

Vue DAY01

徐铭 xuming@tedu.cn 15201603213

Vue基础 框架开发

Vue2与百慕大影城后台管理项目

Typescript+OOP

vue3与百慕大影城前台项目

学习目标

前三个阶段重点在于掌握基础的核心知识点。学习方法：更多的是边听边写。

后两个阶段重点在于学会写项目：理解一个合理的业务功能需求，从以前学过的知识当中找到需要的知识点，串联起来形成完整代码的**能力**。学习方法：课上重点在于听，课下重点在于练。出了错，有了异常，多翻译异常信息。上课跟着写的代码出了bug，不要让改bug的时间占用新知识点的学习时间。

Vue

web 三大框架之一。 vue react Angular 。

1. 早期，HTML、css。
2. Javascript，动态页面，动态操作 DOM。
3. JQuery，封装js的常用功能：DOM处理、动画、发请求等。
4. Angular。MVM 框架。目前市场占有率非常低。
5. React。目前极其活跃，国外、国内一线城市占有率较高。
6. vue。国内市场占有率第一，各线城市都在用。

Vue 基础

现阶段，vue常用的版本：

vue2：过渡期，逐步转向 vue3。

vue3：官方主推版本，以后的主流。

vue 的官方文档：

<https://v2.cn.vuejs.org/v2/guide/index.html>

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，例如：

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

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

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

vue的写法：

```
<div>欢迎: {{name}}</div>
<p>欢迎: {{name}}</p>
<span>欢迎: {{name}}</span>
<i>欢迎: {{name}}</i>
<u>欢迎: {{name}}</u>
.....
```

```
new Vue({
  data: {
    name: '亮亮'
  }
})
```

vue的设计思想，**不再是**先找DOM元素，然后调用方法，修改DOM。而是使用 `{{}}` 语法将页面内容与一个js变量绑定，如果需要修改DOM时，只需要修改绑定的变量即可。

第一个 vue 案例，写一个电影详情页。

实现步骤：

1. 准备一个 html 页面。下载并引入 vue.js 。

```
<script src="./js/vue.js"></script>
```

2. 编写代码，基于vue声明式渲染的语法完成一个电影详情页内容的展示。

```
<!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>01vue.html</title>
</head>
<body>
  <div id="app">
    <h2>电影详情</h2>
```

```

    <div>{{name}}</div>
    <div>{{score}}</div>
    <div>主演: {{actors.join(' / ')}}</div>
    <div>{{showingon}} 上映</div>
  </div>
  <!-- 引入vue.js -->
  <script src="./js/vue.js"></script>
  <script>
    new Vue({
      data: { // data中用于定义页面所需要的数据
        name: '这个杀手不太冷静',
        score: 9.6,
        actors: ['魏翔', '沈腾', '马丽'],
        showingon: '2021-10-15'
      },
      el: '#app' // el用于指明到底让Vue管理哪一个div #app
    })
  </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，大家可以通过网址去仓库中找每一节课的代码，看日志。晚上复习使用。每天的代码晚上都会上传 `ftp`。

<https://gitee.com/mingxuchn/code2209>

Vue 中的插值语法

```

<span>{{ js表达式 }}</span>
<span>{{ score * 10 }}</span>
<span>{{ actors.join('/') }}</span>
<span>{{ isShangying ? '已上映': '暂未上映' }}</span>

```

在 `js` 表达式中可以直接访问 `vue` 的 `data` 中声明的变量，也可以直接进行数据的运算，方法的调用等操作。

`vue` 在背后做了大量的工作，通过声明式语法在 `data` 中声明变量，这些变量将与页面中引用该变量的 `{{}}` 建立关联，一旦建立好这个关联关系，那么 `data` 中定义的这些变量将会拥有**响应式变量**的特点（一旦 `data` 中这些变量的值有变化，`vue` 将会操作关联的 `DOM` 元素，自动更新 `UI`，而不需要程序员寻找 `DOM` 元素，然后手动更新）。

编写一个按钮，点击后换一部电影

在页面中提供一个按钮，为该按钮绑定事件，点击按钮后，修改 `vue` 对象中 `data` 里定义的变量的值。从而达到修改 `dom` 的目的。

```
<!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>01vue.html</title>
</head>
<body>
  <div id="app">
    <h2>电影详情</h2>
    <div>{{name}}</div>
    <div>{{score}}</div>
    <div>主演: {{actors.join(' / ')}}</div>
    <div>{{showingon}} 上映</div>
    <button id="btn">换一部电影</button>
  </div>
  <!-- 引入vue.js -->
  <script src="./js/vue.js"></script>
  <script>
    let vm = new Vue({
      data: { // data中用于定义页面所需要的数据
        name: '这个杀手不太冷静',
        score: 9.6,
        actors: ['魏翔', '沈腾', '马丽'],
        showingon: '2021-10-15'
      },
      el: '#app' // el用于指明到底让Vue管理哪一个div #app
    })
    btn.onclick = function(){
      // 修改vm对象的相应属性,即可修改掉在data中声明的变量的值
      vm.name = '功夫'
      vm.score = 8.5
      vm.actors = ['周星驰', '黄圣依', '火云邪神']
      vm.showingon = '2018-10-21'
    }
  </script>
</body>
</html>

```

Vue 中的事件处理

vue 中给元素绑定事件的方式如下:

