

数据绑定

v-bind：动态绑定

**v-bind:href="website"**

**website**:[**https://www.baidu.com**](https://www.baidu.com)

<div v-bind:class="{ active: isActive }"></div> 绑定class

isActive为true，则存在active类。



v-html：

**v-html="websitetag"**

**websitetag**:**"<a href='https://www.baidu.com'>webtag</a>"**

v-on: //@

方法method——function

获取方法时要加括号 {{ function（）}}

*<!--vue-app是跟容器-->*<**div id="vue-app"**>  
 <**p**>{{greet('morning')}}</**p**>  
 <**h1**>{{ name }}</**h1**>  
 <**h2**>{{ job }}</**h2**>  
 <**a v-bind:href="website"**>web</**a**>  
 <**input type="text" v-bind:value="job"**>  
 <**p v-html="websitetag"**></**p**>  
  
 <**button v-on:click="add(1)"**>add</**button**>  
 <**button @dblclick="add(10)"**>add\*10</**button**>  
 <**p**> my age is {{age}}</**p**>  
 <**div id="canvas" v-on:mousemove="updateXY"**>  
 {{x}},{{y}} -<**span v-on:mousemove="STOP"**> Stop</**span**>  
 </**div**>  
 <**a v-on:click.prevent="alert()" href="https://www.baidu.com"**> the new</**a**>  
</**div**>

事件修饰符：

**v-on:click.prevent 阻止弹出**

*alert 警告*

键盘事件：

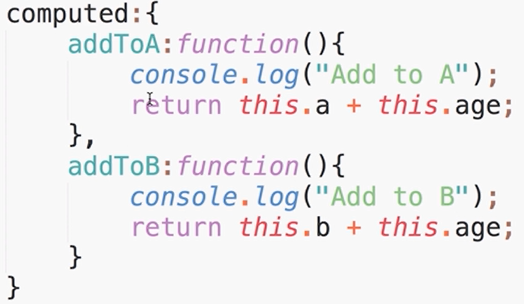
**v-on:keyup.enter** 点enter才会触发

**v-model双向绑定**

**ref="name"**

计算属性：更优化 （用于耗时，大量搜索的时候）用法和方法（methods）

Computed





只有true和flase两个值

v-if

v-show 只是不显示，一直存在

v-for:

**<li v-for=“user in users”>**

**{{user}}**

**</li>**

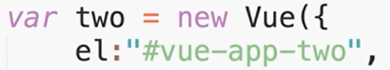
**<li v-for=“(user,index)下标 in users”>**

**{{index}}-{{user}}**

**</li>**

**实例化多个vue对象**

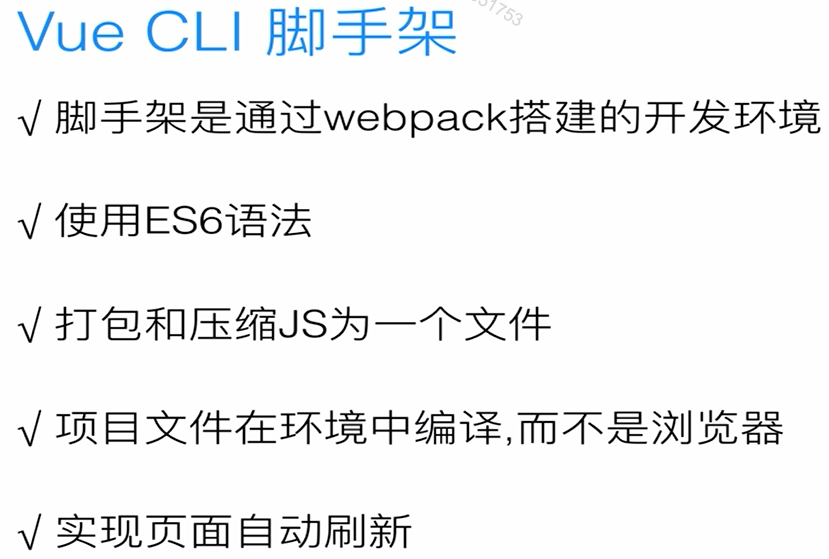




**Vue-app-two改变vue-app-one的title**





****

**npm run dev**

**Src文件：**

**Assets里面放图片**

**Component里面放组件**

****

**Main.js与index.html直接相关**

**App.Vue是组件（包含模板：html结构，行为：处理逻辑，样式：解决样式）**

**Template标签内只能有一个根标签**

注册组件：

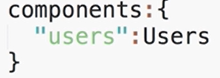
**import** Users **from './components/Users'**

