

Vue 数据响应式

从入门到工作 - Vue 全解

版权声明

本内容版权属杭州饥人谷教育科技有限公司（简称饥人谷）所有。

任何媒体、网站或个人未经本网协议授权不得转载、链接、转贴，
或以其他方式复制、发布和发表。

已获得饥人谷授权的媒体、网站或个人在使用时须注明「资料来源：饥人谷」。

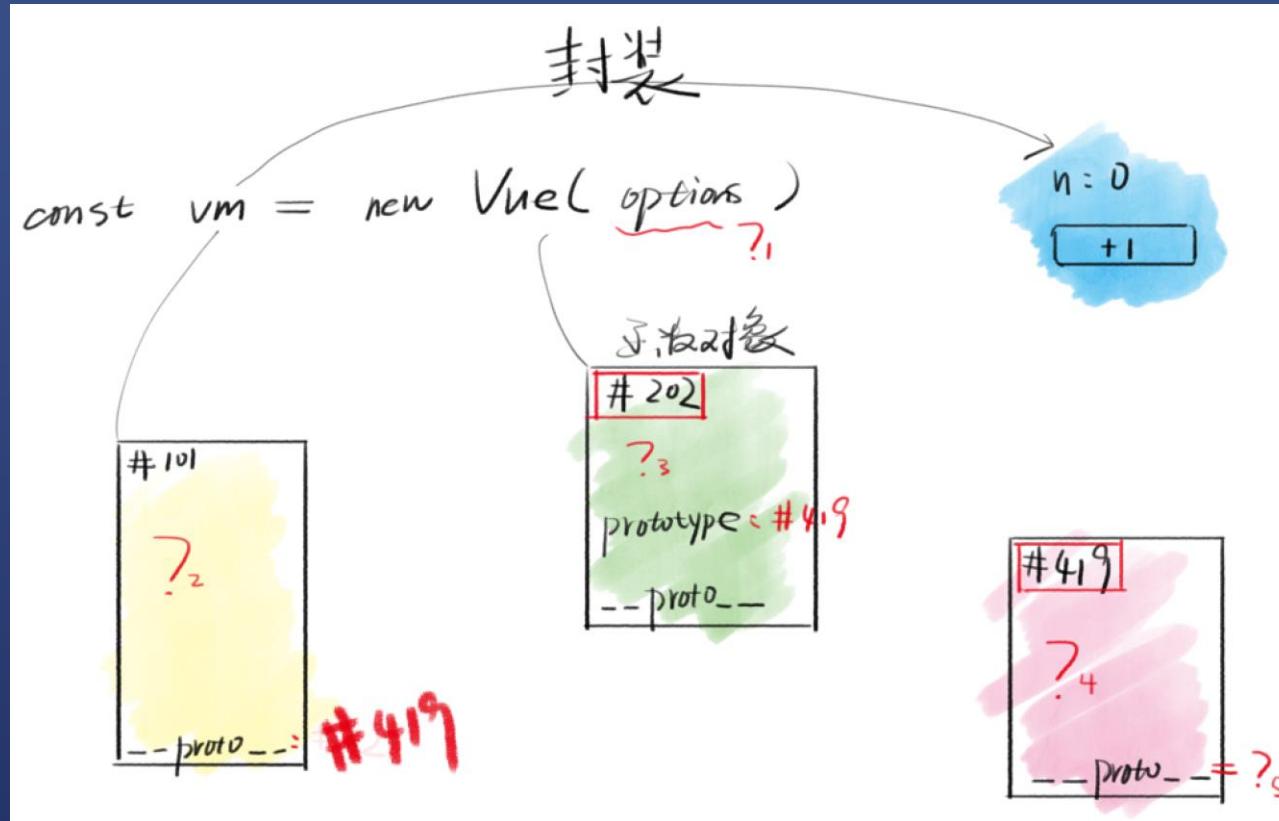
对于违反者，饥人谷将依法追究责任。

联系方式

如果你想要购买本课程
请微信联系 **xiedaimala02** 或 **xiedaimala03**

如果你发现有人盗用本课程
请微信联系 **xiedaimala02** 或 **xiedaimala03**

内存图



今天这节课，深入理解一下 `options.data`
方便我们后续学习 `options` 的进阶属性
主要原理在文档的「深入响应式原理」章节可以找到
但我扩充了非常多内容

Vue 对 data 做了什么

重要议题

小实验

- **data 变了**

- ✓ 示例代码: myData 居然变了
- ✓ 一开始是 {n:0}, 传给 new Vue 之后立马变成 {n: (...)}
- ✓ {n: (...) } 是个什么玩意, 为什么表现和 {n:0} 一致?
- ✓ 我们需要先学习一下 ES 6 的 getter 和 setter
- ✓ 示例代码
- ✓ 学完之后呢? 还是不理解 {n: (...) } 呀
- ✓ 我们需要再学习一下 Object.defineProperty
- ✓ 学完之后呢? 依然不理解
- ✓ 继续实验, 现在理解了吗? (见下页)

