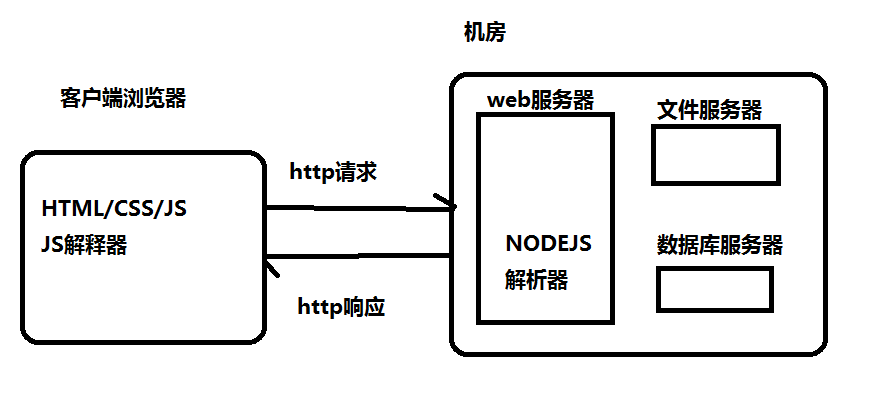
1:复习昨天重点内容

#NodeJS是什么:

NodeJS是个软件开发平台运行服务器，

使用开发语言是JS,适用IO密集型应用.



#JS和NODEJS区别

JS运行于客户端浏览中，存在兼容性问题;数据类型：值类型+引用类型(ES+DOM+BOM+自定义);

NODEJS运行服务端(V8),不存在兼容性问题,数据类型:值类型+引用类型(ES+扩展对象+自定义);

#NODEJS模块系统

NODEJS中每一个.js文件都是一个"Module",每个模块可以引入其它的模块;也可以导出自己的成员供其它模块来使用.

NODEJS中的每一个模块都是一个独立构造函数，解释器会为每个.js文件添加如下代码.

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

var age = 20;

var stop = function(){

}

exports.age = age;

module.exports.stop = stop;

})

练习:仿写NODEJS提供一个模块:URL

创建模块文件myurl 向外导出一个方法resolve(url),该方法接收参数形如下:

http://www.jd.com:80/ad/index?uname=qiangdong

返回对象：形如

{

protocol:'http',

hostname:'www.jd.com',

port:80,

url:"/ad/index?uname=qiangdong"

}

再编写一个主模块，引入上述模块并调用.

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导出需要公开的数据，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,在m4目录中创建名为index.js的文件，导出需要公开数据。其它模块可以require("./m4");

方式2:创建一个目录，假设名为m5,其中创建package.json文件，其中声明main属性指定默认执行js文件，如5.js,其中导出需要公开数据。其它模块可以require("./m5");

方式3:创建一个目录,必须名**node\_modules**,其中创建目录模块，假设m6,其中创建文件**package.json**文件,其中声明**main**属性指定默认执行启动js文件,如：6.js 其中导出需要公开数据，其它模块可以require("m6");

练习:使用方式3：创建两个目录模块，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

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

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

2:NPM 包管理器-了解

Node Package Manager Node.js 的第三方模块/包管理器，可用于下载，更新，删除，维护依赖关系的工具。

npm工具默认到www.npmjs.org网站下载所需的第三方模块包。

npm install 包名

npm uninstall 包名

3:NodeJS官方提供的原生模块--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

4:NodeJS官方提供原生模块-url

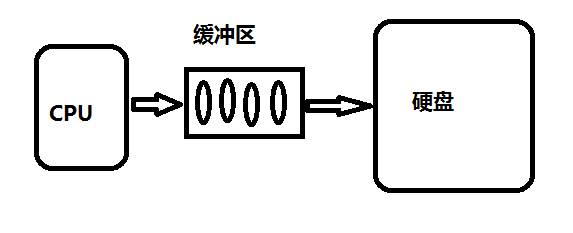
url模块用于解析一个http请求地址，获取其中各个不同部分。

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

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

5:NodeJS官方提供的原生模块--Buffer

Buffer :缓冲区本质是一块**内存**区域，用于暂存以后要用到的数据(可能数字，字符串，图片，音频，视频)，该区域称为“缓存”。



#分配一个指定大小缓冲区 {1024单位字节}

var buf1 = Buffer.alloc(1024);

#使用一个数字数组创建缓冲区

var buf2 = Buffer.from([1,3,5]);

#使用一个字符串创建一个缓冲区

var buf3 = Buffer.from('abcde');

#把一个缓冲区中数据转字符串

var str = buf3.toString();

6:Nodejs官方提供原生模块--fs--重点

#var data = fs.readFileSync(file) ;同步取读文件的内容

#fs.writeFileSync(file,str); 同步向文件写内容

#fs.appendFileSync(file,str); 同步向文件追加内容

#异步读取文件的内容

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

#异步写文件的内容

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

提示:后续的项目中，文件读写可以同步或异步，但推荐使用异步方法--最大限度发挥nodejs优势.

7:nodejs官方提供的原生模块--http -->重点

HTTP模块可用于编写基于HTTP协议客户程序(即浏览器)，也可以编写基于HTTP协议的服务器程序(即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(request,response){

request 请求消息

response 响应消息

});

练习:NodeJS创建一个web服务器，根据客户请求的地址不同，输出不同html页面内容,如

http://127.0.0.1:8080/login.html

服务器应该返回 ./public/login.html 中读内容

http://127.0.0.1:8080/news.html

服务器应该返回 ./public/news.html 中读取内容

若请求资源在public目录不存在,则返回404响应消息.