1 npm install -g live-server 用来起动本地服务 安装好之后，直接在项目里面运行live-server

2 npm init 初始化项目 设置一些基本信息  
  
v-if 判断是否徐渲染

v-else 要配合v-if一起使用，当v-if为false时，显示v-else

v-show 判断是否显示 默认display:none;

v-for (item,index) in items ｛｛item.age｝｝ v-for 里面必需要绑定:key值，默认可以绑定:key=”index”

v-text 和用字面量是同一个效果{{ message }} 主要区别 在于就是当我们网速很慢或者javascript出错时，会暴露我们的{{xxx}}; v-text会把html标签默认当string输出  
v-html 能输出html标签 不推荐使用v-html 不太安全，因为容易导致XSS攻击。  
v-on @click v-on:click

v-model   
.lazy 懒加载 （文本框中，失去焦点才渲染数据）  
.number输入数字才双向数据绑定 （默认如果是字符串的话，会失效）  
.trim去空格（删除前后空格）

v-bind 单个直接使用key,多个情况下用[key,key]，

v-pre直接原样输出，切断vue的双向绑定方法

v-cloak渲染完成后再显示

v-once只渲染一次，渲染成功后，为静态元素

computed 文档加载完后，执行方法

methods 默认不执行方法,只是实例化

## Vue.directive( id, [definition] ) 自定义指令

bind(只调用一次，指令第一次绑定到元素时调用)  
inserted（被绑定元素插入父节点时调用）  
update（指令所在组件的 VNode 及其子 VNode 更新之前）  
componentUpdated（指令所在组件的 VNode 及其子 VNode 全部更新后调用）  
unbind（解除事件绑定）



## Vue.extend( options ) 扩展实例构造器

先var 一个变量，然后把要写的div组件定入template，data里面传值，注意data是一个方  
法，方法里面直接return一个对象  


## Vue.set( target, key, value ) 全局操作

当你利用索引直接设置一个项时，vue不会为我们自动更新 当你修改数组的长度时，vue不会为我们自动更新 使用 Vue.set()触发视图更新



## Vue的生命周期（钩子函数）

console.log('1-beforeCreate 初始化之前');

console.log('2-created 创建完成');

console.log('3-beforeMount 挂载之前');

console.log('4-mounted 被挂载之后');

console.log('5-beforeUpdate 数据更新之前');

console.log('6-updated 数据更新之后');

console.log('7-activated');

console.log('8-deactivated');

console.log('9-beforeDestroy 销毁之前');

console.log('10-destroyed 销毁之后');



## Template模版

Template 模版的三种引用方式：

1. template: `<h3 style=”color: red”> 我是template模版</h3>` 注意``是~~不是单引号
2. <template id=”dd2”><h3 style=”color: red”>我是template标签模版</h3></template> template: ”#dd2”
3. <script type=”x-template”><h3 style=”color: red” id=”dd3”>我是script标签模版</h3></script>   
   template:”#dd3” 特点：script标签模版可以使用src引用

Template 全默认替换掉app里面的所有内容



## Component 初识组件

Vue.component({‘jspang’,

Template:`<div style=”color: red”>我是组件</div>`  
})

// 全局定义

Component:{“panda”,

Template:`<div style=”color: green”>我是组件</div>`  
}

// 局部组件



## Component 组件属性props

Props:[‘title’,likes]

