

## 25.Module 模块化

学习要点：

1. 浏览器加载
2. Node 加载

本节课我们来开始学习 ES6 新增的模块化导入导出的方法。

### 一. ES6 模块化

1. ES6 支持模块化设计，也能其它后端语言一样使用导入导出的功能；
2. 我们首先，创建一个要被导入的模块 `module.js`，具体如下：

```
export let name = 'Mr.Lee';           //导出这个变量
```

3. 再创建一个普通的 `.js` 文件，比如 `25.js`，然后导入 `module.js`；

```
import {name} from './module.js';  
console.log(name);
```

4. 最后一步，在 `.html` 文件要加载 `25.js` 文件，才有小，注意 `type` 格式；

```
<script type="module" src="js/25.js"></script>
```

PS：注意，这种导入导出的方式属于 ES6 模块，仅支持浏览器模式；

5. 除了导出变量、常量之外，还可以导出函数、类等功能；

```
export let name = 'Mr.Lee';
```

```
export function sum(x, y) {  
    return x + y;  
}
```

```
export class Person {  
    constructor(name) {  
        this.name = name;  
    }  
  
    run() {  
        return 'name : ' + this.name;  
    }  
}
```

```
import {name, sum, Person} from './module.js';
```

```
console.log(name);
console.log(sum(10, 20));
console.log((new Person('Mr.Lee')).run());
```

6. 也支持使用\*号，将所有导出的内容全部加载进来；

```
import * as module from './module.js';
```

```
console.log(module.name);
console.log(module.sum(10, 20));
console.log((new module.Person('Mr.Lee')).run());
```

7. 支持别名设定，设定别名后，源名即失效了；

```
import {name as user} from './module.js';
console.log(user);    //name 无效了
```

8. 统一导出方案，不需要再每一个导出的内容设置 export；

```
export {
  name,
  sum,
  Person
}
```

9. 可以给导出设置一个默认值，导入部分就可以不用花括号了；

```
export default name;
import name from './module.js';
import name, {sum, Person} from './module.js';
```

## 二. Node 加载

1. Node 下有自己的导出和导入的加载模式：CommonJS 规范；

```
let name = 'Mr.Lee';
```

```
module.exports = {
  name : name,
};
```

```
const name = require('./common.js');
console.log(name);
```