

途虎养车

养车 就是途虎

# 《Vue混入与渲染函数》

郑琴--门店开发组

# 大纲总览

- 混入

- 基础

- 选项合并

- 全局混入

- 渲染函数

- 思考

- render函数

- render参数介绍

- 渲染函数的好处

- QA

混入

虎

# 1、混入定义

定义：分发Vue组件中的可复用功能

一个混入对象可以  
包含任意组件选项

data

props

computed

watch

methods

生命钩子函数

template

mixins

等

## 思考

```
var mixin = {  
  data: function () {  
    return {  
      message: 'mixin混入',  
    }  
  },  
  created: function () {  
    console.log('混入对象的钩子被调用')  
  }  
}
```

```
new Vue({  
  mixins: [mixin],  
  data: function () {  
    return {  
      message: 'inside',  
    }  
  },  
  created: function () {  
    console.log('内部钩子调用')  
  }  
})
```

## 2、选项合并

### 一、数据对象



组件数据优先

递归合并

## 2、选项合并

## 二、钩子函数



结论：同名钩子函数将合并为一个数组，因此都将被调用。另外，混入对象的钩子将在组件自身钩子之前调用。

## 2、选项合并

## 三、值为对象的选项



取组件对象的键值

递归合并



## 2、选项合并

## 四、其他

例如：mixins、el、template等

```
var defaultStrat = function (parentVal,childVal) {  
  return childVal === undefined  
    ? parentVal  
    : childVal  
};
```

### 3、自定义合并策略

```
Vue.config.optionMergeStrategies._my_option  
= function (parent, child, vm) {  
    //自定义策略函数处理  
}
```

## 4、全局混入

```
Vue.mixin({  
  mounted() {  
    console.log('全局混入');  
  }  
})
```

**慎用：** 影响创建的每一个Vue实例，多应用于插件开发

# 渲染函数

## 1、思考

根据不同的等级，渲染出不同大小的标题



## 2、render函数

首先render函数生成的内容相当于template的内容，故使用render函数时，在.vue文件中需要先把template标签去掉。只保留逻辑层

### 3、render函数

```
render: function(createElement){  
    return createElement(tag, options, children);  
}
```

返回结果：Vnode（虚拟节点）

### 3、render参数

tag: {String | Object | Function} 定义的元素

```
//String 标签名 上述任何一种的一个 async 函数
return createElement(h() {});
// Object 组件选项对象
return createElement({
  template: "<div>锄禾日当午</div>"
})
return createElement(domFun())
```



### 3、render参数

options: 属性对应的数据对象

```
{  
  'class': {foo: true}  
  style: {color, width...}  
  attrs: {id, href, src...}  
  props  
  domProps: {innerHTML...}  
  on: 事件监听器  
  nativeOn: 监听原生事件  
}
```

### 3、render参数

children: 子级虚拟节点(VNodes)    String | Array

```
return createElement('div', {  
  attrs: {  
    id: "content"  
  }  
}, 'hello world' )
```



<div id="content">hello world</div>

### 3、render参数

children: 子级虚拟节点(VNodes)    String | Array

```
return createElement('div', {  
  attrs: {  
    id: "content"  
  }  
}, [  
  createElement('h1', '我是H1标题'),  
  createElement('h6', '我是H6标题')  
]  
)
```

## 4、约束

组件树中的所有 VNode 必须是唯一的

```
var ele = createElement('h1', '我是H1标题');  
return createElement('div', {  
  attrs: {  
    id: "content"  
  }  
}, [  
  ele, ele  
)
```

