# 一、v-for

## 1.普通用法（适用于遍历数组）

<li v-for=" 数据项 in 在vue对象中定义的数据源">{{数据项}}</li>

## 2.带索引的普通用法（适用于遍历数组）

<li v-for=" (数据项,i) in 在vue对象中定义的数据源" :key="i">{{i}}:{{数据项}}</li>

i表示当前循环的下标

## 3.高端用法（适用于遍历单个对象有多个属性的情况）

<li v-for="value in 是一个对象的数据源">{{value}}</li>

value是该对象中所有键值对的值

## 4.高端用法（适用于遍历多个对象有多个属性的情况）

<li v-for="(v,k,i) in 是一个对象的数据源"></li>

v 键值对中的值

k 键值对中的键

i 当前键值对的索引

## 5.嵌套的for

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="utf-8">  <title></title>  <link href="https://cdn.bootcss.com/twitter-bootstrap/4.3.1/css/bootstrap.min.css" rel="stylesheet">  </head>  <body>  <div id="app">  <table class="table table-hover" align="center" border="1px">  <tr>  <th>序号</th>  <th>姓名</th>  <th>年龄</th>  <th>对象</th>  </tr>  <tr v-for="(student,i) in students" :key="i">  <td>{{i+1}}</td>  <td>{{student.username}}</td>  <td>{{student.age}}</td>  <td>{{student.girl}}</td>  </tr>      </table>  <hr/>    <ul>  <li v-for="(v,k,i) in stu">  {{i+1}}=={{k}}=={{v}}  </li>  </ul>  <hr/>  <ul v-for="student in students">  <li v-for="(v,k,i) in student">  {{i+1}}=={{k}}=={{v}}  </li>    </ul>    </div>    </body>  <script src="https://cdn.bootcss.com/twitter-bootstrap/4.3.1/js/bootstrap.min.js"></script>  <script src="https://cdn.bootcss.com/vue/2.6.10/vue.js"></script>  <script>    new Vue({    el:'#app',  data:{  stu:{  username:'小鱼',  age:20,  girl:'如花'  },  students:[  {  username:'小鱼',  age:20,  girl:'如花'  },  {  username:'小飞',  age:21,  girl:'翠花'  },  {  username:'小鹏',  age:22,  girl:'志玲'  }    ]  },  methods:{    }  });      </script>  </html> |

# 二、vue实例

## 1.vue实例之间可以相互调用属性和方法

1）一个实例调用另一个实例的属性

2）一个实例调用另一个实例的方法

|  |
| --- |
| <body>  <div id="app1">  {{message}}  </div>  <div id="app2">  <button @click="btnfn" type="button">悄悄话</button>  </div>    </body>  <script src="https://cdn.bootcss.com/vue/2.6.10/vue.js"></script>  <script>    var v1 = new Vue({    el:'#app1',  data:{  message:'晚上约吗'  },  methods:{  dosomething:function(){  alert("很开心！");  }  }  });    var v2 = new Vue({    el:'#app2',  data:{    },  methods:{  btnfn:function(){  v1.message = "go!你家我家还是如家？";  v1.dosomething();  }  }  });  </script> |

## 2.实例属性

vue的属性，指的是vue中data中的内容。

vue的实例属性

### 1)$data

vue中的直接的内容：比如 v1.$el v1.$data 等等 ，可以直接调用

### 2)$refs

标签中的ref相当于是id

通过$refs.id找到某个元素

### 3）$mount

实现vue对象和html元素的动态绑定

|  |
| --- |
| var v1 = new Vue({  template:"<h3>晚上好！</h3>"//定义好内容  });  v1.$mount("#app"); |

# 三、Vue组件

## 1.组件化的概念

万事万物皆组件

所有的东西都是组件，之后我们开发vue就一个组件就够了。

将一个vue对象要实现的html的内容 进行封装，可以实现复用的目的。

### 1）全局注册

注册的组件可以在全局的div中（只要被vue绑定过）使用

步骤一：组件的注册

|  |
| --- |
| Vue.component("hello",{  data:function(){  return {  message:'晚上好'  }  },  template:"<div><h3>{{message}}</h3><button @click='comeon' type='button'>快进来吧！</button></div>",  methods:{  comeon:function(){  // alert("我来啦！");  this.message = "贵宾楼上请";  }  }  }) |

