03月05日 日报

1. 已完成的内容

//获取当前文件的位置

console.log(\_\_dirname);

//查看当前文件的位置

console.log(\_\_filename);

//同步事件和异步事件在一起的时候，优先执行的是同步事件。

//node提供的方法，查看路径

//\_\_dirname 文件夹路径

//\_\_filename 文件路径

//path: 在cmd里面输入path查看环境变量

//配置：一般都是找到软件的bin目录的路径，粘贴到 我的电脑--右键属性--高级电脑配置--环境变量--path栏

//模块化的由来：

1.function name(a，b，c) {//设计者

逻辑

}

2.团队过大--设计者过多

A:添加商品 add（）a.js

B：注册人员 add（）b.js

C:添加关注 add（）c.js

调用的时候 -- 命名冲突

3.java 命名空间

var org = {};

org.collection = {};

org.collection.utils = {};

org.collection.utils.add = function(){} 设计者都不一定记得住

文件依赖：

玉伯 sea.js 模块加载器 -- 避免命名冲突 明确文件依赖

nodejs 自己的模块化 -- 一个js文件（可以被任何人拿来使用里面的函数，没有命名冲突，明确文件依赖）

没有命名冲突：

1.暴露方式

module.exports = {} 使用

2.暴露方式

exports. 属性的名字 = 暴露的内容

例：exports.qq=add -- add=function(){}

var xxx = require('') 引入

//nodejs模块的使用（单线程）：

//fs模块：文件系统模块

//\*适应使用回调函数

//fs模块（内置模块）

//npm的使用（相对应的工具 bower cnpm 。。。）

//npm init 初始化 package.json -- vue

//npm install 名字 下载操作（当前打开cmd的文件夹里面）

//npm install -g 名字 下载操作（下载到npm全局路径下面）

//npm uninstall 名字 卸载文件

//npm install -g cnpm --registry=http://registry.npm.taobao.org

//express-generator:cnpm install express-generator -g

//脚手架 -- node封装的内库express的工具，搭建服务器

//异步事件（多核的使用）

//同步事件 异步末尾＋Sync，没有回调函数 例：unlinkSync()

var fs = require('fs')//如果是自己写入的模块 require（‘./fs’）注意路径

//删除文件

//fs.unlink('目标文件',function(err){执行了删除指令之后要做的事情})

fs.unlink('删除测试.txt',function(err){

//if(err) throw err;//throw 抛出错误

if(err){

console.log('删除错误')

}else{

console.log('删除成功')

}

})

//rename('目标','修改名称',function(err){})

fs.rename('修改测试.txt','修改完成.txt',function(err){

if(err){

console.log('修改错误')

}else{

console.log('修改完成')

}

})

//读取文件：readFile('目标文件','utf-8',function(err，data){ data 读取到的内容 })

fs.readFile('nodejs02.js','utf-8',function(err,data){

if(data){

fs.writeFile('写入测试.txt',a,function(err){

if(err){

console.log('写入失败')

}else{

console.log('写入成功')

}

})

}

})

//写入文件：writeFile('目标文件','写入内容',function(err){})

//wirteFile() 是覆盖式写入 appendFile()是追加式写入

// fs.writeFile('写入测试.txt','hhhh',function(err){

// if(err){

// console.log('写入失败')

// }else{

// console.log('写入成功')

// }

// })

//如何编写一个复制文件的脚本

//流文件：边读边写（有了封装好的插件很少用到）

//文件上传，

//读取流

var rs = fs.createReadStream('目标文件')

//写入流

var ws = fs.createWriteStream('目标文件')

//data :你读取的文件里面是否有数据

rs.on('data',function(chunk){

ws.write(chunk)

})

rs.on('end',function(){

console.log('写入完毕')

})

1. 教师评语：