js不足

- js没有 模块 系统，不支持 封闭作用域 或 依赖管理
- 没有 标准库 ，没有文件系统API
- 没有 包 管理系统，不能自动加载和安装依赖

什么是CommonJS?

CommonJS 是javascript模块化编程的一种规范，主要是在服务器端模块化的规范，一个单独的文件就是一个模块。每一个模块都是一个单独的作用域，也就是说，在该模块内部定义的变量，无法被其他模块读取，除非定义为global对象的属性。

1.模块的使用

1.1 模块

- 定义模块 一个单独的文件就是一个模块。每一个模块都是一个单独的作用域
- 导出模块 使用module.exports或者exports

- 使用模块 通过require使用模块

1.2 文件的作用域

每个模块都是一个单独的作用域,通过闭包的形式产生独立作用域

```
(function(exports,require,module,__filename,__dirname){
return module.exports;
});
```

1.3 文件加载缓存

- 加载模块后会缓存，多次加载后得到同一对象
  ```
  require('./test.js');
  ```
- 查看模块缓存
  ```
  console.log(require.cache);
  ```
- 查询模块绝对路径
  ```
  require.resolve('./test.js');
  ```
- 查看单个的模块缓存
  ```
  require.cache[require.resolve('./test.js')]
  ```
- 删除模块缓存
  ```
  require.cache[require.resolve('./test.js')];
  ```

1.4 导出类和对象

module.exports和exports的区别

```
(function(exports,require,module,__filename,__dirname){
    exports = module.exports = {};
    return module.exports;
});
```

2.包和npm

- 多个模块(多个文件)可以封装成一个包
- npm是node.js默认的模块管理器,用来安装和管理node模块 网址为 http://npmjs.org
- 可以用包的方式通过npm安装、卸载、发布包

3.初始化项目

```
$ npm init

{
"name":"包的名称",
"description": "包的简要说明。",
"version":"版本号",
"keywords": "关键字",
"licenses":"许可证",
```

```
"repositories": "仓库地址 ",
"dependencies":"包的依赖，一个关联数组，由包名称和版本组成。"
}
```

4.安装第三放包

4.1 全局安装

直接下载到Node的安装目录中，各个项目都可以调用,适合工具模块，比如webpack

```
$ npm install -global webpack
```

> 全局安装可以在命令行下直接使用

4.2 本地安装

将一个模块下载到当前目录的node_modules子目录，然后只有在当前目录和它的子目录之中，才能调用这个模块

```
$ npm install  webpack --save-dev/--save
```

> 后期我们会在package.json里使用本地模块，我们可以用require使用本地模块

4.3 查看全局路径

```
$ npm root -g
```

5.卸载第三方包

- 卸载全局

  ```
  npm uninstall -g [package name]
  ```

- 卸载本地

  ```
  npm uninstall [package name] --save-dev
  ```

6.发布项目

6.1 发布步骤

- 创建并进入目录

  ```
  $ mkdir jiang && cd jiang
  ```

- 初始化项目

  ```
  $ npm init
  ```

- 注册用户

  ```
  $ npm adduser
  ```

- 发布项目

```
$ npm publish
```

6.2 管理源

nrm NPM registry 管理工具

```
$ npm install -g nrm 安装此工具
$ nrm ls 显示所有的源
$ nrm use cnpm 切换到中国源
$ nrm help 显示帮助
```

7.模块的分类

7.1 内置模块

这里我们使用util(内置模块)举例说明:

- inherits
  默认为原型链继承(不继承私有属性)

```
ctor.prototype.__proto__ = superCtor.prototype;
ctor.prototype = Object.create(superCtor.prototype);
```

- inspect 解析对象

```
var obj  = {name:'zfpx',age:7};
util.inspect(obj,{showHidden:true,depth:1,colors:true});
```

  给对象定义属性

```
Object.defineProperty(obj,'age',{
        value:100,
        enumerable:true,
        writable:true,
        configurable:true
});
```

- is方法 isArray isRegExp isDate isError...

8.模块查找规则

当没有以'/'或者'./'来指向一个文件时，这个模块要么是核心模块,要么就是从node_modules文件夹加载的

8.1 查找

- 内置模块
- 第三方模块
  i. 从module.paths取出第一个目录开始。
  ii. 直接从目录中查找，存在结束，不存在下一条。
  iii. 尝试添加.js、.json查找
  iv. 尝试将require的参数作为一个包来查找，先查找index.js,读取package.json，取得main配置项指定的文件查找，不存在进行3
  v. 继续失败查看下一个目录