**Vue中export和export default的区别和用法**

Vue是通过webpack实现的模块化，因此可以使用import来引入模块，例如：

import Vue from 'vue'

import Router from 'vue-router'

import util from '@assets/js/util'

此外，你还可以在 bulid/webpack.base.conf.js 文件中修改相关配置：

    resole:{

        extensions:['.js','.vue','.json'],

        alias:{

            'vue$':'vue/dist/vue.esm.js',

            '@':resolve('src'),

        }

    },

意思是，你的模块可以省略 ".js"，".vue"，“.json” 后缀，weebpack 会在之后自动添加上；可以用 "@" 符号代替 "src" 字符串等。

　　export 用来导出模块，Vue 的单文件组件通常需要导出一个对象，这个对象是 Vue 实例的选项对象，以便于在其它地方可以使用 import 引入。而 new Vue() 相当于一个构造函数，在入口文件 main.js 构造根组件的同时，如果根组件还包含其它子组件，那么 Vue 会通过引入的选项对象构造其对应的 Vue 实例，最终形成一棵组件树。

　　export 和export default 的区别在于：export 可以导出多个命名模块，例如：

*//demo1.js*

    export *const* str = 'hello world'

    export *function* f(*a*){

        return a+1

    }

对应的引入方式：

*//demo2.js*

    import { str, f } from 'demo1'

　export default 只能导出一个默认模块，这个模块可以匿名，例如：

*//demo1.js*

    export default {

        a: 'hello',

        b: 'world'

    }

　对应的引入方式：

*//demo2.js*

    import obj from 'demo1'

**切记，一个js文件中，只能有一个export default；   
但是，一个js文件中，可以有多个export。**

**vue中，new vue({})与export default {} 两者的区别？**

1. new vue({}) 只在入口文件 main.js里使用，而其余组件的里的属性和方法的使用，为什么都要放在export default{}中，而不是每个组件都用new Vue({})来生成呢？
2. 放在export default{}中它是作为一个class被导出的么？
3. 另外两者的写法也不一样。就data来说，给作为new Vue的参数，它是对象。如：

new Vue({

  el: '#app',

  data: {

    message: '<h1>我是vue</h1>'

  }

})

而在 export default中，它就成了方法，这是为什么呢？

*export* default{

     data() {

         return {

             seller1:{}

         };

      },

      components: {'v-header': header}

};

1 & 2： export default 的是一个对象 Object，然后父组件通过 components 属性注册，其实是内部调用了 Vue.extend 方法，把这个 Object 传入，然后得到的也是一个 Vue 的实例。为啥用 Vue.extend 而不是直接new Vue，因为他们要建立父子关系，形成一个 Vue 的组件树。  
3. 组件里的 data 必须是一个方法，因为组件是多个实例，如果 data 是一个同一个 object，那么一个组件的修改会影响另一个，因此它必须返回一个方法。