Vue01

WEB的发展史

- 原生DOM
  - 原生方法的单词长，难以记忆
  - 如何选中/找到 想要操作的元素
- jQuery
  - 利用封装技巧，对原生DOM进行了简化
  - 自带遍历操作，方便查找元素，不分多个/单个
- Vue
  - 完全抛弃 DOM 的语法，做了一套自己的语法--更简单

Vue

jQuery的格言： Write Less, Do More 写得少做得多

Vue的格言：让不会DOM的人一样能写WEB -- 让傻壮快乐编程

> 官网网站：**cn.vuejs.org**

Vue的开发方式分两种：

- 脚本方式： 同 jQuery，适合入门
- 脚手架方式：更适合实际开发

> Vue从 2014 年诞生，到现在共有3个版本
>
> - Vue1：已经淘汰
> - Vue2：目前的主流版本，但是出于版本末期
> - Vue3：目前最新版本，过渡状态中 -- 以后是主流

插值语法

```
<!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>插值语法 14:29</title>
</head>

<body>
  <!-- 尤雨溪 在Vue中，为大家提供了一套新的语法，允许直接在HTML中书写JS代码 -->
  <div id="app">
    <!-- 壮壮可以控制的范围： 在 app 里，可以为所欲为 -->
    <p>主人：{{uname}}</p>
    <!-- {{}}：类似 模板字符串中的 ${}，其中的代码是JS的 -->
    <p> 8+8={{8+8}}</p>
    <p>贾维斯，我的女孩是谁？ <b>{{girl}}</b></p>
    <p>我的存款有多少？ <b>{{money}}</b></p>
  </div>
```

```
  <!-- 注意: vue的语法只能在 vue对象服务的元素里书写 -->
  <p>贾维斯, 我的女孩是谁? <b>{{girl}}</b></p>


  <!-- 脚本分两种 -->
  <!-- vue.js :开发版, 给人看的. 在使用时提供更多的报错 -->
  <!-- vue.min.js: 产品版, 没有报错 体积更小. 适合上线时使用 -->
  <script src="./vendor/vue.js"></script>
  <script>
    // Vue哪来的? 脚本中引入的
    // new: 触发构造函数, 得到 Vue 构造的实例对象
    // el: 是固定的配置项, 指定Vue实例管理的元素, 值是 id选择器

    // 相当于:
    // new 钢铁侠战甲( { 主人: '陶壮壮', 工具:{ 钱,武器,范凯 } } )


    // 返回值 v 就是创造好的对象, 拥有强大的功能

    // data: 固定属性, 用于提供各种数据
    var v = new Vue({
      el: '#app',
      // 通过后台查看: data中的数据会加载到 生成的vue对象里
      data: {
        money: 9999,
        uname: "壮壮",
        girl: '范凯'
      }
    })

    console.log(v)
  </script>
</body>

</html>
```

插值练习

```
<!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>插值练习 15:00</title>
</head>

<body>
  <div id="box">
    {{eid}}, {{ename}}, {{age}}
  </div>

  <script src="./vendor/vue.js"></script>
  <script>
    // 制作一个 Vue对象 服务于 id=box 的元素
```

```
    // 初始化时提供一些数据:  eid='1001'  ename=文青  age=26
    // 最后把数据显示在 box 里
    new Vue({
      el: '#box',
      data: { eid: '1001', ename: "文青", age: 26 }
    })

  </script>
</body>

</html>
```

属性绑定语法

```
<!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>属性绑定语法 15:25</title>
</head>

<body>
  <div id="app">
    <!-- 属性绑定语法 分两种 -->
    <!-- vue1  v-bind:属性名 -->
    <a v-bind:href="href" v-bind:title="title">{{title}}</a>
    <!-- vue2 :属性名 -->
    <a :href="href" :title="title">{{title}}</a>
    <!-- 注意区分 -->
    <!-- 不带: 是普通的字符串 -->
    <!-- 带 : 是 JS代码 -->
    <button x="8+8" :y="8+8">壮壮</button>
  </div>

  <script src="./vendor/vue.js"></script>
  <script>
    new Vue({
      el: "#app",
      data: {
        title: "Tmooc",
        href: "http://tmooc.cn"
      }
    })
  </script>
</body>

</html>
```

事件

```
<!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>事件 15:41</title>
</head>

<body>
  <div id="app">
    <!-- 标签内容的JS代码，用{{}} -->
    <!-- 点击事件：click -->
    <!-- 两种语法：  vue1   v-on:事件名 -->
    <button v-on:click='num++'>{{num}}</button>
    <!-- 自动化：num变化后，自动更新相关的页面元素，例如按钮的内容 -->

    <!-- vue2语法：@事件名 -->
    <button @click="num++">{{num}}</button>

    <!-- vue的特色：同一个数据出现在不同的元素里，一旦变化，大家一起变 -->
    <button @click="count++">{{count}}</button>
  </div>

  <script src="./vendor/vue.js"></script>
  <script>
    new Vue({
      el: '#app',
      data: { num: 1, count: 10 }
    })


    // 原生写法
    // var btn = document.querySelector('button')
    // btn.onclick = function () {
    //   var x = this.innerHTML * 1
    //   x += 1
    //   this.innerHTML = x
    // }
  </script>
</body>

</html>
```

