

前端框架的三国时代-Vue.js(一)

- 快速上手vue,了解数据的双向绑定和简易应用
- Vue.js (读音 /vju:/, 类似于 view) 是一套构建用户界面的 渐进式框架。
- 1.Vue介绍
 - 渐进式的理解
- 2.Vue的两个核心点
 - 响应的数据变化
 - 组合的视图组件
- 3.MVVM模式
- 4.vue的安装
 - 查看所有版本
 - 下载并进行安装
- 5.hello,vue.js
- 6.指令
- 7.双向数据绑定v-model
- 8.表达式
- 9.v-once/v-html
- 10.v-for循环
 - 循环数组
 - 循环对象
- 11.v-on事件
- 12.v-if/v-show
- 13.实现简易购物车

快速上手vue,了解数据的双向绑定和简易应用

Vue.js (读音 /vju:/, 类似于 view) 是一套构建用户界面的 渐进式 框架。

1.Vue介绍

- 核心只关注视图层(view)
- 易学, 轻量, 灵活的特点
- 适用于移动端项目
- 渐进式框架

渐进式的理解

- 声明式渲染(无需关心如何实现)
- 组件系统
- 客户端路由
- 大规模状态管理
- 构建工具

可以根据情况选择, 随意搭配组成自己想要的框架结构。

2.Vue的两个核心点

响应的数据变化

- 当数据发生改变 ->视图的自动更新

组合的视图组件

- ui页面映射为组件树
- 划分组件可维护、可复用、可测试

3.MVVM模式

- M : Model数据模型
- V: view 视图模板
- vm: view-Model 视图模型

将数据绑定到DOM上，监听视图的变化更改数据

4.vue的安装

1.0 -> 2.0

[查看所有版本](#)

```
$ npm info vue
```

下载并进行安装

```
$ npm init -y  
$ npm install vue --save
```

默认下载最新版本使用。

5.hello,vue.js

```
<div id="app">
  {{message}}
</div>
<script>
  var vm = new Vue({
    el: '#app',
    data: {
      message: 'hello,vue.js'
    }
  })
</script>
```

6.指令

- 什么是指令

指令是一种特殊自定义行间属性，在Vue中，指令以v-开头

7.双向数据绑定v-model

```
<input type="text" v-model="message">
```

当你改变数据上的属性，视图也会随之变化。相反视图变化也会影响数据

8.表达式

插值表达式，使用Mustache语法(双大括号)

```
{{message=="hello,vue.js"? "yes": "no"}}  
<div v-text="hello,vue.js"? "yes": "no"></div>
```

当数据改变时，插值内容会自动更新,在表达式中可以赋值、三元、和展示返回值

9.v-once/v-html

- 有时候我们想让数据只绑定一次,当数据在更改时不会更新内容(v-once)
- 当绑定的数据是html字符串时，展示成正常的html(v-html)

```
<div v-once>{{message}}</div>  
<div v-html="html"></div>  
data:{  
  html: '<h1>hello</h1>'  
}
```

10.v-for循环

循环数组

循环的顺序是value,index

```
<ul>
  <li v-for="(phone,index) in phones">{{phone}} {{index}}</li>
</ul>
data:{
  phones:['apple','xiaomi','huawei']
}
```

循环对象

循环的顺序是value,key

```
<li v-for="(value,key) in json" >
  {{value}}
</li>
data:{
  json:{name:'zfx'}
}
```

11.v-on事件

- v-on:click简写@click
- 执行方法时加上()事件源默认不传递，需要手动传入\$event
- methods中的this永远指向Vue的实例

```
<button v-on:click="addFruit">按钮</button>
<ul>
```

```

<li v-for="value in fruits" >
  {{value}}
</li>
</ul>
var vue = new Vue({
  el: '#box',
  data:{
    fruits:['香蕉','苹果','橘子']
  },
  methods:{
    addFruit(){
      this.fruits.push('苹果');
    }
  }
});

```

12.v-if/v-show

- v-show 如果为false时通过css样式将元素隐藏

```
<div v-show='false'>jw, handsome</div>
```

- v-if 如果为false时移除DOM节点

```
<div v-if='false'>jw, handsome</div>
```

v-if后面可以跟v-else-if或者v-else

13.实现简易购物车

商品列表

商品名称	商品价格	操作
iphone7	6777	<button>删除</button>
<button>删除全部</button>		

商品名称

商品价格

添加重置商品