## 5、v-if&v-for

只要在原生的 JavaScript 中可以轻松完成的操作，Vue 的渲染函数就不会提供专有的替代方法

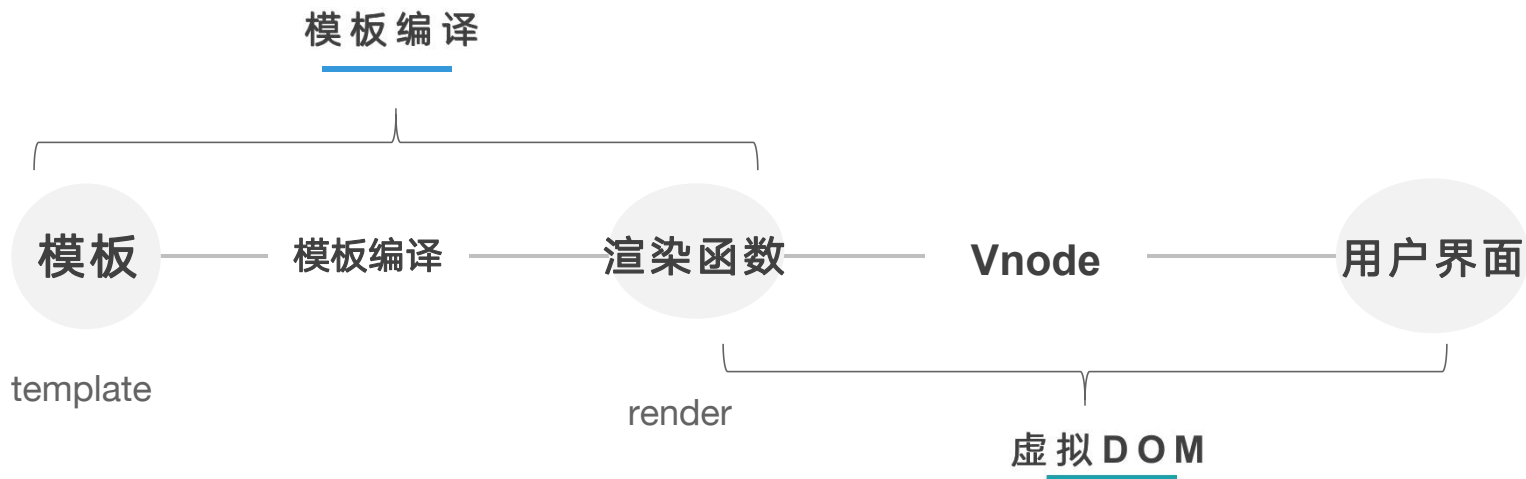
```
<ul v-if="items.length">  
  <li v-for="item in items">{{ item.name }}</li>  
</ul>  
<p v-else>No items found.</p>
```

## 5、v-if&v-for

渲染函数中用 JavaScript 的 if/else 和 map 来重写

```
if (this.items.length) {  
  return createElement('ul', this.items.map(function (item) {  
    return createElement('li', item.name)  
  })))  
} else {  
  return createElement('p', 'No items found.')  
}
```

## 6、渲染过程

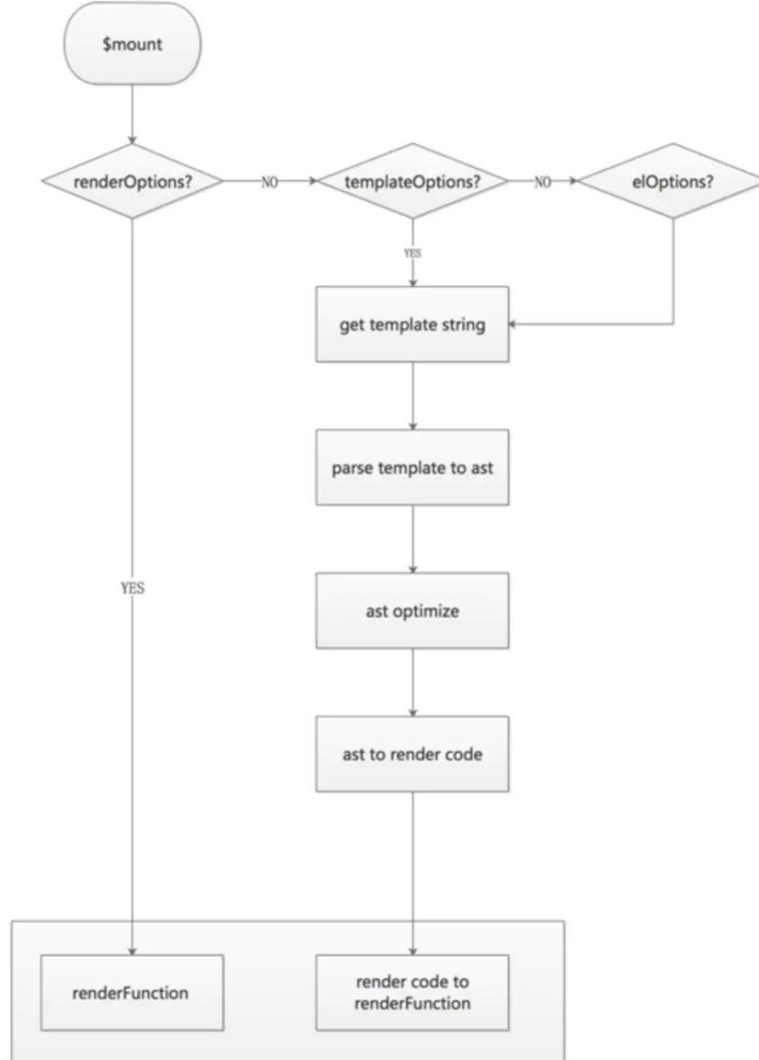






## 7、与template对比

- Template适合逻辑简单，render适合复杂逻辑。
- 使用者template理解起来相对容易，但灵活性不足；自定义render函数灵活性高，但对使用者要求较高。
- render的性能较高，template性能较低。
- 基于上一点，我们通过vue组件渲染流程图知道，使用render函数渲染没有编译过程，相当于使用者直接将代码给程序。所以，使用它对使用者要求高，且易出现错误
- Render 函数的优先级要比template的级别要高，但是要注意的是Mustache(双花括号)语法就不能再次使用



# 课后练习



Q&A

# 途虎养车

养车 就是途虎