1.better-scroll和fast-click插件点击事件冲突：给img图片添加一个class=“needslcik”;

2.左右联动逻辑视频，第五章

3.开发时候，vuex调试工具方法---const debug = process.env.NODE\_ENV !== 'production'

store/index.js

import Vue from 'vue'

import Vuex from 'vuex'

import \* as actions from './actions'

import \* as getters from './getters'

import state from './state'

import mutations from './mutations'

import createLogger from 'vuex/dist/logger'//调试vuex 打印数据

Vue.use(Vuex)

const debug = process.env.NODE\_ENV !== 'production'//调试vuex

export default new Vuex.Store({

actions,

getters,

state,

mutations,

strict: debug,//调试vuex

plugins: debug ? [createLogger()] : []//调试vuex

})

4.图片按10:7的比例缩放样式：

img{

width: 100%

height: 0

padding-top: 70%

}

5.vuex操作步骤：

1，state初始化数据，

2.getter相当于计算属性，是一个函数。外部组件需要用到vuex里面的state值都是通过mapGetter([])获取出来。

3.mutation是一个操作改变state操作，有两个参数：（state,传入参数），一般都是mutations-type管理mutation,组件调用mutation是可以用mapMutation({组件用到函数名：‘mutation-type里面的相应名’}),里面是对象形式的，调用mutations-types里面统一操作状态。mapMutation({setSimner:'SET\_SINGER'})。调用：this.setSimner(传入参数)

4.action:异步或者改变mutation里面的状态（多个或者一个（一个时候，可以直接操作mutation））,mapAction{[]}.

6.axios传递参数是 params格式的，需要改为form-data格式，可以在main.js设置header.

7.uni-app的异步请求发送传递参数格式是form-data格式的。

8.audio音乐播放标签，有音乐加载完成事件@ready=""

9.css引用变量样式，stylus,===>@import '~styles/varibles.styl'格式是：~@/../

git如何建立分支和使用：

1.在仓库上创建分支，然后git pull拉下分支代码；

2.gitcheckout 分支名；

3.就可以在这分支上开发过程，使用git run dev；

4.git status

5.提交分支代码（正常的提交方式：git add . ===>git commit -m'描述'==》git push）

6.合并分支代码（git checkout master==>git merge 分支名===》git push）

其他命令符：

git status=🡺查看状态（如果是on master 🡺证明是在主分支上，假如需要提交代码A到分支B上，则需要查看分支 git branch，然后git checkout 分支名B，这样就可以切换到对应的分支上，然后是正常git add .==> git commint –m ‘’ 🡺git push ==）git chckout master🡺git merger 分支名==》git push 搞掂）

git remote add origin 远程地址

git push -u origin master(添加到远程地址)

第二种方法：===》https://www.cnblogs.com/linjiqin/p/7756164.html

git branches ==>查看分支

如果有有分支没有合并到master中，又创建了心得分支，则使用git merge 上一个没有合并分支的分支名称，即可搞掂。

10.图片等比例缩放写法：

.wrapper{//这种写法是占位宽高的比例，相当于是30vw.(是有具体高度了，而且是等比例的)

overflow: hidden

height: 0

padding-bottom: 30.25%====》是宽/高\*100%

// 或者直接用高30vw

.swiper-img{

width: 100%

}

}

11.APP开发设计稿都是750px，而我们打开项目时候浏览器是375px，二倍图，所以高和实际的情况差2倍。应该乘以2.

12.static是放静态文件，外部可以通过链接打开。（localhost:8080/static/mock/index.json）

这个static文件下可以模拟静态数据，.json格式数据存储，可以通过axios请求。let url="/static/mock/index.json"

http://localhost:8080/static/mock/index.json===>记得不能带#

13.盒子模型：box-sizing:border-box; with:100%; padding:0 0.5rem. 这样设置也不会改变盒子width宽度的大小。

14。v-for遍历数组时候是(item,index) in arr 遍历对象时候：（item,key）in object :key=key key==>是对象的key值

15.this.$refs.dom或者是this.$refs[dom]这两种都是或者元素的写法。

16.better-scroll插件点击A事件，B滚动到对应的位置用scrollToElement（）方法。

17.updated生命周期，是在数据改变渲染完成了，再去调用的更新updated的生命周期。可以在这个生命周期里面执行某个稳定的值得逻辑。

18.offsetTop==>偏移上一个元素的距离

19.节流方法：1.给标注 2.给定时器 ===》去哪儿App-city模块就有用到定时器做节流的优化性能。

20.跨页面（A页面传值到B页面）传值方法：eventBus,vuex，路由传参数

也是可以直接出发mutation里面的方法的，1、 ...mapMutation([A函数名：‘vuex中mutation定义的函数名’]) 2、this.$store.commit('mutation里面定义的函数名'，参数)。

21.mapState(可以是数组['A','B']或者是对象{A(自定义名字):"B（是vuex中state定义的名字）"}),mapGetter（）一样写法。这两个都是在computed计算属性中使用的。 mapAction(),mapMutation()这两个是在motheds中使用

22.vuex中的getters这个方法就像computed这个计算方法一样的。如果逻辑上要计算出新的值，或者要过滤那些没用值，就可以从mapGetter()获取值出来使用，如果只是改变原来的值，即可以直接从state中获取出来使用。