**Vue**.component(**"users"**,Users); **Users为创建的Users.vue文件**

**直接使用<users>标签**

Main.js中是全局注册组件

App.vue中<script>中是局部注册组件：



Scoped控制css样式的作用域



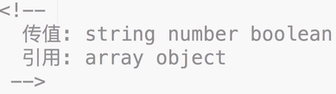
**组件属性传值：**

**将users.vue中的内容放到app.vue**

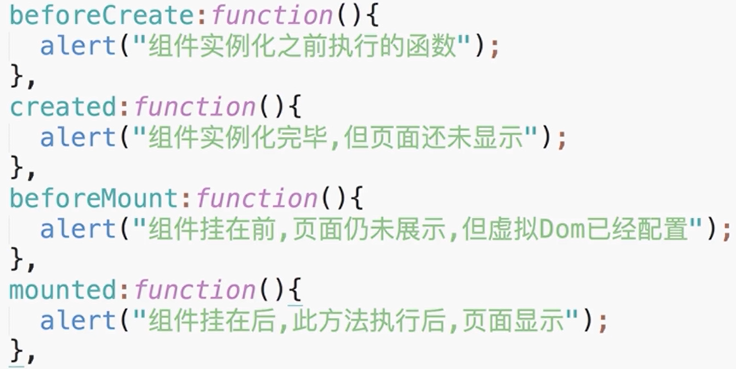
**App.vue:**

**users.vue：**

****

****

**生命周期**





**路由**

*//配置路由***const** router = **new** VueRouter({  
 **routes**:[  
 {**path**:**"/"**,**component**:Home}, 找到‘/’执行Home组件  
 {**path**:**"/helloworld"**,**component**:HelloWorld},  
 ]，

**mode**:**"history"** 取消‘#’  
})

<**ul**>  
 <**li**><**a href="/"**>home</**a**></**li**>  
 <**li**><**a href="/helloworld"**>hello</**a**></**li**>  
</**ul**>

<**ul**>  
 <**li**><**router-link to="/"**>home</**router-link**></**li**>  
 <**li**><**router-link to="/helloworld"**>hello</**router-link**></**li**>  
</**ul**>

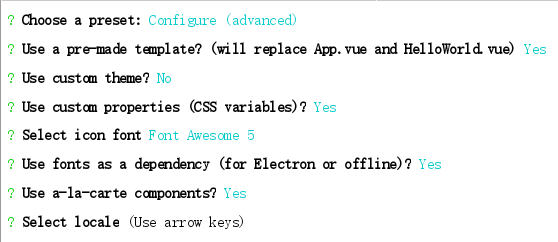
**点击不刷新；**

created() {  
 **this**.**$http**.**get**(  
 **"http://jsonplaceholder.typicode.com/users"** ).then((data)=>{  
 **this**.**users**=data.**body**;  
 })  
}

**vue3.0：**

**.vuerc文件 存储cli的文件**

**vue add添加新的插件**



**环境变量**

**.env文件**

**.env.development 开发环境变量**

**VUE\_APP\_URL=**

**vue serve 独立运行单个vue文件**

**图形页面创建项目：**

**Vue ui**

**Vue.config.js的配置：**

**Asset文件里面放图片，css文件**

****

devServer:{  
 open:true,*//自动开启浏览器* host:"localhost", *//本机名字* port: 8081,*//端口号* https: false, *//true可能有警告* hotOnly: false, *//热模块更新，更新时自动加一些配置* proxy: {  
 *//配置跨域* '/api':{  
 target:"http//localhost:5000/api/" *//跨域地址* ws:true,  
 changeOrigin: true,  
 pathRewrite:{  
 '^/api':''  
 }  
 }  
 }  
}