plugins，插件本身为函数

6）. devtools：store 实例配置中的 devtools 选项用来设置是否在 devtools 调试工具中启用 Vuex，默认值为 true，表启用

简述 vuex 中的 actions 的含义

actions 选项用来定义事件处理方法，用于处理 state 数据。actions 类似于 mutations，不同之处在于 actions 是异步执行的，事件处理函数可以接收 {commit} 对象，完成 mutation 提交，从而方便 devtools 调试工具跟踪状态的 state 变化。

在使用时，需要在 store 仓库中注册 actions 选项，在里面定义事件处理方法。事件处理方法接收 context 作为第 1 个参数，payload 作为第 2 个参数（根据需要进行选择）。

请简述什么是服务器端渲染。

服务器端渲染（简称 SSR），是将组件或页面通过服务器生成 html 字符串，再发送到浏览器，最后将静态标记"混合"为客户端上完全交互的应用程序，简单理解就是将页面在服务器中完成渲染，然后在客户端直接展示。

请简述服务器端渲染的代码逻辑和处理步骤。

Vue 进行服务器端渲染时，需要利用 Node.js 搭建一个服务器，并添加服务器端渲染的代码逻辑。

使用 webpack-dev-middleware 中间件对更改的文件进行监控，使用 webpack-hot-middleware 中间件进行页面的热更新，使用 vue-server-renderer 插件来渲染服务器端打包的 bundle 文件到客户端

请简述 Nuxt.js 中，声明式路由和编程式路由的区别。

声明式路由：在页面中使用 完成路由跳转。

编程式路由：在 JavaScript 代码中实现路由的跳转。

请简单列举一个项目从开始到上线的开发流程需要的步骤。

1、产品创意：在这一阶段就是决定要做一个什么产品（What），为什么要做这个产品（Why），解决 What 和 Why 的问题

2、产品原型：在这一阶段，通过由产品经理对原型进行设计包括功能、页面，最重要的是用户体验

3、美工设计

在这一阶段，美工设计人员根据产品经理提供的原型图实现 PSD 设计图稿，并切图

4、前端实现

在这一阶段，前端开发工程师拿到美工设计好的 psd 图，负责具体的 HTML、CSS

静态页面的实现，以及动态特效、动态数据的绑定和交互的实现

5、后端实现

在这一阶段，实现数据处理、业务逻辑代码

6、测试、试运营、上线

由测试人员进行项目测试。将所有的问题解决后，就可以试运行，将项目上线

2．请简单列举 6 个“微商城”项目中用到的重点知识。

1、路由的配置

2、Vuex 的配置

3、axios 的配置以及接口的调用

4、购物车功能的实现

5、底部导航栏的实现

6、分类列表的实现

7、图片预览的实现 vue-preview 插件

下面选项中，主要用来编写网页的结构的是（）。

A、php

B、HTML

C、CSS

D、JavaScript

下列选项中，代表视图部分的是（）。

A、Element

B、DOM

C、Model

D、View

、

下面选项中，可以引入 vue.js 文件的是（）。

A、<a>

B、<script>

C、<style>

D、<link>

Windows 系统内置的命令行工具是（）。

A、npm

B、cmder

C、git-bash

D、cmd

下面选项中，表示一个 Node.js 的包管理工具，用来解决 Node.js 代码部署问题

的是（）。

A、webpack

B、vue

C、Node

D、npm

下面选项中，可以用来创建实例对象的关键字是（）。

A、let

B、var

C、data

D、new

下面选项中，在创建 Vue 实例时，用来表示唯一根标签的是（）。

A、watch

B、components

C、data

D、el

下面选项中，可以通过插值语法将 data 初始数据绑定到页面中的是（）。

A、[[]]

B、{}

C、{{}}

D、[]

1、

以下代码

```
 var vm = new Vue({
     el: '#app',
      data: {
          price: 20,
          num: 1
     },
     computed: {
        //  总价格 totalPrice
        totalPrice () {
           return this.price * this.num
        }
     }
})
```

console.log(vm.totalPrice);执行后的结果是（）。

A、0

B、2

C、20

D、1

下列选项中，在插值表达式中可以使用过滤器来对数据进行处理，管道符号是（）。

A、/

B、>

C、|

D、\

下列选项中，可以实现绑定事件的指令是（）。

A、v-show

B、v-on

C、v-html

D、v-text

下列选项中，可以实现获取随机数的是（）。

A、Math.max()

B、Math.min()

C、Math.random()

D、Math.floor()

下面选项中，能够实现自定义指令的是（）。

A、Vue.use()

B、Vue.set()

C、Vue.component()

D、Vue.directive()

下面选项中，能够实现在 Vue 中安装插件的是（）。

A、Vue.use()

B、Vue.set()

C、Vue.component()

D、Vue.directive()

下面选项中，用于向响应式对象中动态添加一个属性，并确保这个新属性同样是响应式的，且触发视图更新的是（）。

A、 Vue.extends

B、Vue.use

C、Vue.extend

D、Vue.set

下面选项中，用来获取用户自定义选项的是（）。

A、vm.$data

B、 vm.$props

C、vm.$attrs

D、vm.$options

下面选项中，用来访问 vm 实例使用的根 DOM 元素的是（）。

A、 vm.$el

B、 vm.$props

C、 vm.$slots

D、vm.$attrs

下面选项中，用来访问子组件实例的是（）。

A、vm.$children

B、 vm.$props

C、vm.$slots

D、vm.$attrs

Vue 全局配置对象 config 中用于控制生产信息的显示和隐藏的是（）。

A、devtools

B、productionTip

C、tip

D、silent

Vue 提供了内置的过渡封装组件是（）。

A、transition

B、Velocity

C、animate

D、group

下面选项中，transition 类名中表示进入过渡的开始状态的是（）。

A、v-leave

B、v-enter-to

C、v-enter-active

D、v-enter

下面选项中，可以通过百分比的形式来规定动画的状态的是（）。

A、v-enter-to

B、CSS

C、animate

D、@keyframes

下面选项中，可以通过 transition 组件的钩子函数实现动画的是（）。

A、v-enter

B、v-enter-active

C、enter-class

D、@before-enter

下面选项中，属于 JavaScript 动画库的是（）。

A、vue.js

B、Velocity.js

C、animate.css

D、react.js

下面选项中，可以实现不同标签名的过渡效果的是（）。

A、v-show 和 v-hidden

B、v-else 和 v-show

C、v-show 和 v-if

D、v-if 和 v-else

下面选项中，相同标签名元素的过渡需要为元素定义的属性是（）。

A、class

B、key

C、id

D、name

下面选项中，用来实现<transition>组件过渡模式切换的是（）。

A、pattern

B、mode

C、change

D、type

动态组件需要通过 Vue 中的<component>元素绑定（）属性来实现多组件的过渡。

A、contains

B、to

C、name

D、is

下面选项中，可以与 v-for 结合使用并实现列表过渡的组件是（）。

A、item-list

B、<transition>

C、<transition-group>

D、list

下面选项中，可以实列表循环渲染的是（）。

A、v-on

B、v-else

C、v-for

D、v-if

下面选项中，可以实现列表排序平滑过渡的是（）。

A、v-mode

B、v-move

C、v-enter

D、v-leave

下面选项中，能够判断字符串中是否存在某一字符的是（）。

A、inof

B、indexof

C、index

D、indexOf