学习网址：<https://cn.vuejs.org/>

2016年12月之前是1.0的版本

2.0和1.0版本的区别：

（1）2.0版本删除了系统自带的过滤器

vue.js简介：

vue是是一套构建用户界面的**渐进式和只关注视图层的框架。**vue.js不支持IE8及以下版

vue.js特点：

1. 易用

基于css、html和javasctipt。简而言之就是好学、门槛低

1. 灵活

vue一样是渐进式的技术栈（由易变难），能够和任何第三方库进行结合不会造成相互的影响，只关注视图层（简而言之就是只帮我们做渲染），采用的是传统的DOM模版

1. 性能

vue.js文件足够小，压缩后大概20KB

vue.js的使用：

1. 引入vue.js文件
2. 下载到本地引入（建议使用）
3. 绝对路径引用

<script src="https://unpkg.com/vue"></script>

1. 创建vue应用。一个DOM模版、一个Vue实例、再在实例配置对象中传一个data属性，组件也是放到DOM处进行渲染。如下：



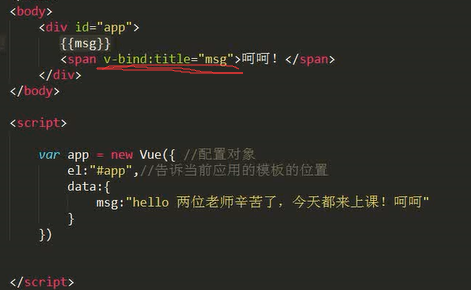
声明式渲染：如何在模版中使用对象中的数据

el：通过id把DOM模板和实例相关联

vue.js的指令：

vue.js的指令都带有前缀 v-，以表示它们是 Vue 提供的特殊属性，会在渲染的 DOM 上应用特殊的响应式行为（指令是用来帮我们做DOM操作的地方）

（1）v-bind：指令，将这个元素节点的 xxx属性和 Vue 实例的xxx属性保持一致”。（简而言之是将数据的变化映射到DOM的行为，想要使属性值是一个变量，就需要在属性名前面加v-bind前缀，如：v-bind:title=”message”）



上图解释：vue.js把data里面的属性全部挂载到实例app中，所以直接打点调用msg（app.msg）的属性。

（2）v-if条件指令（控制切换DOM节点显示与隐藏）

在DOM节点添加一个属性指令（v-if=”变量”），在实例data属性中添加此变量（变量:true），true表示显示，false表示隐藏。如下：



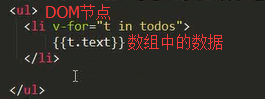
（3）v-for循环指令

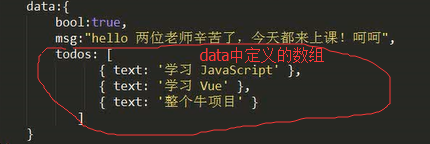
在data属性中定义一个数组todos→在DOM节点添加一个属性指令（if-for=”t in todos”）。如下：

t：表示数组的每一项

in：关键字

todos：所遍历的数组

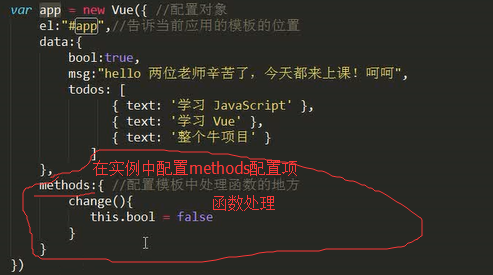




数据驱动：可以往todos中添加一组数据，app.todos.push(text:”学习”)，在页面中同样跟新显示数据，此时会显示4条相对应的数据

（4）v-on:绑定事件指令，处理用户输入

在DOM节点上绑定一个事件（v-on:click=”change”）→再在实例中配置methods配置项→在methods配置项中配置处理函数（methods配置项中是专门用来配置模板中处理函数的地方）

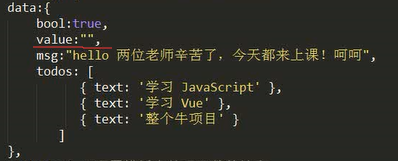


上图this指向的是当前实例。

（5）v-model指令，数据的双向绑定

在input节点上绑定指令（v-model=”value”）→在实例中的data配置项中配置此value属性（value:””），此时data中的value值就与input中的value值绑定。如下：

