1.同步和异步

同步：阻止后续代码的执行，先执行完当前的同步操作才能往后执行，通过返回值获取结果

异步：不会阻止后续代码执行，在一个独立的线程，执行完以后把结果放入到回调函数中，通过回调函数获取结果

2.fs模块

(1)查看文件状态

statSync( 文件路径 ) / stat(文件路径, 回调函数)

回调函数 获取文件状态

err 可能产生的错误

s 文件的状态

(2)读取目录

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

结果是一个数组

(3)覆盖写入文件

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

如果文件不存在，先创建文件然后写入数据

如果文件已经存在，覆盖原来内容写入数据

(4)追加写入文件

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

如果文件不存在，先创建文件然后写入数据

如果文件已经存在，会在文件末尾追加写入数据

练习：创建一个数组，包含有多个姓名，使用同步方法将数据写入到文件stu.txt中。

(5)读取文件

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

读取的文件数据格式为Buffer

(6)删除文件

unlinkSync(文件的路径)/unlink(文件的路径,回调函数)

删除1.txt 2.txt

(7)检测文件是否存在

existsSync(文件的路径)

存在-true 不存在-false

(8)拷贝文件

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

(9)文件流

createReadStream() 创建可读取的流

createWriteStream() 创建可写入的流

pipe() 用于将读取的流添加到写入流

|  |
| --- |
| 监听事件：一旦触发事件，自动调用回调函数  on( 事件名称, 回调函数 ) |

2.http协议

http协议是浏览器和web服务器之间的通信协议

(1)通用的头信息

Request URL: 请求的URL

Request Method: 请求的方法 get/post

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，跳转

4\*\*: 客户端请求错误

5\*\*：服务器错误

(2)响应的头信息response

Content-Type: 响应的内容类型 text/html

Location: 响应后要跳转的URL ，结合着状态码302使用

(3)请求的头信息request

(4)请求的主体

可有可无，如果没有传数据不显示这项

请求过程中表单传递的数据

3.http模块

可以用来创建web服务器

|  |
| --- |
| http.createServer() 创建web服务器  listen() 设置端口  //给服务器添加事件，用于接收请求，以及做出响应  app.on('request', (req,res)=>{  req 请求的对象  req.url 请求的URL 对应端口号后的部分，格式 /login  req.method 请求的方法  res 响应的对象  res.writeHead() 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，设置端口，根据浏览器请求的URL做出响应

/index 响应 '<h2>这是首页</h2>'

/list 响应文件 1.html 先同步读取文件数据，把读取结果响应

/study 跳转到 http://www.tmooc.cn

其它 响应404 not found

(3)预习nodejs第4天