此时就能使用hello标签，一个hello标签就能展示后面的vue组件的内容

步骤二：注册好的组件只能在被vue绑定的div中使用。

步骤三： 在div中使用组件，直接使用标签即可

<div id="app">

<hello></hello>

<hello></hello>

<hello></hello>

<hello></hello>

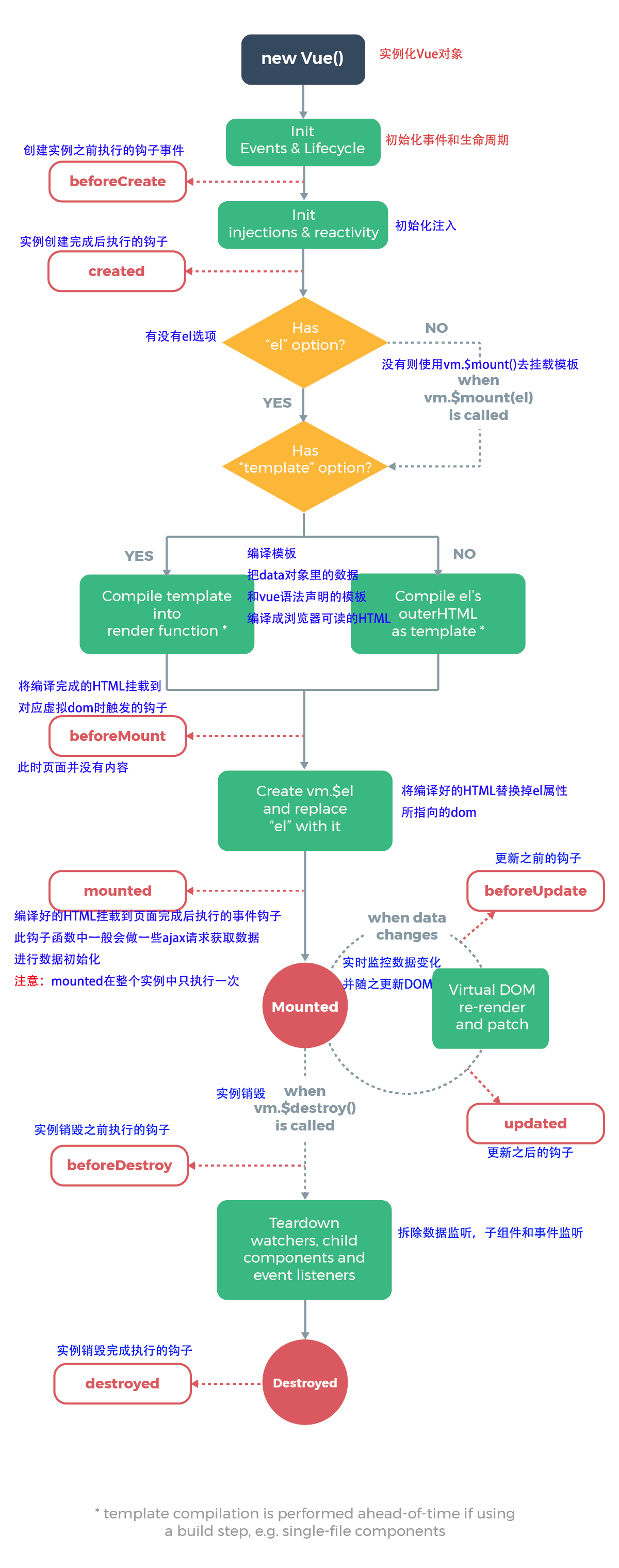
</div>

### 2）本地注册

注册的组件，只能在当前vue所绑定的那个div中使用，其他div中不能使用。

|  |
| --- |
| <body>  <div id="app">  **<myheader></myheader>**  </div>  <div id="app1">  <!--该组件只能在app的div中使用，因为该组件是被本地注册的，而非全局注册-->  <myheader></myheader>  </div>  </body>  <script src="https://cdn.bootcss.com/vue/2.6.10/vue.js"></script>  <script>  var myheader = {  template: "<h3>Java1904\*\*院</h3>"  }  new Vue({  el: '#app',  **components:{**  **"myheader":myheader //本地注册**  **}**  });    new Vue({  el:'#app1'  })    </script> |

