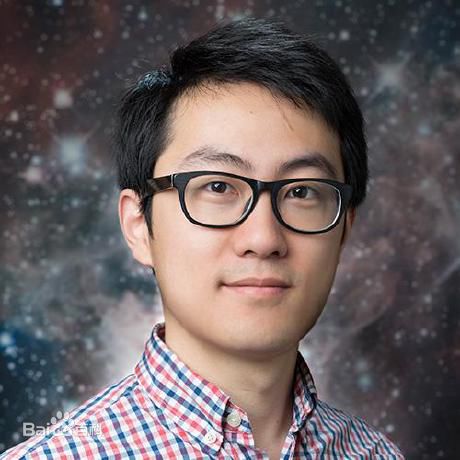
**Vue记录者:--白建平**

**尤雨溪**



是[Vue.js](https://baike.baidu.com/item/Vue.js)框架的作者，HTML5版Clear的打造人。他认为，未来App的趋势是轻量化和细化，能解决问题的应用就是好应用。而在移动互联网时代大的背景下，个人开发者的机遇在门槛低，成本低，跨设备和多平台四个方面。

1.vue是一个渐进式得javaScript得框架

渐进式:vue.js -->组件化-->vue-rputer(跳转路由)->vuex->构建工具webpack-->周边生态环境

**2.mvvm模式 , mvc模式s**

mvvm:

m:model(数据层)

v:view(视图层) -->指向html（试图）

vm:view-model(试图模型) --> m和v得一个桥梁

mvc:

m:model(数据层)

v:view(视图层) -->指向html（试图）

c:controller(控制器)

3.vue得核心特点：组件化，数据驱动，虚拟dom

**4.vue指令**

常用指令:s

v-show:和v-if

v-show:切换频率高时用

v-if:切换频率低时用

v-for遍历

v-on:绑定事件

v-bind:绑定属性 用“ ： ”冒号代表v-bind了 :key=“i”

v-model:双向绑定

v-cloak:为了防止网速加载慢，导致未加载完，页面上出现{{}}

用户体验并不好

具体写法：

<div v-cloak>{{ list }}</div>

在便签上加 v-clack 然后在加样式 display：none；

div【v-cloak】{

Display:none;

}

修饰符:

.stop 阻止冒泡(嵌套div 案例)

.prevent 阻止默认事件(a标签体跳转 可以组织默认事件 当按钮用)

.once 只能点一次

5.v-for key=“i”？什么意思

Key作用：提高遍历性能

唯一性

如果数据相同 好辨别

6**.vue 响应式得原理**

主要利用ES5得object.defineProperty()中得getter和setter来实现得

如果要让数据响应，必须放在将数据放在data对象中提前定义好，但如果后面新添加的属性，而不会响应，如果让后来添加的数据也响应呢？？？这回要请出$set来实现

$set:

格式:this.$set(要添加得对象,属性,值)

列如:this.$set(v,’num’,1)

7.计算属性computed 和 监听watch和 methods方法

Methods: 方法s 每次触发才能调用里面得代码

Computed : 只要依赖项数据不发生改变，则不会执行其中代码（即有缓存数据的作用），直接返回，除非依赖项发生变化才执行其中代码

Watch:监听属性 即监听,可以监听data和computed中数据的新旧的变化

  ```

   watch:{

        title(newvalue,oldvalue) {

            console.log('oldvalue:',oldvalue)

            console.log('newvalue:',newvalue)

            if(newvalue>20) {

                alert('库存不足')

            }

        },

        total(newvalue,oldvalue) {

             if(newvalue>200) {

                alert('库存不足')

            }

        }

```

8. **## 样式处理**

  vue处理样式：class样式，行内样式

  1.class样式：

     （1）对象写法：例如：

     格式：<li  :class="{ hot:布尔值}">

     ```

        <li :class="{ hot:item.num>5}">

     ```

     （2）数组写法:适合添加多个类

     格式：<li  :class="[ 类1,类2,.... ]">

     总结：$set，计算属性，动态样式

**Vue过滤器:**

1. Vue过滤器是什么 用于进行数据得转换方法(格式转换)
2. 使用场景：例如：性别得处理，货币得处理，字母大小得处理，日期的处理
3. 具体实现2中方法
4. 全局过滤器:

一定要写在数据名的后面

<span class="span">{{v.price | qian}}.00</span>

Vue.filter(‘过滤名称’,function(函数){

})

后台请求到的数据未必是

(2) 局部过滤器

通过在vue实例中添加filters属性来实现局部过滤器的

<span class="span">{{v.price | qian}}.00</span>

Filters:{

qian(vv){

  return "￥"+vv

}

}

**##vue自定义指令**

什么是自定义指令:

v-for,v-if,v-on.v-bind ---是内置指令（是有限得）

提供了除了内置指令之外得，封装dom操作所需要得指令

v-自定义指令名 --来调用指令

**具体实现**:

全局自定义指令:

