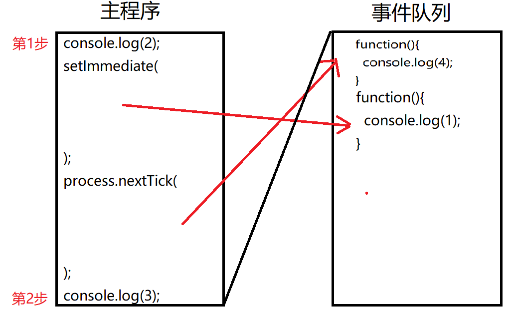
一.立即执行定时器

process.nextTick()



二.文件系统模块

用于文件操作

文件分为目录形式和文件形式

1.查看文件的状态

statSync(文件的路径)/stat(文件的路径, 回调函数)

isDirectory() 是否为目录

isFile() 是否为文件

2.创建目录

mkdirSync(目录的路径) / mkdir(目录的路径, 回调函数)

3.移除目录

rmdirSync(目录的路径) / rmdir(目录的路径, 回调函数)

只能移除空目录

练习：分别使用两个方法移除mydir1和mydir2

4.读取目录

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

5.清空写入文件

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，会先创建文件然后写入

如果文件已经存在，会清空内容然后写入

练习：使用同步往文件2.txt中写入数据

6.追加写入

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，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

练习：使用同步方法往文件3.txt写入数据

7.读取文件

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

练习：使用异步方法读取stu.txt文件

8.删除文件

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

练习：删除1.txt

9.检测文件(目录)是否存在

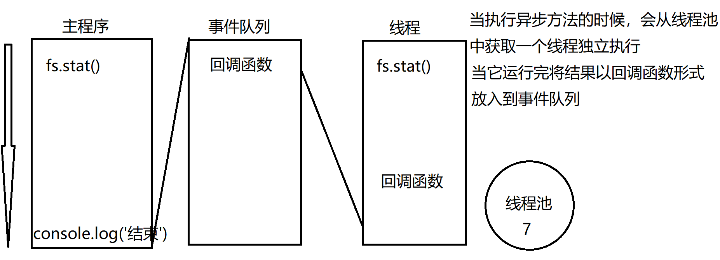
existsSync(文件路径)

只有同步一种方法，返回布尔型的值

三.同步和异步

同步：是在主程序中执行，会阻止后边代码的执行，通过返回值获取结果

异步：是在一个独立的线程执行，不会阻止主程序后续代码执行，通过回调函数获取结果



四.文件流

createReadStream(文件的路径) 创建可读取的文件流，分段读取文件

createWriteStream(文件的路径) 创建可写入的文件流，会创建文件

pipe() 管道，通过管道可以将读取的流添加到写入的流，完成文件的拷贝

on('事件名称', 回调函数) 添加事件，监听某一个动作，会自动调用回调函数；事件名称是固定的字符串。

五.http协议

超文本传输协议

是客户端和WEB服务器端之间的通信协议

1.通用头信息(general)

Request URL：请求的URL，要请求的资源

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

Status Code：响应的状态码，请求的结果

1\*\*：接收到了请求，还没有结束响应

2\*\*：成功的响应

3\*\*：响应的重定向，发生跳转

4\*\*：客户端错误

5\*\*：服务器错误

2.响应头信息(response)

Location：设置要跳转的URL，通常配合3\*\*使用

Content-Type：设置响应的内容类型，中文乱码解决(text/html;charset=utf-8)

3.请求头信息(request)

六.http模块

可以用来创建WEB服务器

createServer() 创建WEB服务器

listen() 设置端口

|  |
| --- |
| 通过事件接收请求并作出响应  on('request', (req,res)=>{  req 请求的对象  res 响应的对象  res.setHeader() 设置响应的头信息  res.write() 设置响应到浏览器的内容  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：使用http模块创建WEB服务器并设置端口，添加事件接收浏览器的请求并作出响应 ‘<h1>欢迎来到学子商城~</h1>’

(3)预习Node.js第4天

http://www.codece.com/archives/145