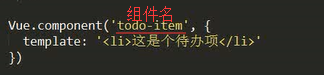




vue.js的组件构建：

最外层始终是一个实例，实例中告诉一个模板。模板中插入的全是组件。

组件的创建，如下：



vue周期函数

1. 使用方法将data配置项挂载到实例之前的beforeCreate函数
2. 实例创建好之后的created函数
3. 插入真实DOM节点之前的beforeMount函数
4. 渲染模板之后的mounted函数

1-4的周期函数是初始化的函数

1. 当有数据改变时，虚拟DOM重新渲染之前的beforeUpdate、updated函数
2. 实例或组件销毁时的beforeDestroy-destroyed函数，删除子组件和事件监听（app.$destroy()）

注意：带$的方法都是vue内部的方法

Vue模板的语法

1. 模板中插入文本，使用双大括号{{msg}}，可以在{{}}使用简单的js表达式，如+-\*/ 三目运算符
2. 纯HTML，使用 v-html 指令，.
3. 指令，指令就是对DOM进行操作

指令的参数，如事件指令v-on:后面所跟的事件类型（click、focus）就是指令的参数

指令的修饰符，如：.prevent 修饰符告诉 v-on 指令对于触发的事件调用 event.preventDefault()

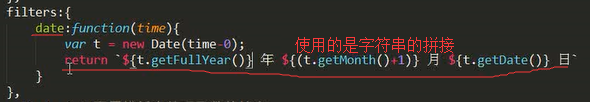


Vue的过滤器

当我们需要格式化显示文本的时候就要用过滤器（文本转换）

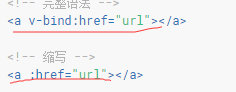
在data配置项中添加time属性（date:new Date().getTime()）→此时就需要格式化转换为我们常用的时间格式，在data配置项中创建filters属性，在次属性中创建一个data的过滤器，创建一个匿名函数（第一个参数是所操作的数据），如下：



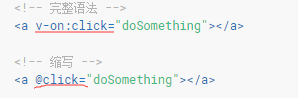


Vue的缩写

（1）v-bind属性的缩写



（2）v-on事件绑定的缩写



Vue的计算属性（重点）

计算属性出现的原因：由于在模板中使用太多逻辑会让代码变得难以维护，同时当你想监听多个数据变化的时候就需要使用计算属性

注意：计算属性计算好的值必须做return处理

使用：在实例中配置computed配置项computed:{}（computed是配置计算属性的），如下：

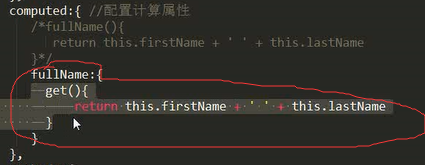


计算属性的特点：

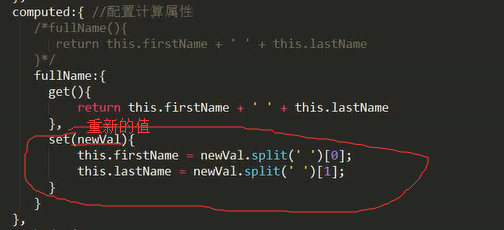
1. 会监听计算属性里面所依赖的任何属性（简而言之同时监听多个属性）
2. 计算属性会被缓存，当我们在DOM模板多次使用时不会再调用函数
3. 当监听依赖的属性发生变化的时候，计算属性的值会被重新计算并重新赋值

计算属性的get（取计算属性的值），如下：



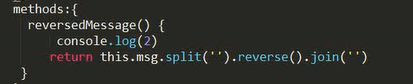


计算属性的set，当计算属性的值发生改变的时候执行一个函数，同时可以改变计算属性所依赖的数据值，如下：



函数调用的方法：



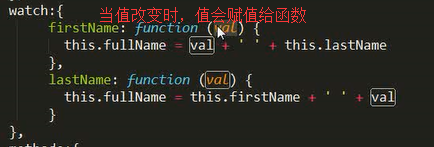


计算属性和函数调用的方法的区别：

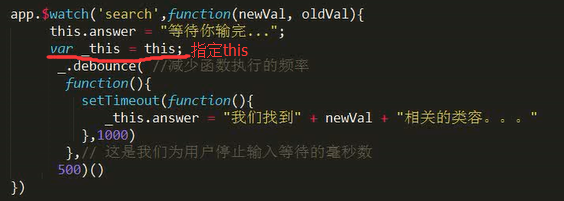
1. 计算属性只调用一次（缓存），函数调用的方法是调用多少次就是多少次（造成资源浪费）
2. 计算属性在数据改变时会从新更新，函数调用的方法在数据改变时无法从新更新

watch属性，当有异步操作的时候就需要用watch属性，首先需要引入lodash.js文件（lodash库是对象操作的拓展，是工具库），当input中的值改变的时候，对应的内容也会跟着改变。有两种方法，具体使用如下：

