今天学习的内容

1:复习昨天重点

2:作业

3:今天的目标

练习1:创建文件模块 circle.js 其中常量PI,

声明二个方法 getSize(r) 求面积

getPerimiter(r); 求周长

创建主模块 02\_app.js 调用其公开二个方法

练习2:创建文件模块 arrayutils.js

包含一个sum(arr); 返回数组值和,

包含一个avg(arr); 返回数组值平均值

创建主模块 03\_app.js 调用其公开二个方法

3.1:nodejs 模块-目录模块

自定义模块的两种形式

(1)**文件模块**

创建一个JS文件,如m3.js,导出其公开数据，

其它模块可以require("./m3");引入

(2)**目录模块**

方式一:创建一个目录,假设m4,其中创建index.js文件，导

出需要公开数据。其它模块引入

var m = require(“./m4”);

方式二:创建一个目录,假设m5,其中创建5.js文件，导出需要

公开数据数据,创建package.json文件 main指定启动

文件 5.js 其它模块引入

var m = require(“./m5”);

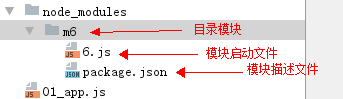
方式三:创建一个目录,必须名为 node\_modules

,其中再创建目录模块,假设 m6 其中创建 package.json

文件，其中声明 main属性指定默认执行启动js,如6.js，

其中导出需要公共数据，其它模块引入

require(“m6”);



|  |  |  |
| --- | --- | --- |
|  | 文件模块 | 目录模块(方式三) |
| 程序结构 | m1.js | node\_modules  m2  package.json  2.js |
| 模块名称 | m1 | m2 |
| 导出数据 | module.exports | module.exports |
| 导入模块 | var m = require(“./m1”); | var m = require(“m2”); |

练习1:创建二个目录模块,circle,rectangle都对外公开两个方法size(); perimeter();返回指定图形面积和周长

最后在最外层主模块中引入上述二个模块四个方法

circle.size(r); circle.perimeter(r);

rectangle.size(w,h);

rectangle.perimeter(w,h);

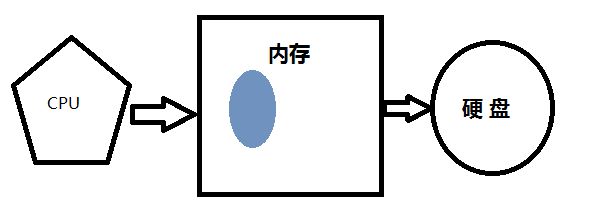
3.2:nodejs 原生模块-->(重点)

Node.js 解析引擎自带

3.3: Node.js 原生模块—Buffer(缓冲区)

Buffer:缓冲区,本质上是一块内存区域，用于暂存以后要用到数据

(数字,字符串,二进制图片/音频/视频),该区域称为“缓存”



#分配置一个指定大小缓冲区 1024字节

建议：缓冲区大小不超过 512KB 512\*1024

var buff1 = Buffer.alloc(1024);

#使用一个数组创建缓冲区

var buff2 = Buffer.from([1,2,3]);

#使用一个字符串创建缓冲区

var buff4 = Buffer.from(“abc”);

#将一个缓冲区内容转换字符串{如果缓冲区数据是字符串}

var str = buff4.toString();

3.4: Node.js 原生模块—QueryString

querystring模块用于处理HTTP请求URL中查询字符串

var obj = qs.parse(str); 把查询字符串解析js对象

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

3.5: Node.js 原生模块—URL

url模块用于解析一个HTTP请求地址，获取其中各个部分

var obj = url.parse(str); 把一个URL地址为js对象

var obj = url.parse(str,true); 功能同上，并且把其中查询字符串转换

js obj

3.6: Node.js 原生模块—fs—(重点)

文件:在node.js 我们系统文件或者目录统称为文件

fs 模块提供对文件和目录进行，增删改查，读定文件内容

1. 同步读写文件(阻塞)

#同步读取文件内容 var data = fs.readFileSync(fileName);

#同步向文件写内容 fs.writeFileSync(fileName,data);

注意:(1)如果当前写入文件并不存在，自动创建一个

a.txt

(2)如果当前写入目录并不存在，报错

d:/abc/a.txt

(3)如果当前文件己经存在内容，清空

#同步向文件追加内容

fs.appendFileSync(fileName,data);

练习:使用上述方法，实现文件复制操作

./public/data.log-->backup\_data.log

1. 异步读写文件(非阻塞)

#异步读取数据 fs.readFile(fileName,function(err,data){ });

函数:文件读取完成后调用

err:读取文件不存在，权限不足,

自动创建错误对象

#异步写数据 fs.writeFile(fileName,data,function(err){})

函数:文件写结束调用

#异步追加数据 fs.appendFile(fileName,data,function(err){})

函数:文件追加结束

练习:使用非阻塞方法,实现public/data.txt文件,复制backup\_data.txt

3.7: Node.js 原生模块--http

HTTP 模块可用于编写基于HTTP协议客户端程序(即浏览器),也可以编写一个基于HTTP协议服务器(APACHE)

APACHE[1000并发量] nginx[28000] IIS[微软]

用http模块编写一个web服务器

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

(2)绑定监听端口 1-65535 server.listen(3000);

(3)接收客户端请求， server.on(“request”,(req,res)=>{..})

(4)并且响应客户消息 res.end(str...);

req:请求对象:保存客户端请求消息

req.url 请求url地址

res:响应对象:服务器发送数据客户端,修改响应格式

作业1:使用node.js创建服务器，根据客户端的请求地址不同

输出不同的html页面内容

http://127.0.0.1:3000/login.html

http://127.0.0.1:3000/admin.html

若请求资源不存在，则返回 404.html