# day03

## 1.fs模块

(1)创建文件/写入文件

writeFile(fd,data,callback)/writeFileDync(fd,data)

fd 文件的路径

data要写入的数据

callback 回调函数

如果文件不存在会创建并写入，如果稳健已经存在，清空里边内容然后写入

(2)追加写入

appendFile(fd,data,callback)

练习：使用追加写入数据abcde到文件2.txt,运行多次。

练习：将一组数据写入到文件3.txt[‘biubiu’,’平凡而伟大’,’到南方去’,’到北方去’];

(3)读取文件

readFile(fd,callback)/readFileSync(fd)

读取的结果是buffer数据，需要转为字符串

(4)删除文件

unlink(fd,callback)/unlinkSync(fd)

(5)判断文件/目录是否存在

existsSync(fd) 存在->true 不存在->false

(6)文件流Stream

var readStream=createReadStream(文件路径) 创建可读流

readStream.on(‘data’,function(chunk){

//当有数据进入到内存中，自动触发该事件

//分块来读取数据

chunk 就是读取某一块数据

})

readStream.on(‘end’,function(){ })

当数据读取结束，触发该事件

var writeStream=fs.createWriteStream(文件路径)

pipe 管道，可以将数据使用管道写入到文件

readStream.pipe(writeStream);

## 2.http协议

是浏览器和web服务器之间的通信协议

(1)通用头信息 General

Request URL:请求的URL，对应的浏览器地址栏，向服务器获取哪些内容

Request Method:请求的方法，获取数据的方式post/get

Status Code:响应的状态码

1\*\*：正在响应，还没有结束

2\*\*：成功的响应

3\*\*：响应的重定向，会跳转到另一个网址

4\*\*：客户端请求错误

5\*\*：服务器端错误

Remove Adress：请求的服务器的ip地址和端口号

(2)响应头信息 Response headers

Connection：连接方式 keep-alive 持久连接

Content-type:响应的文件类型

Content-Encoding:压缩形式 gzip

Transfer-Encoding:传输方式(流Stream),chunked分块传输

Location:跳转的URL，常配合着状态码3\*\*使用。

(3)请求头信息 Request Headers

Accept:浏览器端接收的文件类型有哪些

Accept-Encoding:浏览器端接收的压缩形式

User-Agent:发送请求所使用的浏览器

(4)请求主体

可有可无，浏览器向服务器发请求传递的数据

## 3.http模块

既可以模拟浏览器向服务器发请求，还可以创建web服务器

(1)模拟浏览器

http.get(url,callback)

get 请求的方法

callback 回调函数，用于获取服务器端的响应

res 响应的对象

res.statusCode 获取响应的状态码

res.on(‘data’,function(chunk){ })

事件：监听服务器端是否有数据流传输进来

chunk 就是传输的数据，格式默认为buffer

(2)创建web服务器

var app=http.createServer();//创建web服务器

app.listen(8080);//监听端口

app.on(‘request’,function(req,res){

//监听浏览器的请求

req 请求的对象

req.method 请求的方法

req.url 请求的URL

req.headers 请求的头信息

res 响应的对象

res.writeHead(statusCode,headers) 设置响应的状态码和头信息

res.write() 设置响应的内容

res.end() 结束并发送响应到浏览器端

});

练习：创建web服务器，监听端口8081，接收浏览器的请求，根据请求，做出不同的响应 (/login 响应内容 this is login page;/reg 响应内容 this is reg page;/study 跳转到 <http://www.tmoo.cn)以上都没有，响应内容> 404 not found

## 4.express框架

基于NodeJS,快速、开发、极简的web开发框架

<http://www.npmjs.com/package/express>

http://www.expressjs.com.cn

下载安装：nmp install express

|  |
| --- |
| const express=require(‘express’);//引入第三方的express  var app=express();//创建web服务器  app.listen(8080);//监听端口 |

课后任务：

1. 复习，整理思维导图
2. 练习：使用http模块，创建web服务器，接收浏览器的请求，根据请求做出不同的响应（/index 响应内容 this is homepage;/login 响应内容 <h2>please login</h2>;/ 跳转到 /index;其它 响应 404 not found）
3. 预习：express中的路由，get和post