Vue项目总结：

1. get请求，通过引入vue-resourse 插件，向vue注册$http属性，可以向对应的地址发送请求；注意：我们在引用了插件后，需要调用use方法进行注册，才能在发送请求的时候，调用this.$http的方法；一般来说，在引入了js的库以后，需要进行use，才能将其的属性方法注册到vue对象上面；而css方法只需要将其引入即可；

// 7.0 将vue-resource在vue中绑定，自动在vue对象实例上注入一个$http对象就可以使用ajax方法了  
import vueResource from 'vue-resource';  
Vue.use(vueResource);

1. 关于url地址的存储，将所有数据的api域名地址存储在当前应用的公共配置上面；kits文件夹下面的common.js文件中；

export default{  
 apidomain:'http://www.lovegf.cn:8899' // 所有数据api的域名地址  
}

1. 在每个子组件中使用的时候，直接使用import引用进来；

import common from '../../kits/common.js';

1. 对于主页面的公共样式，我们要进行统一设置，放置在statics文件夹里，用来设置中间的样式，以便引用过来的组件的样式统一，规定好上下的margin值；以及背景色等样式；
2. 定义路由规则，在当前应用的main.js主入口文件中,先引入vue-router插件，并全局注册；然后导入个路由对象所需的的子组件，然后通过路由配置，将刚才定义的路由组件引入；
3. 引入UI插件，本次项目所使用的插件是mint-ui和mui这两个插件；一般来说插件的引用方式，有两种，可以直接将所需文件直接拷贝到本地；然后直接通过import来引用，也可以使用npm的方式install；引入的时候，可以使用路径查找和关键字查找的方式进行引用；
4. 已经下载到本地的文件，路径查找引用：

import 'mint-ui/lib/style.min.css';

1. 通过npm的install方式安装的插件，通过关键字的方式进行引用；

import mintui from 'mint-ui';

1. 定义过滤器：过滤器是需要全局进行注册的，这样在其他的子组件中也可以使用；
2. 在进行时间的格式化的功能时，引入了moment这个时间标准化的插件，引入以后，通过Vue.filter(‘’,function (){})来进行全局注册；

Vue.filter('datefmt',function(input,fmtstring){  
// 使用momentjs这个日期格式化类库实现日的格式化功能  
 return moment(input).format(fmtstring);  
});

1. 实例化vue对象，记着需要引入路由对象，相当于注册的效果；

new Vue({  
 el:'#app',  
 // 使用路由对象实例  
 router:*router1*,  
 // render:function(create){create(App)} //es5的写法  
 render:c=>c(App) // es6的函数写法 =>：goes to  
});

1. 在具体的子组件中，使用了v-bind指令进行绑定；
2. V-bind指令详解：v-bind指令的功能是用于响应的更新html特性；比如href特性，就可以写成 v-bind:href 写成 :href，也就是给这些元素的属性加上一个可以变化的值，让其可以改变；简写形式 ：属性名，但是需要注意的是,像router-link这种路由的指令，它的to属性，后面是一个具体的路径，就不能直接时候用:的简写形式了，需要使用对象的形式来表示；里面的内容形式仍然是常量+变量的形式；

<router-link v-bind=**"{to:'/news/newsinfo/'+item.id}"**></router-link>

1. 将需要路由的对象，统一改写成router-link的指令，将里面的属性改成to属性；
2. 在使用ui插件的时候，如果需要修改它自带的样式，一定要将整个类名取过来，这样才能把权重保持一致，否则无法覆盖该属性；
3. 在当前的子组件中使用自定义的样式的时候，一定要加上scoped；保证只有当前的组件使用整个样式；背后的工作就是给当前的组件的所有元素都加上了data-v-1233这样的唯一的标识；

<style scoped>

</style>

1. 新闻评论的内容采用子组件的形式，在外部定义子组件，然后引入到当前的子组件，再通过exports注册进来，

export default {  
 components:{},

1. 当然，父组件是需要将当前的相关数据传递到子组件，通过props的方法传递；将接受到的参数作为url中的请求参数，累加到url的后面，进行数据的请求；也就相当于是一种新的页面之间传值的方式；

props:['id'],

1. 评论区采用自定义组件的方式进行集成封装，一共要实现，发表评论，获取评论列表，加载更多的几个功能；
2. 首先，像这种提交数据的功能，内容不能为空，为空就提示用户，并且return；
3. Vue中的难点在于组件之间的传值问题，包括子组件和父组件之间，还有不是共享一个vue实例的，例如点击加入购物车的操作，是直接与底部的app.vue进行交互的，这两个组件并不是属于一个实例vue，所以还需要构建中间的桥梁；
4. 要弄清楚这点，首先需要搞清楚整个项目的结构，那像这种情况，就是需要建立一个中间的桥梁，一般采用建立eventBus 全局组件的方式，在这个全新的组件中建立一个新的vue实例，来供两个非父子关系的组件传值；
5. 难点：购物车组件；购物车组件要实现的功能：
6. 1.要从本地存储中获取对应的goodsid和最初的count值；然后传递到购物车的儿子组件中；
7. 在购物车的子组件中通过自定义的click事件操作中间的数值，但是需要注意的是，返回值是一个对象，中间会包含一个属性和goodsid，这个属性用来返回给父组件，传递到父组件以后，需要进行两个操作，一个是更新当前页面的数据，原理是根据返回的数据的属性值，来实现增加和减少的动作；本地存储的值也是采用这种方式进行判断；需要注意的是，特殊值1，当值为1时，就不能继续进行，直接return；
8. 删除当前这一列，需要先分析，当前这一列的数据都由哪些构成，然后依次删除；当前购物车的列表项包含3部分数据，前面的按钮的value值数组，第二部分是dalalist数据，第三部分是本地存储的数据，需要删除后，再进行重新保存；
9. 还需要注意的是，再进行排查删除操作的时候，循环的时候，需要从后面到前面一次进行，这样不会漏掉数据，否则删除不完整，因为数组的索引一直在发生变化；