# 重构的收获

1. String.fromCharCode（numX） 可接受一个指定的 Unicode 值，然后返回一个字符串。 1

参数：numX ，必填的，一个或多个 Unicode 值。

2. @click="$router.go(-1)" 返回上一个页面

### 3. 移动端点击延迟事件

1. 移动端浏览器在派发点击事件的时候，通常会出现300ms左右的延迟

2. 原因: 移动端的双击会缩放导致click判断延迟

# 解决方法：Fastclick

// 安装 cnpm install fastclick -S

// 引入 import FastClick from 'fastclick'

    // 使用 FastClick.attach(document.body);

使用js 引用：<script type='application/javascript' src='/path/to/fastclick.js'></script>

使用jquery ： $(function() {

        FastClick.attach(document.body);

    });

// 没使用jquery的时候

    if ('addEventListener' in document) {

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

            FastClick.attach(document.body);

        }, false);

    }

# 4.go back与return有什么区别

go back回去,回到某一个地方  
 return是从A地到B地,然后强调了又从B地回到A地  
 而且return还有一个用法就是指借了东西,归还回去

# 5. [Fetch](https://www.cnblogs.com/libin-1/p/6853677.html)

**Fetch获取数据**

假设我们想通过GitHub获取一个仓库，我们可以像下面这样使用：

fetch('https://api.github.com/users/chriscoyier/repos');

Fetch会返回Promise，所以在获取资源后，可以使用.then方法做你想做的。

fetch('https://api.github.com/users/chriscoyier/repos')

.then(response => {/\* do something \*/})

Github返回的响应是JSON格式的，所以调用response.json方法来转换数据。还有其他方法来处理不同类型的响应。如果请求一个XML格式文件，则调用response.text。如果请求图片，使用response.blob方法

# 列如： fetch('https://api.github.com/users/chriscoyier/repos')

# .then(response => response.json())

# .then(data => {

# // data就是我们请求的repos

# console.log(data)

# });

# Fetch发送数据

使用Fetch发送也很简单，只需要配置三个参数。

第一个参数是设置请求方法（如post、put或del），Fetch会自动设置方法为get。

第二个参数是设置头部。因为一般使用JSON数据格式，所以设置ContentType为application/json。

第三个参数是设置包含JSON内容的主体。因为JSON内容是必须的，所以当设置主体时会调用JSON.stringify。

列如： let content = {some: 'content'};

// The actual fetch request

fetch('some-url', {

method: 'post',

headers: {

'Content-Type': 'application/json'

},

body: JSON.stringify(content)

})

// .then()...

**6. async**

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

async 函数返回的是一个promise 对象，如果要获取到promise 返回值，我们应该用then 方法。Promise 有一个resolved，这是async 函数内部的实现原理，如果async 函数中有返回一个值 ,当调用该函数时，内部会调用Promise.solve() 方法把它转化成一个promise 对象作为返回，但如果timeout 函数内部抛出错误呢？ 那么就会调用Promise.reject() 返回一个promise 对象， 这时修改一下timeout 函数

async function timeout(flag) {

if (flag) {

return 'hello world'

} else {

throw 'my god, failure'

}

}

console.log(timeout(true)) // 调用Promise.resolve() 返回promise 对象。

console.log(timeout(false)); // 调用Promise.reject() 返回promise 对象**。**