

# 前端组件模块化思考与实践

——requirejs VS seajs

# 思考

- ◆ 为什么需要模块化
- ◆ 模块化的优缺点

# 实践

- ◆ 模块定义规范
- ◆ AMD VS CMD
- ◆ requirejs VS seajs

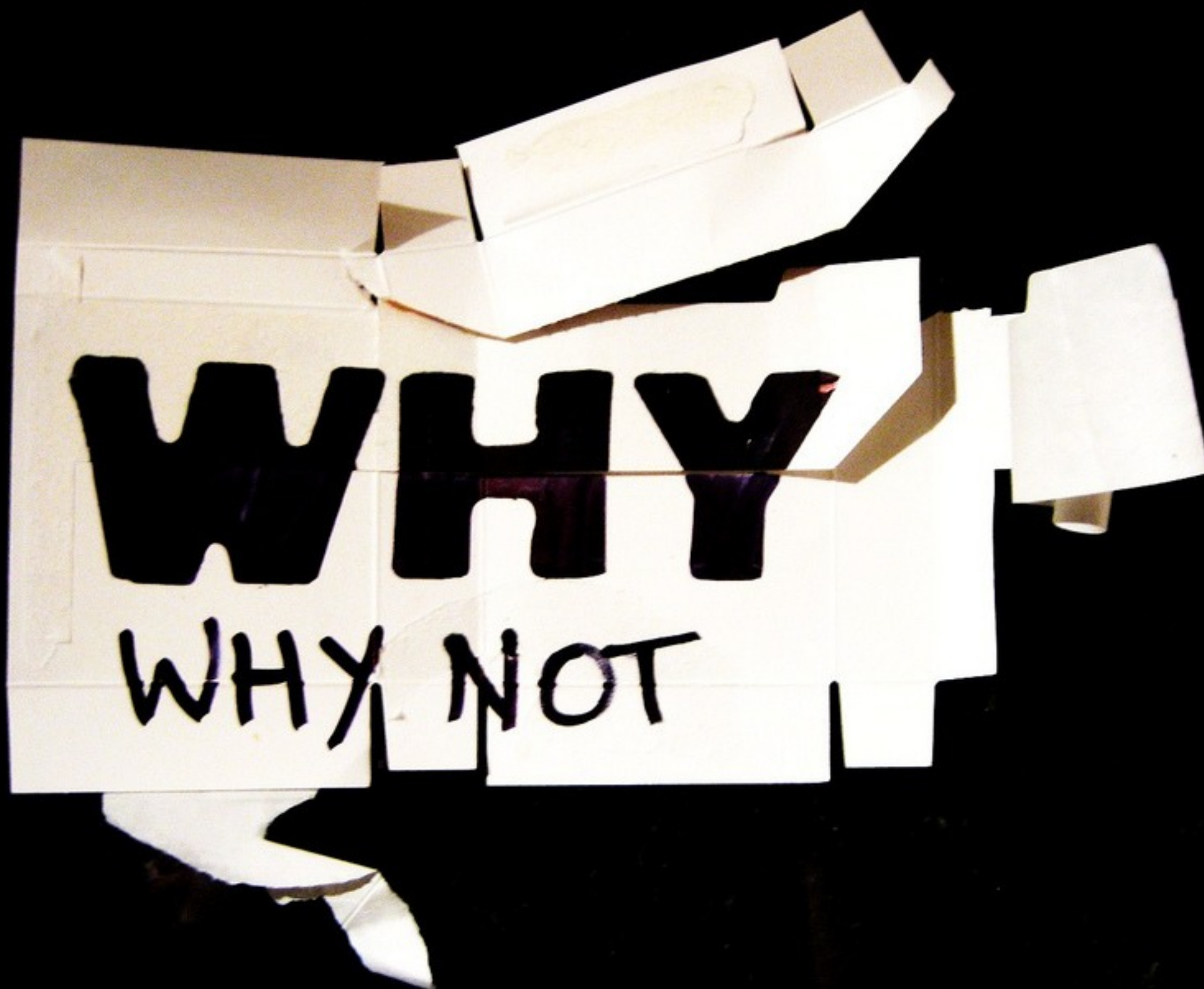
# 思考

- ◆ 为什么需要模块化
- ◆ 模块化的优缺点

# 实践

- ◆ 模块定义规范
- ◆ AMD VS CMD
- ◆ requirejs VS seajs





# 原因 「为什么需要模块化」

# 原因 「为什么需要模块化」

## ◆前端开发流程规范化



# 原因 「为什么需要模块化」

◆前端开发流程规范化

◆业务与需求更复杂

# 原因 「为什么需要模块化」

- ◆前端开发流程规范化
- ◆业务与需求更复杂
- ◆代码复用、架构升级



# 思考

- ◆ 为什么需要模块化
- ◆ 模块化的优缺点

# 实践

- ◆ 模块定义规范
- ◆ AMD VS CMD
- ◆ requirejs VS seajs

# 优缺点 「模块化的优缺点」

# 优缺点 「模块化的优缺点」

- ✓ 代码复用强、提高开发效率
- ✓ 有利于开发规范与文档系统
- ✓ 模块依赖关系系统解决使开发更专注



# 优缺点 「模块化的优缺点」

- ✓ 代码复用强、提高开发效率
- ✓ 有利于开发规范与文档系统
- ✓ 模块依赖关系系统解决使开发更专注
- ◆ 模块粒度过高不利于管理
- ◆ 额外的学习成本



# 思考

- ◆ 为什么需要模块化
- ◆ 模块化的优缺点

# 实践

- ◆ 模块定义规范
- ◆ AMD VS CMD
- ◆ requirejs VS seajs



# CommonJS模块规范



# CommonJS模块规范



CommonJS 是一个有志于构建 JavaScript 生态圈的组织。它有一个 邮件列表，很多开发者参与其中。整个社区致力于提高 JavaScript 程序的可移植性和可交换性，无论是在服务端还是浏览器端。——lifesinger

# CommonJS的两个子规范

CommonJS

```
graph TD; A[CommonJS] --> B[Module/AMD]; A --> C[Module/Wrappings];
```

