

## 1.计算属性的基本属性:(computed)

```
<div id="app">
  <!-- 之前的方法 -->
  <h2>{{firstName + ' ' + lastName}} </h2>
  <h2>{{firstName}} {{lastName}} </h2>
  <h2>{{getFullName()}} </h2>
  <!-- 计算属性方法 -->
  <h2>{{fullName}}</h2>
</div>
<script>
  const app = new Vue({
    el: '#app',
    data: {
      firstName: 'Lebron',
      lastName: 'James',
      books: [{ ...
    },
    // computed:计算属性
    computed: {
      fullName: function() {
        return this.firstName + ' ' + this.lastName
      },
    },
    methods: {
      getFullName() {
        return this.firstName + ' ' + this.lastName
      }
    },
  })
```

## 2.计算属性的复杂操作

```
// computed:计算属性
computed: {
  fullName: function() {
    return this.firstName + ' ' + this.lastName
  },
  totalPrice: function() {
    let result = 0
    for (let i = 0; i < this.books.length; i++) {
      result += this.books[i].prices
    }
    return result
  }
}
```

### 3.计算属性的setter和getter

每个计算属性都包含一个getter和一个setter

- 下面的例子中，我们只使用getter来读取
- 某些情况下，也可以提供一个setter方法(不常用)
- 在需要写setter的时候，代码如下：

```

computed: {
  //简写版
  /*
  fullName: function() {
    return this.firstName + ' ' + this.lastName
  },
  */
  //完整版
  fullName: {
    //计算属性一般是没有set方法，只读属性
    /*
    set: function(newValue) {
      //console.log('-----', newValue)
      const names = newValue.split(' ');
      this.firstName = names[0];
      this.lastName = names[1];
    }, */
    get: function() {
      return this.firstName + ' ' + this.lastName;
    }
  }
},

```

#### 4.计算属性的缓存

methods和computed的区别：

计算属性会进行缓存，如果多次使用时，计算属性只会调用一次；methods会调用多次  
开发中多用computed

kobe Bryant

kobe Bryant

kobe Bryant

kobe Bryant

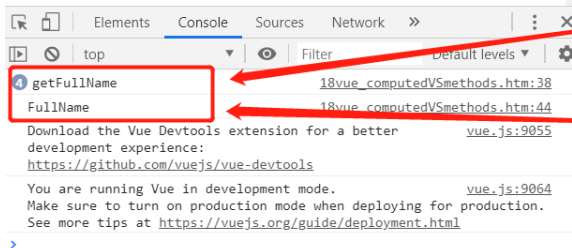
kobe Bryant

kobe Bryant

kobe Bryant

kobe Bryant

kobe Bryant



```
13 <div id="app">
14   <!-- 方案一：直接拼接,语法过于繁琐 -->
15   <h2>{{firstName}} {{lastName}}</h2>
16   <hr>
17   <!-- 方案二：通过定义Methods -->
18   <h2>{{getFullName()}}</h2>
19   <h2>{{getFullName()}}</h2>
20   <h2>{{getFullName()}}</h2>
21   <h2>{{getFullName()}}</h2>
22   <hr>
23   <!-- 方案三：通过计算属性computed -->
24   <h2>{{fullName}}</h2>
25   <h2>{{fullName}}</h2>
26   <h2>{{fullName}}</h2>
27   <h2>{{fullName}}</h2>
28 </div>
29 <script>
30   const app = new Vue({
31     el: '#app',
32     data: {
33       firstName: 'kobe',
34       lastName: 'Bryant',
35     },
36     methods: {
37       getFullName: function() {
38         console.log('getFullName')
39         return this.firstName + ' ' + this.lastName
40       }
41     },
42     computed: {
43       fullName: function() {
44         console.log('FullName')
45         return this.firstName + ' ' + this.lastName
46       }
47     }
48   })
49 </script>
```