

export、export default和import用法详解，这一篇就够了

吃自己家大米 [关注](#)

0.413 2020.06.11 08:50:57 字数 692 阅读 3,182

这三个名词功能的出现是为了实现 js 的模块化的，属于es6语法。

这仨没出现之前，我们给 js 按功能分为不同的文件，在 html 文件页面里引用的时候要使用大量的 script 标签，如果都放在一个文件里呢，又可能出现只使用一两个功能，却要引入相对大的一个文件，还有运行环境不同产生的等等问题。

接着 js 模块化即可在js中按需导入模块功能就出现了。

字面意思就可以看出 **export**、**export default** 负责 **导出**，**import** 则负责 **导入**。

export 在一个 js 文件中可以有多个，**export default** 最多只能有一个。

1、export

方式1：先声明后导出，假设 test1.js

```
1 | const a1 = 'a1' //var 、let
2 | function fn1() {
3 |   console.log('我是fn1');
4 | }
5 | export { a1, fn1 }
```

方式2：导出与声明一起，假设 test2.js

export的函数必须命名

```
1 | export const a2 = 'a2' //var 、let
2 | export function fn2() {
3 |   console.log('我是fn2');
4 | }
```

混用：不限个数当然是可以混用的，假设 test3.js

```
1 | const a3 = 'a3'
2 | export function fn3() {
3 |   console.log('我是fn3');
4 | }
5 | export {a3}
```

不管上面哪种形式都可以用下面两种方式使用。

使用1：用哪个取哪个，还可以命名别名

```
1 | import {fn1} from './test1.js'
2 | import {a2 as ha,fn2} from './test2.js'
3 | import {a3} from './test3.js'
4 | fn1()
5 | console.log(ha)
6 | fn2()
7 | console.log(a3)
```

使用2：全部取出，对象方式使用

```
1 | import * as test1 from './test1.js'
2 | import * as test3 from './test3.js'
3 | test1.fn1()
4 | test3.fn3()
5 | console.log(test3.a3)
```

2、export default

方式1：先声明后导出，假设 test4.js

```
1 | function fn4() {
2 |     console.log('我是fn4');
3 | }
4 | export default fn4
```

方式2：导出与声明一起，假设test5.js

```
1 | export default function fn5() {
2 |     console.log('我是fn5');
3 | }
```

方式2这里 fn5 可以是匿名函数如下，这与 **export** 必须命名函数是一个区别。为什么这里可以是匿名呢？原来 **export default** 导出的函数在 **import** 导出的时候都会重新命名，具体意思看下面的示例。

```
1 | export default function() {
2 |     console.log('我是fn5');
3 | }
```

使用：**export default** 只有一种使用方式

```
1 | import test5 from './test5.js'
2 | test5()
```

这个 test5 就是我们给 export default function fn5() 或者 export default function() 重新定义的名字。

有心的小伙伴可能注意到这里并没有给出 **export default** 导出 `const | let | var` 这种变量的示例，那是因为这种变量是不可以直接像 **export** 一样直接出现在后面的，多值则需要以对象的形式导出，如下 (test6.js)：

```
1 | const a7='a7' // let,var
2 | export default {
3 |     a6:'a6',
4 |     a7:a7,
5 |     fn6(){
6 |         console.log('我是fn6');
7 |     }
8 | }
```

使用还是跟使用 test5.js 一样的

```
1 | import moreEle from './test6.js'
2 | console.log(moreEle.a6)
3 | moreEle.fn6()
```

3、export 和 export default 混用，test7.js

```
1 | const a1 = 'a1' //var , let
2 | function fn1() {
3 |     console.log('我是fn1');
4 | }
```

```
5 | export { a1, fn1 }
6 |
7 | export const a2 = 'a2' //var 、let
8 | export function fn2() {
9 |     console.log('我是fn2');
10 | }
11 |
12 | const a3 = 'a3'
13 | export function fn3() {
14 |     console.log('我是fn3');
15 | }
16 | export { a3 }
17 |
18 | export default function fn5() {
19 |     console.log('我是fn5');
20 | }
```

以上 test7.js 使用的时候，要使用 **export default** 定义的就只能是 **import [name] from './test7.js'** 形式，**export** 也是用自己的那两种方式 **import {[name],[name] as [alias]} from './test7.js'**(即必须有花括号) 或者 **import * as [alias] from './test7.js'**，注意后一种 * 尽管是导出全部的意思，但这并不包括 **export default** 导出的。

感谢阅读，喜欢的话点个赞吧：)

更多内容请关注后续文章。。。

禁止转载，如需转载请通过简信或评论联系作者。



7人点赞 >

js

...



吃自己家大米 前端小马甲，努力码ing

总资产7 共写了2.1W字 获得137个赞 共70个粉丝

关注