23.keep-alive缓存页面的标签，是跟activated生命周期一起使用的。如果keep-alive缓存的页面，只有执行activated生命周期。

如果在App.vue页面设置了<keep-alive><router-view/></keep-alive>,缓存过的页面，如何根据参数变化，而去改变页面数据渲染呢？首先在：data中定义一个lastData:''.然后在mounted的生命周期中，初始化调用数据接口时候，赋值初始化参数的值给this.lastData=初始化参数的值。最后在activated中判断初始化参数值是否跟lastData的值是否相等，不相等的时候，则重新去调用获取数据的接口并且传递对应的参数值。

# 24.keep-alive===》包裹的router-view缓存过的页面，只能执行activated,deactivated生命周期。

注意监听全局事件 window.addEventListener('scroll', this.handleScroll)，跳转页面时候，需要清除掉。

  activated () {

        window.addEventListener('scroll', this.handleScroll) //绑定一个window scroll事件

    },

             //上面的window的是全局函数,其他页面也可以执行;一直计算数值需要配合生命周期销毁函数 deactivated

    deactivated(){

            window.removeEventListener('scroll', this.handleScroll)

    }

25. .vue页面中每个页面都定义一个name值得作用：1.递归组件写法2.keep-alive中两个属性,exclude和include这两个属性可以配合一起使用。

<template>

  <div>

      <div class="item" v-for='(item,index) of list':key='index'>

      <div class="item-title border-bottom" >

        <span class="item-title-icon"></span>

        {{item.title}}

      </div>

       <div class="item-children" v-if='item.children'>

            <!--  name: 'DetailList', 其实这个名字很大的一个用处

            就是为了我们使用递归组件的时候

            来使用，假设一个组件要用自己的时候，那么我就可以通过自己的名字

            来使用自己，那怎么使用呢？  v-if如果我自身有children，

            我呢就把children当做例子。的一个循环放出来

             组件的自身调用自己组件的自身

            -->

              <detail-list :list='item.children'></detail-list>

       </div>

      </div>

  </div>

</template>

<script>

export default {

  name: 'DetailList',组件名字就是使用了name定义值。

 props:{

   list:Array

 }

}

</script>

第二种写法

 <div id="app">

      <!-- keep的用法是如果我的路由被加载一次就放在内存之中,

      下一次加载从内存取出不需要渲染函数  表示所有路由组件都做了缓存, exclude除了detail不做缓存-->

    <keep-alive exclude='Detail'>//也是name定义值名字

        <router-view />

    </keep-alive>

  <!-- <router-view /> -->

  </div>

</template>

：export default {

    name: "Detail",

    components: {

        DetailBanner,

        DetailHeader,

        DetailList

},

}

26. // import "babel-prolify"  兼容低版本浏览器白屏问题

27.better-scroll插件在手机上不同浏览器滑动时候，会把浏览器拖动出一大截白屏。可以使用修饰符-------- .prevent 阻止默认事件发生。

   @touchstart.prevent="handleTouchStart"

28.部署上线过程：

assetsPublicPath: '/',//表示打包上线项目的目录是放在服务器的根目录。即是把dist文件中的index.htm和static.css内容复制出来放到放服务器的根目录上。

 assetsPublicPath: '/project',//如果后台开发人员是把你的项目部署在服务器上的另一个文件夹中（把dist文件中的index.htm和static.css内容放project的文件夹中），

//则需要在这个目录下放多一层project可以了，即是改为/project，然后再重新打包一次项目,npm run build把dist文件夹名字改为project，最后直接把这个文件夹放到后台服务器上可以。

29.uni-app使用的是rpx单位，设计稿是750px，ps工具测量是多少，页面就写多少。会自动转换为px单位的。例如：设计稿测量到30px,则代码就写30rpx，会自动转换的。移动端字体大小一般都是12px,14px。 浏览器检查元素是12px，则页面代码要写24rpx.(2倍图)

30.样式如果在一行并且超出会换行的，可以使用float:left来实现，设置给子元素浮动时候，父元素要清除浮动。clear:both;

31.移动端设计稿是750px(2倍图)，直接使用rem方法。（重要）

/ 1rem=html font-size=50px 因为是2倍图(750px)，所以设计稿高度是86px,实际上是43px。0.86rem\*50=43px（86/2/50=86/100）巧妙转化

第二种方法是使用插件。

32.vue中@表示src下根目录（是表示src/下的根目录）

 /\* @是表示根目录下即是src目录下 \*/./表示当前目录下

33.字体图标使用：记得修改iconfont.css文件的样式的路径。

@font-face {

  font-family: "iconfont";

  src: url('./iconfont/iconfont.eot?t=1585229860257');

  /\* @是表示根目录下即是src目录下 \*/

  /\* IE9 \*/

  src: url('./iconfont/iconfont.eot?t=1585229860257#iefix') format('embedded-opentype'),

    /\* IE6-IE8 \*/

    url('./iconfont/iconfont.woff?t=1585229860257') format('woff'),

    url('./iconfont/iconfont.ttf?t=1585229860257') format('truetype'),

    /\* chrome, firefox, opera, Safari, Android, iOS 4.2+ \*/

    url('./iconfont/iconfont.svg?t=1585229860257#iconfont') format('svg');

  /\* iOS 4.1- \*/

}

表示在当前文件夹下iconfont的字体。

34.