# 创建标签

var btn=document.createElement("BUTTON");

# 元素添加事件

# btn.addEventListener("click", function(){}) ---🡪 removeEventListener()

# .html() ---＞这个可以添加标签　　　.text()-🡪这个是只能是文字

# 在遍历DOM时，通常用$(selector).each(function(index,element))函数；

# 在遍历数据时，通常用$.each(dataresource,function(index,element))函数。

# var jsonResourceList = '

# [{"id":"1","tagName":"apple"},

# {"id":"2","tagName":"orange"}]';

# if(jsonResourceList.length >0){ $.each(JSON.parse(jsonResourceList), function(index, obj) {     alert(obj.tagName); }); }

# 阻止事件冒泡：

# <div id="app">

# <div @click="dj1">

# <button type="button" class="btn" @click="dj2($event)">按钮</button>

# </div>

# </div>

# <script>

# var app = new Vue({

# el:'#app',

# data:{

# },

# methods:{

# dj1:function(){

# console.log(1);

# },

# dj2:function(ev){

# console.log(2);

# }

# }

# })

# </script>

# 会返回 2 后还会返回 1；

# 阻止冒泡：

# 1：ev.cancelBubble=true; （原生）

# 2：@click.stop="dj2()" （vue 推荐）

# 阻止右键默认事件：没有会有刷新什么的出现

# 1:ev.preventDefault(); (原生)

# 2：@contextmenu.prevent="dj2()" （vue 推荐）

# 键盘事件：

# 1：ev.keyCode

# 2:@keyup.13=’dj(2)’ 13的意思是回车 或是@keyup.enter

# 属性；

# v-bind 比直接 {{}} 好 v-bind:src ----🡪 简写 ：src