# **vm.$attrs 【Vue 2.4.0新增inheritAttrs，attrs详解】**

[](https://www.jianshu.com/u/aabdb035b04c)

[\_小坦克\_](https://www.jianshu.com/u/aabdb035b04c) 关注

2018.01.22 17:23\* 字数 620 阅读 2283评论 3喜欢 5

#### **1、vm.$attrs简介**

首先我们来看下vue官方对vm.$attrs的介绍：  
包含了父作用域中不作为 prop 被识别 (且获取) 的特性绑定 (class 和 style 除外)。当一个组件没有声明任何 prop 时，这里会包含所有父作用域的绑定 (class 和 style 除外)，并且可以通过 v-bind="$attrs" 传入内部组件——在创建更高层次的组件时非常有用。  
猛一看有点看不明白....

#### **2、场景介绍**

vue中一个比较令人烦恼的事情是属性只能从父组件传递给子组件。这也就意味着当你想向嵌套层级比较深组件数据传递，只能由父组件传递给子组件，子组件再传递给孙子组件...像下面这样：

<parent-component :passdown="passdown">

<child-component :passdown="passdown">

<grand-child-component :passdown="passdown">

....

就这样一层一层的往下传递passdown这个变量，最后才能用{{passdown}}。

假如我们需要传递的属性只有1,2个还行，但是如果我们要传递的有几个或者10来个的情况，这会是什么样的场景，我们会在每个组件不停的props，每个必须写很多遍。有没有其它方便的写法？有，通过vuex的父子组件通信，的确这个是一个方法，但是还有其它的方法，这个就是我们要说的。通过inheritAttrs选项，以及实例属性$attrs

#### **3、实例：**

<template>

<div class="home">

<mytest :title="title" :massgae="massgae"></mytest>

</div>

</template>

<script>

export default {

name: 'home',

data () {

return {

title:'title1111',

massgae:'message111'

}

},

components:{

'mytest':{

template:`<div>这是个h1标题{{title}}</div>`,

props:['title'],

data(){

return{

mag:'111'

}

},

created:function(){

console.log(this.$attrs)*//注意这里*

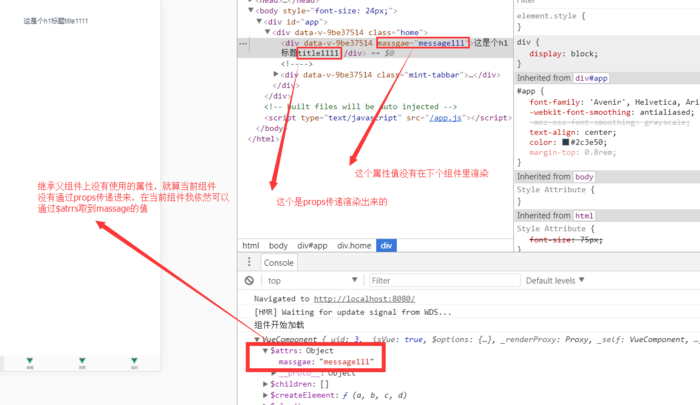
}

}

}

}</script>

上边的代码，我们在组件里只是用了title这个属性，massgae属性我么是没有用的，那么下浏览器渲染出来是什么样呢？如下图：



attrs.png

我们看到：组件内未被注册的属性将作为普通html元素属性被渲染，如果想让属性能够向下传递，即使prop组件没有被使用，你也需要在组件上注册。这样做会使组件预期功能变得模糊不清，同时也难以维护组件的DRY。在Vue2.4.0,可以在组件定义中添加inheritAttrs：false，组件将不会把未被注册的props呈现为普通的HTML属性。但是在组件里我们可以通过其$attrs可以获取到没有使用的注册属性，如果需要，我们在这也可以往下继续传递。

如果我们在子组件里设置 inheritAttrs: false：

components:{

'mytest':{

template:`<div>这是个h1标题{{title}}</div>`,

props:['title'],

inheritAttrs: false,

data(){

return{

mag:'111'

}

},

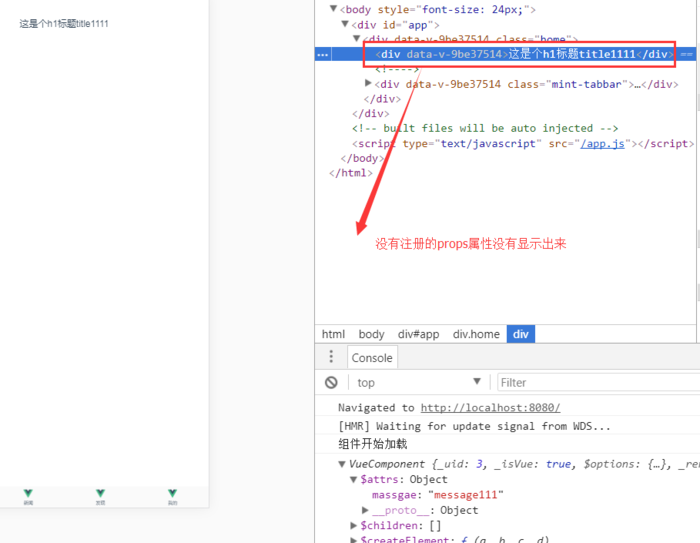
created:function(){

console.log(this.$attrs)*//注意这里*

}

}

渲染效果如下：



不继承的情况.png