1. export function getIndex(){} 抛出的本来就是一个函数，为什么后来引入时还要解构: import { getIndex } from "../assets/apis/index.js"？

答: export确实是向外抛出一个函数的意思。但是一个js文件其实也是一个模块对象整体。所以export仅仅是向当前文件模块中抛出一个可被外界引用的成员而已。并不能真正形成多个模块。引入时，也是以.js文件模块整体方式引入，是一个对象中包含很多成员。所以，要想使用.js文件模块中的一个成员，必须用对象解构。

2. 什么时候导出用export default，什么时候导出用export？

答: 导出页面组件或子组件对象，希望作为一个整体引入并使用时，就加default。如果希望自定义的js文件模块中的成员，将来被解构出来，分别单独使用时，就不要加default。而是每个成员前分别加export …

3. 如果一个.js文件中，同时写多个export xxx，但是名称不同。后边的会覆盖前边的吗？

答：不会覆盖！同一个js文件中的多个export xxx，最终会放在一个.js文件模块里一起抛出。每个export其实只是.js文件模块的一个成员而已。

4. 为什么

getIndex()

.then(data=>{

[this.p1,this.p2]=data

})

可以解构成功。但是，用es7:

(async function(){

var data=await getIndex();

[this.p1,this.p2]=data //报错

})()

答: 因为this！function(){ … }里的this已经和外部的Vue组件中的this不同了。所以this.p1就无法访问到data中的p1变量了。所以报错。

解决: 改成箭头函数，保证内外this一致！

(async ()=>{

var data=await getIndex();

[this.p1,this.p2]=data //报错

})()