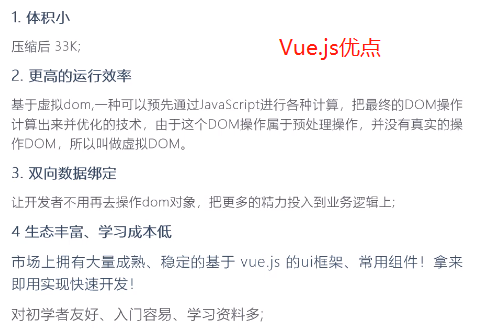
###### 简介、优点、应用场景、安装方式



渐进式：由浅入深、简单到复杂，这都不影响对这个框架的使用。





新手推荐使用script标签之间引入，对框架有一定了解后可使用命令行工具(CLI)进行创建，这样创建项目结构跟文件后缀都会发生改变

###### 第一个Vue应用



注：实例中没声明的数据不能在视图中使用

可以通过实例来更改变量数据，当值不存在时自动生成变量

eg: 在实例内另外生成一个name变量



若需要锁定数据属性，可通过Object.freeze()来进行，锁定后数据属性无法动态更新



###### Vue实例暴露的实例属性跟方法



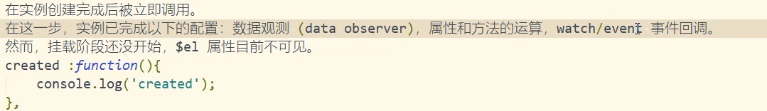
###### watch(监听的属性数据, function(新值, 旧值) {})方法



###### 实例生命周期钩子，在Vue初始化时进行自己的一些个人操作，生命周期钩子函数需要写到传进实例对象的属性中

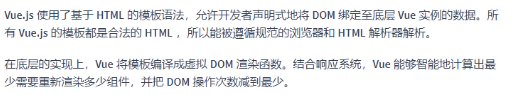








模板语法



###### 模板语法 - 插值 - 文本与v-once指令





###### 模板语法 - 插值 - 原始html与v-html指令，这里是在span标签内插入了一个div标签



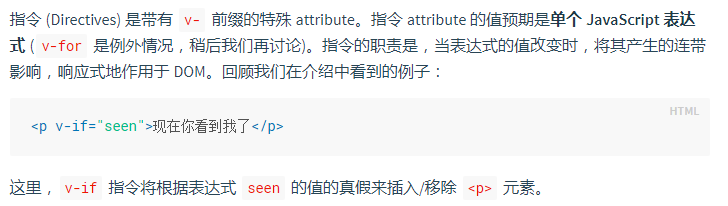
###### 模板语法 - 插值 - 属性与v-bind指令，这里改span标签绑定了一个class的属性，class的对应的属性数据为color它的值为font-color



###### 模板语法 - 插值 - 使用JavaScript表达式



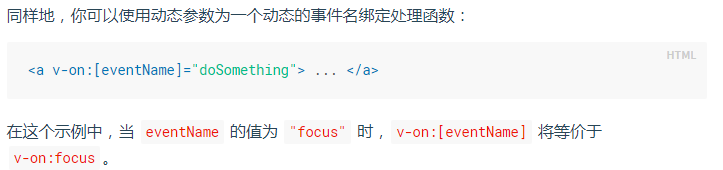
###### 模板语法 - 指令



###### 模板语法 - 指令 - 参数，能绑定标签的属性、标签的DOM事件



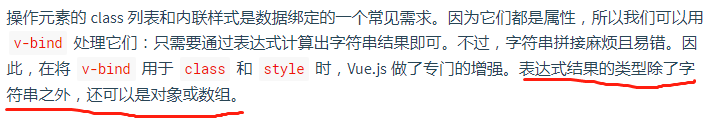
###### 模板语法 - 指令 - 动态参数，支持动态改变参数，标签属性或者标签DOM事件，参数不能使用引号和空格，不能使用非字符串值



###### 模板语法 - 指令 - 修饰符，以半角句号 . 指明的特殊后缀，表明一个指令以特殊的方式绑定



###### Class与Style绑定



###### Class与Style绑定 - 绑定HTML Class - 对象语法



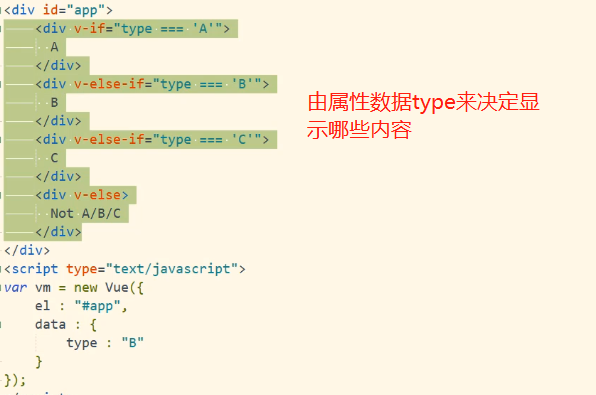
###### Class与Style绑定 - 绑定HTML Class - 数组语法



