



# Vue.js详解

## Lesson 5

### Vue计算属性



授课老师：爱米

# 计算属性 ——

○ 在模板中绑定表达式是非常便利的，但是它们实际上只用于简单的操作。

如果需要多于一个表达式的逻辑，应当使用计算属性。

# Vue.js 计算属性

- 1. 基础例子
- 2. 计算属性 vs. \$watch
- 3. 计算 setter

# 基数例子 ——

○ `<div id= "demo"> a = {{ a }}, b = {{ b }} </div>`

○ 

```
var vm = new Vue({  
  el: '#demo',  
  data: { a: 1 },  
  computed: { b: function () { return this.a + 1; } };  
});
```

○ 结果:  $a = 1, b = 2$

# 基数例子 ——

- 这里我们声明了一个计算属性**b**。我们提供的函数将用作属性**vm.b**的**getter**。

- ```
console.log(vm.b) // -> 2  
vm.a = 2  
console.log(vm.b) // -> 3
```

# 基数例子 ——

- 像绑定普通属性一样在模板中绑定计算属性。
- 这里可以是声明式地创建这种依赖关系——计算属性的 **getter** 是干净无副作用的，因此也是易于测试和理解的。

# 计算属性 vs. \$watch —

○ Vue.js提供了一个方法\$watch，用于观察Vue实例上的数据变动。

○ `<div id="demo">{{fullName}}</div>`

○ 更好的办法是使用计算属性而不是一个命令式的\$watch回调。

# 计算setter ——

○ 计算属性默认只是getter，不过在需要时也可以提供一个setter：

```
○ computed; {  
  fullName; {  
    // getter  
    get: function () {  
      return this.firstName + ' ' + this.lastName;  
    },  
    // setter  
    set: function (newValue) {  
      var names = newValue.split(' ');  
      this.firstName = names[0];  
      this.lastName = names[names.length - 1];  
    }  
  }  
}
```



# THANKS FOR WATCH