**JavaScript模块化**

# 第1章：模块化入门

## 理解什么是模块

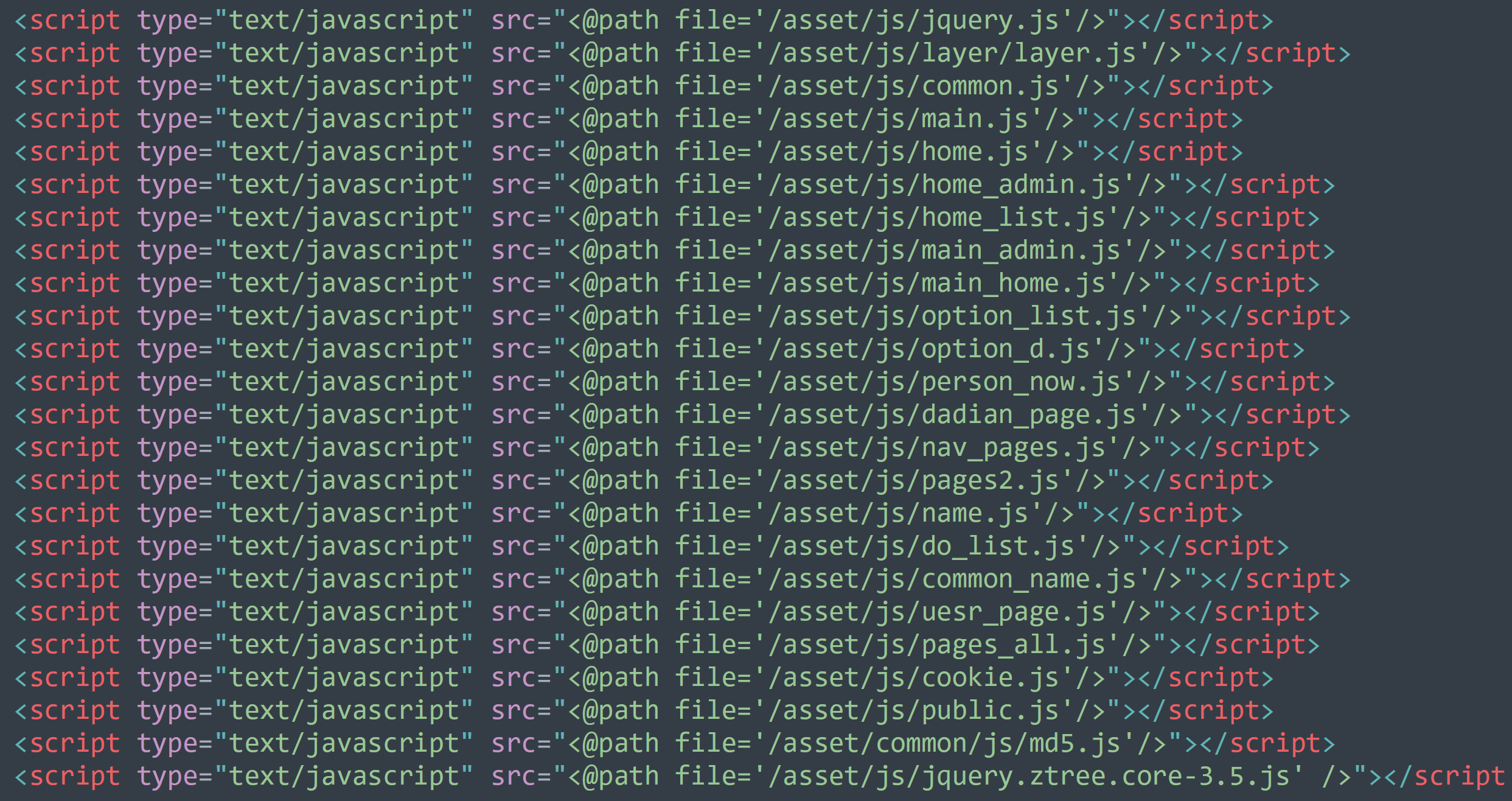
1. 将一个复杂的程序依据一定的规则**拆**分成单个文件, 并最终组**合**在一起
2. 这些拆分的文件就是模块，模块内部数据是私有的, 只是向外部暴露一些方法与外部其它模块通信

