

html中引入模块要加上`type="module"`

```
<script type="module" src="/static/moudleIn.js">
```

```
1、 export const name = 'hello';  
export const addr = 'chunhua';  
export const age = '22';
```

```
import {name,age,addr} from "/static/moudleOut.js"  
  console.log(name,age,addr);
```

```
2、 const name = 'hello';  
    const addr = 'chunhua';  
    const age = '22';  
    export {  
        name,addr,age  
    }
```

```
import {name,age,addr} from "/static/moudleOut.js"  
  console.log(name,age,addr);
```

```
3、 const name = 'hello';  
    const addr = 'chunhua';  
    const age = '22';
```

`export default name` // 导出默认值 很重要 引入时就不用加{}了，允许自定义名字，不用知道原文件中的名字，`export`后面跟花括号，`export default`后面直接跟名称

```
export {  
    addr,age  
}
```

```
import name,{age,addr} from "/static/moudleOut.js"
```

4、 `import name,{age as age1,addr}` from "/static/moudleOut.js" // 花括号中呢更改名字，花括号中导出必须知道源文件中变量名

5、引入方法 同 引入变量规则

```
export function say(content){  
    console.log(content);  
}
```

```
import {say} from '/static/moudleOut.js'  
say('引入方法')
```

6、导出对象 注意歧义问题

7、导出类 同上

8、引入多个

```
import * as Mod from './lesson2-14-mod'  
  
let test = new Mod.Test()  
console.log(test.id)  
let animal = new Mod.Animal()  
console.log(animal.name)
```

在*引入中，mod的默认导出的必须通过下面的方式去访问，不能通过名字访问

```
let people = new Mod.default()
```