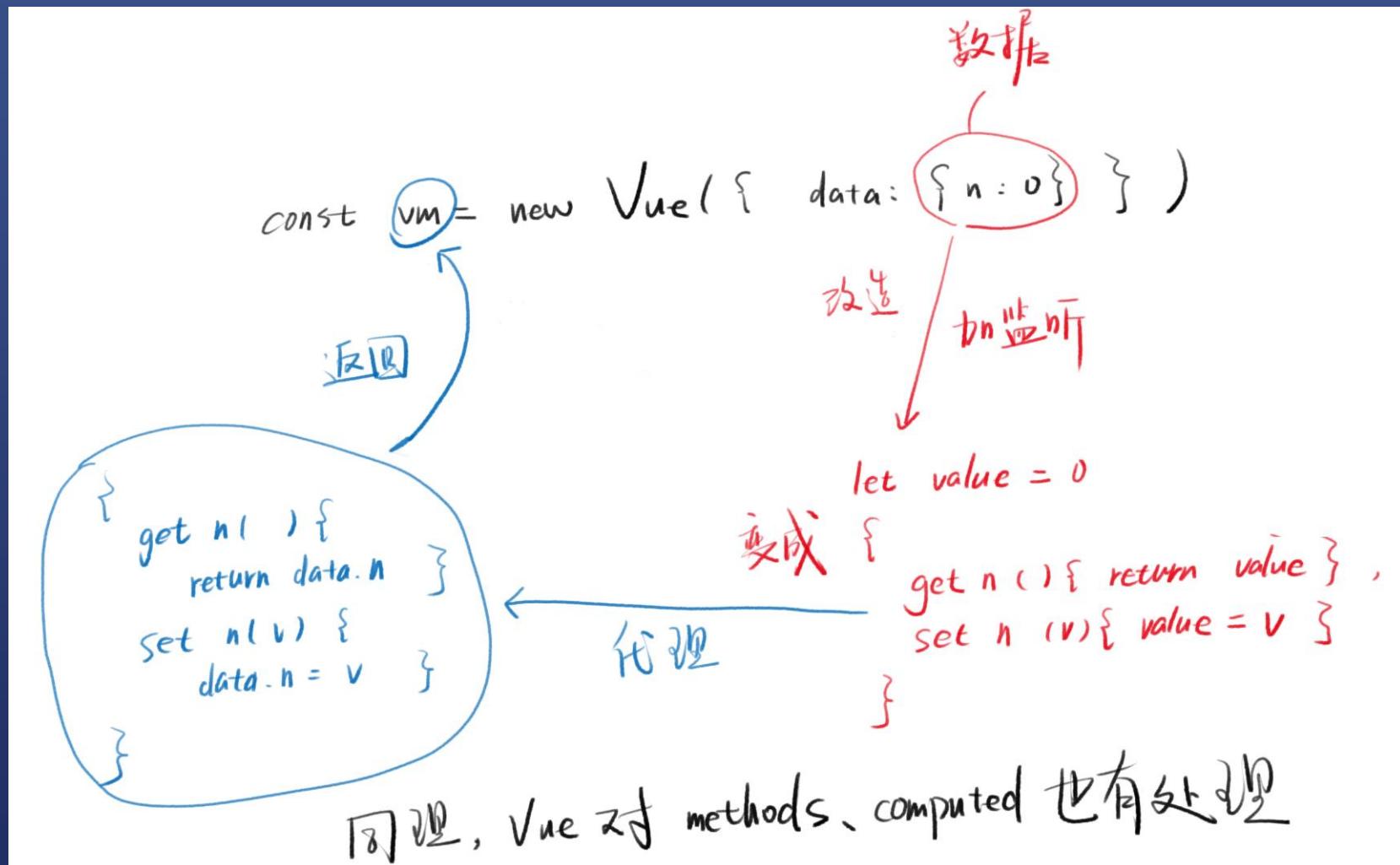
- **注意**

- ✓ 本节课的研究方法比知识本身更重要
- ✓ 这些方法能让你不读源码, 也能了解真相

小结

- **Object.defineProperty**
 - ✓ 可以给对象添加属性 value
 - ✓ 可以给对象添加 getter / setter
 - ✓ getter / setter 用于对属性的读写进行监控
- **啥是代理（设计模式）**
 - ✓ 对 myData 对象的属性读写，全权由另一个对象 vm 负责
 - ✓ 那么 vm 就是 myData 的代理（类比房东租房）
 - ✓ 比如 myData.n 不用，偏要用 vm.n 来操作 myData.n
- **vm = new Vue({data: myData})**
 - ✓ 一、会让 vm 成为 myData 的代理（proxy）
 - ✓ 二、会对 myData 的所有属性进行监控
 - ✓ 为什么要监控，为了防止 myData 的属性变了，vm 不知道
 - ✓ vm 知道了又如何？知道属性变了就可以调用 render(data) 呀！
 - ✓ UI = render(data)

