复习

模块:自定义模块、核心模块、第三方模块

require('./01.js')

const querystring=require('querystring')

parse/stringify

url

parse/format

query

fs

fs.stat(path, callback) / fs.statSync(path)

fs.mkdir(path, callback)

npm install mysql

require('mysql')

http://www.codeboy.com:80/detail.html?lid=5&a=2

setImmediate/clearImmediate

process.nextTick

1.fs模块

(1)创建目录

fs.mkdir(path, callback) / fs.mkdirSync(path)

(2)移除目录

fs.rmdir(path, callback) / fs.rmdirSync(path)

(3)读取目录

fs.readdir(path,callback)/fs.readdirSync(path)

callback

err

result 读取的结果，格式为数组

(4)写入文件

fs.writeFile(path,data,callback)/fs.writeFileSync(path,data)

如果文件不存在，会先创建文件，然后写入；

如果文件已经存在，会先清空内容，然后写入

(5)追加写入

fs.appendFile(path,data,callbak)/

fs.appendFileSync(path,data)

如果文件不存在，会先创建文件然后写入

如果文件已经存在，会在末尾追加写入

(6)读取文件

fs.readFile(path, callback)/fs.readFileSync(path)

callback

err

data 读取的数据，格式为buffer

(7)删除文件

fs.unlink(path, callback)/fs.unlinkSync(path)

(8)检测是否存在

fs.existsSync(path)

存在返回true ，不存在返回false

练习：如果文件data2.txt存在，同步删除它

(9)文件流

fs.createReadStream(path) 创建可读取的流

|  |
| --- |
| readStream.setEncoding('utf8');//设置编码类型  readStream.on('data',function(chunk){ }); //监听是否有数据流，自动触发，通过回调函数获取数据；chunk就是读取数据，是分段的。  readStream.on('end',function(){ }); //监听是否读取结束 |

fs.createWriteStream(path) 创建可写入的流

pipe() 管道，可将数据从读取的流添加到写入的流

2.http协议

浏览器和web服务器之间的通信协议

(1)通用头信息

Request URL: 向服务器获取哪些内容，对应浏览器地址栏

Request Method: 请求的方法 get/post，打开网页默认都是get

Status Code: 响应的状态码

1\*\*: 正在请求，响应还没有结束

2\*\*: 成功的响应

3\*\*: 响应的重定向，跳转到另一个URL

4\*\*: 客户端请求错误

5\*\*: 服务器端错误

Remote Address: 请求的服务器的IP地址和端口号

(2)响应头信息

Connection: 建立的连接方式，keep-alive 持久连接

Content-Encoding: 服务端内容压缩形式 gzip

Content-Type: 文件的类型

Transfer-Encoding: 传输方式 chunked 分段传输

Location: 跳转的URL，常结合着状态码3\*\*使用

(3)请求头信息

Accept: 浏览器接收的文件类型有哪些

Accept-Encoding: 接收的压缩类型有哪些

User-Agent: 客户端使用的浏览器

(4)请求主体

可有可无，用于浏览器向服务器发送的数据

3.http模块

既可以创建web服务器，也可以模拟浏览器向其它的服务器发请求

(1)模拟浏览器

http.get(url, callback) 向服务器发请求

url 请求的url

callback 回调函数，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){ })

监听服务器端是否有数据传输

chunk: 就是传输的数据，格式为buffer

(2)创建web服务器

|  |
| --- |
| var app=http.createServer(); //创建web服务器  app.listen(8080); //设置监听的端口  //监听浏览器的请求,使用回调函数处理请求  app.on('request', function(req,res){  req 请求的对象  req.url 请求的URL，端口号后的部分，表示要获取的内容  req.method 请求的方法  res 响应的对象  res.writeHead(状态码, 头信息对象) 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  }) |

课后任务

(1)复习今天内容，整理思维导图

(2)练习

创建web服务器，监听端口8080，接收浏览器的请求，根据请求作出响应

/list 响应内容 'this is product list'

/index 响应内容 'this is homepage'

/ 跳转到 /index

/study 跳转到 http://www.tmooc.cn

其它 响应状态 404 内容 not found

(3)express第三方模块 mysql模块 数据库