

## 学习记录

### 1. 学习vue \$router go replace push 的区别

router.go(n) 意思是history记录返回n步，相当于window.history.go(n)

router.push() 导航到不同的url，像history栈中添加新纪录 点浏览器的返回按钮  
返回

router.replace() 导航到不同的url不同的是不会像history中添加，而是替换掉当前history

### 2. 学习element-ui 尤其是dropdown组件中的方法 和属性

属性：

type	菜单按钮类型，同 Button 组件(只在 split-button 为 true 的情况下有效)	string	—	—
size	菜单尺寸，在 split-button 为 true 的情况下也对触发按钮生效	string	medium / small / mini	—
split-button	下拉触发元素呈现为按钮组	boolean	—	false
placement	菜单弹出位置	string	top/top-start/top-end/bottom/bottom-start/bottom-end	bottom-end
trigger	触发下拉的行为	string	hover, click	hover
hide-on-click	是否在点击菜单项后隐藏菜单	boolean	—	true
show-timeout	展开下拉菜单的延时（仅在 trigger 为 hover 时有效）	number	—	250
hide-timeout	收起下拉菜单的延时（仅在 trigger 为 hover 时有效）	number	—	150

方法：

#### Dropdown Events

事件名称	说明	回调参数
click	split-button 为 true 时，点击左侧按钮的回调	—
command	点击菜单项触发的事件回调	dropdown-item 的指令
visible-change	下拉框出现/隐藏时触发	出现则为 true，隐藏则为 false

下拉菜单项属性：

### Dropdown Menu Item Attributes

参数	说明	类型	可选值	默认值
command	指令	string/number/object	—	—
disabled	禁用	boolean	—	false
divided	显示分割线	boolean	<a href="#">—</a>	<a href="#">false</a>

3. 熟悉供应链pas平台代码结构 主要用到的点为vue ES6 element-ui v-charts等。进一步学习其中的各个知识块 争取本项目结束后能够运用熟练

已解决

1. 看vue官方文档中对于.native的描述 不是很理解 这是文档的截图

### 将原生事件绑定到组件

你可能有很多次想要在一个组件的根元素上直接监听一个原生事件。这时，你可以使用 `v-on` 的 `.native` 修饰符：

```
<base-input v-on:focus.native="onFocus"></base-input>
```

HTML

在网上查了查什么叫直接监听一个原生事件，大致就是vue中的组件是没有原生所包含的一些方法的例如click等 所以我们需要对其添加.native 来达到类似原生的效果 而其实内部也是对其添加了addEventListener方法类似实现

2



已采纳


原生就类似于你直接用

```
$element.addEventListener(click, callback);
```

绑定事件。

`router-link` 会阻止 `click` 事件，你可以试试只用 `click` 不用 `native`，事件是不会触发的。  
而 `a` 标签不会阻止。

2016年12月23日回答 2 评论 赞赏 编辑

 sdykjym  
110 声望

1 是的。

我是这么认为的，因为 `router-link` 的作用是单纯的路由跳转啊，如果不阻止事件的话，会有很多坑吧。你可以试试在 `router-link` 绑定其他事件看看，都不会触发的。除非加了 `native`

— sdykjym · 2016年12月23日

[展开评论](#)

16

意思就是当你给一个vue组件绑定事件时候，要加上`native`！如果是普通的html元素！就不需要

2017年06月12日回答 5 评论 赞赏 编辑

 hackjus  
72 声望

2 感觉`native`就是把组件变回原生DOM的一种方式，不知这样说对不对 — LuFeng · 2017年10月17日

3 精辟！非常精辟！！这才是真理 — 啦啦啦 · 2017年07月28日

[展开评论](#)