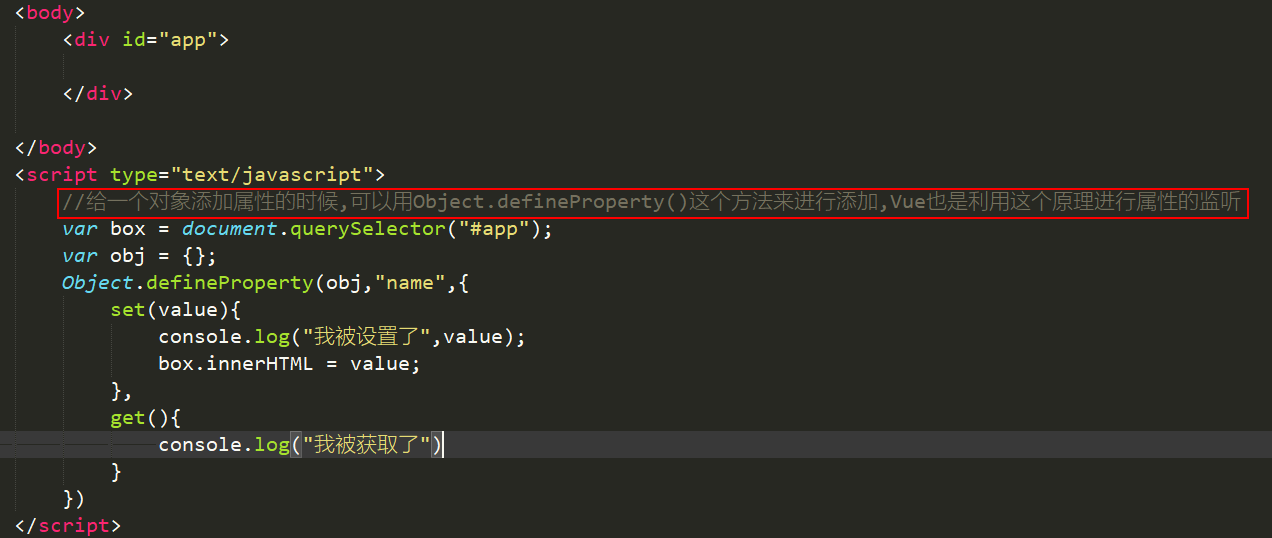
VUE从入门到精通

# VUE原理

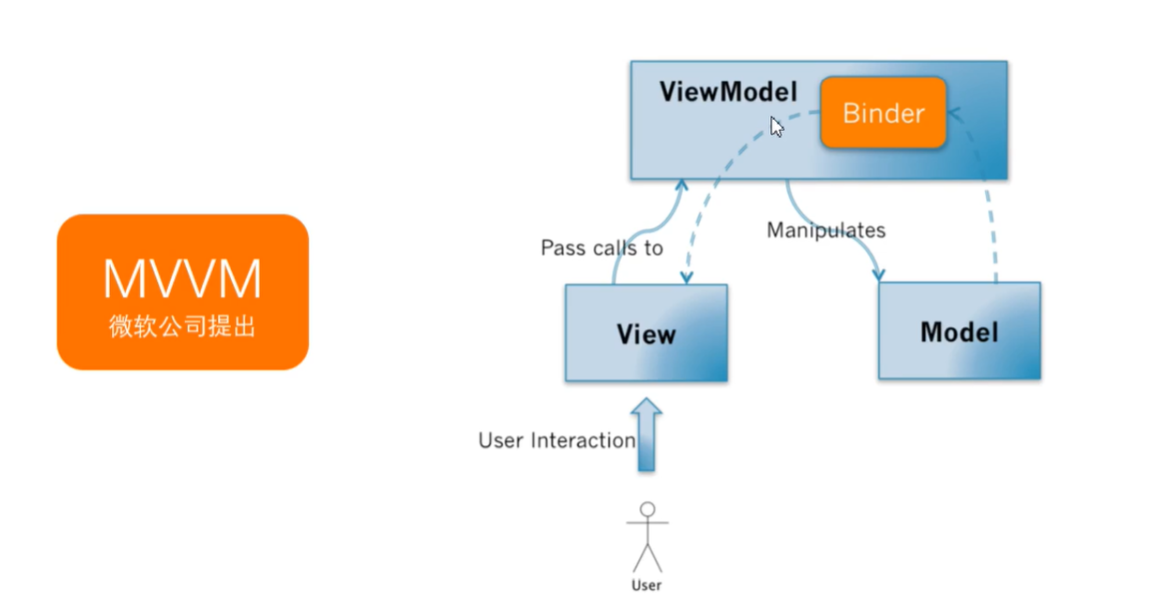
给一个对象添加属性的时候,可以用Object.defineProperty()这个方法来进行添加,Vue也是利用这个原理进行属性的监听





# MVVM

Vue就是基于MVVM模式的一种框架，中间的vm对视图（View）和Module（模型）进行双向数据绑定



# JS数组方法

## Splice()

## Push()

## Unshift()

## Shift()

## Filter()

对数组进行过滤

arr = ["aaa","bbb","ccc","ddd","add","cee","eee"]

