一、一个 JS 文件就是一个模块

一个js文件，可以理解成一个模块。
这个模块可以被任意其他的模块引入，引入的结果，就是对这个模块进行执行后，所持有的对象。
那么随之而来就有一个问题，文件模块被引入后，所有的东西，都是在自己的作用域中，主动发起引入行为的那个文件，虽然获取到了被引入的对象，但是并不能访问作用域里的东西。
所以提供了export，来决定一个模块对外暴露什么东西。

1、引入的结果，就是对这个模块进行执行后，所持有的对象是什么意思?

1.1） 不导出内容

title.js文件

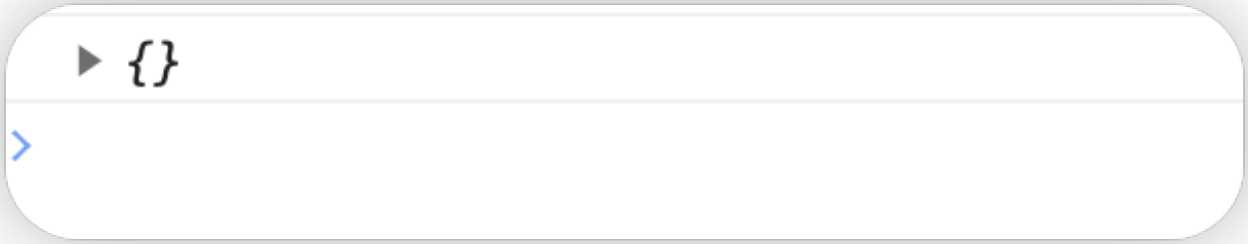
```
let i = 0;
```

index.js文件

```
import * as anything from "./title";

console.log(anything); // ???
```

问题来了：这个anything是什么？？？



没错，我们用import导入后获取到的文件对象是一个空对象，因为title.js中没有导出什么东西

1.2 ）导出内容

title.js文件

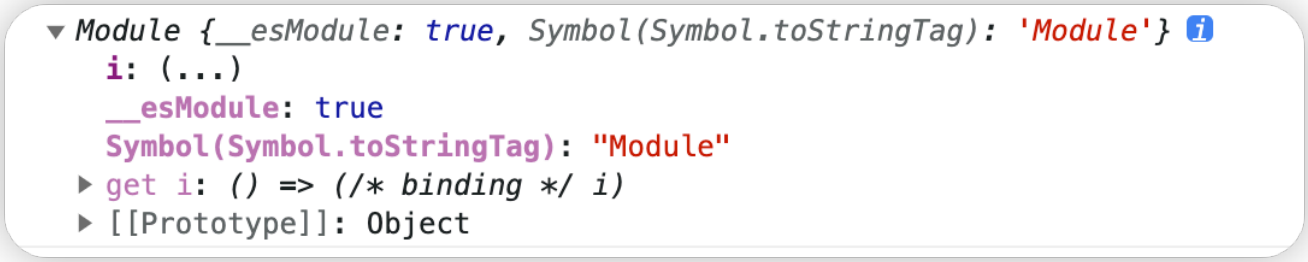
```
export let i = 0;
```

index.js文件

```
import * as anything from "./title";

console.log(anything); // ???
```

问题又来了：这时anything是什么？？？



我们可以看到import导入后获取到的是一个Module实例，实例上的i属性就是title.js导出的值

1.3 ） 默认导出

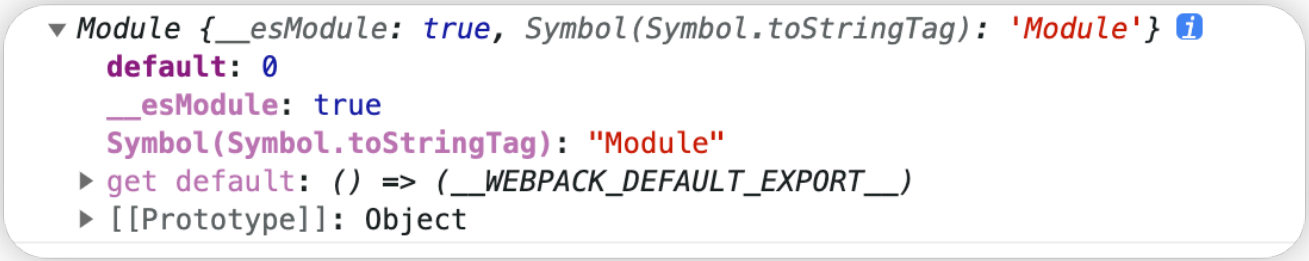
title.js文件

```
let i = 0;
export default i; // 默认导出不能导出一个声明语句
```

index.js文件

```
import * as anything from "./title";

console.log(anything); // ???
```



我们可以看到import导入的export default就是默认在Module给我们加个default属性，把值赋值给default

总结：

```
1. 导入模块会先执行这个模块；
2. 导入的模块可以不导出内容；
3. import导入模块时，声明的变量叫做文件对象（Module的实例）：
   3.1）如果被导入的模块没有导出任何东西，那么import导入的这个对象就是个空对象{};
   3.2）如果被导入的模块使用 export 导出东西，那么import的导入会是Module实例，然后会将export的导出
        放到Module实例上；
   3.3）如果被导入的模块使用 export default导出东西，那么import的导入也是Module实例，不过这时
import
        会在Module实例上增加一个default属性，属性值就是export default的内容；
```

二、 export和export default语法的使用

```
1、export与export default均可用于导出常量、函数、文件、类等；

2、需要特别注意的是，export命令规定的是对外的接口，必须与模块内部的变量建立一一对应关系；

3、在一个文件或模块中，export、import可以有多个，export default仅有一个；

4、通过export方式导出，在导入时要加{ }，export default则不需要，因为它本身只能有一个；

5、 export 可以直接导出或者先定义后导出都可以，export default只能先定义后导出；
```

三、import语法的使用

3.1) 导入默认导出模块

title.js文件

```
let i = 0;
export default i;
```

index.js

```
import i from './title'; // 0
```

3.2) 导入export导出模块

title.js文件

```
let a = 'zf';
let b = '10年';
export {a, b};
```

index.js文件

```
import { a, b } from "./title";

console.log(a, b); // 'zf' '10年'
```

3.3）混合导入导出

title.js文件

```
export const name = '《React进阶实践指南》'
export const author = '我不是外星人'

export default  function say (){
    console.log('hello , world')
}
```

index.js文件

```
import theSay , { name, author as  bookAuthor } from './a.js'
console.log(
    theSay,     // ƒ say() {console.log('hello , world') }
    name,       // "《React进阶实践指南》"
    bookAuthor  // "我不是外星人"
)
```

```
import theSay, * as mes from './a'
console.log(
   theSay, // ƒ say() { console.log('hello , world') }
   mes // { name:'《React进阶实践指南》', author: "我不是外星人", default: ƒ say() {
console.log('hello , world') }}
)
```

```
import * as anything from './title';
console.log(anything);
```

3.4) 重定向导入

```
export * from 'module' // 第一种方式
export { name, author, ..., say } from 'module' // 第二种方式
export { bookName as name, bookAuthor as author, ..., say } from 'module' //第三种方式
```

3.5) 无需导入模块，只运行模块

```
import './module'
```

3.6) 动态导入

```
const promise = import('module');
// import('module')，动态导入返回一个 Promise。为了支持这种方式，需要在 webpack 中做相应的配置处理。
```