1.dropdown列表显示/隐藏方法

效果：

（1）可配置menu的显示方式为hover或click。当为hover时，triggerElm和dropdownElm当mouseenter时触发show，mouseleave时触发hide。

（2）如果是click，点击触发handleClick，toggle下拉menu。

（3）如何控制点击下拉列表以外时，隐藏menu？是通过引入clickoutside.js实现，本质还是通过document代理mousedown和mouseup，触发绑定方法实现。

首先在dropdown.vue中将clickoutside作为自定义指令，然后再下拉菜单元素上应用自定义指令。

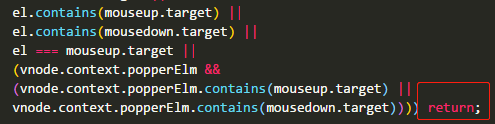




在clickoutside.js中，首先需要在document上代理下拉菜单的mousedown和mouseup事件。



当事件目标是下拉菜单，不触发事件，不是，触发事件，进而隐藏下拉菜单。



其他说明：

（1）tabindex只是控制tab用的，没有其他作用。

（2）focus、blur也没有其他作用，只是用于给dropdown添加1个class。

2.modal

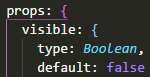
（1）dialog、message-box等都使用到了modal。modal的基础组件为utils/popup。

（2）以dialog为例，首先将Popup作为mixins注入到组件中。



（3）控制dialog显示的是visible变量，在popup中定义。





（4）visible会在多个地方被watch。在dislog中，主要控制dislog自身，包括设置dislog的scrollTop为0，以及是否将dislog挂载到body下。

（5）在Popup中，主要控制modal的显示。具体为：

首先在beforeMount中，将dislog保存到popup-manager中的静态变量中。





如果观察到visible为true时，在第1次渲染时，首先标识dialog的rendered属性为true。此时也没有zIndex，先不管（后面从PopupManager中获取）。然后调用PopupManager的openModal方法创建modal。

openModal中做了3件事：创建唯一的modalDom，作为PopupManager的属性，以后就直接返回它。将dialog的id等属性作为1个对象保存到modalStack数组中。设置zIndex。

（6）zIndex相当重要，除自定义zIndex外，都是通过PopupManager的nextZIndex方法控制：首先基础的zIndex为2000，第1次用到modal就是2000，之后的如dialog第1次创建时，通过nextZIndex自增1。如果自定义了zIndex（通常都会自定义），会先重置PopupManager中的zIndex，然后设置modalDom为nextZIndex自增1。

关闭modal，调用PopupManager的closeModal，首先获取modalStack最上面的那个，重置modalDom的zIndex为modalStack保存的zIndex。当最后1个需要modal的弹窗关掉时，移除modal（动画）。