###### Class与Style绑定 - 绑定内联样式 - 对象语法



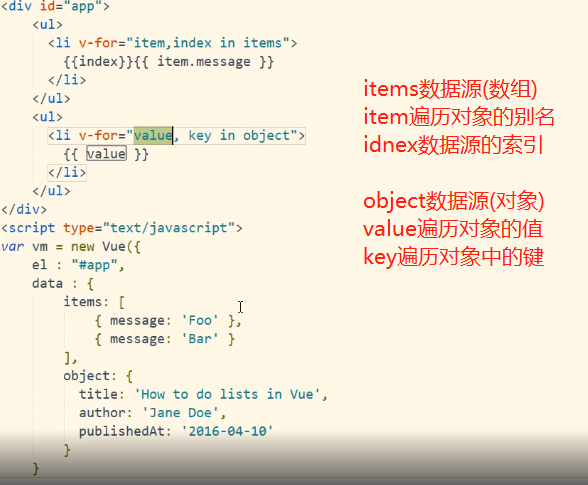
###### 条件渲染 - v-if



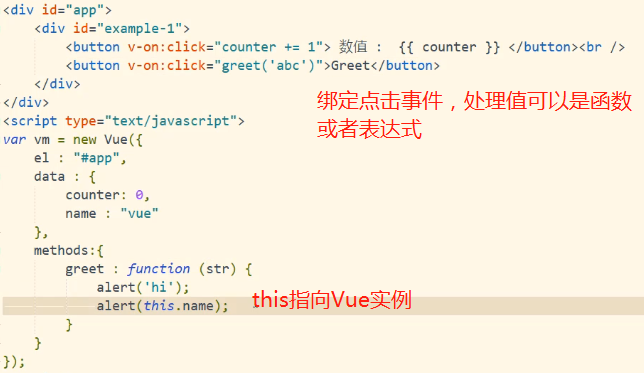
###### 条件渲染 - v-if与v-show



###### 列表渲染 - v-for，索引/键可以省略不要，在列表渲染后面加入:key属性是为了方便Vue更新数据项时查找，:key后面跟的值建议是字符串类型跟数值类型并且最好是唯一



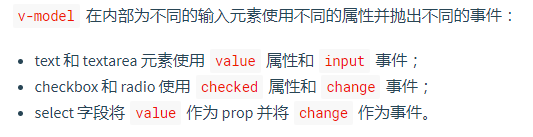
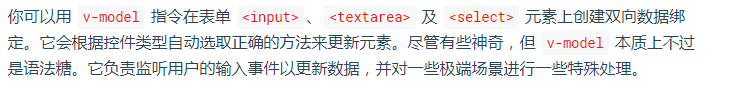
###### 事件处理 - 事件绑定 - 监听事件



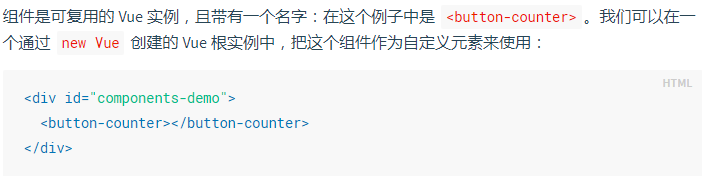
###### 事件处理 - 事件修饰符



###### 表单输入绑定



###### 组件基础





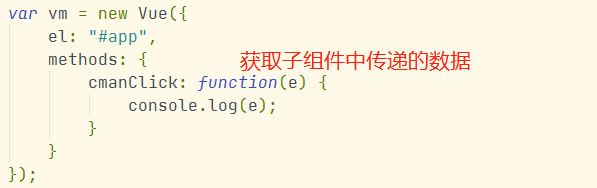
组件的data属性描述了模板的数据，它必须是个函数并需要return

props表明定义子组件的属性，并可以让你向子组件传递数据

template：一个组件有且只有一个根节点

###### 组件基础 - 子组件向父组件传递数据

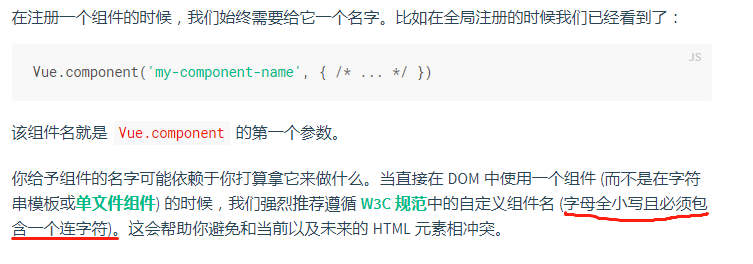




###### 组件基础 - 插槽 - slot标签



###### 组件注册

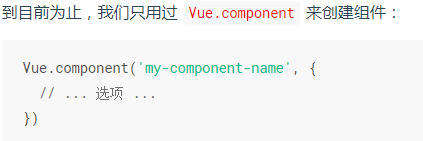


###### 组件注册 - 组件名大小写



###### 组件注册 - 全局注册

全局组件注册后可以用在任何新创建的Vue根实例(new Vue)创建的模板中，但是不管你有没有使用到该组件，它都会在你打包的时候出现



###### 组件注册 - 局部注册

在创建根实例(new Vue)传进的对象参数中写入component属性



###### 模块系统 - 在模块系统中局部注册



###### 单文件组件



###### 单文件组件 - 结构

由template、script、style组成



###### 单文件组件 - 使用

