

写js经常遇到的问题

- 命名冲突
- 代码之间的依赖关系

CMD (Common Module Definition) 通用模块

定义，分为：

- require()用来引入外部模块
- exports对象用于导出当前模块的方法或变量，唯一的导出口
- module对象就代表模块本身

“

- CMD 推崇 依赖就近 ， 延迟执行

为什么选择seajs

- 为了 解决冲突
- 解决代码 依赖关系
- seajs提升代码可维护性(维护)

了解模块的概念

- 在 Sea.js 里，**一切皆是模块**，所有模块协同构建成 **模块系统**
- 一个模块，可以是JS 模块，也可以是 CSS 模块，或是 Template 等模块。在 Sea.js 里，我们专注于 **JS 模块**

规则

- 模块是一段 JavaScript 代码，具有 **统一的基本书写格式**。
- 模块之间通过基本 **交互规则**，能彼此引用，协同工作。

seajs的使用

下载seajs

```
npm install seajs
```

简单的seajs

- 1.引入seajs
- 2.变成模块化 `define`
- 3.调用模块 `seajs.use`
- 4.导出模块 `exports`
- 5.如何依赖 `require`

定义模块

```
define(function(require, exports, module){  
  
    //require exports module参数不可更改  
    //模块代码  
  
});
```


调用模块

加载一个模块，在加载完成时，执行回调

```
seajs.use('./a', function(a) {  
    a.doSomething();  
});
```

加载多个模块，在加载完成时，执行回调

```
seajs.use(['./a', './b'], function(a, b) {  
  
    //当然seajs 也可以通过seajs.require()方法来动态获取参数  
    a.doSomething();  
    b.doSomething();  
});
```

导出模块

```
define(function(require, exports) {

    // 对外提供 foo 属性
    exports.foo = 'bar';

    // 对外提供 doSomething 方法
    exports.doSomething = function() {};

});
//当然可以用return/modules.exports导出方法
```

如何依赖

```
define(function(require) {  
  
    // 获取模块 a 的接口  
    var a = require('./a');  
    // 调用模块 a 的方法  
    a.doSomething();  
    //异步依赖的方法  
    require.async()  
});
```

seajs小例子

- 一个可拖动的蓝色div盒子
- 当鼠标按下的时候盒子变黄，离开时变回蓝色
- 限制盒子拖动的范围

构建工具GRUNT

安装grunt

```
npm install -g grunt-cli
```

```
npm install grunt
```

```
npm -version
```

使用GRUNT

grunt插件

```
npm install grunt-contrib-concat --save-dev
```

```
npm install grunt-contrib-uglify --save-dev
```

解决seajs压缩和并问题

提取 `id` 和提取 `依赖`

```
npm install grunt-cmd-transport@0.3.0 --save-dev
```

简单了解下spm压缩代码

- 安装

```
$ npm install spm@2.x -g
```

```
$ npm install spm-build -g
```

- 初始化package.json

```
{  
  "family": "jwseajs",  
  "name": "hello",  
  "version": "1.0.0",  
  "spm": {  
    "output": ["main.js", "css/index.css", "css/style.css"]  
  }  
}
```


seajs的配置

```
seajs.config({
  base:'', //配置根目录
  path:'', //配置路径
  vars:'', //声明变量
  map:'', //设置映射关系
  preload:'', //预先加载
  debug:'', //调试
  charset:'' //编码格式
})
```

配置详细介绍

seajs一些常用的方法

- `require.resolve`
- `module.id`
- `module.uri`
- `module.dependencies`
- `seajs.cache`

SEAJIS还为我们提供了一些插件

[seajs模板](#) [seajs导入样式表](#)

[更多插件](#)