Vue入门学习

# Vue加载流程图



# Vue数据双向绑定

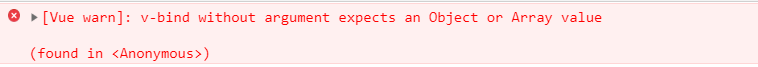
## 双花括号语法



## v-bind绑定值

Vue中不存在这种用法，v-bind用于元素的属性绑定

使用这种格式会报错



效果图如下所示

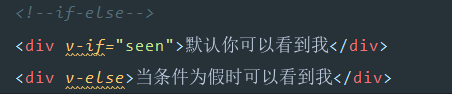


# 条件与循环

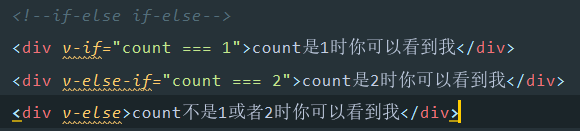
## if



## if-else

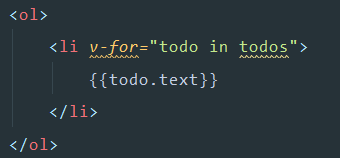


## if-else if-else

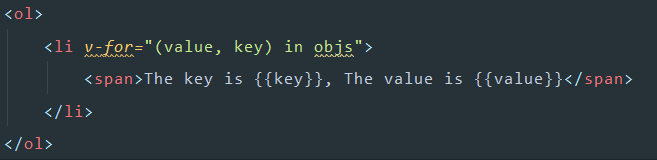


## for

### 遍历数组



### 遍历对象

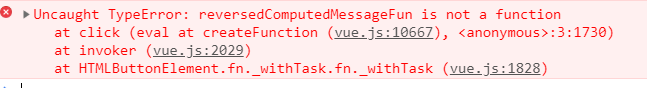


# 方法

## method



这里不能加括号即不能这样使用reversedMessageFun(),否则会报错



当方法有形参时可以添加括号例如reversedMessageFun(‘abc’)



## computed

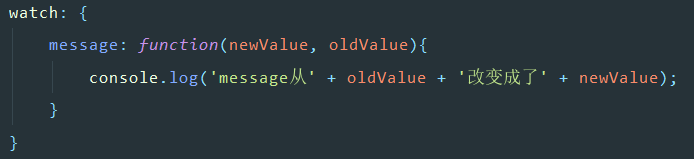


二者的区别是：method只要条用了这个方法则这个方法体里的内容就会被执行。而computed只有依赖的属性改变了这个方法才会被执行，并且不能作为click或其他事件的回调函数使用。

计算属性存在setter方法



## 侦听器的使用



# 组件

## 全局注册





两者的区别是my-component-head在页面中只能<my-component-head></ my-component-head>这样使用而MyComponentBody在页面中可以<MyComponentBody></MyComponentBody>这样使用也可以<my-component-body></my-component-body>这样使用

## 局部注册

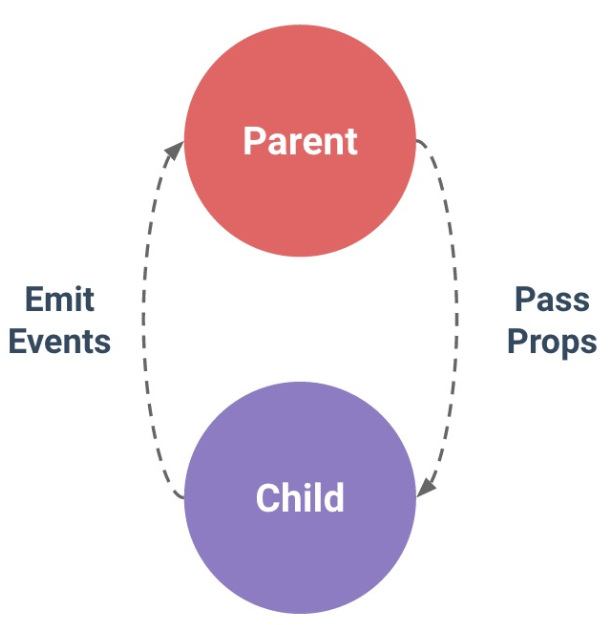


可以通过一个普通的JavaScript对象定义组件

## 根组件和普通组件的区别

除了普通组件没有el选项和data:{return {}}必须通过一个函数返回一个对象之外，其他的选项都一直例如data, computed, watch, methods以及生命钩子等。