Vue.directive(‘指令名’,function(){

事件名:{

Inserted(el){

Console.log(‘el’,el)

}

}

})

局部自定义指令：

**Vuej脚手架 安装**

什么是vue组件:

就是可以独立使用的视图单元，每个组件包括 template script style三部分

注意:

父组件能改变子组件，但子组件不能改变父组件 这就是所谓的单向数据流(推荐)

父组件能改变子组件，子组件也能能改变父组件 这就是所谓的双向数据流

组件的好处:

方便重复使用和维护，分离关注点，提高开发效率

组件通信

父传子:

在父组件的子组件标签上定义个属性名 + 值

<Shou :arr="arr"></Shou>

在子组件里通过 props来接受这个数据

export default {

        props:['arr']

props:{

arr:{

type:[String,Number]

}

}

Arr: 数据只是简单的接受

对象: 可以验证数据类型和默认值

子传父:

在子组件里定义个事件名:然后在方法里通过this.$emit((属性名：值)

<button @click="del()">哈哈</button>

del(){

                this.$emit('del',this.num)

            }

在父组件中监听子组件派发的事件来接受这个数据

 <Shou @del="rect"></Shou>

用@监听子组件理的事件名 = “父组件的名“

   methods: {

     rect(v){

       console.log(v)

     }

   },

兄弟传值:

1. 在同一个父组件的情况下 A组件 =>父组件 =>B组件
2. Bus中央同信

创建一个空的vue实例担任总线的位置 Bus.Js文件里

A组件:( 引入总线（引入Bus.js 文件） )

通过Bus.$rmit(‘ 自定义属性名 ’, + 值)

qq(){

                 Bus.$emit('fun',this.tou)

             }

B组件(引入总线（引入Bus.js 文件)

采用钩子函数 created 来监听A组件

在通过Bus.$on(‘自定义属性名’ + 值)

注意：自定义属性名一定要是一样的

组件slot:

1. props
2. 事件
3. Slot:

Slot作用: 提供一个组件占位符的作用,封装一个有共性,也带有差异性的组件时比较常用

<slot></slot>

如果插件多个slot，通常每个slot用name命名，以区分

例如:<slot name=”上面定义的name名“></slot>

嵌套数据:

通过template标签添加v-slot来条用上面你的name名称

<template v-slot:name=“名称“></template>

Keep-alive:

主要用于缓存组件，保留组件内的数据状态

<Keep-alive include=”tab1,tab2”>

要缓存的组件

</Keep-alive>

Include 名称匹配的组件缓存

Exclude 名称匹配的组件不会被缓存

**生命周期**

第一阶段:创建阶段

创建前:beforeCreate

创建后:created

第二阶段:挂载阶段

创建前:beforeMount

创建后:Mounted

第三阶段:创建阶段

创建前:beforeUpdate

创建后:updated

第四阶段:创建阶段

创建前:beforeDestroy

创建后:destroyed

**Vue路由**

路由是什么?

英文:router 通过路由可以分发，跳转到不同的位置

SPA是什么?

SPA : Sing Page Appliaction,即单页面应用,通过路由技术实现在一个页面中展现对应的内容，提高打开速度

路由原理:

无论是 vue，react，angular，前端路由实现原理都是相同的

Hash + onhashchange实现

H5

1. history 和 hash 得区别

默认hash 兼容性好

缺点：地址栏有 # 号 不美观

History:

后台需要配置页面路径

1. 嵌套路由
2. {
3. path:'/tou',
4. name:"Tou",
5. component:Tou,
6. children:[
7. {
8. path:'nan',
9. component:Nan
10. },
11. {
12. path:'box',
13. component:Box
14. },
15. {
16. path:'tong',
17. component:Tong
18. },
19. {
20. path:'',
21. redirect:'box'
22. }
23. ]
24. },
25. 路由传参 Query params

Query传参

传：<router to=”/a/b?c=100”></router>

接:路由中通过冒号+属性姓名配置一下

在要接受得组件中 this.$route.query

params传参

传：<router to=”/a/b/c”></router>

接：在要接受得组件中 this.$route.parmas

1. 导航路由

Route 获取信息

Router 跳转路由

This.$router.push(‘ /要去得路由 ’)

This.$router.replace()

This.$router.go()

1. 路由钩子函数

**Vuex**

1.什么样的数据应该放到vuex中？

组件之间有可能会被共享的数据放到vuex仓库中

2.