nodeJS的一些方法

目录：

Event

FileSystem

Net

Http

主体：

Event

操作事件的

FileSystem

操作文件

I/O 操作

I : Input

O : Output

注意：

在node中有一个约定，如果一个操作是异步的，同时如果他还支持回调的话，那么回调函数的第一个参数就是error对象，如果没有错误，则第一个参数是null，那么这个约定又称为：错误优先 Error first

fs对象下的方法和属性

**读**

fs.writeFile（file，内容,fn(error)）覆盖

fs.writeFileSync（file，内容,fn(error)）覆盖

**写**

fs.readFile（file,fn(error,chunk)）

fs.readFileSync（file,fn(error,chunk)）

**追加**

fs.appendFie（file，内容,fn(error)）追加

**删除**

fs.unlink（file）删除

fs.unlinkSync（file）删除

**生成文件夹或文件**

fs.mkdirSync（file）——vue的脚手架，生成模板

fs.mkdirSync（index.css）——vue的脚手架，生成模板

**读取文件夹**

fs.readdir()

fs.statSync()——资源管理器

**生成promise对象**

fs.promises——新建一个promise对象

Net

服务端

创建方法一：new net.Server();

创建方法二：server = **net**.createServer(**function**(client){})

——当socket接收到数据的时候触发的事件,,client

监听：server.listen(4444, **'0.0.0.0'**)

client : 也是一个socket对象，就是当前连接到服务器的客户端socket

当有客户端连接到这个server，那么server就会触发一个connection事件

client.on(**'data'**,fn(chunk){})

客户端

创建：**net**.createServer(**function**(client){})

——返回的一个Socket对象,,client

client.on(**'data'**,fn(chunk){})

HTTP

服务端

创建方法一： new http.Server()

创建方法二：http.createServer(**function**(request, response))

监听：server.listen(4444, **'0.0.0.0'**)

——当socket接收到数据的时候触发的事件

http的回调函数中会接收到两个对象

request：包含了当前这次连接的客户端的信息，包括客户端的IP，请求地址，请求协议，请求头等等

response：封装服务端的一些方法，输出

注意：无论是客户端还是服务器，当调用了end方法，就表示这次请求结束了，http协议：一次性消费，请求响应完成以后结束了，同时客户端与服务器的连接就会被断开

response.write(content);

response.end();

http的思想：

无论是客户端还是服务器，当调用了end方法，就表示这次请求结束了，http协议：一次性消费，请求响应完成以后结束了，同时客户端与服务器的连接就会被断开

客户端

创建： **http**.request（{}，fn(response)）

——当socket接收到数据的时候触发的事件

fn这个函数是连接成功以后的回调函数

response这个函数的第一个参数就是服务端对象

response.on(**'data'**, (chunk)

response.on(**'end'**,fn)