## 2.vue的生命周期



# 四、Vue-cli骨架的开发

vue-cli可以帮我们快速的生成一个项目的模板。

## 1.安装node.js

要想使用vue-cli，就必须先安装node.js



## 2.安装vue-cli脚手架工具

npm install vue-cli -g

使用淘宝镜像来执行npm命令会更快一些

安装淘宝镜像

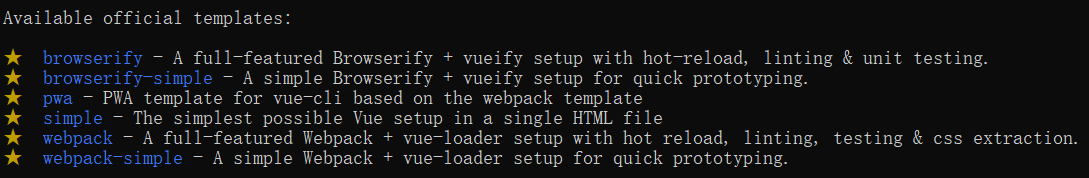
npm install -g cnpm --registry=https://registry.npm.taobao.org

安装好淘宝镜像以后，就可以使用 cnpm来安装vue-cli

cnpm install vue-cli -g

## 3.查看当前可以使用的vue模板有哪些

vue list



我们要使用的模板是webpack

4.如何安装webpack模板

步骤一： 创建一个文件夹，用于拉取该模板

步骤二：使用命令拉取模板：

vue init webpack 项目名

拉取过程中需要输入配置信息



步骤三：进入到该工程文件夹内，然后使用npm install来安装整个项目所需要的依赖环境。