示意图



如果 data 有多个属性 n、m、k，那么就会有 get n / get m / get k 等

数据响应式

- **什么是响应式**

- ✓ 我打你一拳，你会喊疼，那你就是响应式的
- ✓ 若一个物体能对外界的刺激做出反应，它就是响应式的

- **Vue 的 data 是响应式**

- ✓ `const vm = new Vue({data: {n: 0}})`
- ✓ 我如果修改 `vm.n`，那么 UI 中的 n 就会响应我
- ✓ Vue 2 通过 `Object.defineProperty` 来实现数据响应式

- **响应式网页是啥？**

- ✓ 如果我改变窗口大小，网页内容会做出响应，那就是响应式网页
- ✓ 比如 <https://www.smashingmagazine.com/>
- ✓ 但是你要注意，用户没事不会拖动网页大小的

你好好像说 Vue 有 bug

- **Object.defineProperty 的问题**

- ✓ Object.defineProperty(obj, 'n', {...})
- ✓ 必须要有一个 'n'，才能监听 & 代理 obj.n 对吧
- ✓ 如果前端开发者比较水，没有给 n 怎么办
- ✓ 示例一: Vue 会给出一个警告
- ✓ 示例二: Vue 只会检查第一层属性
- ✓ 此时如果我点击 set b，请问视图中会显示 1 吗？
- ✓ 答案是：不会
- ✓ 为什么：因为 Vue 没法监听一开始不存在的 obj.b

- **解决办法**

- ✓ 一、那我把 key 都声明好，后面不再加属性不就行了
- ✓ 二、使用 Vue.set 或者 this.\$set

Vue.set 和 this.\$set

- **作用**

- ✓ 新增 key
- ✓ 自动创建代理和监听 (如果没有创建过)
- ✓ 触发 UI 更新 (但并不会立刻更新)

- **举例**

- ✓ `this.$set(this.object, 'm', 100)`

data 中有数组怎么办？

- 你没法提前声明所有 key
 - ✓ 示例1: 数组的长度可以一直增加，下标就是 key
 - ✓ 你看，你没有办法提前把数组的 key 都声明出来
 - ✓ Vue 也不能检测对你新增了下标
 - ✓ 难道每次改数组都要用 Vue.set 或者 this.\$set
- 尤雨溪的做法
 - ✓ 篡改数组的 API，见文档中「变异方法」章节
 - ✓ 这 7 个 API 都会被 Vue 篡改，调用后会更新 UI
 - ✓ 怎么篡改的？见下页 PPT

ES 6 写法

```
class VueArray extends Array{
  push(...args){
    const oldLength = this.length // this 就是当前数组
    super.push(...args)
    console.log('你 push 了')
    for(let i = oldLength; i<this.length; i++){
      Vue.set(this, i, this[i])
      // 将每个新增的 key 都告诉 Vue
    }
  }
}
```

注：这不代表 Vue 的真实实现，此代码仅用于教学目的
实际上我没看过相关源代码

ES 5 写法 - 原型

```
const vueArrayPrototype = {
  push: function(){
    console.log('你 push 了')
    return Array.prototype.push.apply(this, arguments)
  }
}
vueArrayPrototype.__proto__ = Array.prototype
// 上面这句话用的不是标准属性，仅学习使用
```

```
const array = Object.create(vueArrayPrototype)
array.push(1)
// 估计你看不懂，看不懂就算了，反正面试官也看不懂
```

总结

- 对象中新增的 key
 - ✓ Vue 没有办法事先监听和代理
 - ✓ 要使用 set 来新增 key，创建监听和代理，更新 UI
 - ✓ 最好提前把属性都写出来，不要新增 key
 - ✓ 但数组做不到「不新增 key」
- 数组中新增的 key
 - ✓ 也可用 set 来新增 key，~~且创建监听和代理~~，更新 UI
 - ✓ 不过尤玉溪篡改了 7 个 API 方便你对数组进行增删
 - ✓ 这 7 个 API 会自动处理监听和代理，并更新 UI
 - ✓ 结论：数组新增 key 最好通过 7 个 API

给大家出一个刁钻的题

- **题目链接**

- ✓ 请回答第2题、第3题和第4题
- ✓ 答案解析见这篇文章

理解了 data 才能继续学习
computed 和 watch

options 还没讲完

下节课见