FastClick : 移动端点击延迟事件。（清除这个延迟）

移动端浏览器在派发点击事件是，会出现300ms的延迟

原因：移动端的双击会缩放会导致点击事件延迟

解决方法：

1.禁用缩放  `<meta name = "viewport" content="user-scalable=no" > `

缺点: 网页无法缩放

2.更改默认视口宽度 `<meta name="viewport" content="width=device-width">`

缺点: 需要浏览器的支持

3.css touch-action touch-action的默为 auto，将其置为 none 即可移除目标元素的 300 毫秒延迟

缺点: 新属性，可能存在浏览器兼容问题

4. tap事件 zepto的tap事件, 利用touchstart和touchend来模拟click事件

  缺点: 点击穿透

5.FastClick 原理: 在检测到touchend事件的时候，会通过DOM自定义事件立即出发模拟一个click事件，并把浏览器在300ms之后真正的click事件阻止掉

缺点: 脚本相对较大

使用方法

1. js 引入 <script type='application/javascript' src='/path/to/fastclick.js'></script>
2. Jquery

$(function() {

    FastClick.attach(document.body);

 });

1. 原生js

if ('addEventListener' in document) {

        document.addEventListener('DOMContentLoaded', function() {

            FastClick.attach(document.body);

        }, false);

    }

String.fromCharCode() unicode编码传唤为字符串

$router.go(-1) 返回上一级路由

路由传参数 :to="'/city/' + item.id" this.$route.params.cityid 接收

fetch请求 传统 Ajax 指的是 XMLHttpRequest（XHR），现在已被 Fetch替代;

优点：语法简洁，更加语义化

基于标准 Promise 实现，支持 async/await

同构方便，使用 [isomorphic-fetch](https://github.com/matthew-andrews/isomorphic-fetch" \t "https://www.cnblogs.com/liufei1983/p/_blank)

aSync 用来处理异步

它作为一个关键字放到函数前面，用于表示函数是一个异步函数，因为async就是异步的意思， 异步函数也就意味着该函数的执行不会阻塞后面代码的执行。

async 函数返回的是一个promise 对象 应该用then 方法

async function timeout() {

return 'hello world'

}

timeout().then(result => {

console.log(result);

})

console.log('虽然在后面，但是我先执行');

1013082-20180205151306873-401823311、

Await 用来处理同步

await的含义为等待。意思就是代码需要等待await后面的函数运行完并且有了返回结果之后，才继续执行下面的代码。

注：await必须放在async中

scrollBehavior(to,from,savePosition){

to 指要进入的路由对象

from 指要离开的路由对象

savePosition 会记录滚动条的坐标

If(savePosition){

return

}else{

Return {x:0,y:0}

}

}

页面刷新 滚动条依旧停留在原地（浏览器默认行为，会记录浏览器滚动条默认位置），但你希望它从头开始，就可以用这个函数