# [JS 对象之扩展、密封及冻结三大特性](https://segmentfault.com/a/1190000003894119)

扩展特性

Object.isExtensible 判断对象是否可扩展

Object.preventExtensions 将对象变为不可扩展

密封特性

Object.isSealed 判断对象是否密封

Object.seal 将一个对象密封

冻结特性

Object.isFrozen 判断一个对象是否冻结

Object.freeze 冻结一个对象

获取计算后的样式属性(IE9以上支持，只读不可写)

getComputedStyle(document.getElementByClassName(‘…’))[‘top’]

# VUE

## 阻止默认事件

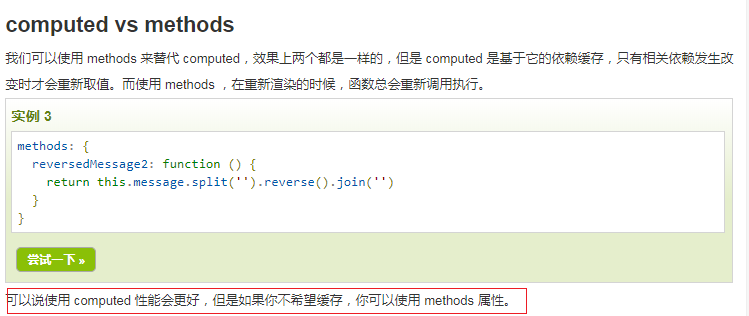
<form v-on:submit.prevent="onSubmit">...</form>

.prevent 修饰符告诉 v-on 指令对于触发的事件调用 event.preventDefault()

## Computed

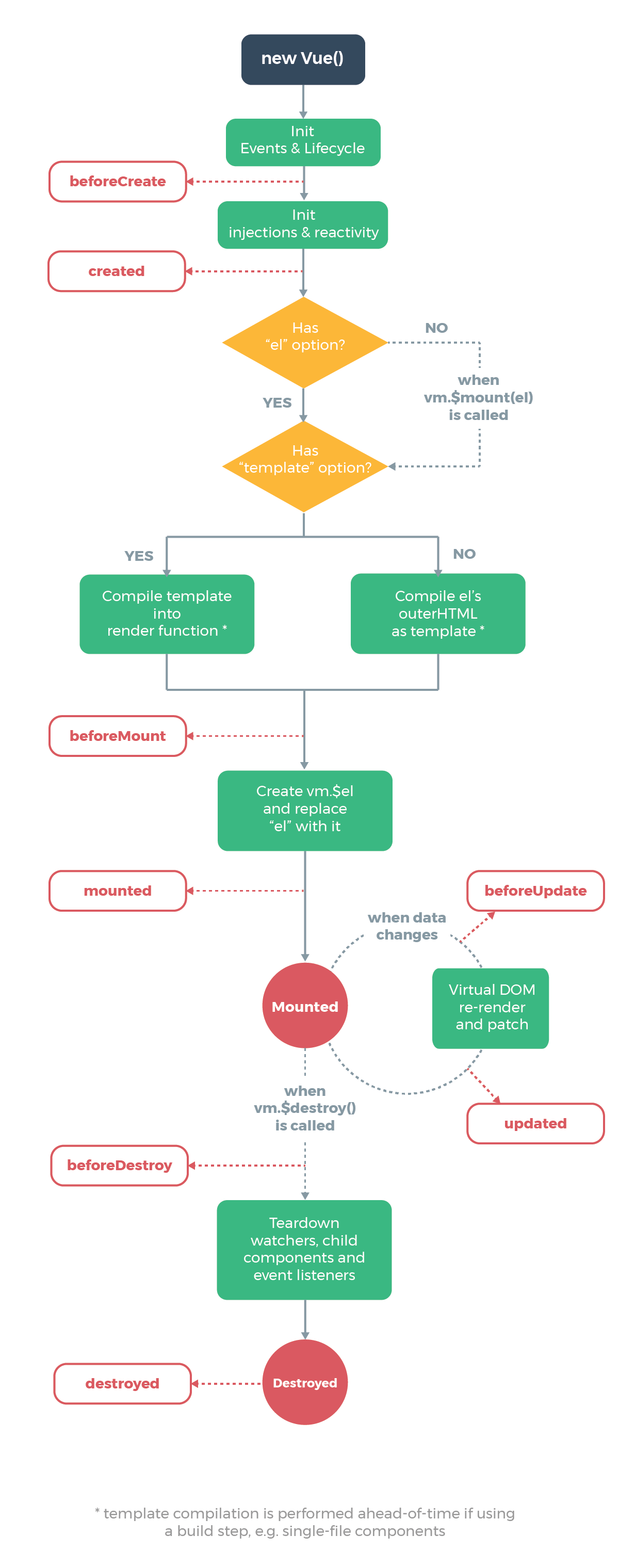
假设我们有一个性能开销比较大的的计算属性 **A**，它需要遍历一个巨大的数组并做大量的计算。然后我们可能有其他的计算属性依赖于 **A** 。如果没有缓存，我们将不可避免的多次执行 **A** 的 getter！如果你不希望有缓存，请用方法来替代。

