响独构代插绑指计类

条列数滤监修表生过

组创传事索分动内活模复异递响

自指自过混插

vue是一个响应的数据与DOM绑定系统

组件系统：可以用独立的可复用的小组件来构建大型应用

构造器

var vm =new Vue({

})

扩展Vue构造器：所有的Vue组件都是扩展Vue实例

var MyComponent = Vue.extend({

})

var MyComponentInstance = new MyComponent()

属性与方法

Vue实例会代理该data对象里所有的属性

Vue实例暴露了一些属性和方法，这些属性和方法都有前缀$，以便区分代理该data对象里所有的属性

数据绑定语法

插值：{{ }}将数据解析为纯文本，可用在text内

v-html指令将数据解析为html

v-once指令只处理单次插值，以后的数据变化不会引起更新

绑定表达式： 表达式，只能包含单个表达式

过滤器，{{ | }}可以串联，可以接受参数（始终以表达式的值

作为第一个参数，带引号的参数视为字符串，不带引号的参数按表

达式计算）

指令：带有v-，值限定为绑定表达式，当绑定表达式的值改变时操作DOM

有些指令可以在指令名称后加冒号和参数

修饰符是以 . 开始的后缀，用以表示指令应当以特殊方式绑定

v-bind:缩写成：

v-on：缩写成@

计算属性 conputed:{

属性名:function(){

}

}

计算属性默认只是getter

conputed:{

属性名:{

get:function(){}

set:function(){}

}

}

class与style绑定：对象语法和数组语法

条件渲染：单个元素直接用v-if，v-else，v-show=””,v-show=”!”

多个元素在外层包<template></template>并用v-if，v-else

区别：v-if局部编译或缺载，v-show切换元素display

运行时条件不太可能改变用v-if，频繁切换用v-show

v-show不支持<template>

列表渲染

v-for支持<template>

v-for操作数据数组：

v-for=”当前数据数组元素 in 数据数组”

v-for=“（当前数据数组元素 ， 索引别名） in 数据数组”

v-for操作数据对象：

v-for=”当前数据对象元素的值 in 数据对象”

v-for=”（当前数据对象元素的值 ， 键别名） in 数据对象”

v-for=”（当前数据对象元素的值 ， 键别名 ， 索引别名） in 数据对象”

v-for操作值：

v-for=”当前数据值 in 整数”

数组变动检测

变异方法：能触发视图更新，方法有pop，push，shift，unshift，sort，splice，

reverse

重塑数组：能触发视图更新，返回新数组，不改变旧数组，方法有concat，slice，

filter

就地复用为:key=””

$set方法：能触发视图更新，应用对象为数据数组对象

$get方法：能触发视图更新，应用对象为数据数组对象

想显示过滤/排序的结果，同时不实际修改或重置原始数据，可以：

1. 创建一个计算属性
2. 在计算属性不适用的情况下（如在嵌套v-for循环中）使用method方法

方法与事件处理器

v-on指令监听DOM事件

1. v-on：事件属性=”方法名”，在Vue实例中

methods：{

方法名：function（event）{

}

}

1. v-on：事件属性=”方法名（参数）”，在Vue实例中

methods：{

方法名：function（参数）{

}

}

1. v-on：事件属性=”方法名（参数，$event）”，在Vue实例中

methods：{

方法名：function（参数，event）{

}

}

事件修饰符（可以串联）

方法中event.preventDefault（）可用.prevent代替，功能是：不在重载页面

方法中event.stopPropagation（）可用.stop代替，功能是：阻止事件冒泡

.self功能是：只当事件在该元素本身（而不是子元素）触发时触发回调

按键修饰符：支持自定义按键别名，支持单字母按键别名

.up .down .left .right

.esc .tab .delete .enter .space

表单控件绑定

1. <input type=”text” v-model=”名称” />

{{名称}}对应input内容

1. <textarea v-model=”名称”></textarea>

{{名称}}对应textarea内容

1. <select v-model=”名称”>

<option value=”value1” selected></option>

<option value=”value2” ></option>

</selected>

{{名称}}对应option的value值

（4）<input type=”radio” value=”value1” v-model=”名称”/>

<input type=”radio” value=”value2” v-model=”名称”/>

{{名称}}对应radio的value值

（5）<input type=”checkbox” value=”value1” v-model=”名称”/>

<input type=”checkbox” value=”value2” v-model=”名称”/>

<input type=”checkbox” value=”value3” v-model=”名称”/>

{{名称}}对应true/false。

若Vue实例的data中有名称：[]则{{名称}}对应选中的value值

绑定值

对于select和input（radio类型）有v-bind：value=”数据中的变量名”

对于 input（checkbox类型）有v-bind：true-value=”数据中的变量名a”

对于 input（checkbox类型）有v-bind：false-value=”数据中的变量名b”

参数特性

.lazy：默认情况下v-model在input事件中同步输入框值与数据

当<input v-model.lazy=”名字”/>时，改在change事件中同步

.number:<input v-model.number=”名字”/>时，自动将用户的输入转为number类型（如果值的转换结果为NaN则返回原值）

实例生命周期

new Vue（）

beforeCreate

Observe Data

Init Events

created No

Has “el” option

Yes when vm.$mount(el) is called

Has “template” option

Yes No

Compile template Compile el`s outerHTML as template

into render function

beforeMount

Create vm.$el and replace “el” with it

mounted

when data changes beforeUpdate

Mounted Virtural Dom

re-render and patch

updated

When vm.$destroyed() is called

beforeDestroy

Teardown watchers

child components and

event listeners

destroyed Destroyed