Vue1.x 和 Vue2.x 的区别

1. 用index代替$index
2. 过滤器

常用指令 : 扩展html标签功能,属性

v-model 一般表单元素(input) 双向数据绑定

循环:

v-for="name in arr"

{{$index}}

v-for="name in json"

{{$index}} {{$key}}

v-for="(k,v) in json"

事件:

v-on:click="函数"

v-on:click/mouseout/mouseover/dblclick/mousedown.....

new Vue({

el:'#box',

data:{ //数据

arr:['apple','banana','orange','pear'],

json:{a:'apple',b:'banana',c:'orange'}

},

methods:{

show:function(){ //方法

alert(1);

}

}

});

显示隐藏:

v-show=“true/false”

bootstrap+vue简易留言板(todolist):

bootstrap: css框架 跟jqueryMobile一样

只需要给标签 赋予class，角色

依赖jquery

确认删除？和确认删除全部么?

-----------------------------------------

事件:

v-on:click/mouseover......

简写的:

@click=" " 推荐

事件对象:

@click="show($event)"

事件冒泡:

阻止冒泡:

a). ev.cancelBubble=true;

b). @click.stop 推荐

默认行为(默认事件):

阻止默认行为:

a). ev.preventDefault();

b). @contextmenu.prevent 推荐

键盘:

@keydown $event ev.keyCode

@keyup

常用键:

回车

a). @keyup.13

b). @keyup.enter

上、下、左、右

@keyup/keydown.left

@keyup/keydown.right

@keyup/keydown.up

@keyup/keydown.down

.....

-----------------------------------------

属性:

v-bind:src=""

width/height/title....

简写:

:src="" 推荐

<img src="{{url}}" alt=""> 效果能出来，但是会报一个404错误

<img v-bind:src="url" alt=""> 效果可以出来，不会发404请求

-----------------------------------------

class和style:

:class=""v-bind:class=""

:style=""v-bind:style=""

:class="[red]" red是数据

:class="[red,b,c,d]"

:class="{red:a, blue:false}"

:class="json"

data:{

json:{red:a, blue:false}

}

--------------------------

style:

:style="[c]"

:style="[c,d]"

注意: 复合样式，采用驼峰命名法

:style="json"

-----------------------------------------

模板:

{{msg}} 数据更新模板变化

{{\*msg}} 数据只绑定一次

{{{msg}}} HTML转义输出

-----------------------------------------

过滤器:-> 过滤模板数据

系统提供一些过滤器:

{{msg| filterA}}

{{msg| filterA | filterB}}

uppercase eg: {{'welcome'| uppercase}}

lowercase

capitalize

currency 钱

{{msg| filterA 参数}}

....

-----------------------------------------

交互:

$http （ajax）

如果vue想做交互

引入: vue-resouce

get:

获取一个普通文本数据:

this.$http.get('aa.txt').then(function(res){

alert(res.data);

},function(res){

alert(res.status);

});

给服务发送数据:√

this.$http.get('get.php',{

a:1,

b:2

}).then(function(res){

alert(res.data);

},function(res){

alert(res.status);

});

post:

this.$http.post('post.php',{

a:1,

b:20

},{

emulateJSON:true //必须要加上

}).then(function(res){

alert(res.data);

},function(res){

alert(res.status);

});

jsonp:

<https://sug.so.360.cn/suggest?callback=suggest_so&word=a>

<https://sp0.baidu.com/5a1Fazu8AA54nxGko9WTAnF6hhy/su?wd=a&cb=jshow>

this.$http.jsonp('https://sp0.baidu.com/5a1Fazu8AA54nxGko9WTAnF6hhy/su',{

wd:'a'

},{

jsonp:'cb' //callback名字，默认名字就是"callback"，此时在百度接口中是

}).then(function(res){

alert(res.data.s);

},function(res){

alert(res.status);

});

**if**(ev.keyCode==38||ev.keyCode==40)**return**;//在按向上或向下的箭头的时候不会发送请求  
 **if**(ev.keyCode==13) { //按回车键会自动在百度上搜索你选择的内容  
 window.open('http://www.baidu.com/s?wd=' + **this**.t1);  
 }

[https://www.baidu.com/s?wd='你](https://www.baidu.com/s?wd=s)要搜索的内容’ 按enter键在百度自动搜索

作业:

1. 简易留言-> 确认删除? 确认删除全部

2. 用vue get 写个例子 weibo

vue制作weibo

交互

vue-> 1.0

vue-resource ajax php

服务器环境(node)

this.$http.get()/post()/jsonp()

this.$http({

url:地址

data:给后台提交数据,

method:'get'/post/jsonp

jsonp:'cb' //cbName

});

----------------------------------

vue事件:

@click=""

数据:

添加一条留言：

获取某一页数据:

getPageData(1);

----------------------------------

vue生命周期:

钩子函数(可以使用的函数) 执行的时期

created -> 实例创建好之后 √

beforeCompile -> 编译之前

compiled -> 编译之后

ready -> 插入到文档中 √

beforeDestroy -> 销毁之前

destroyed -> 销毁之后

----------------------------------

用户会看到花括号标记:（闪烁）

v-cloak 防止闪烁, 比较大段落

<span>{{msg}}</span> -> v-text 也可以防止闪烁，但是不能转义html

-> v-html 转义html

----------------------------------

ng: $scope.$watch

计算属性的使用:

computed:{

b:function(){ //默认调用get ，b的值取决于返回值

return 值

}

}

--------------------------

computed:{

b:{

get:

set:

}

}

\* computed里面可以放置一些业务逻辑代码，一定记得return

---------------------------------

vue实例简单方法:

var vm = new Vue({

el:’#box’,

data:{

a:1

}

});

vm.$el -> 就是元素

vm.$data -> 就是data

vm.$mount -> 手动挂在vue程序, 相当于el:’ ’

vm.$options -> 获取自定义属性

vm.$destroy -> 销毁对象

vm.$log(); -> 查看现在数据的状态

---------------------------------

循环：

v-for="value in data"

会有重复数据？

track-by='索引' 提高循环性能

track-by='$index/uid(数据id)'

---------------------------------

过滤器:

自定义过滤器: model ->过滤 -> view  
 {{ a | name 1 2}}

Vue.filter(‘name’,function(input, a, b){

a：参数1； b：参数2；

});

时间转化器：

双向过滤器:\*（过滤html标记）

Vue.filter('filterHtml',{

read:function(input){ //model-view

return input.replace(/<[^<+]>/g,'');

},//（model -> view）

write:function(val){ //view -> model

return val; //未执行

} //(view -> model)

});

数据 -> 视图

---------------------------------

v-text

v-for

v-html

指令: 扩展html语法

自定义指令:

属性:

Vue.directive(指令名称,function(参数){

this.el -> 原生DOM元素

});

<div v-red="参数"></div>

指令名称: v-red -> red

\* 注意: 必须以 v-开头

拖拽:

-------------------------------

自定义元素指令:（用处不大）

Vue.elementDirective('zns-red',{

bind:function(){

this.el.style.background='red';

}

});

<zns-red></zns-red>

------------------------------------------------

@keydown.up

@keydown.enter

@keydown.a/b/c....

自定义键盘信息:

Vue.directive('on').keyCodes.ctrl=17;

Vue.directive('on').keyCodes.myenter=13;

------------------------------------------------

监听数据变化:

vm.$el/$mount/$options/....

vm.$watch(name,fnCb); //浅度

vm.$watch(name,fnCb,{deep:true}); //深度监视

------------------------------------------------

vue组件:

组件间各种通信

slot

vue-loader webpack .vue

vue-router