## 为什么要模块化？

1. 降低复杂度，提高解耦性
2. 避免命名冲突
3. 更好的分离, 按需加载
4. 更高复用性，高可维护性

## 1.3 模块化概念带来的问题

1. 请求过多
2. 依赖模糊
3. 难以维护



# 第2章：模块化规范

前言：一个大的项目必定会使用模块化技术，使用模块化就会使用相应的模块化规范，现在比较流行的模块化规范有以下2种：CommonJS、ES6

## 2.1. CommonJS

### 2.1.1 规范

1. 官网：http://wiki.commonjs.org/wiki/Modules
2. 每个文件都是一个模块。
3. CommonJS模块化的代码既可在服务端运行，也可在浏览器端运行
4. 服务器端: 模块化的代码可直接运行。
5. 浏览器端: 模块化的代码要经过Browserify（http://browserify.org）编译。

### 2.1.2 基本语法

1. 暴露模块：

第一种方式：module.exports = value

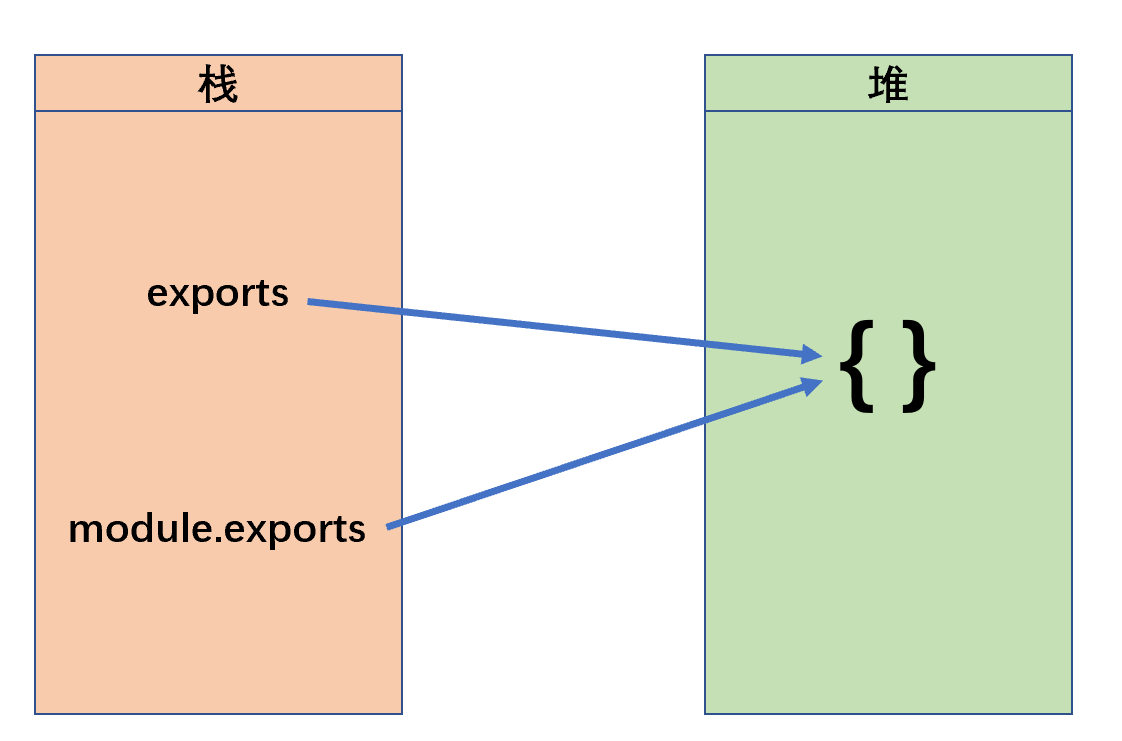
第二种方式：exports.xxx = value

1. 引入模块：

引入第三方模块：require(xxx)，xxx为模块名

引入自定义模块：require(xxx)，xxx为模块文件路径

1. 备注：内置关系



## 2.2 ES6模块化规范

### 2.2.1 规范

1. 每个文件都是一个模块。
2. 要借助Babel和Browserify依次编译代码，才能在浏览器端运行。
3. Babel中文网：https://www.babeljs.cn/

### 2.2.2 基本语法

1. 暴露模块：

1. **分别暴露：**export 暴露内容

2. **统一暴露：**export { 暴露内容1, 暴露内容2 }

3. **默认暴露：**export defalut 暴露内容：

1. 引入模块：

1. 方法1**：**import {xxx,yyy} from './module1'

2. 方法2：import module3 from './module3'

1. 使用规则：

若使用**分别暴露**、**统一暴露**的方式暴露内容，那么就要用方法1引入。

若使用**默认暴露**的方式暴露内容，那么就要用方法2引入。