Module/AMD

Asynchronous Module Definition

Module/Wrappings

Common Module Definition



# CommonJS的两个子规范

CommonJS



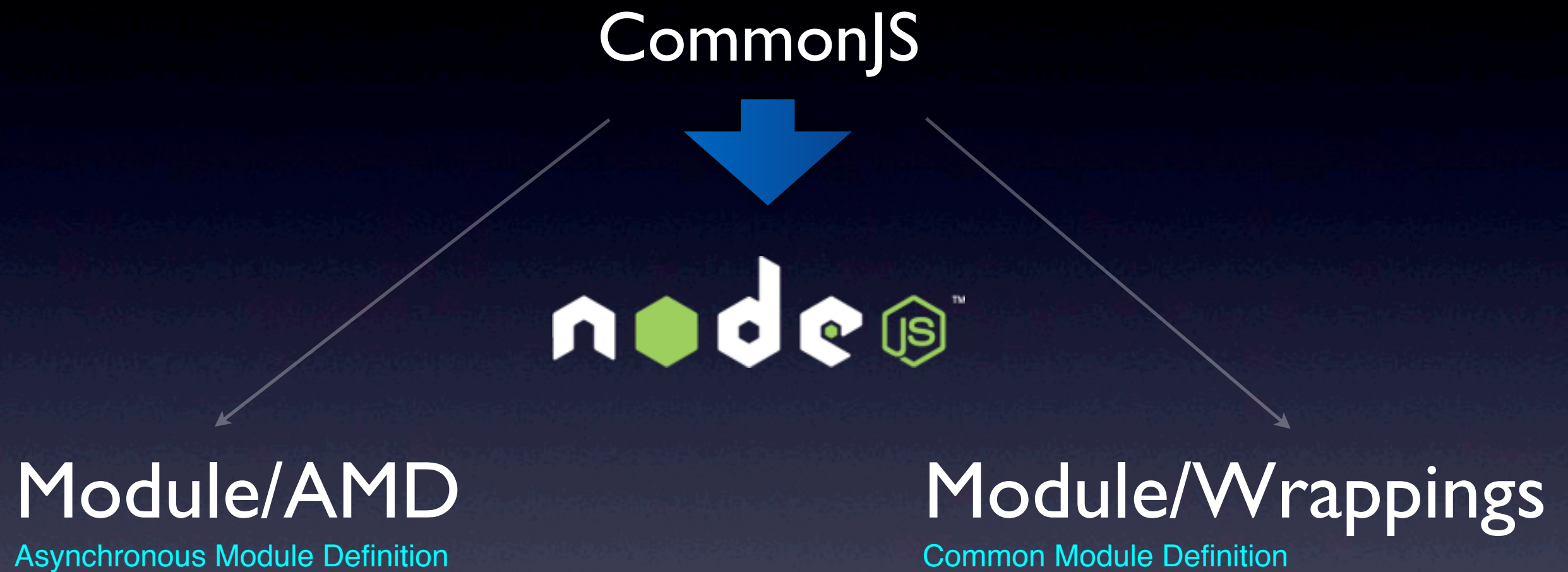
Module/AMD

Asynchronous Module Definition

Module/Wrappings

Common Module Definition

# CommonJS的两个字规范



# CommonJS的两个子规范

CommonJS



Module/AMD

Asynchronous Module Definition

Module/WRappings

Common Module Definition





# CommonJS的两个子规范

CommonJS



Module/AMD

Asynchronous Module Definition

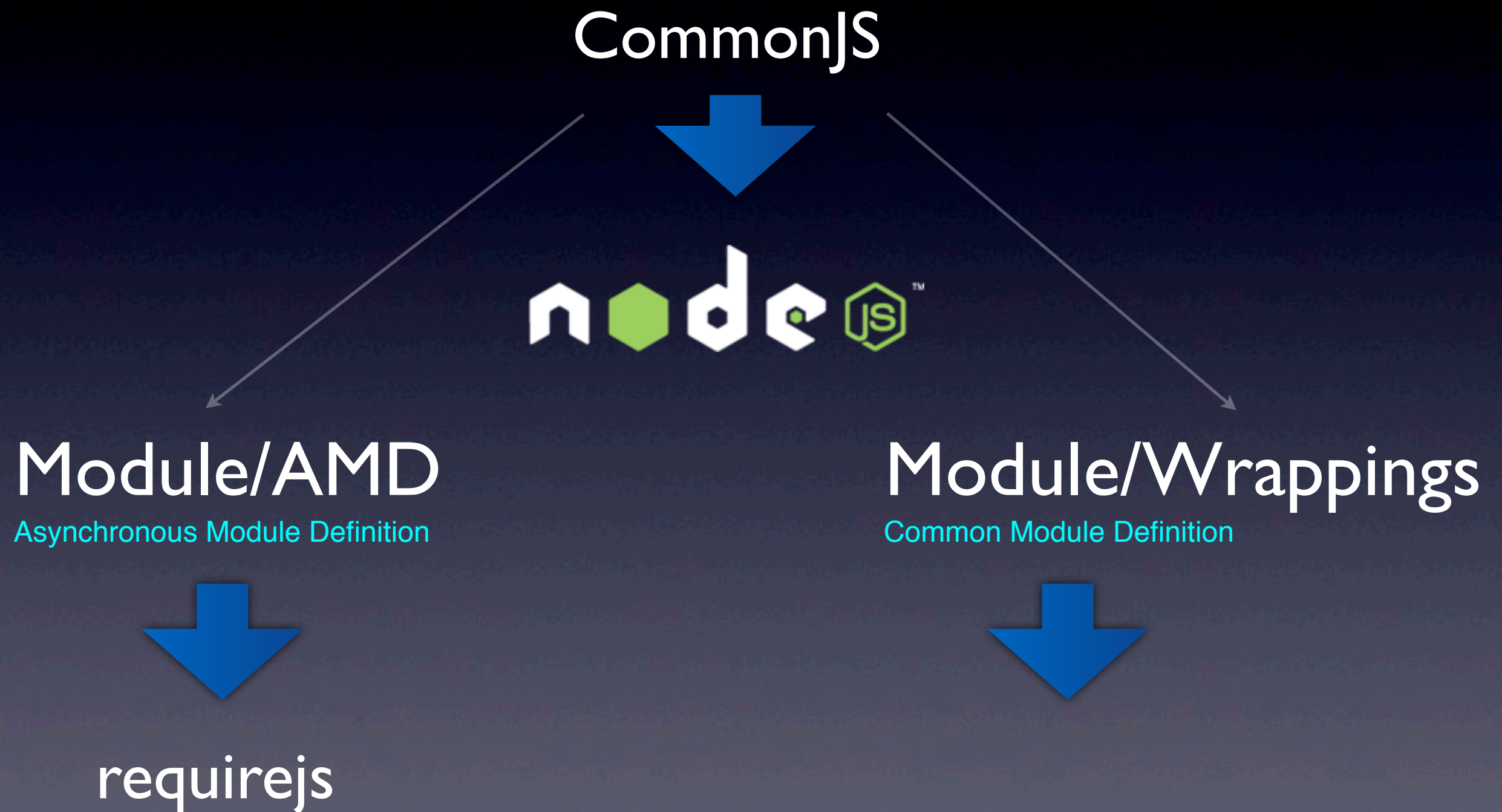
Module/Wrappings

Common Module Definition

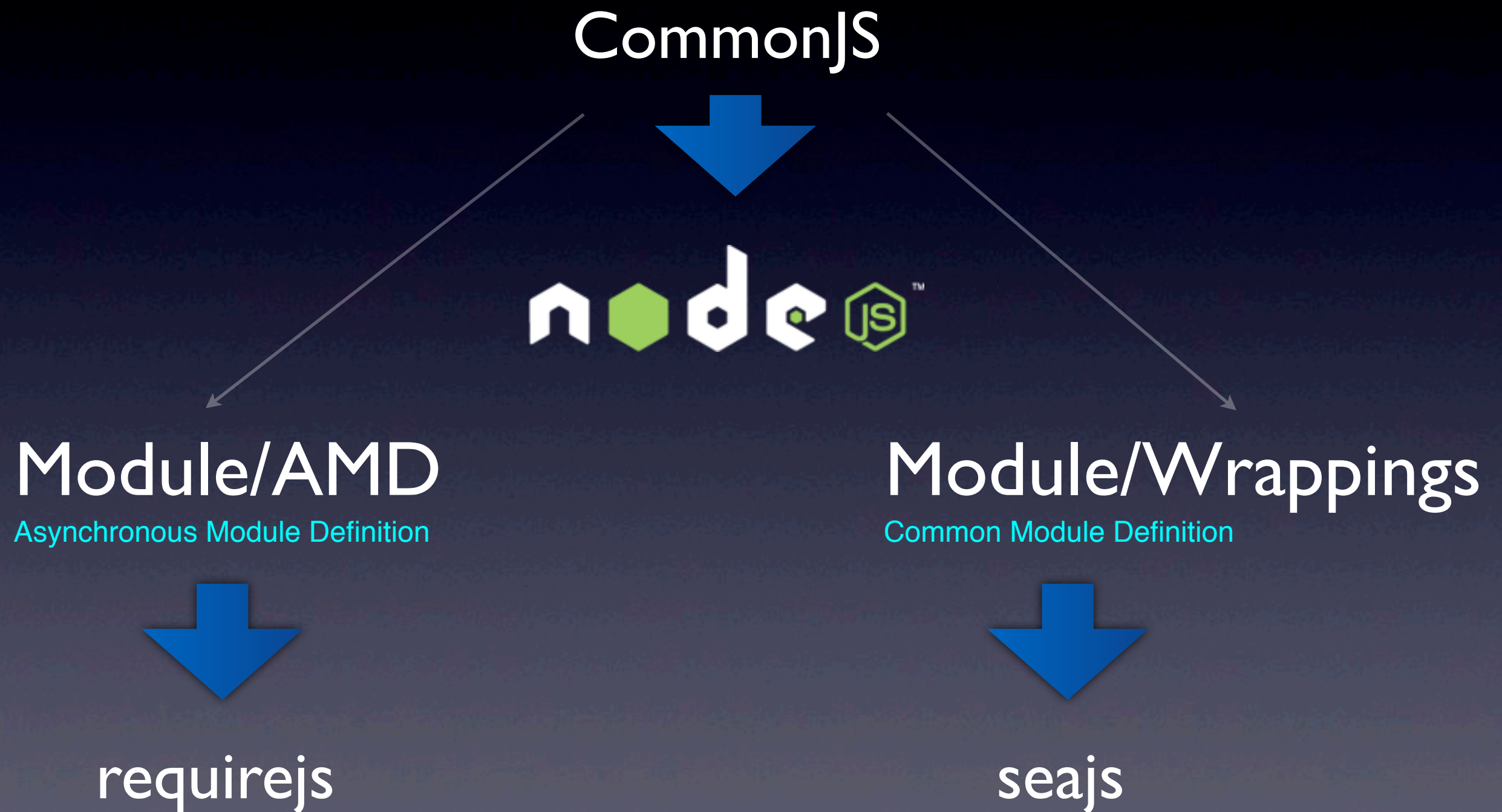


requirejs

