1.fs模块

|  |
| --- |
| 回调函数  参数1：可能产生的错误  参数2：成功的结果，只有结果是数据形式才会出现 |

(1)读取目录

readdirSync(目录的路径) / readdir(目录的路径, 回调函数)

读取的结果为数组

(2)清空写入文件

writeFileSync(文件的路径,数据) / writeFile(文件的路径,数据,回调函数)

如果文件不存在，则先创建文件然后写入数据

如果文件已经存在，会清空文件内容然后写入数据

(3)追加写入文件

appendFileSync(文件的路径,数据) / appendFile(文件的路径,数据,回调函数)

如果文件不存在，则先创建文件然后写入数据

如果文件已经存在，会在文件的末尾追加写入数据

练习：使用同步的方法往2.html写入数据 '这是一条新闻'

(4)读取文件

readFileSync(文件的路径) / readFile(文件的路径, 回调函数)

读取的结果为buffer数据格式

(5)删除文件

unlinkSync(文件的路径) / unlink(文件的路径, 回调函数)

练习：分别删除文件1.txt 和 2.html

(6)判断文件是否存在

existsSync(文件的路径)

存在->true 不存在 -> false

(7)拷贝文件

copyFileSync(源文件路径, 目标文件路径) / copyFile(源文件路径, 目标文件路径, 回调函数)

练习：如果目录mydir不存在则创建该目录，如果文件stu.txt存在，则把文件拷贝到mydir目录

2.文件流

createReadStream() 创建可读取的文件流

createWriteStream() 创建可写入的文件流

on(事件名称, 回调函数) 添加事件，事件名称是固定的字符串，一旦监听到事件会自动执行回调函数

pipe() 管道，可以将读取的流添加到写入的流

3.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：请求的服务器端的资源

Request Method：请求的方法，对资源的操作方式 get/post/delete/put

Status Code：响应的状态码

1\*\*：接收到了请求，还没有做出响应

2\*\*：成功的响应

3\*\*：响应的重定向，跳转到另一个URL

4\*\*：客户端错误

5\*\*：服务器错误

(2)响应头信息(response)

Content-Type：响应的内容类型 text/html;charset=utf-8

Location：设置要跳转的URL

(3)请求头信息(request)

(4)请求主体

显示传递的数据，不是每一次都出现

4.http模块

可以用来创建WEB服务器

(1)创建服务器

|  |
| --- |
| const http = require('http'); //引入http模块  const app = http.createServer();//创建WEB服务器  app.listen(8080,()=>{ });//设置端口 |

(2)接收请求作出响应

|  |
| --- |
| //给服务器添加事件  app.on('request', (req,res)=>{  req 请求对象  req.url 获取请求的URL，显示的端口号后的部分，例如 /jianbing  req.method 获取请求的方法  res 响应对象  res.writeHead(状态码, 头信息) 设置响应的状态码和头信息，第二个参数可以为空  res.write() 设置响应到浏览器的内容  res.end() 结束并发送响应  }); |

练习：编写文件05\_http.js，创建WEB服务器，设置端口为8080，根据请求作出响应

/ 响应 '<h1>这是首页</h1>'

/study 跳转 'https://www.tmooc.cn'

/login 响应网页 1.html （先读取文件的数据，然后作为响应的内容）

其它 响应 404 not found

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，设置端口8080，根据请求作出响应

/index 响应 '<h1>学子商城</h1>'

/ 跳转到 /index

/reg 响应 reg.html

其它 404 not found

(3)预习nodejs第4天 express