五、项目

代码缩进：2

命名：

标签中小写和-，标签使用语义标签

css要初始化，css顺序：display=定位-盒子-文本-文字-背景-C3 0.5可写.5

js：变量/函数小驼峰，对象/类大驼峰，组件大驼峰

js代码优化：

\*可维护性：js/css/html分离，行为/结构分离，数据/代码分离

\*可扩展性：模块化开发

\*可调试性：consolo.log aleft try...catch

\*DOM优化：减少DOM操作，减少script标签或给script标签异步属性

defer(按script顺序执行)或async(不按script顺序

前端资源优化：

SEO

\*精灵图：整合了小图标，较少服务器请求次数

\*代码压缩（打包）：将代码丑化，去除空格和换行

\*预加载：所有资源提前加载到本地，提高页面的加载速度img标签和js的Image对象都是预加载

\*懒加载：路由和图片，需要的时候在加载，原理是监听滚动事件预加载是提前加载，懒加载是延迟加载

1、准备工作：

（1）创建项目，运行测试一下，没有问题后将helloworld相关代码删除

（2）src目录结构

assets

img

css

common 本项目公用的js文件

components 公共组件

common 可以在别的项目用的组件

content 只能在本项目用的组件

views 各个页面独立的组件

router

store

network

App.vue

main.js

（3）基本配置：

①拿到.editorconfig文件，配置代码风格

②创建vue.config.js，配置路径别名，当成绝对路径用

module.export = {

configurewebpack:{

resolve:{

extension:[], //配置后倒入路径可不加后缀，默认已配载就hi，可不写。

alias:{

'@':'src', //默认已配置，可不写

'assets':'@/assets',

'components':'@/components',

'common':'@/common',

'network':'@/network',

'views':'@/views'

}

}

}

}

（4）基本css：

①Github下载normalize.css，统一不同浏览器的html标签模样

②创建base.css，@import “assets/css/normalize”; 并初始化css

然后在App.vue中的script，@import “assets/css/base.css”;

（5）

0、设置缩放的最大最小宽度px

1、css中无法引入css

2、路径别名，引入css可以，其他不行

4、滚动

5、变量保存网络请求的数据，在请求到数据前报错

原因是通过props把数据传给子组件时，使用.调用，但是我定义的是空对象，没有可以调用的键，所以报错

解决：props传入时，就只传入一个对象名

有时候，数据是很杂乱的，一个组件需要的数据可能在请求数据的不同对象中，子组件需要配多个props才能得到数据，非常不好

只要记住一个原则，子组件只接收一个对象处理

所以最终的解决方案：

在封装的网络请求js文件中，导出一个类，类的构造器通过传入各个对象来，将多个对象的属性封装成一个单独的类，使用时只需要导入这个类，new传入相应的对象，赋值给一个对象变量obj，在将obj传入子组件处理。

8、商品列表图片高度不同的情况

7、手机和电脑连一个wifi，手机进入电脑ip:端口

17、网络请求最后一页，以及tabcontrol跳转回来是不是要第一页

19、使用better-scroll后里面tabcontrol实现吸顶

思路：获得tabcontrol距离顶部的距离，与滚动的position比较来决定是否吸顶

this.$refs.refname.$el.offsetTop返回refname组件距离顶部的距离，但是如果在网络请求得到图片并显示前执行，得到的是没有图片撑开的高度，所以需要在图片加载完成后在执行，一般tabcontrol前面的组件中，轮播图的营销比较大，其他比较小，所以一般可以只处理轮播图

轮播图的img绑定 @load 并发射自定义事件，home接收比哦并执行 this.变量=this.$refs.refname.$el.offsetTop 即可

然后再有position参数的函数中，与position比较大小并执行一个解决法。

解决方法：

在better-scroll实现固定定位有一些bug，所以不能用，一个思路：

复制tabControl组件标签粘贴到tabnav下面，并设置相对定位和定位等级，用v-show的布尔值控制它的显示隐藏，用顶部距离和position比较来改变这个布尔值

最后，需要解决两个tabControl活跃样式不同步的问题，在yabControl点击时发射事件，home通过参数同时修改两个tabControl的活跃标志

22、为了避免数据没有及时请求过来，而导致没有数据时就展示元素的报错，可以做类似下面的判断

v-if="goodsInfo.image!=null

有时候数据是缺失的，可以判断 v-if="'keyname' in objname"，也有经常判断值是否为空的

所以要注意，看是否要判断对象，键，值是否为空

扩

23、时间戳格式化

为了保证各种设备都能使用且有扩展性，网络请求得到的时间都是时间戳，自己需要将时间戳格式化为需要的时间格式，js没有内置的函数，需要自己写。

最终使用过滤器展示

30、数组.find(箭头函数) 遍历数组，返回的是箭头函数return为true（true后就不执行）的那个元素

34、选中按钮可以自己用div定义，封装成组件，这样有更多的扩展，是否选中的状态一般在被选择的对象中的一个属性，由他来决定选中前后的逻辑

36、vuex的actions的方法，可以return一个promise，在dispatch().then()，以保证actions的操作完成后做一些操作

38、弹窗

自己定义一个Toast组件作为弹窗，能根据需要进行布局，显示的内容和显示隐藏通过外部传入参数，fixed定位到相应位置。

但是组件定义后，使用时要导入，注册，定义变量等，十分麻烦，可以进一步封装成插件的方式封装Toast，就能this.$Toast直接调用

通过组件内定义methods改变data内的变量

39、其他细节

图片懒加载：

在图片需要在屏幕上显示时，才加载图片，优化性能。

npm install vue-lazyload --save

//main.js

impoet VueLavyLoad from 'vue-lazyload'

Vue.use(VueLazyLoad,{

//此对象可写可不写，用来设置加载时显示的图片，加载失败时的图片等

})

img标签的:src换成v-lazyload

px转vw

npm install postcss-px-to-viewport

因为vue项目结构复杂，想找到一个文件很难找，所以组件命名最好在

前面加上该组件所在的视图，如Home，Shopcart等，找起来就好找多了