Vue

## Vue的介绍

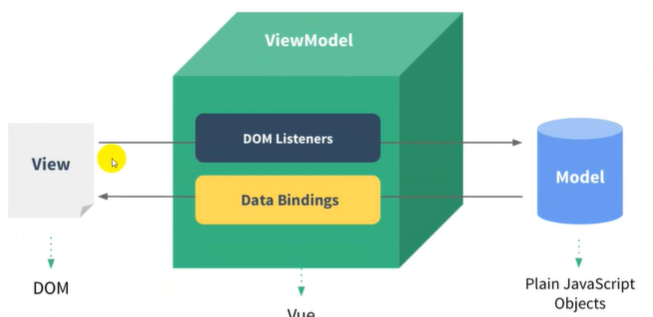
Vue是一个渐进式的js框架，只注重图层，结合了HTML+CSS+JS，非常的易用，并且有很好的生态系统，Vue体积很小，速度很快

SSM的整合，搞定了后端一套：增删改查

### MVVM

Vue技术时MVVM开发模式的实现者。

* Model层：模型层，在这里表示JavaScript对象
* View层：视图层，在这里表示DOM（HTML操作的元素）
* ViewModel：连接视图和数据的中间件，**Vue.js就是MVVM中的ViewModel层的实现者**



cdn内容分发网络：这是一种加速策略，能够从离自己最近的服务器上快速的获得外部的资源。

在MVVM架构中，是不允许数据和视图直接通信的，只能通过ViewModel来通信，而ViewModel就是定义了一个observer观察者

* ViewModel能观察到数据的变化，并对视图对应的内容进行更新
* ViewModel能够监听到视图的变化，并能够通知数据发生改变
* MVVM通过ViewModel实现了双向数据绑定

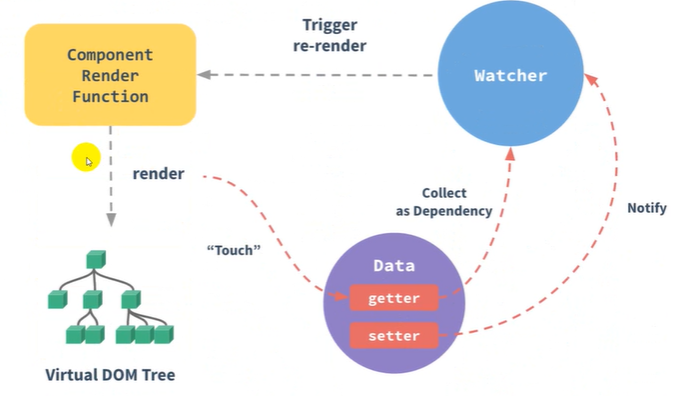
Vue.js就是一个MVVM的实现者，她的核心就是实现了DOM监听和数据绑定

其他MVVM实现者：

1. AngularJS
2. ReactJS
3. 微信小程序

### Vue的两大核心要素

#### 数据驱动



#### 组件化

1. 页面上每个独立的可交互区域视为一个组件
2. 每个组件对应一个工程目录，组件所需的各种资源在这个目录下就近维护
3. 页面不过是组件的容器，组件可以嵌套自由组合（复用）形成完整的页面

## 快速开始

1. 在页面引入Vue的js文件

注：cdn是一种加速策略，能够快速的提供js文件

1. 绑定Vue元素
2. 创建Vue对象，设计对象内容



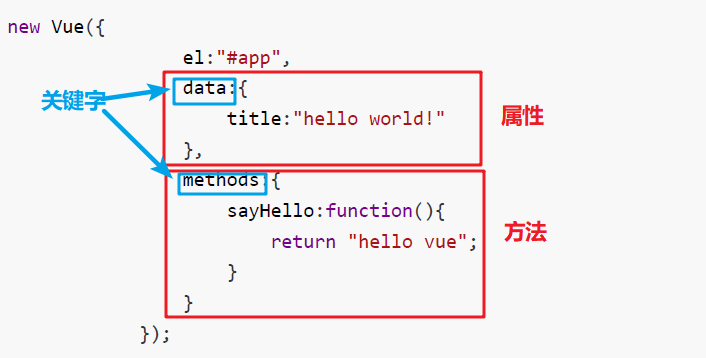
1. 页面元素中使用插值表达式来使用Vue中的内容



## 插（差）值表达式

插值表达式是用在HTML中被绑定元素中的，目的是通过插值表达式来获取Vue对象中的属性和方法

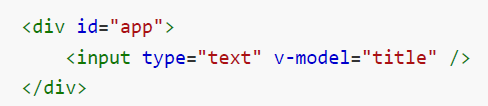
语法格式：{{vue内容}}。插值表达式不可以写在HTML的标签中作为属性值



### Vue中的关键字

#### MVVM双向数据绑定：v-model

用在标签内部，将标签的value值与Vue实例中的data属性值进行绑定



#### 事件绑定v-on



通过配合具体的事件名，来绑定Vue中定义的函数

补充知识：

1. 在响应函数里可以使用event内置的参数对象，该对象表示当前事件，可以通过event.target.value来获得当前事件对象的value的值
2. this的用法：this表示当前Vue对象，可以通过this来调用当前Vue对象的属性和方法
3. v-on还可以简写