```

<div id="app">
  <!-- vue1的写法 -->
  <button v-on:click="clickMe()">点我</button>
  <!-- vue2的写法 -->
  <button @click="clickMe">点我</button>
  <button @click="clickMe()">点我</button>
  <button @click="clickMe(10, 20)">点我</button>
</div>

```

```

new Vue({
  data: {},
  methods: {
    clickMe = (a, b)=>{
      console.log('clickMe...')
    }
  },
  el: '#app'
})

```

以后再vue中为元素绑定事件后，需要在 `methods` 中声明事件处理函数。一旦触发了元素的事件，将会立即调用 `methods` 中声明的事件处理函数。

this

在事件处理函数中，可以直接使用 `this` 来访问vue对象中的属性与其他方法：

```

new Vue({
  data: {
    name: xxx,
    actors: []
  },
  methods: {
    next(){
      console.log(this.name)    // 访问vue对象的属性
      console.log(this.actors)  // 访问vue对象的属性
      this.clickMe()           // 调用vue对象的方法
    },
    clickMe(){...}
  }
})

```

vue 在底层代码工作的过程中，发现事件被触发，则将调用相应事件处理方法，而事件处理方法是vue对象的成员方法，所以真正在调用的时候相当于在执行：

```
vm.next()
```

此时，在 `next` 方法中如果遇到了 `this`，则 `this` 代表的就是当前 `vm` 对象。

Vue 中元素属性的动态绑定

```

<a href="http://www.baidu.com">链接</a>
<button>换一部</button>
```

vue中的动态修改组件属性的语法设计：

```

 // 使用data中定义的url变量

```

```
data: {
  url: 'http://xxx.com/xxx/xxx/xx.jpg',
  n: 1
}
```

如果想要修改图片路径，则直接修改 `url` 即可。

结论：

一旦为属性添加了冒号（`:src`），那么 `vue` 在解析该标签时，将会把该属性当做是动态属性，此时双引号中的代码将被解析为 `js` 代码段，`vue` 会执行这个代码段并将返回值动态设置给标签的属性。

例如：

```
<input :readonly="r" :disabled="d" :placeholder="p" >
```

```
data: {
  r: true,
  d: false,
  p: '文本'
}
```

Vue 中元素样式的动态绑定

在页面中为元素添加css的方式，常见有两种：

```
<div class="d1"></div>
<div style="color:red; background:blue; font-size:1.1em;"></div>
```

动态修改class类名

```
.d1 { color: red; }
.d2 { color: gray; }
```

```
<div :class="active?'d1':'d2'"></div>
```

```
data: {  
  active: true  
}
```

vue为了防止出现语法错误，还提供了通过对象设置class的方式：

```
<div :class="{d1: active, d2: active2}"></div>
```

div 是否包含类名：d1，取决于 active 变量是否为 true。是否包含类名：d2，取决于 active2 变量是否为 true。如果都为true：

```
<div :class="d1 d2"></div>
```

动态修改style内联样式

以拼接字符串的方式来修改style属性：

```
<div :style="`display:${now==0?'block':'none'};`">介绍模块相关内容</div>
```

以对象的方式设置 style，对象的属性相当于css属性名：

```
<div :style="{  
  fontWeight: fw,  
  'font-size': '50px',  
  display: now==0?'block':'none'}">介绍模块相关内容</div>
```

```
data: {  
  fw: 'bolder'  
}
```

Vue 的常用指令

在 vue 解析所管理的 dom 节点时（#app），如果发现 dom 元素身上有 v-xxxx 开头的属性，将会把这些属性当做是 vue 指令来进行特殊处理。指令属性的值会被 vue 当做是 js 代码解析执行。不同的指令有不同的功能。

1. v-on:click 用于绑定事件。
2. v-bind:src 用于动态绑定属性。
3. v-show 可根据指令值是true或false，动态显示/隐藏当前元素。
4. v-if v-else
5. v-for
6. 等...

经典错误集合

✖ ▶[Vue warn]: Property or method "clickMe" is not defined on the instance but referenced during render. Make sure that this property is reactive, either in the data option, or for class-based components, by initializing the property. See: https://v2.vuejs.org/v2/guide/reactivity.html#Declaring-Reactive-Properties .	vue.js:5108
(found in <Root>)	
✖ ▶[Vue warn]: Invalid handler for event "click": got undefined	vue.js:5108
(found in <Root>)	

属性或方法 `clickMe` 并没有在vue对象中定义。要么在data段中声明，要么定义成组件的成员。