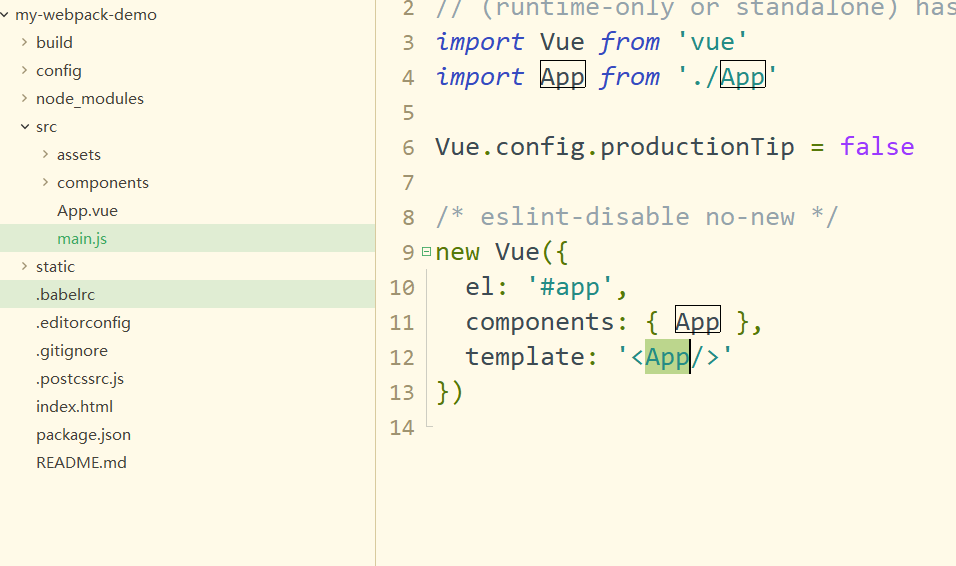
cd my-webpack demo

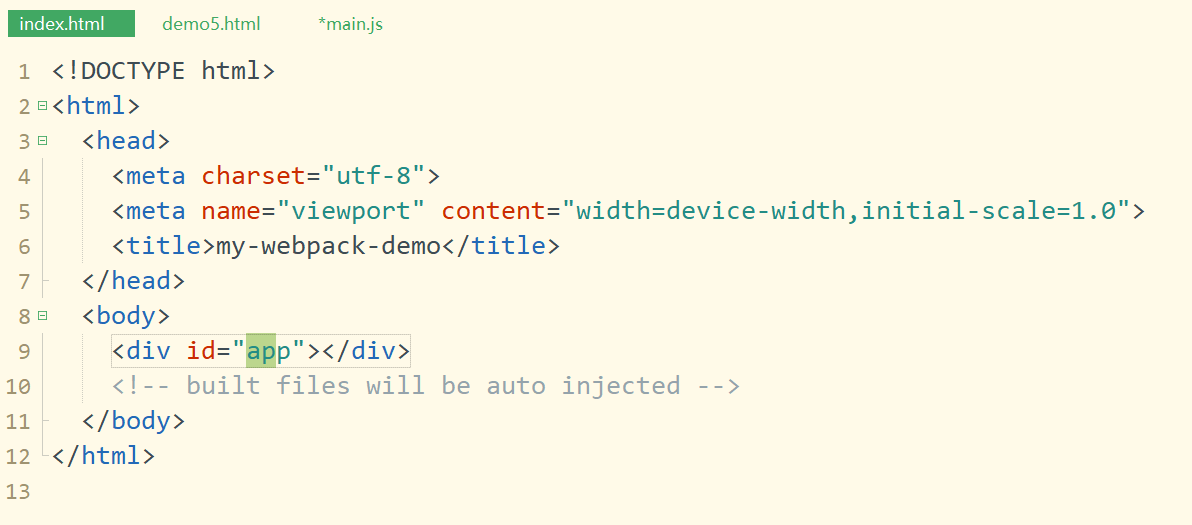
npm install 如果网速慢。可以使用 cnpm install

安装依赖以后，整个项目的结构 如图：



4.webpack项目结构



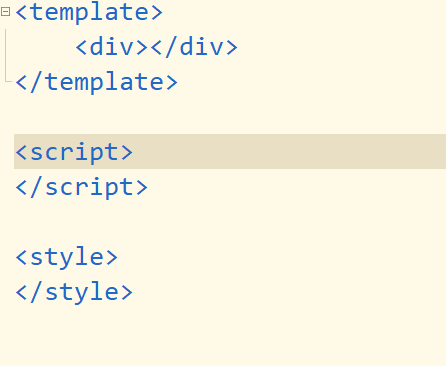


Vue项目里面,所有的内容都是组件,而App这个组件 是个主组件

# 五、使用webpack进行开发

webpack项目里所有的内容都是组件。开发过程就是把其他的组件组装（塞到）在App.vue中。所以webpack项目只有一个index.html，index.html只绑定了一个App.vue.

## 1.webpack中组件的组成部分：



其中template==> html，注意 template必须有且只能有一个根标签。

script==>js

<script>

export default{

el:'',

data:{}

...