**v-on:input=XXX可以替换成@input=XXX**

#### 属性绑定v-bind

HTML里面所有的属性都不能使用插值表达式，要想在HTML的元素的属性中使用Vue对象中的内容要使用v-bind进行属性绑定。

**v-bind也可以简写为冒号**：

<a v-bind:href="link">baidu</a>

可以缩写成 冒号

<a **:**href="link">baidu</a>

#### v-once指令

指明此元素的数据只出现一次，**数据内容的修改不影响此元素**

<p v-once>{{titile}}</p>

#### v-html指令和v-text指令

v-html类似于innerHTML，将Vue中的属性的值作为HTML的元素来使用；v-text将Vue中的属性的值作为纯文本内容输出

## 事件

### 事件绑定

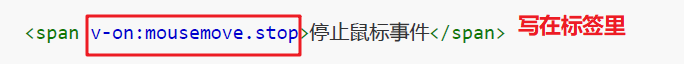
例1：<p v-on:mousemove="mo">mooooooo</p>

### 参数传递



### 停止鼠标事件





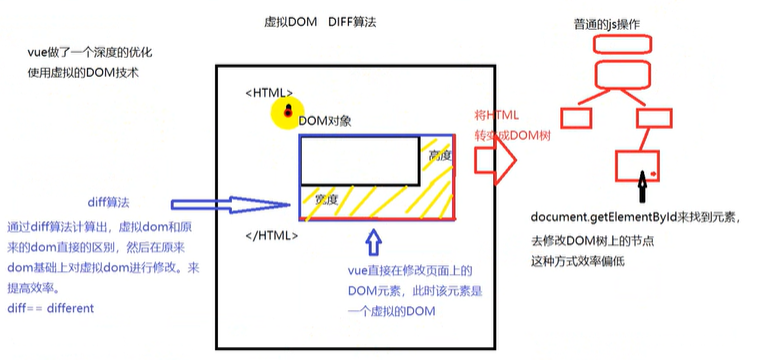
### 事件修饰符

@clicl.stop阻止点击事件传播

@mousemove.stop阻止鼠标移动事件

@keyup.space空格键弹起时

## Vue改变内容—虚拟DOM和diff算法



直接在DOM上修改需要通过getElementById找到元素然后修改DOM树上的节点，效率低，Vue直接修改页面上的DOM，通过diff算法得出新旧DOM区别，然后在原来基础上进行修改

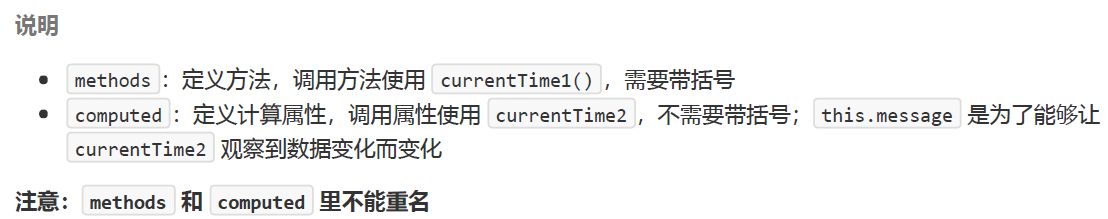
### 计算属性：computed（面试重点）

#### 什么是计算属性

计算属性的重点突出在 属性 两个字上（属性是名词），首先它是个 属性 其次这个属性有 计算 的能力（计算是动词），这里的 计算 就是个函数；简单点说，**它就是一个能够将计算结果缓存起来的属性（将行为转化成了静**

**态的属性）**，仅此而已；

#### 计算属性与方法的区别

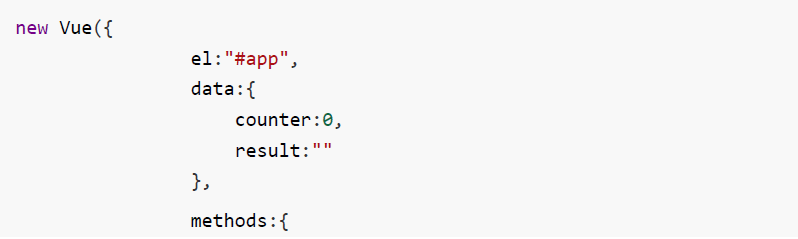


结论：

调用方法时，每次都需要进行计算，既然有计算过程则必定产生系统开销，那如果这个结果是不经常变化的呢？此时就可以考虑将这个结果缓存起来，采用计算属性可以很方便的做到这一点；**计算属性的主要特性就是为了将不经常变化的计算结果进行缓存，以节约我们的系统开销**

