

# ES6的模块化



jCodeLife



0.257 2020.09.04 22:52:01 字数 1,013 阅读 28

编辑文章

我们知道很多语言都有属于自己的模块化规则，可以按照特定的规则来定义和使用引用模块。但是在ES6之前，js是不支持模块化的。那怎么办？其实前端也是有自己的社区的，他们也想用得使用模块化，因为确实好。于是社区制定了一套模块加载方案，有CommonJS、AMD和CMD三种规范。其中Commonjs的代表是nodeJs；AMD的代表是requireJs, curlJs；CMD的代表是seaJs。比较主流的是CommonJS和AMD。

其实也别想着那么复杂，它们只不过是按照自己特定规则来定义和引入模块而已，实现都是一样只是规则有所有差异而已。

因为社区制定的模块化规范比较乱，所以ES6出来就统一了这个问题。那我们就来看看ES6中的模块规范的详细规则

总结两个关键字

- `export`
- `import`

`export`用于导出模块，`import`用于导入模块；

当然根据不同的`export`和不同的`import`语句，导入和导出的结果会有所差异，下面一起来看看

我们知道一个模块就是一个独立的文件，该文件内部的所有变量，外部无法获取。如果想从外部能够读取模块内部的某个变量，显然使用`<script src=""></script>`在src中指定要引入的文件的形式会引入整个文件中的所有东西，这样不好。ES6中使用`export`导出命令可以导出特定的某些需要的变量

## 1. 批量导出导入

在需要导出的js文件中，使用 `export {}` 批量导出变量

```
1 let x = 100;
2 let y = [1,2,3,4,5];
3 let str = "asd123"
4 let fun1 = ()=>{
5   console.log("fun1")
6 }
7 export{x,y,str,fun1}
```

对应的使用`import {} from "文件地址"` 批量引入变量

```
1 import {x,y,str,fun1} from '文件目录';
```

## 2. export导出变量时，使用as改变变量的变量名

```
1 //lib.js文件
2 let a = "apple"
3 let b = "banana"
4 export {a as apple, b as banana}
```

### 推荐阅读

如何理解JavaScript中的对象？

阅读 183

vue数据更新了，视图不更新，怎么解决？

阅读 284

从事7年前端开发，有些经验想对转行学习前端的伙伴说说！

阅读 136

前端面试基础题：介绍js的基本数据类型

阅读 69

12个前端开发必备开发的工具

阅读 486

### 3. 可以在变量定义处单独导出一个或者多个变量

```
1 //lib.js文件
2 export let foo = ()=> {
3   console.log("fnFoo");
4   return "foo"
5 },
6 bar = "s,stringBar";
```

对应使用import{} from "文件目录"导入

```
1 import {foo, bar} from "./lib";
2 console.log(foo());
3 console.log(bar);
```

### 4. 当需要单独导出一个数据时，该数据没有变量名，可使用export default导出

```
1 //lib.js文件
2 export default "abc";
```

对应的使用，使用import xxx from "文件目录"来导入。此时的xxx就是给导出数据取得变量名

```
1 import defaultString from "./lib";
2 console.log(defaultString);
```

### 5. 将要导出的变量as为default，在import的时候，该变量的名字就可以自定义了。但要注意每个模块默认接口就一个

```
1 //lib.js
2 let fn = () => "string";
3 export {fn as default};
4
5 //main.js
6 import defaultFn from "./lib";
7 console.log(defaultFn());
```

### 6. 可以使用通配符\*号批量导入所有导出的变量

```
1 import * as circle from './circle';
```

### 7. 使用import '文件路径'的方式，相当于引入文件；可以是相对路径或者绝对路径

```
1 import 'https://code.jquery.com/jquery-3.3.1.js'
```

并且这种形式，无论你引入多少次，只会导入执行一次

### 8. 使用import()函数动态引入

默认import语法不能写到if之类里面，使用import()可以动态引入,类似node里面require(); import()返回值，是个promise对象，可以使用then()

#### 推荐阅读

[如何理解JavaScript中的对象?](#)

阅读 183

[vue数据更新了，视图不更新，怎么解决?](#)

阅读 284

[从事7年前端开发，有些经验想对转行学习前端的伙伴说说!](#)

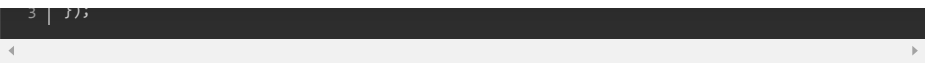
阅读 136

[前端面试基础题：介绍js的基本数据类型](#)

阅读 69

[12个前端开发必备开发的工具](#)

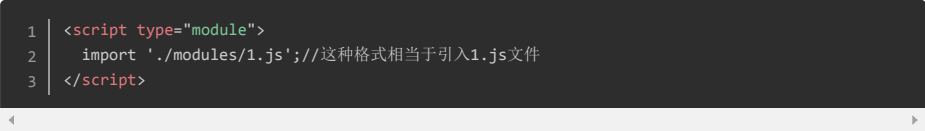
阅读 486



以上差不多就是模块的所有格式！

总的来说，使用模块化还需注意几点：

a. 使用模块需要添加type="module"，告知浏览器这里是跟模块化相关的代码



b. import有提升效果，会自动提升到顶部，首先执行

c. ES6导入的模块都是属于引用，每一个导入的js模块都是活的， 每一次访问该模块的变量或者函数都是最新的。也就是说出去模块内容，如果里面有定时器更改，外面也会改动。这跟社区定义的Commonjs、AMD、CMD有所不同

d. 大括号里面的变量名，必须与被导入模块对外接口的名称相同。跟对象的解构类似

e. 文件路径，.js后缀可以省略



2人点赞 >



笔记 (2)



"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



jCodeLife



书山有路勤为径，学海无涯苦作舟

总资产30 (约2.65元) 共写了13.6W字 获得329个赞 共25个粉丝



写下你的评论...

全部评论 o

只看作者

关闭评论

按时间倒序

按时间正序

被以下专题收入，发现更多相似内容

投稿管理

+ 收入我的专题



前段开发那些事儿

写下你的评论...



评论0



赞2

