计算属性`computed`

1. 定义：要用的属性不存在，要通过已有属性计算得来。
2. 原理：底层借助了 Object.definepropery 方法提供的 getter 和 setter
3. get函数什么时候执行?
    1. 初次读取时会执行一次。
    2. 当依赖的数据发生变化时会被再次调用
4. 优势： 与 method 实现相比，内部有缓存机制(复用),效率更高，调试方便
5. 备注
    1. 计算属性最终会出现在 vm 上，直接读取使用即可。
    2. 如果计算属性要被修改，必须写 set 函数去响应修改，并且 set

get() 正常写法

读取`computed`中的方法时，使用`get()`函数

```
    <div id="a">
        姓：<input type="text" v-model:value="first">
        <br>
        名：<input type="text" v-model:value="lastname">
        <br>
        <span>{{fn}}</span>
    </div>

    <script>
        const vm = new Vue({
            el:"#a",
            data:{
                first:"张",
                lastname:"三",
            },
            computed:{
                fn:{
                    // get的作用？当有人读取 fn方法时，get就会被调用，并且返回值就作为 fn
的值
                    // get什么时候调用？
                        // 1.初次读取 fn 方法时
                        // 2.所依赖的数据发生变化
                    get(){
                        console.log('get被调用了');
                        // this 指向 vm
                        return this.first+""+this.lastname;
                    }
                }
            }
        })
    </script>
```

get() 简写形式

计算属性`computed`中只用到`get`读取属性，就可以使用简写形式，**只读不写**

```
<div id="a">
    姓：<input type="text" v-model:value="first">
    <br>
    名：<input type="text" v-model:value="lastname">
    <br>
    <span>{{fn}}</span>
</div>

<script>
    const vm = new Vue({
        el:"#a",
        data:{
            first:"张",
            lastname:"三",
        },
        // computed 里面 fn函数的 简写形式
        computed:{
            fn(){
                console.log('get被调用');
                return this.first+'-'+this.lastname;
            }
        }
    })
</script>
```

set()

如果计算属性`computed`需要被修改，需要使用使用`set()`函数

```
<div id="a">
    姓：<input type="text" v-model:value="first">
    <br>
    名：<input type="text" v-model:value="lastname">
    <br>
    <span>{{fn}}</span>
</div>

<script>
    const vm = new Vue({
        el:"#a",
        data:{
            first:"张",
            lastname:"三",
        },
        computed:{
            fn:{
```

```
                    get(){
                        return this.first+'-'+this.lastname
                    },
                    //  set 什么时候被调用？ fn 被修改时调用
                    set(value){
                        console.log('set',value);
                        const arr = value.split('-');
                        this.first = arr[0];
                        this.lastname = arr[1];
                    }
                }
            }
        })
    </script>
```