练习

```
<!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>练习 15:56</title>
</head>

<body>
```

```
  <div id="zz">
    <button @click="num--">-</button>
    <span>{{num}}</span>
    <button @click="num++">+</button>
    <p>单价: {{price}}</p>
    <!-- 单价x数量 -->
    <p>总价: {{price * num}}</p>
    <!-- vue特点: 当数据变化时, 相关的所有元素都会`自动`更新 -->
  </div>

  <script src="./vendor/vue.js"></script>
  <script>
    new Vue({
      el: '#zz',
      data: { num: 6, price: 999 }
    })
  </script>
</body>

</html>
```

事件

```
<!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>事件 16:20</title>
</head>

<body>
  <div id="app">
    <!-- 对于简单的操作, 可以把代码直接在HTML中书写 -->
    <button @click="num++">{{num}}</button>
    <!-- 希望 -1 , 最小是1 -->
    <button @click="jian">num-1</button>
  </div>

  <script src="./vendor/vue.js"></script>
  <script>
    var v = new Vue({
      el: "#app",
      data: { num: 10 },
      // 固定属性: methods
      // 事件触发的函数,都放在这里.  最终会混入到vue实例对象中
      methods: {
        jian() {
          console.log('----');
          console.log('this:', this);
          // 当前的实例对象
          console.log(this == v)

          // 修改 num 的值, -1
          console.log(v.num);
```

```
          // this就是当前实例对象，所以
          if (this.num > 1) this.num--
        }
      }
    });

    console.log(v)
  </script>
</body>

</html>
```

练习

```
<!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>练习 16:42</title>
</head>

<body>
  <div id="app">
    <p>count:{{count}}</p>
    <!-- 如果方法没有参数，()可以省略不写 -->
    <button @click="jia1">+1</button>
    <button @click="jia5">+5</button>
    <button @click="jia10">+10</button>
    <button @click="jia10()">+10</button>
    <hr>
    <!-- ()中就是实参 -->
    <button @click="jia(1)">+1</button>
    <button @click="jia(5)">+5</button>
    <button @click="jia(10)">+10</button>
    <button @click="jia(30)">+30</button>
  </div>

  <script src="./vendor/vue.js"></script>
  <script>
    new Vue({
      el: '#app',
      data: { count: 10 },
      methods: {
        jia(n) { this.count += n },

        jia1() { this.count += 1 },
        jia5() { this.count += 5 },
        jia10() { this.count += 10 }
      }
    })
  </script>
</body>

</html>
```

练习

```
<!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>练习 17:15</title>
</head>

<body>
  <div id="app">
    <!-- 需求: 点击按钮后, 让图片发生对应的变化 -->
    <!-- 做法分两种 -->
    <!-- 1. DOM方案: 用自定义属性存储名称 -->
    <!-- 2. vue方案: 依赖事件传参 -->
    <button @click="pickHero('Annie')">安妮</button>
    <button @click="pickHero('Olaf')">奥拉夫</button>
    <button @click="pickHero('Zoe')">佐伊</button>
    <button @click="pickHero('Gwen')">格温</button>
    <br>
    <!-- :src="JS代码"    JS代码中怎么拼接字符串? 模板 -->
    <img :src="`./heros/${heroName}.png`" alt="">

    <!-- 需要你区分 -->
    <!-- 标签内容用{{}}   属性用 : -->
    <a href="" :x="3+3">{{3+3}}</a>
  </div>

  <!-- 学习vue的思想: HTML中会变化的内容,要存储在data里, 作为数据项 -->

  <script src="./vendor/vue.js"></script>
  <script>
    new Vue({
      el: '#app',
      data: {
        // 分析: 图片地址中的 英雄名 会变化
        heroName: 'Annie'
      },
      methods: {
        // 声明形参 接收 实参
        pickHero(name) {
          this.heroName = name
        }
      }
    })
  </script>
</body>

</html>
```

练习

```html
<!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>练习 17:40</title>
</head>

<body>
  <!-- 大小图切换 -->
  <div id="app">
    <div>
      <img src="./heros/Annie.png" @click="chooseHero('Annie')" alt="">
      <img src="./heros/Galio.png" @click="chooseHero('Galio')" alt="">
      <img src="./heros/Gwen.png" @click="chooseHero('Gwen')" alt="">
      <img src="./heros/Zoe.png" @click="chooseHero('Zoe')" alt="">
      <img src="./heros/Seraphine.png" @click="chooseHero('Seraphine')" alt="">
    </div>

    <!-- 变化 -->
    <img :src="`./heros/${heroName}.jfif`" alt="">
  </div>

  <script src="./vendor/vue.js"></script>
  <script>
    new Vue({
      el: '#app',
      data: { heroName: 'Annie' },
      methods: {
        chooseHero(suibian) {
          this.heroName = suibian
        }
      }
    })

    // 猜一猜都什么意思
    // zz   变量
    // 'zz' 字符串
    // /zz/ 正则
    // [zz] 数组
    // {zz} 对象
  </script>
</body>

</html>
```

作业

- 提前预习剩下内容： 从 FTP 下 2202 班的笔记
- jQuery考试题： 最终月考题会从每个小阶段的考试题中抽取
  - 2202班级的大神：月考 8分钟交卷 96分
  - jQuery考试题： https://ks.wjx.top/vm/QWbfacF.aspx