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

Vue-mixins

mixins的概念的理解：混入 (mixins) 是一种分发 Vue 组件中可复用功能的非常灵活的方式。混入对象可以包含任意组件选项。当组件使用混入对象时，所有混入对象的选项将被混入该组件本身的选项。

mixins的特点:

1 方法和参数在各组件中不共享

2 值为对象的选项，如methods,components等，选项会被合并，键冲突的组件会覆盖混入对象的

混入对象中的方法

3 值为函数的选项，如created,mounted等，就会被合并调用，混合对象里的钩子函数在组件里的钩子函数之前调用

混入对象函数中的console

与vuex的区别:

vuex：用来做状态管理的，里面定义的变量在每个组件中均可以使用和修改，在任一组件中修改此变量的值之后，其他组 件中此变量的值也会随之修改。

Mixins：可以定义共用的变量，在每个组件中使用，引入组件中之后，各个变量是相互独立的，值的修改在组件中不会相 互影响。

js刷新页面location.reload()用法

注意：

window.location.reload;

刷新时如果提交数据的动作，则会出现对话框！

解决

代码示例:

window.location.href=window.location.href;

window.location.reload;

router-view 也是一个组件，如果直接被包在 keep-alive 里面，所有路径匹配到的视图组件都会被缓存：在APP.js中，

https://www.jianshu.com/p/0b0222954483

// routes 配置

export default [

{

path: '/',

name: 'home',

component: Home,

meta: {

keepAlive: true // 需要被缓存

}

}, {

path: '/:id',

name: 'edit',

component: Edit,

meta: {

keepAlive: false // 不需要被缓存

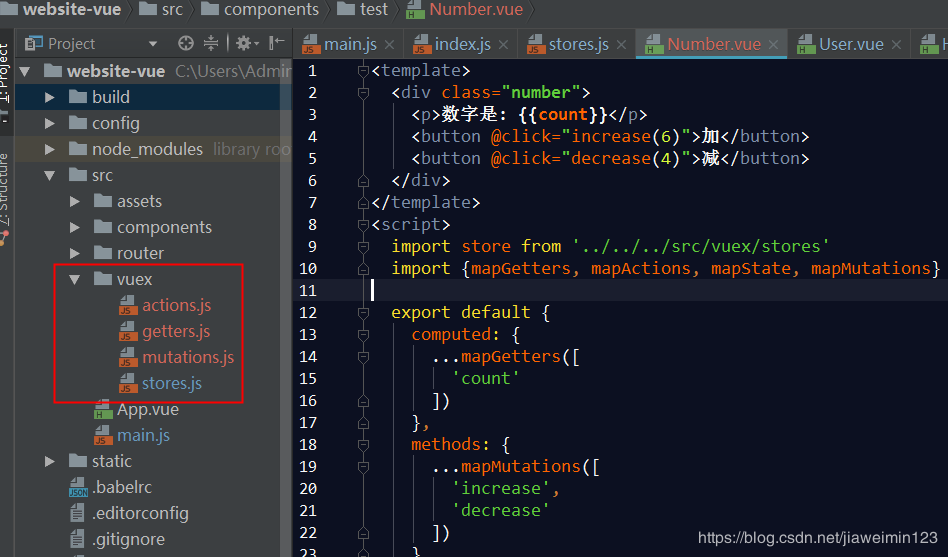
}

}

]

mapMutations映射方法的问题 /mapMutations辅助函数传参

1. 新建文件夹，vuex，里面存放的是store相关的js，跟redux一样，相关的文件分离开来。



1. store.js的内容，该js为主入口，也是默认导出的，全局可import这个js来获取和修改state数据。示例是加减计数的例子， 所以有个初始值count=0.

import Vue from 'vue'

import Vuex from 'vuex'

import \* as getters from './getters'

import \* as actions from './actions'

import \* as mutations from './mutations'

Vue.use(Vuex);

const state = {

count: 0

};

const store = new Vuex.Store({

state,

getters,

actions,

mutations

});

1. getters.js文件是获取默认state的值的，该文件将所以state绑定到store里去，别的地方都可以共享state。

export const count = state => state.count;

4、actions.js是用户操作，跟react-redux里面的action很相似，这里面有2个提交操作，点击增加和点击减少。

export const increase = ({commit}) => commit('increase');

export const decrease = ({commit}) => commit('decrease');

5、mutations.js文件是真实的数据变化，比如增加多少，减少多少，或者对值的修改，这里的写法是传参的形式。第一个参数 state是默认的，第二个参数则是点击时绑定传过来的参数。

export const increase = (state,m) => {

state.count += m

};

export const decrease = (state,n) => {

state.count -= n

};

这样的话，vuex的相关js文件以及添加完毕，现在需要对view层进行展示的代码修改。

6、新建一个Number.vue组件，用了展示效果。两个按钮，对应两个action，绑定点击事件并把参数传入。我的是加6，减4。 通过mapGetters来拿到每次的count的值，通过mapMutations来拿到以及写好的两个点击事件，并绑定到对应的按钮上。

<template>

<div class="number">

<p>数字是: {{count}}</p>

<button @click="increase(6)">加</button>

<button @click="decrease(4)">减</button>

</div>

</template>

<script>

import store from '../../../src/vuex/stores'

import {mapGetters, mapMutations} from 'vuex'

export default {

computed: {

...mapGetters([

'count'

])

},

methods: {

...mapMutations([

'increase',

'decrease'

])

},

}

</script>

<style scoped>

</style>

7、在主入口文件main.js，import store，再在new的构造器里传入store即可。