1. 在实例配置中配置watch属性，如下：



1. 用实例调$watch，首先需要引入，如下：



watch属性的缺点：一次只能监听一个数据的变化

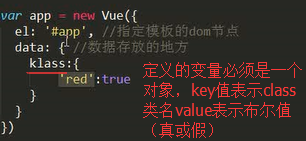
watch属性的优点：可以在watch函数中设置中间状态，可以在watch函数中作异步加载

vue的class（类名）和style（内联样式）的绑定

1、class（类名）的绑定语法，使用v-bind方法绑定：

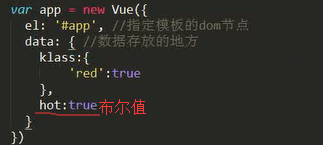
（1）直接变量语法，使用v-bing绑定变量v-bind:class=”klass”，这个变量在data配置项中必须是一个对象（对象的key值表示class类名value值表示一个变量（布尔值）），如下：





（2）对象语法 :class=”{}”，对象的key值表示class类名value值表示一个变量（布尔值），为真时类名会绑定在DOM节点上，假不会绑定。对象中可以使用三目运算，如下：





:class=”{}”可以和class=””共存。如下：



（3）数组语法，数组中每一项就是变量，变量对应的value值是class类名，数组中可以使用三目运算。如下：





数组中三目运算对应的就是类名，下图hot就是class类名。如下：



添加的内容：为标签添加class类还可以使用classList方法。classList是HTML5新增的API，使用如下：

document.body.classList.add(“red”)为body添加一个为red的class类名

document.body.classList.remove(“red”)删除body中为red的class类名

2、使用style内联样式绑定

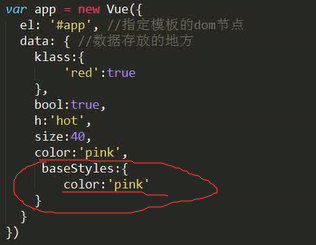
（1）对象语法v-bind:style=”{}”，对象中的key值表示style的属性，value值表示一个变量（变量对应的值是style属性的属性值（value））。如下：





（2）数组语法，数组中的每一项是一个变量，变量在data配置项中必须是一个对象（key值是style样式的属性value值是style样式属性的属性值）。如下：





注意：比重层次，v-bind的对象语法>v-bind的数组语法>v-bind的直接变量>内联style对象语法>内联style数组语法

vue条件渲染

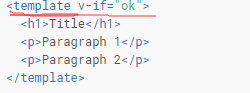
（1）v-if，显示或隐藏。下图ok为true时显示，false时隐藏



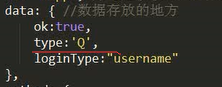
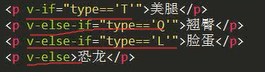
（2）v-if和v-else共同使用，二选一。并且v-else必须紧跟在v-if或者v-else-if的后面，不然会报错。下图ok为true时显示Yes，false时显示No



（3）当用v-if渲染多个的时候就需要<template>配合使用，<template>标签是vue保留的组件。下图ok为ture时全部显示，false时全部隐藏



（4）v-if、v-else-if、v-else的共同使用，多者选一。下图type属性的属性值与哪个相同就显示哪个



（5）当两个DOM节点的属性相同时，这时不想复用当中的值就需要添加不同的key属性（key管理可复用的元素），此时当中的值就不会备复用。如下：



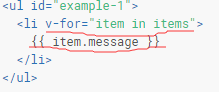
v-if与v-show的区别

v-if删除当前的DOM节点，v-if有更高的切换开销

v-show改变的是DOM节点的display，并不支持v-else、不支持template，v-show有更高的初始渲染开销

vue列表渲染

（1）使用v-for对数组进行遍历v-for=”i in arr”，i表示数组中的每一项，arr表示遍历的当前数组；v-for还接受第二个参数v-for=”(i,index) in arr”，i表示当前项的索引。如下：





