

# Vue DAY01

---

徐铭 [xuming@tedu.cn](mailto:xuming@tedu.cn) 15201603213

## 阶段内容

Vue基础 框架开发基础

vue2与组件库 百慕大影城后台项目开发

Typescript + OOP

Vue3与组件库 百慕大影城前台项目开发

### 学习目标:

前三个阶段重点在于掌握核心基础知识点。

后两个阶段重点在于培养：理解一个合理的业务功能需求，从记忆的深处找到需要的知识点，连接起来形成一段完整代码的**能力**。学习方法：课上重点在于听，课下重点在于练。上课出错不害怕，有异常多翻译，课上先听，课下再调。

## Vue

---

web三大框架之一：vue、react、angular

现阶段，vue有3个大版本：

vue1：淘汰

vue2：过渡期，正在逐步转向vue3。

vue3：主推版本，以后的主流。

<https://v2.cn.vuejs.org/> Vue2官方文档

<https://v2.cn.vuejs.org/v2/guide/installation.html> vue2安装页面，点击**开发版本**按钮，下载 vue.js

## 使用Vue开发web应用的编程方式

传统的dom操作或jquery的开发模式，如果需要更新DOM：

```
<div id="info">欢迎：<span id="name">张三</span></div>
```

```
$('#info').text('欢迎：李四')  
$('#name').text('王五')
```

传统的写法，讲究的是通过选择器找到页面中符合条件的DOM元素，修改它的属性。

## Vue的写法：

```
<div>欢迎: {{name}}</div>
<div>欢迎: {{name}}</div>
<div>欢迎: {{name}}</div>
<div>欢迎: {{name}}</div>
<div>欢迎: {{name}}</div>
```

```
// 创建vue对象来管理DOM树
new Vue({
  data: {
    name: '张三'
  }
})
```

如上定义的名字变量，将会直接显示在页面中，如果通过程序修改了name的值，则页面将会及时刷新。

## 第一个vue案例：写个电影详情页

实现步骤：

1. 准备一个html页面，下载并引入vue.js。
2. 编写代码，基于vue的声明式的语法，完成电影详情内容的展示。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>vue01.html</title>
  <script src="./js/vue.js"></script>
</head>
<body>
  <div id="app">
    
    <h3>电影名称: {{ movie.name }}</h3>
    <h3>电影时长: {{ movie.duration }}分钟</h3>
    <h3>上映时间: {{ movie.showingon }}</h3>
  </div>
  <script>
    // 创建Vue对象来管理页面元素
    new Vue({
      // data选项用于定义页面中需要显示的一些变量
      data: {
        movie: {
          name: '速度与激情10',
          type: '动作 / 剧情',
          duration: 112,
```

```
        show: '2023-05-17'
      }
    },
    el: '#app', // 该属性用于告诉vue，管理页面中的哪个DOM
  })
</script>
</body>
</html>
```

## 未来传代码的方式

**gitee**。白天写的代码都会上传到远程仓库。大家可以通过网址直接去仓库中找每一节课的代码，看日志。晚上复习用。每天晚上将代码和笔记打包上传到ftp。

将代码上传到远程仓库：

1. 去gitee网站创建仓库。code2302
2. 初始化本地仓库目录，提交到本地仓库。
3. 为本地仓库添加远程仓库路径origin。
4. 将本地仓库的版本上传至远程仓库，并开源远程仓库即可。

<https://gitee.com/mingxuchn/code2302>

保存到收藏夹中

## vue中的插值语法

```
<span>{{ js表达式 }}</span>
```

在js表达式中可以直接访问vue对象里data选项声明的变量，也可以进行数据的运算，方法的调用。

## 使用了Vue.js的网页执行流程

浏览器从上到下加载普通的html页面，通过script引入vue.js，浏览器将会加载vue.js脚本。

接下来继续加载body中的dom，加载完毕后了解到所有的DOM元素。

末尾加载script，在该script中通过new Vue()的方式创建Vue对象，让Vue对象来接管#app中需要显示的内容。Vue将会按照自己的方式显示#app中的内容：当vue识别出 `{{}}` 后，将会当做js代码段来执行，进而访问到vue中的变量，动态显示页面内容。

vue在背后做了大量的工作，通过声明式的语法在data中声明变量，这些变量将于页面中引用该变量的DOM元素建立强关联，data中声明的这些变量将会拥有**响应式**的特点（一旦data中的这些变量值被程序所修改，则vue将会操作强关联的DOM元素，自动更新），而不需要程序员先找到某个元素，然后手动更新。

## 点击一个按钮，换一部电影