### Watch的用法：监控

当属性值变化时，该函数就会自动被调用。调用时可以接收两个参数，第一个是属性改变后的值，第二个是属性改变前的值





## Vue改变样式

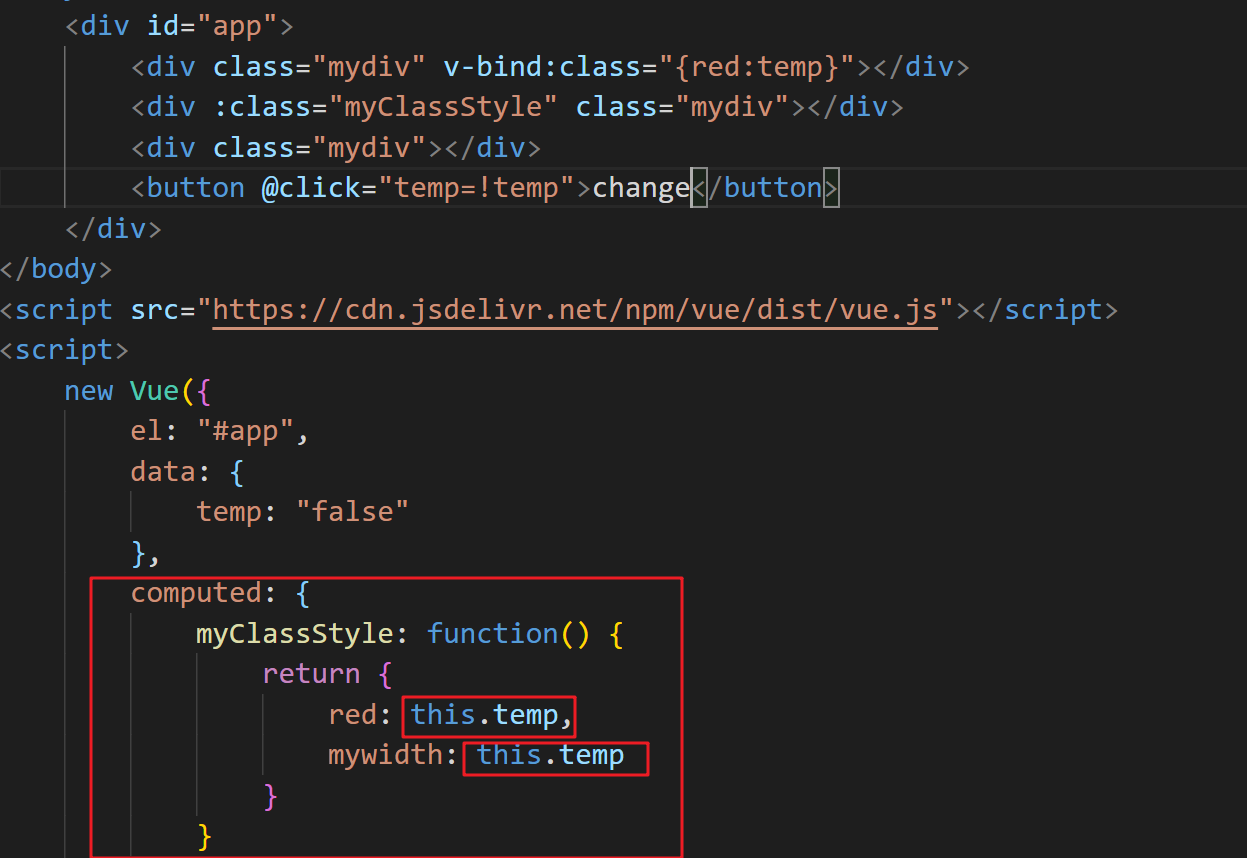
### Class的动态绑定

1. 通过给HTML元素的class属性绑定Vue中的属性值，得到样式的动态绑定



### 加入computed

通过computed返回一个对象，对象里放着多个键值对



### 多个样式的操作

多个样式的组合用数组，不能写两个:class

.red{background-color: red;color: white;}

<div class="demo" :class="[color,{red:attachRed}]">hahaha</div>

### 通过style设置样式

<div class="demo" :style="{backgroundColor:color}"></div>

设置div的style属性的值，style里放json对象，**键是驼峰式写法**，**值是变量**color（Vue中的属性值）

Style中引用了Vue中的内容，因此是一个键值对，所以需要大括号，对象的键不能出现连接符-。

### 通过设置style属性的多个样式



myStyle是computed里的属性

## Vue中的语句

### 分支语句

* v-if：<div v-if=”temp”>+++</div>当temp中的值是true时，div中的内容才会显示
* v-else:v-if的对立面
* v-else-if:在v-else里再嵌套一个if
* v-show: V-show=”布尔值变量”是true时，显示内容，是false时不显示，但是v-show改变的是元素的样式，不显示内容时样式是：display=none，而v-if是直接让元素消失和直接添加元素，效率上v-show更高
* template模板标签的使用，可以使用该标签配合v-if实现多个元素一起出现或一起消失的效果，但是template不能和v-show一起使用

### 循环语句

v-for：

<li v-for="a **in** args">

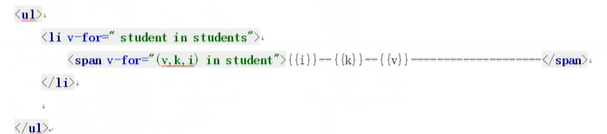
{{a}}

</li>









## 1

### 2

#### 3

##### 4

###### 5