}

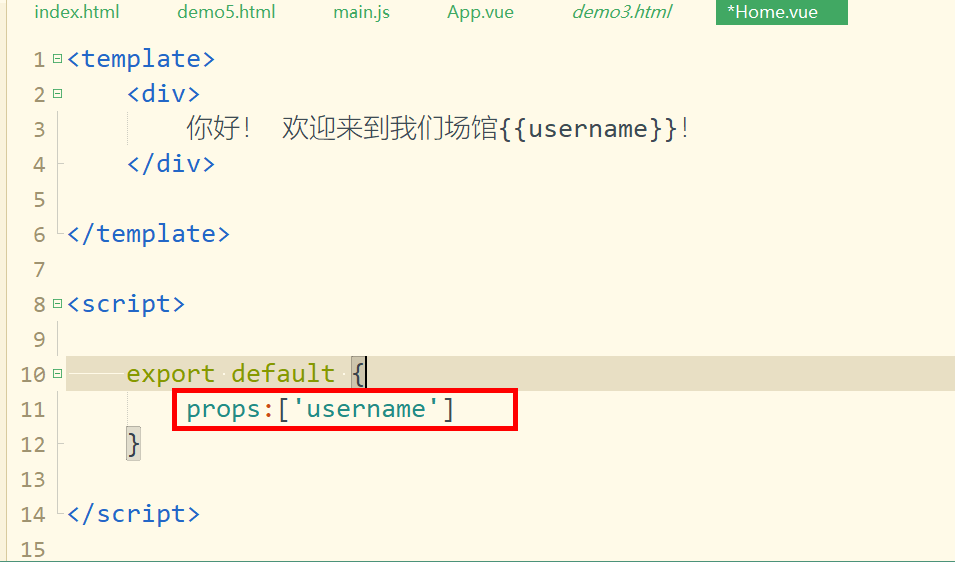
</script>

style==>css

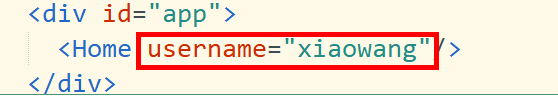
## 2.组件的参数传递

## 1) 父传子

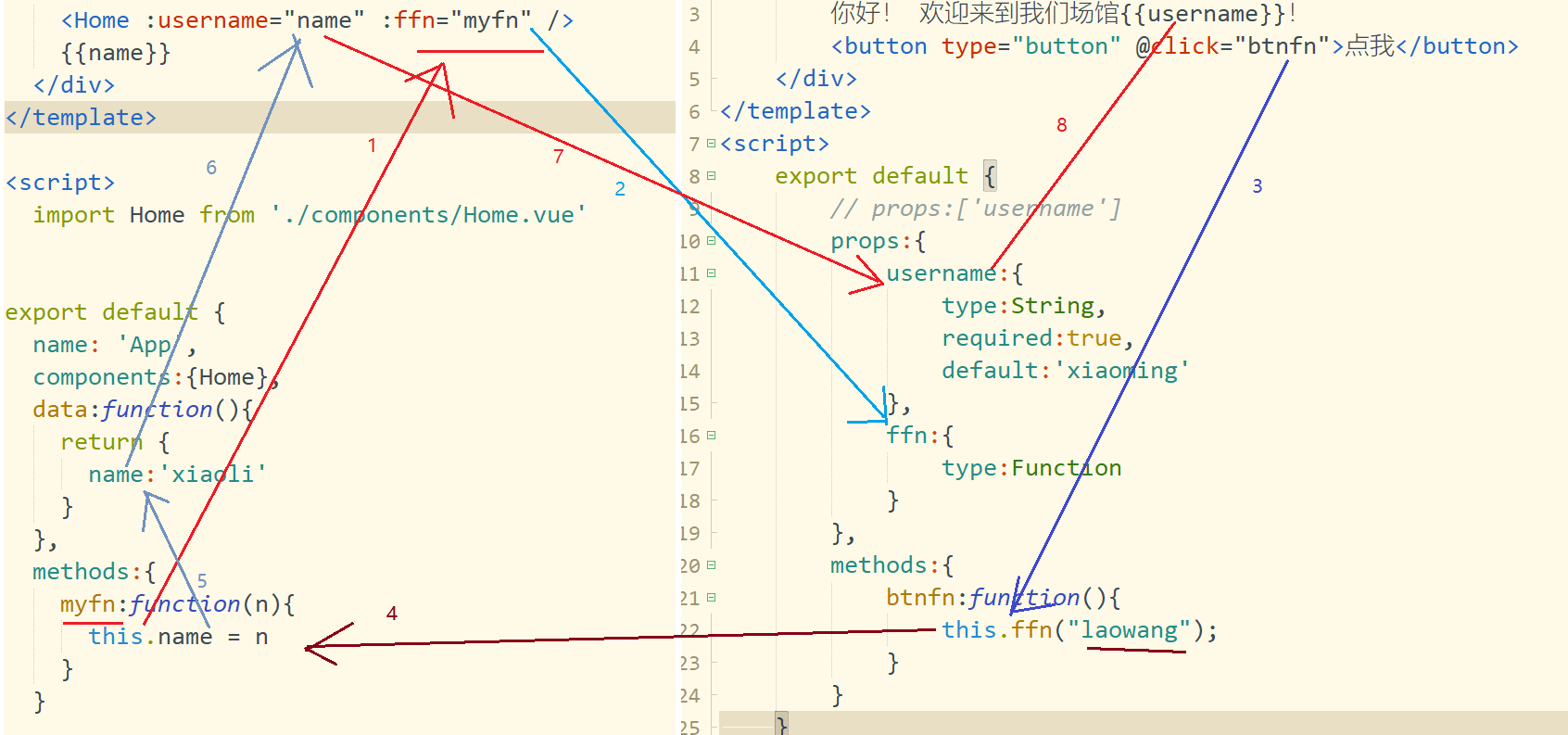
子组件：



父组件：



## 2）子传父



element ui