```
// 创建Vue对象来管理页面元素    #app
let vm = new Vue({
  // data选项用于定义页面中需要显示的一些变量
  data: {
    movie: {
      name: '速度与激情10',
      type: '动作',
      duration: 112,
      showingon: '2023-05-17'
    }
  },
  el: '#app', // 该属性用于告诉vue，管理页面中的哪个DOM
})

let btn = document.getElementById('btn')
btn.onclick = function(){
  // 修改掉vue对象内部data声明的变量的值
  vm.movie = {
    name: '刀剑神域进击篇：暮色黄昏',
    type: '动作 动画',
    duration: 101,
    showingon: '2023-05-26'
  }
}
```

上述代码依然使用了DOM的方式，为按钮绑定事件，实际上按钮已经被vue所接管，可以使用vue的方式为按钮绑定事件。

## vue中的事件处理

```
<div>
  <!-- vue1 的绑定事件的写法 -->
  <button v-on:click="doClick()">按钮</button>
  <!-- vue2 的绑定事件的写法 -->
  <button @click="doClick()">按钮</button>

  <!-- 事件处理函数不用小括号 依然可以执行 -->
  <button v-on:click="doClick">按钮</button>
  <button @click="doClick">按钮</button>

</div>
```

```

new Vue({
  data: { ... },
  methods: {
    doClick(){
      alert('按钮被点击...')
    }
  }
})

```

如果绑定事件处理函数时没有写小括号，也会执行相应事件处理函数；并且在此调用过程中将会自动传递一个 event 事件对象。对象内部包含了触发该事件相关的参数信息。

如果想要同时传递事件对象与自定义参数，则：

```

<button v-on:click="doClick($event, 100, 200)">
  v-on:click 绑定事件
</button>

```

```

doClick(e, arg1, arg2){
  ...
}

```

new Vue() 时，data 中声明的属性，methods 中声明的方法最终都会被混入到 vue 对象中，成为 vue 对象的成员属性、方法。所以在方法中可以直接使用 this 来访问 vue 对象本身。如果需要操作 data 里的数据，则：this.movie.xxxxx；如果需要调用 methods 里的方法：this.sum(10, 20), this.nextMovie() 即可。

## vue 中属性的动态绑定

```


<a href="http://www.baidu.com">链接</a>
<input placeholder="" readonly disabled type="password">

```

vue 框架动态修改属性的语法：

```

    <!-- vue1 -->

    <!-- vue2 -->

<button :disabled="v">按钮</button>

```

```

data: {
  url: 'http://xxx.com/1.jpg',
  n: 1,
  v: true
}

```

一旦为属性添加了 `:` (`v-bind:`), 那么, vue在解析该标签时, 将会把该属性当做动态属性, 此时双引号中的代码将被解析为js代码块, vue会将js代码块的返回值设置给该属性。

## Vue中样式的动态绑定

在页面中为元素绑定css样式的方式, 通常有两种:

```
<div class="c1">阿斯蒂芬啊</div>
<div style="color: red;">直接用style</div>
```

```
.c1 { ... }
.c2 { ... }
```

要研究的就是如何使用vue修改标签的class, 与style属性。

### 动态修改class类名

```
.c1 { ... }
.c2 { ... }
```

```
<div :class="className">阿斯蒂芬啊</div>
<div :class="isc1?'c1':'c2'">阿斯蒂芬啊</div>
<div :class="'tab ${i==0?'active':''}">电影介绍</div>
```

`<!-- vue设计的语法 -->`

```
<div :class="{tab:true, active:true}">电影介绍</div>
```

将会生成:

```
<div class="tab active">电影介绍</div>
```

```
<div :class="{tab:true, active:false}">电影介绍</div>
```

将会生成:

```
<div class="tab">电影介绍</div>
```

```
data: {
  className: 'c1',
  isc1: true,
  i: 0
}
```

### 动态修改style属性

将style当做普通属性，动态拼接属性值字符串即可

```
<div :style="`display:${i==0?'block':'none'};`"></div>
```

<!-- vue设计的语法 -->

```
<div :style="{display:'block', color:'red', border:bstr}"></div>
```

```
<div :style="{display:'none'}"></div>
```

```
data: {  
  bstr: '1px solid black',  
  i: 0  
}
```

## Vue的常用指令

在vue解析所管理的dom节点时，如果发现dom元素身上有v-xxx开头的属性，将会把这些属性当做是**vue指令**来进行处理。每一个指令背后就是一段程序，用于处理当前DOM元素。指令属性的值将会被vue当做javascript代码来解释执行。不同的指令有不同的功能：

1. v-on 动态绑事件
2. v-bind 动态绑定标签属性
3. v-show 动态显示或隐藏元素
4. v-if 动态控制是否输出当前元素

## v-if指令

```
<div v-if="boolean表达式1">分支1</div>  
<div v-else-if="boolean表达式2">分支2</div>  
<div v-else-if="boolean表达式3">分支3</div>  
<div v-else>分支4</div>
```

案例：