# CommonJS的两个字规范



# CommonJS的两个子规范





# AMD模块规范



define

require

# 定义一个模块

```
define(id?, deps?, factory)
```

# 定义一个模块

```
define(id?, deps?, factory)
```

string  
模块标识



# 定义一个模块

```
define(id?, deps?, factory)
```

string	string/array
模块标识	依赖数组

# 定义一个模块

```
define(id?, deps?, factory)
```

string	string/array	function
模块标识	依赖数组	工厂函数

# 加载一个模块

```
require(['modules'], callback)
```



# 加载一个模块

```
require(['modules'], callback)
```

string/array  
加载数组

# 加载一个模块

```
require(['modules'], callback)
```

string/array  
加载数组

function  
加载成功回调函数

# define用例

```
define('fp', ['jquery', 'trimpath'], function() {  
    var footPrint = function() {  
        // code here  
    }  
  
    // 模块对外开放的变量、方法、对象  
    return footPrint;  
});
```



# require用例

```
require(['jquery', 'fp'], function(jquery, footprint) {  
    // dom加载完成后初始化脚印模块  
  
    $(function() {  
        var fp = new footPrint();  
    });  
});
```

# SeaJS ?

# SeaJS ?

coming soon...



# RequireJS 对比 SeaJS

requirejs(15kb)	seajs (7kb)
AMD应用广泛，对AMD支持良好	另辟蹊径选择 Wrappings 自立门户，语法优雅、更适合浏览器端
脚本加载器+模块加载器	纯粹的模块加载器
AMD天然支持JSONP模块	模块内部按需异步加载

# 引用:

<https://github.com/amdjs/amdjs-api/wiki/AMD>

<https://github.com/seajs/seajs>

<https://github.com/jrburke/requirejs>

END