Vue.js 目录结构 Vue.js 样式绑定

Vue.js 计算属性

计算属性关键词: computed。

计算属性在处理一些复杂逻辑时是很有用的。

可以看下以下反转字符串的例子：

实例 1

```
<div id="app">
{{ message.split('').reverse().join('') }}
</div>
```

尝试一下 »

实例 1 中模板变的很复杂起来，也不容易看懂理解。

接下来我们看看使用了计算属性的实例：

实例 2

```
<div id="app">
<p>原始字符串: {{ message }}</p>
<p>计算后反转字符串: {{ reversedMessage }}</p>
</div>
<script>
var vm = new Vue({
el: '#app',
data: {
message: 'Runoob!'
},
computed: {
// 计算属性的 getter
reversedMessage: function () {
// `this` 指向 vm 实例
return this.message.split('').reverse().join('')
}
}
})
</script>
```

尝试一下 »

实例 2 中声明了一个计算属性 reversedMessage 。

提供的函数将用作属性 vm.reversedMessage 的 getter 。

vm.reversedMessage 依赖于 vm.message，在 vm.message 发生改变时，vm.reversedMessage 也会更新。

computed vs methods

我们可以使用 methods 来替代 computed，效果上两个都是一样的，但是 computed 是基于它的依赖缓存，只有相关依赖发生改变时才会重新取值。而使用 methods ，在重新渲染的时候，函数总会重新调用执行。

实例 3

```
methods: {
reversedMessage2: function () {
return this.message.split('').reverse().join('')
}
}
```

尝试一下 »

可以说使用 computed 性能会更好，但是如果你不希望缓存，你可以使用 methods 属性。

---

computed setter

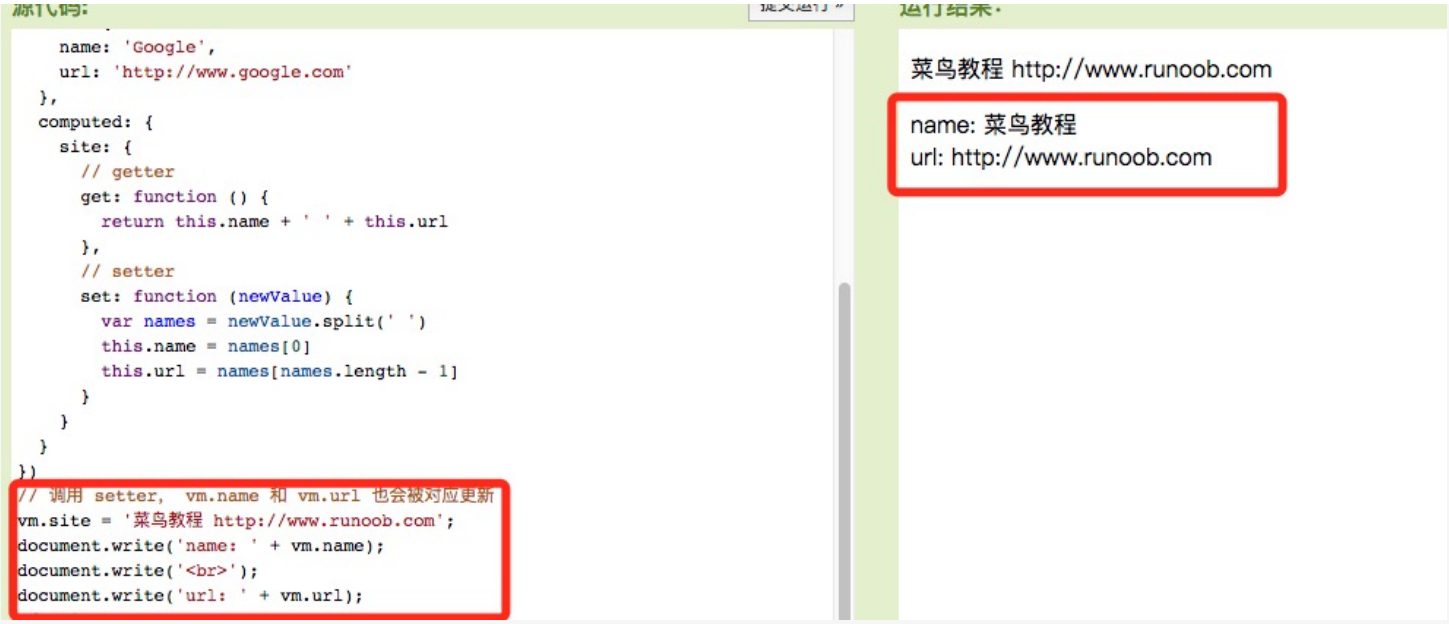
computed 属性默认只有 getter ，不过在需要时你也可以提供一个 setter ：

实例 4

```
var vm = new Vue({
el: '#app',
data: {
name: 'Google',
url: 'http://www.google.com'
},
computed: {
site: {
// getter
get: function () {
return this.name + ' ' + this.url
},
// setter
set: function (newValue) {
var names = newValue.split(' ')
this.name = names[0]
this.url = names[names.length - 1]
}
}
}
})
// 调用 setter， vm.name 和 vm.url 也会被对应更新
vm.site = '菜鸟教程 http://www.runoob.com';
document.write('name: ' + vm.name);
document.write('<br>');
document.write('url: ' + vm.url);
```

尝试一下 »

从实例运行结果看在运行 vm.site = '菜鸟教程 http://www.runoob.com'; 时，setter 会被调用， vm.name 和 vm.url 也会被对应更新。



← Vue.js 目录结构 Vue.js 样式绑定 →

2 篇笔记

写笔记

把代码改了改，应该可以体现 computer 属性“依赖缓存”的概念以及与 method 的差别。如下面代码，cnt 是独立于 vm 对象的变量。在使用 reversedMessage 这个计算属性的时候，第一次会执行代码，得到一个值，以后再使用 reversedMessage 这个计算属性，因为 vm 对象没有发生改变，于是界面渲染就直接用这个值，不再重复执行代码。而 reversedMessage2 没有这个缓存，只要用一次，函数代码就执行一次，于是每次返回值都不一样。

```
var cnt=1;
var vm = new Vue({
  el: '#app',
  data: {
    message: 'Runoob!'
  },
  computed: {
    // 计算属性的 getter
    reversedMessage: function () {
      // `this` 指向 vm 实例
      cnt+=1;
      return cnt+this.message.split('').reverse().join('')
    }
  },
  methods: {
    reversedMessage2: function () {
      cnt+=1;
      return cnt+this.message.split('').reverse().join('')
    }
```

```
  }
})
```

尝试一下 »

softfei 2个月前 (01-16)

当你没有使用到计算属性的依赖缓存的时候，可以使用定义方法来代替计算属性，在 methods 里定义一个方法可以实现相同的效果，甚至该方法还可以接受参数，使用起来更灵活。

```
<div id ="app">My Runoob Application
    <p>原始数据: {{text}}</p>
    <!-- 注意，这里的reversedText是方法，所以要带()-->
    <p>变化后数据: {{reversedText()}}</p>
</div>
<script>
var app = new Vue({
    el: '#app',
    data: {
        text:'123,456',
    },
    methods:{
        reversedText: function(){
            return this.text.split(',').reverse().join(',');
        }
    },
});
</script>
```

尝试一下 »

挑战者666888 2个月前 (01-30)