今天学习的内容

1:复习昨天知识重点

2:作业

3:今天的目标

3.1:目录模块(重点与难点)

3.2:自定义模块两种形式

(1)文件模块

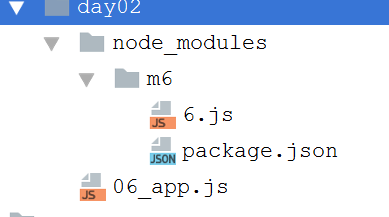
创建js文件，如m3.js 导出需要公开数据，其它的模块

require("./m3");模块

(2)目录模块

方式一:创建一个目录,必须名为node\_modules,其中再创建目录模块，假设目录名m6,其中创建描述文件(名称固定)package.json,其中声明main属性指定默认执行启动js文件,

假:6.js 其中导出需要公共的数据。其它的模块 require("m6");



注意结构

node\_modules 所有目录必须放在此目录下

m6 目录模块

6.js 启动js文件(名称可修改)

package.json 目录模块描述文件(名称不可修改)

{

"name": "m6", m6 模块名称

"version": "0.0.1", 目录模块版本

"author":"tao", 作者

"main":"./6.js" 启动程序

}

练习:使用以上的方式:创建两个模块,

circle/rectangle都对外公开两个方法

size()/perimeter()返回指定图形的面积和周长

最后在最外层的主模块中引入上述两个模块

circle.size(r);

circle.permeter(r);

rectangle.size(w,h);

rectangle.permeter(w,h);

方式二:创建一个目录/假设名为 m4,其中创建名为index.js(固定不变)文件，导出需要公共的数据，其它的模块可以 require("./m4");

方式三:创建一个目录，假设名为m5,其中创建package.json文件，其中声明main属性指定默认启动js文件,如:5.js,其中导出公开数据，

其它的模块可以require("./m5");

|  |
| --- |
| 自己尝试:若有个文件名为m7.js，还有目录  m7/index.js,还有 node\_modules/m7/index.js  最后的app.js  require("./m7"); 引入的哪个模块  require("m7"); 引入的哪个模块 |

小结:nodejs目录模块

(1)node\_modules/m5/package.json/index.js+app require("m5");

(2)m6/index.js + app require("./m6");

(3)m7/index.js/package.json + app require("./m7");

3.3:NPM 包管理器

Node Package Manager;Node.js 管理第三方模块工具,可用于下载,

更新,删除,维护包.

npm 工具默认 www.npmjs.org 网站下载第三方模块

npm install 包名; //安装

npm uninstall 包名;//删除

3.4:原生的模块([http/fs]重点)

3.5:Node.JS官方提供原生模块-- querystring

querystring模块用于处理HTTP请求URL中的查询字符串

var obj = qs.parse(str) ;;把查询字符串解析为JS对象

var str = qs.stringify(obj) ;;把js对象转换为查询字符串

3.6:Node.JS官方提供原生模块-- url

url模块用于解析一个URL请求地址，获取其中不同的部分

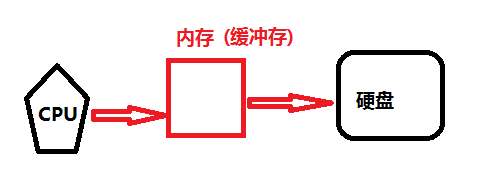
var obj = url.parse(str); 把一个URL地址解析为一个对象

var obj = url.parse(str,true); 把一个URL地址解析为一个对象,并把其中查询字符串也解析为对象

3.7: Node.js官方提供原生模块--Buffer

Buffer:缓冲区，本质是一块内存区域，用于暂存以后要用的数据

(可能是数字，字符串，二进制图片/音频视频),该区域称为缓存.toString()



(1)创建一个指定大小缓冲区

var buf1 = Buffer.alloc(缓冲区大小(字节));

示例:

var buf1 = Buffer.alloc(1024\*2);

(2)使用数字数组创建一个缓冲区

var buf2 = Buffer.from([1,2,3,4]);

(3)使用一个字符串创建一个缓冲区

var buf3 = Buffer.from("abc");

(4)把一个缓冲区的数据转换字符串

var str = buf3.toString();

3.8: Node.js官方提供原生模块--fs---重点

fs模块提供了对文件系统中的文件/目录进行增删改查，读写的功能

同步:二个程序一个等待一个工作

异步:二个程序二个都在工作

同步操作:

var data = fs.readFileSync(file); //同步读取数据

fs.writeFileSync(file,str/buf); //同步写数据

#写操作:默认操作先清除原来的数据/添加

fs.appendFileSync(file,str/buff); //同步追加

#"\r\n" 换行

练习一:使用同步的方法实现 3.txt 文件复制为3.backup.txt

练习二:使用同步的方法实现 p3.png 文件复制为p3.backup.png

练习三:使用同步的方法实现 am.docx 文件复制为am.backup.docx

练习四:使用同步的方法实现birds.mp4 文件复制为birds.backup.mp4

异步操作:

fs.readFile(file,function(err,data){ data}); //异步读取

fs.writeFile(file,str/buf,function(err){err}); //异步写数据

fs.appendFile(file,str/buff,function(err){err});//异步追加

练习一:使用异步的方法实现 3.txt 文件复制为31.backup.txt

练习二:使用异步的方法实现 p3.png 文件复制为p31.backup.png

练习三:使用异步的方法实现 am.docx 文件复制为am1.backup.docx

练习四:使用异步的方法实现birds.mp4 文件复制为birds1.backup.mp4

3.9: Node.js官方提供原生模块--http---重点

http模块可用于编写基于HTTP协议客户端或服务器端程序(浏览器-apache)

用http模块编写一个web服务器

(1)接收客户端http请求

(2)解析客户请求消息

(3)读取客户请请求资源

(4)向客户端发送http响应消息，主体就是客户端请求资源

示例:

var server = http.createServer(); //创建服务器对象

server.listen(80); //绑定监听端口

端口:1-65535 建议 1024以后

server.on("request",function(req,res){

req:请求对象{请求浏览器数据}

res:响应对象{响应并且发送给浏览器数据}

});

作业1：点击操作

作业2：nodejs http模块创建服务器,

如果客户请求 /index.html

读取当前目录下index.html 返回客户端浏览器