一、nodejs基本介绍

1、 阻塞==》同步执行 非阻塞==》异步执行

2、在 Node.js 中 JavaScript 的执行是单线程的，因此并发性是指事件循环在完成其他工作后执行 JavaScript 回调函数的能力。任何预期以并行方式运行的代码必须让事件循环能够在非 JavaScript 操作（比如 I/O ）执行的同时继续运行。

3、事件循环不同于许多其他语言的模型，其它语言创建额外线程来处理并发工作。

4、事件循环机制解析

主线程

客户代码（同步的）

异步请求1

其他线程

异步请求2

异步请求3

取事件

事件队列

执行回调

5、单线程：其实这里的“单线程”指的是我们（开发者）编写的代码只能运行在一个线程当中（可以称之为主线程吧），就像我们在Windows桌面程序开发中一样，编写的所有界面代码均运行在UI线程之中。

在nodejs中写的代码（包括回调方法）均只运行在一个线程中，但是不代表它只有一个线程。nodejs中许多异步方法在具体的实现时，内部均采用了多线程机制（具体后面会讲到）。

二、模块：

1、commnd.js规范

2、定义模块：一个js文件就是一个模块；

3、输出模块：module.exports={};exports.method(or property)=\*\*\*\*;(注意这两种写法)；

4、引入模块：require()函数；注意：要写模块的相对路径。如果只写模块名称的话，则Node会依次在内置模块、全局模块和当前模块下查找hello.js

三、基本模块：

1、global对象：JavaScript有且仅有一个全局对象；浏览器中叫window对象，node中叫global对象。

2、process对象：也是Node.js提供的一个对象，它代表当前Node.js进程。

3、fs对象：文件系统模块，负责读写文件；和所有其它JavaScript模块不同的是，fs模块同时提供了异步和同步的方法。

fs.readFile(url,data,function(err,data){])//异步读取文件

fs.readFileSync(url,data);//同步读取文件；

fs.writeFile(url,data,function(err){});//异步写入文件；

fs.writeFileSync(url,data)//同步写入文件；

fs.stat(url,function(err,data){})//获取文件信息的异步函数；返回一个对象；data.size;data.birthtime;data.mtime;data.isFil(); data.isDirectory();

var sd=fs.stat(url);//同步获取文件信息；

4、1）tream是Node.js提供的又一个仅在服务区端可用的模块，目的是支持“流”这种数据结构。

rs=fs.createReadStream(url,'utf-8');

rs.on('data',function(chunk){}),rs.on('end',fn);rs.on('err',function(data));

ws=fs.createWriteStream(url,'utf-8');

fs.createWriteStream(new Buffer('str','utf-8'));

pipe();连接两个流；rs.pipe(ws);

默认情况下，当rs流的数据读取完毕，end事件触发后，将自动关闭ws流。如果我们不希望自动关闭ws流，需要传入参数：rs.pipe(ws, { end: false }

2） Stream 有四种流类型：Readable - 可读操作。Writable - 可写操作。Duplex - 可读可写操作。Transform - 操作被写入数据，然后读出结果。

3）所有的 Stream 对象都是 EventEmitter 的实例。常用的事件有：data - 当有数据可读时触发。end - 没有更多的数据可读时触发。error - 在接收和写入过程中发生错误时触发。finish - 所有数据已被写入到底层系统时触发。

5、http对象：request对象封装了HTTP请求，我们调用request对象的属性和方法就可以拿到所有HTTP请求的信息；

6、response对象封装了HTTP响应，我们操作response对象的方法，就可以把HTTP响应返回给浏览器。

7、crypto对象: crypto模块的目的是为了提供通用的加密和哈希算法。

8、模块系统：