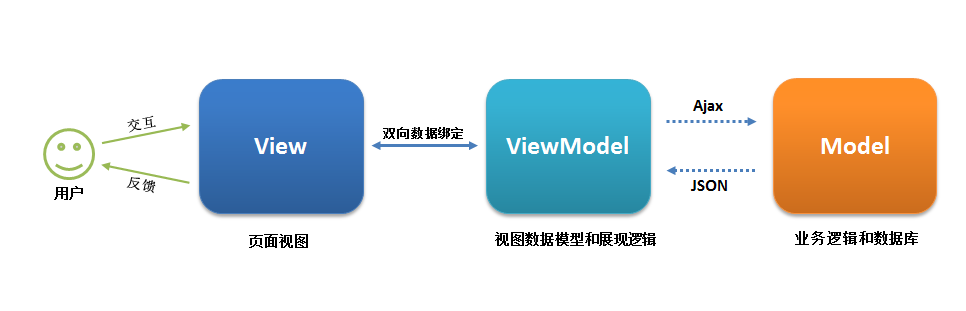
# 一 Vue介绍

## 1 什么是Vue.js

Vue 是目前很火的一个前端框架，用于构建用户界面的框架，只关注视图层，它不仅易于上手，还便于与第三方库或既有项目整合。

## 2 MVVM

Vue使用MVVM架构，将前端的视图层，分为三部分 Model, View , ViewModel



MVVM 的核心是 ViewModel 层，它就像是一个中转站（value converter），负责转换 Model 中的数据对象来让数据变得更容易管理和使用，该层向上与视图层进行双向数据绑定，向下与 Model 层通过接口请求进行数据交互，起呈上启下作用。

# 二 第一个例子

## 1 引入vue

<script type="text/javascript" src="vue.min.js" ></script>

## 2 html中用法

{{}}插值表达式

<div id="app">

<h1>{{message}}</h1> <!-- 文本插值， 获取data对象中数据 -->

<h1>url : {{url}}</h1>

</div>

## 3 创建vue实例对象

<script type="text/javascript">

// 创建vue对象

var vue = new Vue({

el:"#app", // 关联元素

data:{ // 设置el对应的元素内需要用到的数据

message:"hello world"

}

})

</script>

# 三 Vue基本语法

## 1 v-text

设置标签的文本内容，会覆盖原来的内容

与{{}}插值表达式的不同：

插值表达式只会替换自己占的位置的内容

## 2 v-bind

用于绑定属性的指令，不能实现双向绑定

该指令可以简写为 :绑定的属性

<h1 v-bind:title="mytitle">{{msg}}</h1>

<h1 :title="mytitle">{{msg}}</h1>

## 3 v-on

事件绑定机制

可以简写为 @事件

<button v-on:click="start">开始</button>

<button @click="start">开始</button>

事件中触发的函数，在vue实例中的methods中进行定义，可以按照如下两种方式进行定义：

<script type="text/javascript">

var vue = new Vue({

el:"#app",

data:{

msg:"houhou",

mytitle:"haha",

},

methods:{

start: function(){

// this 表示vue对象

this.msg = "123";

alert("start")

},

stop(){

}

}

})

</script>

事件修饰符

.stop 阻止冒泡

.prevent 阻止默认行为

.capture 实现捕获触发事件的机制

.self 实现只有点击当前元素时候，才会触发事件处理函数，另外只会阻止自己身上冒泡行为的触发

.once 只触发一次事件处理函数

例子：周期性显示当前时间

## 4 v-model

可以实现表单元素和数据的双向数据绑定

<input type="text" v-model="msg">

注意：针对表单元素

例子：加法运算处理

## 5 v-if

条件判断指令，还有v-else

<p v-if="flag">haha</p>

<p v-if="flag">heihei</p>

类似的，还有v-show

v-if：每次都会重新删除或创建元素

v-show：只是设置了元素的 display:none样式

## 6 v-for

循环指令

遍历普通数据

<ul>

<li v-for="item in arr">{{item}}</li>

</ul>

遍历列表中对象数据

<ol>

<li v-for="(item,i) in list">{{item.id}}-{{item.name}}</li>

</ol>

遍历对象数据

<ol>

<li v-for="(v,k) in user">{{k}}-{{v}}</li>

</ol>

例子：向表格中动态添加数据

## 7 生命周期

beforeCreate： 执行的时候，data和methods中的数据都还没有没初始化

created：data 和 methods 都已经被初始化

beforeMount：模板已经在内存中编辑完成了，但是尚未把模板渲染到页面中

mounted：内存中的模板，已经真实的挂载到了页面中，用户已经可以看到渲染好的页面。页面加载完成，执行的操作放在该函数中

beforeUpdate： 页面中显示原来的数据，但是此时data中数据是最新的，页面尚未和最新的数据保持同步

Updated： 页面和 data 数据已经保持同步了，都是最新的