# 第一部分 Vue入门篇

## Vue起步知识

儿童相见不相识，笑问客从何处来？这是怎样的一种尴尬啊！如果学习了本章，对Vue会有个感性认识，避免对大名鼎鼎的Vue产生不相识的尴尬。

### 1.1 从Hello World开始

千里之行，始于第一行。先不介绍Vue，直接来一个Vue的Hello World页面。这个过程很简单，只需要两步，**不能跳过去，因为下面的学习需要这么一个简易的开发环境来写代码**。

（1）从Vue官网下载vue.js，官网地址：<https://cn.vuejs.org/>。如图1.1-1：



图1.1-1

下载开发版本后，再新建一个lesson1项目，可以用WebStorm，HBuilder，VSCode等编辑器来开发，推荐使用JetBrains公司的WebStorm开发工具。把下载后的vue.js文件拷贝到项目的static目录，目录结构如图1.1-2：

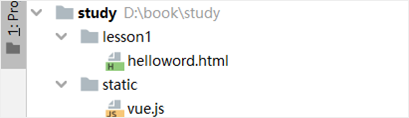


图1.1-2

（2）在lesson1目录下，新建helloworld.html文件，输入如下代码：

<!DOCTYPE html>  
<html>  
 <head>  
 <meta charset**="utf-8"**>  
 <title></title>  
 </head>  
 <body>  
 <div id**="app"**>  
 <p >**我的名字是：{{name}}**</p>  
 <p >**我的年龄是：{{age}}**</p>  
 </div>  
 <!--vue.js的路径-->  
 <script src**="../static/vue.js"**></script>  
 <script>  
 var *model*={  
 name:'Vue',  
 age:4  
 }  
 var *app*=new Vue({  
 el:"#app",  
 data:*model* })  
 </script>  
 </body>  
</html>

注意vue.js文件路径和你的本项目路径一致。将helloworld.html拖入到浏览器，渲染结果如图1.1-3：

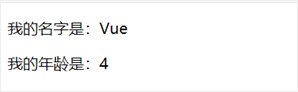


图1.1-3

上面的代码是什么逻辑呢？其实就是MVVM，是Model-View-ViewModel的简写，即模型-视图-视图模型。

### 1.2 Vue的核心概念MVVM

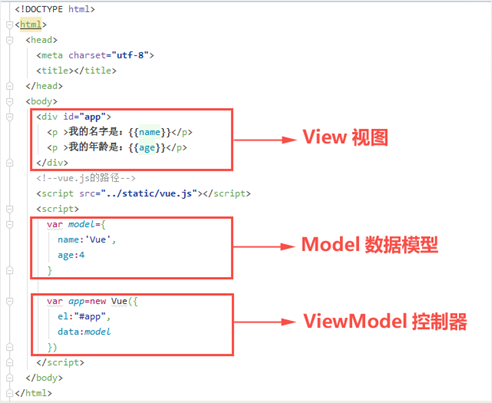
Vue的核心概念是MVVM，即View、Model和ViewModel，ViewModel是View和Model的桥梁。

视图：View

模型：Model

视图模型：ViewModel

上面的helloworld.html代码，体现的就是MVVM这种思路，如图1.1-4所示：

图1.1-4

三者之间的关系是：

View在ViewModel这个桥梁上利用DOM-listener事件绑定监听事件，将DOM节点上发生的事件传给Model，Model在ViewModel这个桥梁上利用Data-bindings数据绑定事件，将Model数据发生改变的事件传给View，这样就实现了视图层和数据层双向数据绑定。原理图如图1.1-5：

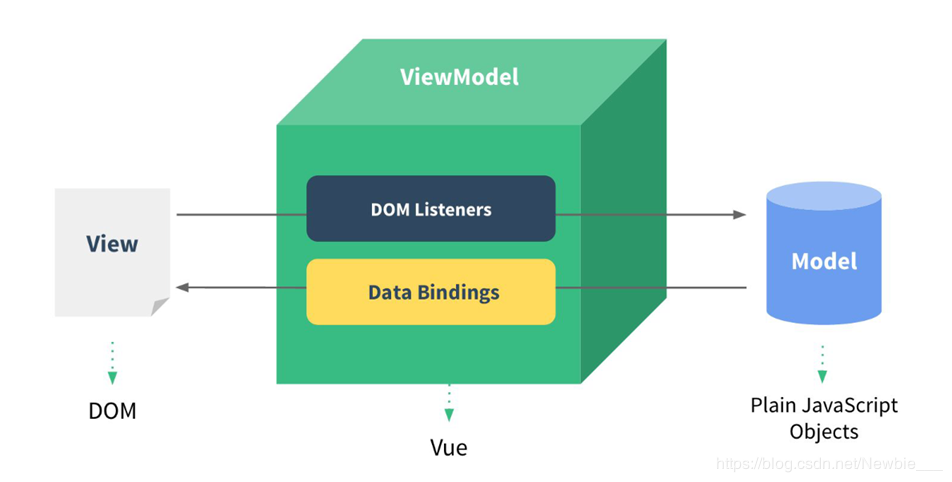


图1.1-5

#### 1.2.1 Vue的模型

模型就是数据模型，是在js里定义的，先看一下下面的代码，主要看<script>部分，这是一个最精简的Vue代码。

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset**="utf-8"**>  
 <title></title>  
 <script src**="../static/vue.js"**></script>  
</head>  
<body>  
<script>  
 var *obj*={msg:'hello'};  
 var *vm*=new Vue({  
 data:*obj* })  
</script>  
</body>  
</html>

上面的obj是数据模型，vm是Vue的实例，就是一个ViewModel，如何访问Vue实例的数据msg呢？通过这种形式vm.data.msg访问是错误的，正确访问数据的方式是vm.msg。通过数据模型对象也能访问，下面的条件成立：

*vm*.msg === *obj*.msg

实例初始化里有data，不能通过.data访问，总感觉不爽。其实Vue里有这个$data对象，作为实例一个属性，可以访问具体的数据内容，可以按照如下写法访问：

*vm*.$data.msg===*obj*.msg

#### 1.2.2 Vue的实例方法

Vue的实例就是类实例，面向对象中的类包含属性和方法，属性是在data里定义，实例方法是在methods里定义。代码清单如下：

<body>  
<div id**="app"**>  
 <button type**="button"** v-on:click**="sayHello"**>**点击**</button>  
</div>  
<script>  
 var *vm*=new Vue({  
 el:'#app',  
 data:{msg:'hello'},  
 methods:{  
 sayHello:function(){  
 *console*.log('hello function')  
 }  
 }  
 })  
</script>  
</body>

Vue的实例方法是在methods对象里定义，和js方法的应用是一样的，需要注意的是方法的写法，正确的方法写法有两种：

（1）sayHello:function(){…}

（2）sayHello(){…}

错误的写法：

（3）function sayHello(){…}