基本用法：arr.filter(function(){}) 参数为一个回调函数

arr.filter = (function(item){

return item.includes(“a”);

})

以上可以使用箭头函数简写为：arr.filter(item=>item.includes(“a”));

# 事件绑定注意

在事件绑定中，调用方法时加括号和不加括号的区别：当加括号时候，Vue会把操作权交给你，所以这时候想在方法中获取事件对象如果没传值得话就会undefine，当不加括号的时候，Vue认为自己该操作，所以就会把事件对象当做默认的一个参数传到被调用的方法中

|  |
| --- |
| <body>  <div id="app">  <input type="text" @input="handleInput"/>  <button @click="handleAdd()">添加</button>  </div>  </body>  <script type="text/javascript">  var vm = new Vue({  el:"#app",  data:{},  methods:{  handleAdd:function(evt){  console.log(111,evt); //打印结果：111 undifine  },  handleInput:function(evt){  console.log(2222222, evt); //打印结果：111 InputEvent对象  }  }  })  </script> |

# Class属性为对象或数组

|  |
| --- |
| <style>  .aa{color: red;}  .bb{font-size: 40px;}  .cc{font-weight: bold;}  </style>  </head>  <body>  <div id="app">  <div :class="classArr">动态切换class-对象写法</div>  <div :class="classObj">动态切换class-对象写法</div>  </div>    </body>  <script type="text/javascript">  var vm = new Vue({  el:"#app",  data:{  classObj:{  "aa":true,  "bb":true,  "cc":false //true为添加该属性，false为不添加  },  classArr:["aa","bb","cc"] //只要在数组中的值都会被添加  }  })  </script> |

# v-for知识

## 对象遍历

用v-for也可以进行对象的遍历，v-for=” value,key,index in obj”

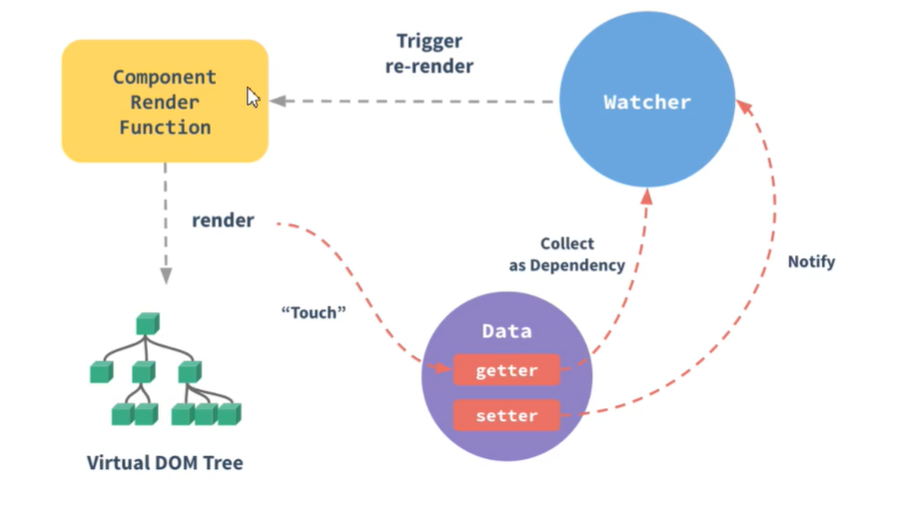
其中，遍历出来的item是obj的value，index是obj的key

数组的本质也是对象，只不过item是每一项，index为索引值

## key值

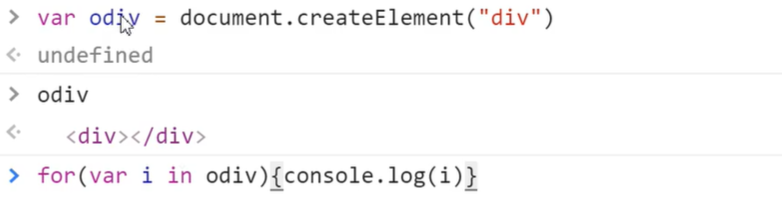
作用：跟踪每个节点的身份，从而重用和重新排序现有元素

理想的key值是每项都有的且唯一的id。(Item.id)



设置虚拟dom是为了能够在对比之后找到哪些节点是可以复用的，而设置虚拟dom又不足以找到快速对比的方法，所以在长列表中必须设置一个有效的key值。这样通过一个虚拟的dom和key值的有效设定才能够用最快速的方法找到哪些节点需要更新，哪些节点需要删除，从而进行下一步真实dom的修改。这也是VUE为什么渲染的这么快，为什么能提高性能的原因。

为什么不用真实dom进行操作呢？原因是里面的东西太多！做如下实验：



# 数组更新检测

* 使用以下方法操作数组，可以检测变动

push()、pop()、shift()、unshift()、splice()、sort()、reverse()

* 无法检测变动的方法，解决的方法时用新数组替换旧数组

filter()、concat()、slice()、map()

* 无法检测以下变动的数组

Vm.items[indexOfItem] = newValue //vm.dataList[0] = “kerwin”

解决方法：

1、splice方法 arr.splice(0,1,”kerwin”) //将数组的第一个值改为kerwin

2、Vue.set(example1.items, indexOfItem, newValue)