（2）当需要渲染多个元素块的时候，就需要使用<template>标签



（3）使用v-for对对象进行遍历v-for=”value in obj”,value表示对象的属性值；提供第二个参数（表示对象的属性key），提供第三个参数（表示索引值）v-for=”(value,key,index) in obj”

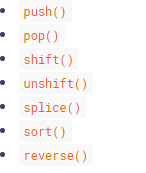
（4）当v-for和v-if处于同一节点（共同使用）时，v-for的优先级比v-if高（简而言之v-if会重复运行到每个v-for循环中），先进行遍历再进行v-if判断其布尔值，结构如下：



（5）key属性，为每一个DOM节点添加一个唯一标识。用变量来提供，如下：



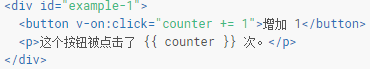
（6）数组跟新检测，Vue 包含一组观察数组的变异方法，当数组中有数据改变时，也会触发视图跟新。方法如下：

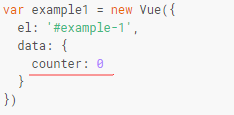
 

vue事件处理器

用 v-on 指令监听 DOM 事件来触发一些 JavaScript 代码

（1）监听事件，如下：

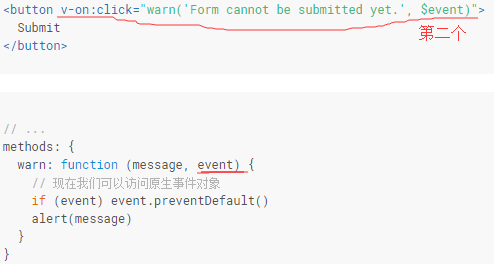




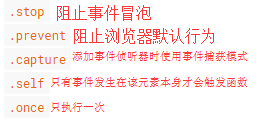
（2）方法事件处理器，在实例中配置methods配置项，在methods配置项配置模板处理函数，如果函数中没有传参数，event对象默认是第一个参数



当在函数中传有参数时，并想在函数中获得event对象，此时就要用变量$event传入，$event是第几个变量那么函数中第几个参数就表示event对象

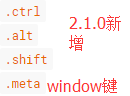
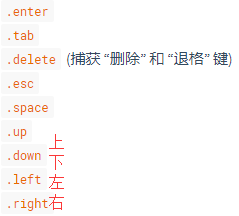


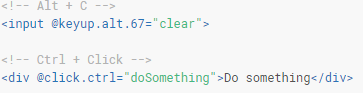
（3）事件修饰符



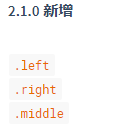
（4）键值修饰符







（5）鼠标按键修饰符



为什么在HTML中监听事件

1. 扫一眼 HTML 模板便能轻松定位在 JavaScript 代码里对应的方法
2. 无须在 JavaScript 里手动绑定事件，你的 ViewModel （实例）代码可以是非常纯粹的逻辑，和 DOM 完全解耦，更易于测试
3. 当一个 ViewModel 被销毁时，所有的事件处理器都会自动被删除。你无须担心如何自己清理它们。

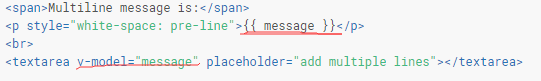
vue表单控件的绑定

使用v-model指令在表单控件元素上进行双向数据绑定

（1）文本



（2）多行文本



（3）复选框

单个复选框，v-model的变量是一个布尔值

多个复选框，v-model的变量相同，在实例的data配置项中变量是一个数组，数组的每一项对应的是复选框的value值。空数组表示都没有勾选，数组中的值等于复选框的value值表示勾选此复选框

（4）单选按钮，v-model的变量相同，value值不同。实例data配置项中变量的属性值是对应单选框的value，空表示都没选择，属性值等于value值时相对应的单选按钮选中

（5）选择列表，当没有value值时变量的属性值表示当前的innerHTML，有value值时表示当前的value值



（6）多选列表，绑定到一个数组

（7）修饰符

1）在默认情况下， v-model 在 input 事件中同步输入框的值与数据，但可以添加一个修饰符 lazy ，从而转变为在 change 事件中同步



2）自动将用户的输入值转为 Number 类型（如果原值的转换结果为 NaN 则返回原值），可以添加一个修饰符 number 给 v-model 来处理输入值



3）如果要自动过滤用户输入的首尾空格，可以添加 trim 修饰符到 v-model 上过滤输入