注意属性命名，如用驼峰命名法，在标签上，应当换成-



## Component父子组件



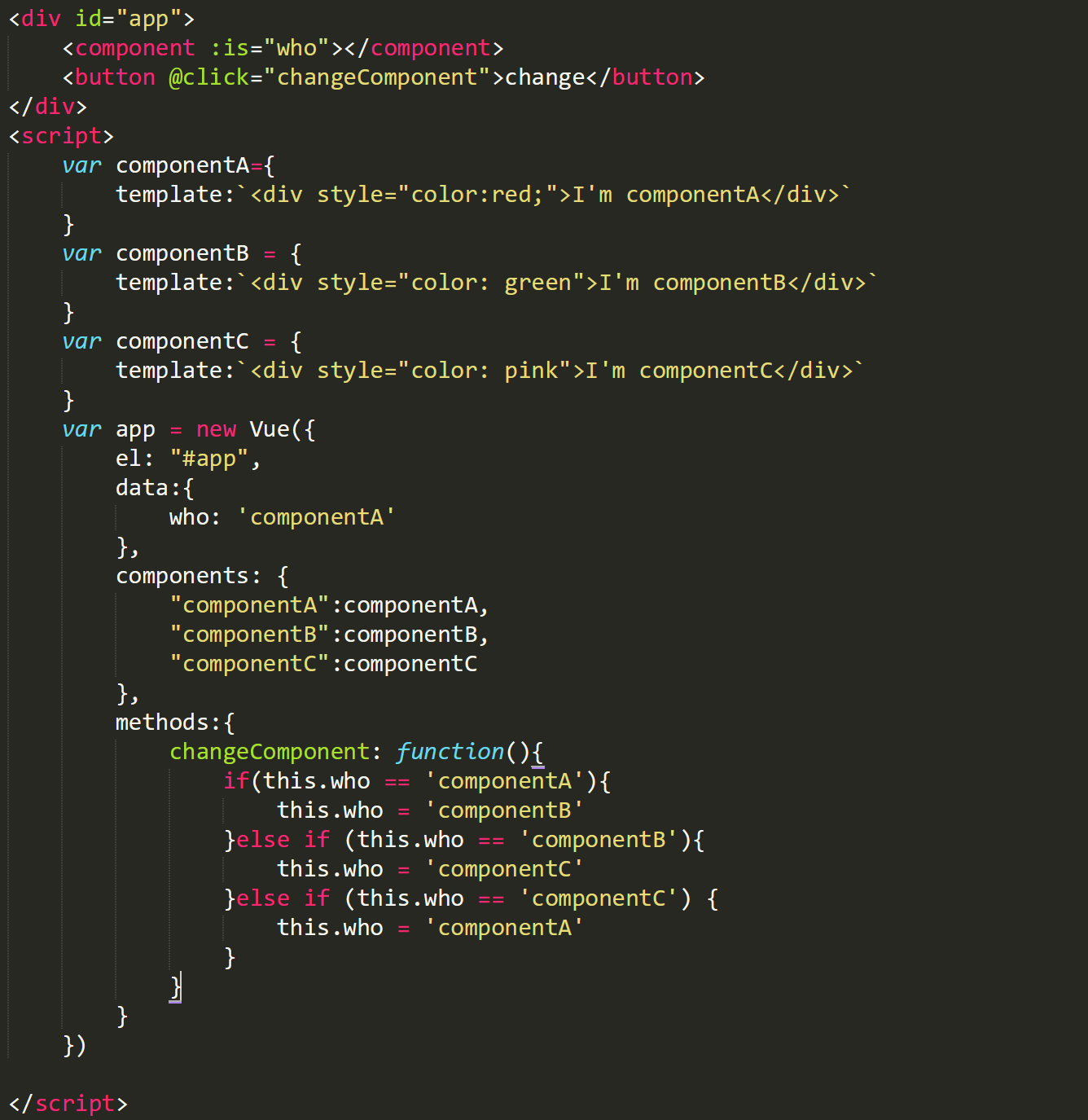
### 父传子

父传子组件要先通过父组件注册，然后子组件再注册后，最后通过父子组件往里传入



## Component 标签

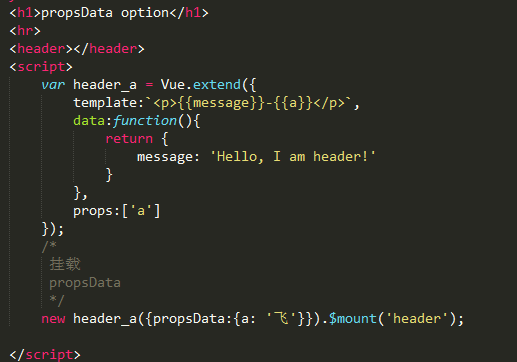
Vue有一个封闭好的component标签



## propsData option

类型：{ [key: string]: any }

限制：只用于 new 创建的实例中



## Computed 计算选项

{ [key: string]: Function | { get: Function, set: Function } }



## Methods 方法

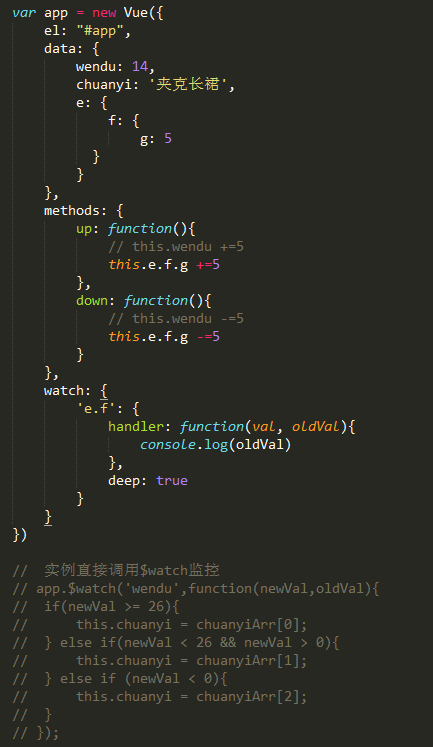
类型：{ [key: string]: Function }



## Watch

类型：{ [key: string]: string | Function | Object | Array }

监听引用类型时需要深入监听，添加deep: true;



## Mixins 混入

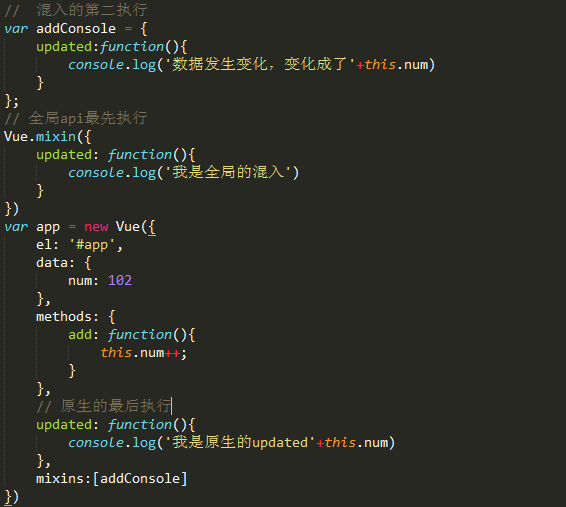
类型：Array<Object>

// 全局api最先执行 Vue.mixin({})

// 混入第二执行 mixins:[addConsole]

// 原生的最后执行 updated





## Extends

类型：Object | Function

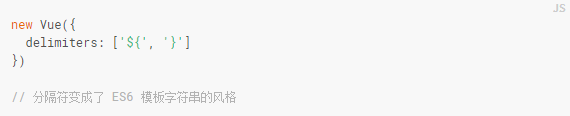
// 混入第二执行 mixins:[addConsole]

// 原生的最后执行 updated



## Delimiters

类型：Array<string> 改变纯文本插入分隔符。



## Example-01 实例属性

引入jquery，直接在mounted方法里调用jquery方法就行

外部调用vue里面的方法直接 app.add();



## Example-02 实例方法

$mount(); 挂载实例

$destroy(); 销毁实例

$forceUpdate(); 刷新方法

$nextTick(); 数据修改后



## Example-03 实例事件

$on() 绑定事件

$once() 只执行一次事件

$emit() 触发当前实例上的事件

$off() 删除事件



## Example-04 实例组件slot

String用于标记往哪个具名插槽中插入子组件内容

<div Slot=”key”>value</div>

<slot name=”key”></slot>

