

1.vue语法模板:

```
<div id="app">
  {{message}}
</div>
<script>
  const app =new Vue({
    el:'#app',
    data:{
      message: '你好哈',
    },
  })
</script>
```

2.v-for: `<li v-for="item in movies">{{item }}`

3.计数器案例:

```

<div id="app">
  <h2>当前计数:{{counter}}</h2>
  <!-- <button v-on:click="counter++">+</button>
  <button v-on:click="counter--">-</button> -->
  <button v-on:click="add">+</button>
  <button @click="sub">-</button>
</div>
<script>
  const obj = {
    counter:0,
    message:'abc'
  }
  const app = new Vue({
    el:'#app',
    data:obj,
    methods: {
      add:function(){
        console.log('add被执行')
        this.counter++
      },
      sub:function(){
        console.log('sub被执行')
        this.counter--
      }
    },
    created:function(){
      console.log('created');
    },
    mounted:function() {
      console.log('mounted');
    },
  })
</script>

```

4.Vue中的MVVM(View Model VueModel)

View层:

- 视图层

- 在我们前端开发中，通常就是DOM层
- 主要的作用是给用户展示各种信息

Model层:

- 数据层
- 数据可能是我们固定的死数据，更多的是来自我们服务器，从网络上请求下来的数据。
- 在我们计数器的案例中，就是后面抽取出来的obj，当然，里面的数据可能没这么简单。

VueModel层:

- 视图模型层
- 视图模型层是View和Model沟通的桥梁
- 一方面它实现了Data Binding,也就是数据绑定，将Model的改变实时的反应到View中
- 另一方面它实现了DOM Listener,也就是DOM监听，当DOM发生一些事件(点击、滚动、touch等)时，可以监听到，并在需要的情况下改变对应的Data.

5.options中可以包含哪些选项

详细解析：vueAPI :<https://cn.vuejs.org/v2/api/index.html#optionMergeStrategies>

目前掌握这些选项:

el:

- 类型:string|HTMLElement
- 作用:决定之后Vue实例会管理哪一个DOM

data:

- 类型:Object|Function
- 作用:Vue实例对应的数据对象

methods:

- 类型:[key:string]|function}}
- 作用:定义属于Vue的一些方法，可以再其他地方调用，也可以再指令中使用