## Computed与methods





## VUE生命周期



## 注意事项

由于 JavaScript 的限制，Vue 不能检测以下变动的数组：

1. 当你利用索引直接设置一个项时，例如：vm.items[indexOfItem] = newValue
2. 当你修改数组的长度时，例如：vm.items.length = newLength

为了解决第一类问题，以下两种方式都可以实现和 vm.items[indexOfItem] = newValue 相同的效果，同时也将触发状态更新：

|  |
| --- |
| // Vue.set Vue.set(example1.items, indexOfItem, newValue) |

|  |
| --- |
| // Array.prototype.splice example1.items.splice(indexOfItem, 1, newValue) |

为了解决第二类问题，你可以使用 splice：

|  |
| --- |
| example1.items.splice(newLength) |

# 正则表达式

// ip (127.0.0.1)

ip: /^(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[1-9]{1}[0-9]{1}|[1-9])\.(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[1-9]{1}[0-9]{1}|[1-9]|0)\.(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[1-9]{1}[0-9]{1}|[1-9]|0)\.(25[0-5]|2[0-4][0-9]|1[0-9]{2}|[1-9]{1}[0-9]{1}|[0-9])$/,

// port (0～65535)

port: /^([0-9]|[1-9]\d{1,3}|[1-5]\d{4}|6[0-5]{2}[0-3][0-5])$/,

// email

email: /(^\w+((-\w+)|(\.\w+))\*@[A-Za-z0-9]+((\.|-)[A-Za-z0-9]+)\*\.[A-Za-z0-9]+$)|(^$)/,

// mac地址

macAddr: /^([A-Fa-f0-9]{2}-){5}[A-Fa-f0-9]{2}$/,

// 数字

number: /^\d\*$/,

// 自然数 0 1 2 3 4

naturalNum: /^(0|[1-9]\d\*)$/,

// 正整数 1 2 3 4

positiveNum: /^[1-9]\d\*$/,

// 1～32个字符；不能包含 ' / \ : \* ? " < > | 等特殊字符

name: /^[^'\/\\:\\*\?"<>\|]\*$/,

// 1-32个字符，只能用字母、数字、汉字、小数点、下划线、连接符、括号 「设备型号」

deviceType: /^[a-zA-Z0-9\.\-\_\(\)\u4E00-\u9FA5]\*$/,

// 字母、数字、汉字、小数点、下划线、连接符 [设备编码]

deviceCode: /^[a-zA-Z0-9\.\-\_\u4E00-\u9FA5#()\s]\*$/,

// 设备登录用户名 1～32个字符，只能用字母、数字、汉字、小数点、下划线、连接符。

deviceUserName: /^[a-zA-Z0-9\.\-\_\u4E00-\u9FA5]\*$/,

// 字母、数字的组合 「设备编号1-32位」

numLetter: /^[\da-zA-Z]\*$/,

// 字母、数字、汉字的组合 「工号1-32」「入场播报」「出场播报」「放行播报」

numLetterAndCN: /^[\da-zA-Z\u4E00-\u9FA5]\*$/,

// 字母、汉字的组合 「职位1-128」

numAndCN: /^[a-zA-Z\u4E00-\u9FA5]\*$/,

// 字母、数字和除:\"之外的特殊字符 「设备账号1-32位」

deviceAccount: /^[a-zA-Z0-9~`!@#\$%^&\*()\_+\-=\[\];',<.>/|?]{1,32}$/,

// 车牌号

licenseNumber: /^.{1,16}$/,

// 手机号

phoneNum: /^\d{1,11}$/,

// 姓名 1-128个数字、字母、汉字、间隔号（·）和空格

personName: /^[\da-zA-Z\u4E00-\u9FA5· ]{1,128}$/,

// 证件号码

IDNumber: /^[0-9a-zA-Z]{0,20}$/,

// 身份证号 18位，前17位为数字，尾号为数字或大写字母X。

identityCard: /^\d{17}[\d|X]$/,

// 密码为8-16位字符，至少由大写字母、小写字母、数字、特殊字符任意两种组成。

password: /^(?![A-Z]+$)(?![a-z]+$)(?!\d+$)(?![\W\_]+$)\S{8,16}$/,

// 卡号 8-18个字符，只能用数字和大写字母。

cardNum: /^[0-9A-Z]{8,18}$/,

// 4-8位数字 「卡密码」「胁迫密码」「超级密码」

simplePassword: /^\d{4,8}$/,

// UUID 32位大写英文字母、数字、连接符。

uuid: /^[a-zA-Z\d]{8}-[a-zA-Z\d]{4}-[a-zA-Z\d]{4}-[a-zA-Z\d]{4}-[a-zA-Z\d]{12}$/

# 对输入框绑定键盘事件

<input v-model="dataInput" v-on:keyup.13="addDataItem()" ></input>

# 英文单词换行问题

word-break: keep-all;

word-wrap: break-word;

white-space: pre-wrap;

# 事件修饰符