## 组件之间的数据传递



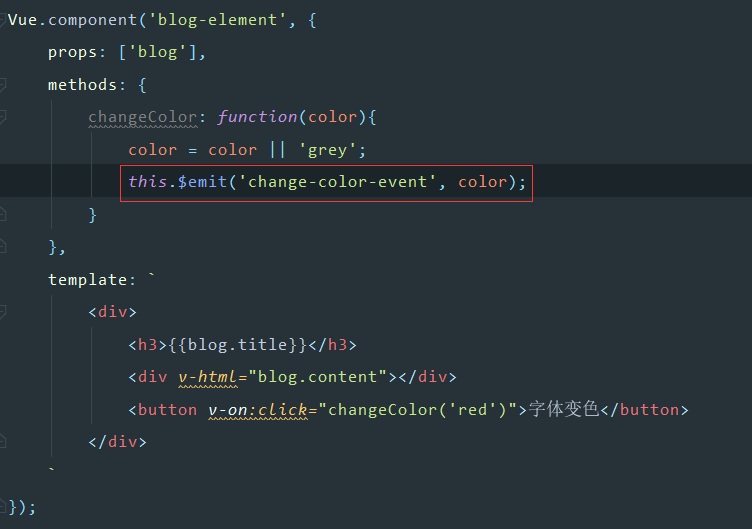
### 父组件->子组件

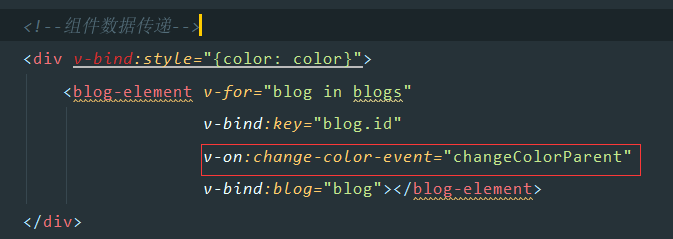
子组件

父组件

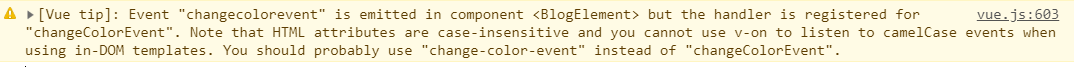
由此可见数据从父组件传递给子组件通过props属性声明

### 子组件->父组件

子组件

父组件

疑惑点：当我用驼峰的方式给事件命名时提示信息如下



而使用webpack搭建的vue项目则不存在这种问题

## 插槽slot

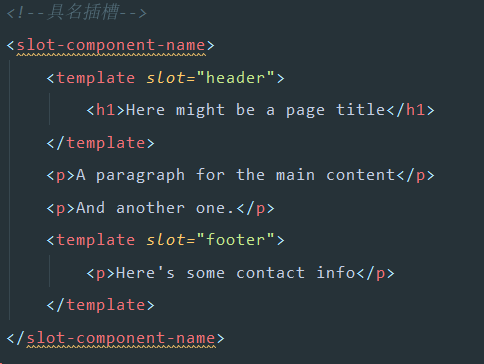
### 插槽内容



组件模板中至少需要存在一个slot

### 具名插槽





通过slot的name属性指定对应的html放置到哪个位置

### 编译作用域



插槽可以访问跟这个模板的其他地方相同的实例属性，但不能访问子组件的作用域

提示：父组件模板的所用东西都会在父级作用域内编译，子组件的所有东西都会在子级作用域内编译

# 指令

# 类绑定

# 样式绑定