目录

[脚手架 2](#_Toc527211244)

[全局api 2](#_Toc527211245)

[vue.extend/vue.component 2](#_Toc527211246)

[vue.extend/ vm.$mount() 2](#_Toc527211247)

[vue.component 2](#_Toc527211248)

[new vue，vue.component,vue.extend 3](#_Toc527211249)

[vue.set/vue.delete 3](#_Toc527211250)

[选项/数据 3](#_Toc527211251)

[data 4](#_Toc527211252)

[watch/vm.$watch 4](#_Toc527211253)

[computed 4](#_Toc527211254)

[选项/dom 4](#_Toc527211255)

[el/ vm.$el 4](#_Toc527211256)

[template 5](#_Toc527211257)

[实例属性 5](#_Toc527211258)

[$refs 5](#_Toc527211259)

[$options 5](#_Toc527211260)

[$attrs 5](#_Toc527211261)

vue.config = { 9个

errorHandler,wranHanddler,slient,performance,productionTip

keyCodes, ignoredElement, optionMergeStrategies

devtools,

}

# 脚手架

vue-cli

vue init webpack xxx

# 全局api

## vue.extend/vue.component

### vue.extend/ vm.$mount()

只是一个构造器。

参数：组件选项，其中data为函数。

返回的是一个“扩展实例构造器”，也就是预设了部分选项的Vue的实例构造器

当在模板中遇到该组件作为标签的自定义元素时，会自动调用“扩展实例构造器”来生产组件实例，并挂在到自定义元素上

### vue.component

获取或注册全局组件。

// 注册组件，传入一个扩展过的构造器

Vue.component('my-component', Vue.extend({ /\* ... \*/ }))

// 注册组件，传入一个选项对象 (自动调用 Vue.extend)

Vue.component('my-component', { /\* ... \*/ })

// 获取注册的组件 (始终返回构造器)

var MyComponent = Vue.component('my-component')

### new vue，vue.component,vue.extend

## vue.set/vue.delete

对象不能是 Vue 实例，或者 Vue 实例的根数据对象（比如说不能是data里的某个属性，可以是data里的某个属性的属性）

这个方法主要用于避开 Vue 不能检测属性被添加的限制。（因为只是浅层的检测）

var vm=new Vue({

el:'#test',

data:{ //data中已经存在info根属性

info:{

name:'小明';

}

}

});

//给info添加一个性别属性

Vue.set(vm.info,'sex','男');

# 选项/数据

data/methods

props/propsData

watch/computed

## data

浏览器 API 创建的原生对象，原型上的属性会被忽略。

Vue 实例代理了 data 对象上所有的属性，因此访问 vm.a 等价于访问 vm.$data.a。

以 \_ 或 $ 开头的属性 **不会** 被 Vue 实例代理，因为它们可能和 Vue 内置的属性、API 方法冲突。你可以使用例如 vm.$data.\_property 的方式访问这些属性。

data在**组件定义时(也就是说new Vue不需要)**必须是一个函数，因为如果 data 仍然是一个纯粹的对象，则所有的实例将**共享引用**同一个数据对象！

如果需要，可以通过将 vm.$data 传入 JSON.parse(JSON.stringify(...)) 得到深拷贝的原始数据对象。

## watch/vm.$watch

复杂写法

d: {

handler: function (val, oldVal) { /\* ... \*/ },

immediate: true // 立即以表达式的当前值触发回调

deep： true

}

## computed

# 选项/dom

## el/ vm.$el

只在由 new 创建的实例中遵守

## template

如果 Vue 选项中包含渲染函数，该模板将被忽略。

# 实例属性

## $refs

类似react的refs

## $options

## $attrs