

简介：将软件产品看作为一系列功能模块的组合，通过特定的方式实现软件所需模块的划分、管理、加载；

适用于业务复杂、重用逻辑非常多、扩展性要求较高的项目

非模块化开发短处：

- 命名冲突
- 文件依赖

意义：

- 协同
- 代码复用
- 文件
 - + 大量的文件引入
 - + 命名冲突
 - + 文件依赖的顺序

特点：

- 高内聚
- 低耦合

演变过程：

1. 函数-原始写法
2. 对象封装
3. 私有共有成员分离
4. 模块的维护扩展
5. 模块的第三方依赖

模块化规范：

- + 服务器端规范
 - CommonJS---node.js
- + 浏览器端规范
 - AMD---RequireJS 国外相对流行
 - CMD---SeaJS 国内相对流行

模块分类：

- 文件模块
- 核心模块
- 第三方模块

SeaJS和RequireJS对比

- 对于依赖的模块，AMD是提前执行，CMD是延后执行
- CMD推崇依赖就近，AMD推崇依赖前置
- AMD的API默认是一个当多个用，CMD的API严格区分，推崇职责单一

AMD是“Asynchronous Module Definition”的缩写，意思就是“异步模块定义”。它采用异步方式加载模块，模块的加载不影响它后面语句的运行。所有依赖这个模块的语句，都定义在一个回调函数中，等到加载完成之后，这个回调函数才会运行。

<https://www.cnblogs.com/chenguangliang/p/5856701.html>