**一.前言**

假设我们有如下这样一段代码

<!DOCTYPE html>

<html>

<head>

<title>Vue中的计算属性</title>

<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>

</head>

<body>

<div id="box">

{{message.split(",").join("-")}}

</div>

<script type="text/javascript">

var vm = new Vue({

el: '#box',

data: {

message: 'A,B,C,D,E,F,G'

},

methods: {

}

})

</script>

</body>

</html>

可以看到message字符进行了分割和重新连接，当对一个数据存在多个操作时，如果直接写在模板中，很容易让代码变得很繁琐。此时，计算属性是一个不错的选择。

**二.计算属性**

我们直接写一个示例来演示计算属性的用法

<!DOCTYPE html>

<html>

<head>

<title>Vue中的计算属性</title>

<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>

</head>

<body>

<div id="box">

<!-- 在模板处理数据 -->

{{message.split(",").join("-")}}

<br>

<!-- 将数据处理写到计算属性中 -->

{{resetMessage}}

<br>

{{upperMessage}}

</div>

<script type="text/javascript">

var vm = new Vue({

el: '#box',

data: {

message: 'A,B,C,D,E,F,G'

},

methods: {

},

computed: {

// 计算属性：函数的名称必须和属性有关联 xxxMessage

resetMessage: function(){

return this.message.split(",").join("-");

}

// 此处可以编写多个关于message的计算属性

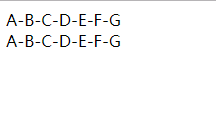
}

})

</script>

</body>

</html>



可以看到计算属性需要写在Vue的computed属性中，计算属性的函数名称必须是"xxx属性名",并且可以编写一个属性的多个计算属性。

**三.计算属性的优势**

上面示例中的计算属性resetMessage也可以通过一个方法实现

<!DOCTYPE html>

<html>

<head>

<title>Vue中的计算属性</title>

<script type="text/javascript" src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>

</head>

<body>

<div id="box">

<!-- 在模板处理数据 -->

{{message.split(",").join("-")}}

<br>

<!-- 将数据处理写到计算属性中 -->

{{resetMessage}}

<br>

<!-- 计算属性使用方法实现 -->

{{m\_resetMessage()}}

</div>

<script type="text/javascript">

var vm = new Vue({

el: '#box',

data: {

message: 'A,B,C,D,E,F,G'

},

methods: {

// 计算属性使用方法去实现

// 方法名不能和计算属性重复，否则计算属性在模板中会解析成函数

m\_resetMessage: function(){

return this.message.split(",").join("-");

}

},

computed: {

// 计算属性：函数的名称必须和属性有关联 xxxMessage

resetMessage: function(){

return this.message.split(",").join("-");

}

// 此处可以编写多个关于message的计算属性

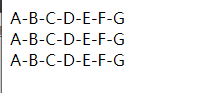
}

})

</script>

</body>

</html>



注意：methods中的方法名不能和computed中的计算属性重复，否则计算属性在模板中会解析成函数

使用方法和计算属性的效果基本一直，不同的是计算属性存在缓存行为。当对应的属性发生变化后，计算属性会重新求值；否则调用计算属性就会返回之前的计算结果。

因此当计算属性依赖的某个属性数据量较大，且计算复杂时，我们就可以使用计算属性避免不必要的开销。