今天学习的内容

1:复习昨天知识重点

NodeJS:开发平台

适用范围:适合IO密集型(微博，微信)项目，不适合CPU密集(天气预报)项目.

前端JS和NodeJS区别

前端JS:运行在客户端浏览器中,存在兼容性问题，数据类型:值类+引用类型(ES+BOM+DOM+自定义)

NodeJS:运行于服务器(V8引擎),不存在兼容性，数据类型:值类+引用类型(ES+第三方对象+自定义)

Node模块系统

Nodejs中每个.JS文件都是一个"Module",每个模块都可以引入其它模块,也可以导出自己的成员(变量/函数/对象)供它的模块使用.

(function (exports, require, module, \_\_filename, \_\_dirname) {

//自己编写内容

var age = 20;

var fn = function(){}

exports.age = age;

module.exports.fn = fn;

//源代码 exports = module.exports;

//##注意事项

module.exports = {age:10,f:fn}; ///OK

exports = {age:10,f:fn}; //ERROR

});

exports与module.exports在使用上最大区别

(1)module.exports可以向外导出对象或者是成员

(2)exports只能向外导出成员

Node模块的类型

(1)官方提供原生模块 global/util/url/fs/http/....

(2)第三方模块 mysql/oracle/exports...

(3)自定义模块 文件模块和目录模块

2:作业

3:新的内容

(1)npm和目录模块管理(有点乱)

(2)常用原生模块---重点&难点

3.1:自定义模块两种形式

(1)文件模块

创建一个js文件如 m3.js 导出需要公开数据，其它模块可以require("m3") 模块

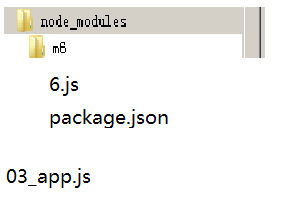
(2)目录模块

方式1:创建一个目录,假设名为m4,其中创建名(必须)为index.js文件，导出公开的数据，主模块可以 require("./m4");

方式2:创建一个目录,假设名为m5,其中创建名package.json文件

,其中声明main属性指定默认执行的js文件,如 5.js,其中导出需要公开的数据。其它模块可以 require("./m5");

方式3:创建一个目录 必须名为node\_modules,在此目录中再创建目录模块，假设m6,在m6目录再创建文件,必须名为package.json文件,其中声明main属性指定默认执行的启动文件名,如:6.js，其中导出公开数据。其它模块可以可以 require("m6")



练习: 使用方式3：创建二个目录模块 circle/rectangle，都对外公开二个方法 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二个模块

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

rectange.size(w,h); rectange.perimeter(w,h);

2:NPM 包管理器

Node Package Manager :NodeJS第三方模块管理器，下载，更新，删除维护包依赖关系的工具.

npm 默认到 www.npmjs.org 网站下载所需第三方模块

标准语法:

(1)下载第三方软件包 npm install 包名

(2)删除指定软件包 npm uninstall 包名

3:NodeJS官方提供原生模块--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(ob); 把js对象转换为查询字符串

4:NodeJS官方提供原生模块-url

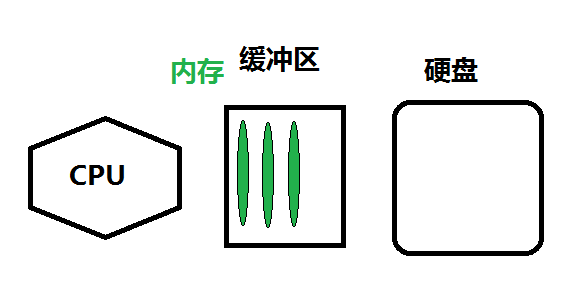
url模块用于解析http请求地址,获取其中各个部分

var obj = url.parse(str); 把一个url字符串解析为一个对象

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

5:NodeJS官司方提供原生模块-Buffer

Buffer:缓冲区,本质是一个内存区域，用于暂存以后要用到数据(可能数字，字符串，图片、音频、视频...)



(1)创建缓存区并且分配指定大小

var buf1 = Buffer.alloc(1024); //字节b

(2)创建缓存区数字数组

var buff2 = Buffer.from([1,4,5,6]);

(3)创建缓存区字符串

var buff3 = Buffer.from("abcde...");

(4)把缓存中数据转字符串

var str = buff3.toString();

6:NodeJS官方提供原生模块--fs----->重点

fs模块提供对文件系统的文件/目录进行增删改查，读写的功能.

(1)同步读取文件中的内容

var data = fs.readFileSync(file);

(2)同步向文件写入内容(删除己有内容)

fs.writeFileSync(file,str/buff);

(3)同步向文件中追加写的内容(不删除己有内容)

fs.appendFileSync(file,str/buff);

(4)异步读取文件的内容

fs.readFile(file,function(err,data){})

(5)异步向文件中写入内容

fs.writeFile(file,str/buf,function(err){});

7:NodeJS官方提供原生模块--http----->重点

HTTP模块可用于编写基于HTTP协议客户端程序(即浏览器);也可以编写基于HTTP协议服务器(web服务器)

用http模块编写一个web服务器

(1)接收客户端http请求消息

(2)解析客户端请求消息

(3)读取客户端请求文件

(4)向客户端发送HTTP响应消息，主体就是客户端请求文件

var server = http.createServer();

server.listen(8080);

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

req 请求对象

res 响应对象

});

作业1:异步复制文件

作业2:使用Node创建一个web服务器，根据客户端请求地址不同，输出不同的html页面内容，如

http://127.0.0.1:8080/login.html

服务器应该返回 ./public/login.html中的内

若请求资源不存在,则返回 404响应消息