官网 <https://nodejs.ogg>

中文手册 nodejs.cn

### 全局对象

node.js的全局对象 global

在脚本模式下声明的变量和函数都是局部作用域下的。

JS的全局对象 window

### console对象

console.log 标准打印

console.warn 打印警告

console.error 打印错误

console.info 打印信息

计算程序运行时间

console.time(‘相对应’)

console.timeEnd(‘相对应’)

### process对象

process.arch 查看CPU架构类型

process.platform 查看操作系统的类型

process.version/versions 查看nodejs版本

process.env 查看当前计算机的环境变量

process.memoryUsage 查看内存使用情况

process.uptime 查看nodejs运行时间

process.pid 查看当前进程的编号

process.kill()杀死某个编号的进程

### buffer对象

是global对象下的一个对象；用于将数据存储在缓冲区中，数据相对较小。

创建buffer Buffer.alloc(5,’缓冲内容’)

### 全局函数

#### 定时器

①一次性定时器

|  |
| --- |
| 开启 setTimeout（参数1 参数2）  参数1是回调函数，间隔时间到了，所执行命令  参数2，间隔时间，单位毫秒  清除  clearTimeout（） |

②周期性定时器

|  |
| --- |
| 开启  timer=setInterval()  清除  clearInterval（） |

③process.nextTick(function(){})

④setImmediate()

### 同步和异步

同步：程序按顺序执行，先执行前面的，后执行后边的，

异步：程序在执行的过程中，遇到异步处理函数，把这些函数放到整个程序的最后进执行。异步非等待。

特点：使用回调函数

### 模块

文件模块，就是一个独立的作用域

|  |
| --- |
| Nodejs中的模块文件的本质，是一个构造函数方法内的执行体  function(exports,require,module,\_\_filename,\_\_dirname){  中间部分是程序员写的代码  }红色部分是nodejs自定添加的代码 |
| //exports 导出  //require 导入  //module 模块  //\_\_filename 显示完整的路径和文件名称  //\_\_dirname 显示路径  // module.exports 作用是导出模块，类型是对象 |

课后练习：创建圆周长和面积（木块1），

预习http协议

进程和线程

虚拟目录部分，第一个/到最后一个/

http协议



锚部分，必须有

参数部分

文件名部分

端口部分

域名部分

协议部分

URI URL URN

复习

全局对象——nodejs(global), JS(window)

console对象 consle.log/warn/info/error/time/timeEnd

process对象

arch/platform/version/versions/env/pid/kill

全局函数

一次性定时器:setTimeOut() clearTimeOut()

周期性定时器：setInterval() clearInterval

process.nextTick()

setImmediate()

同步和异步

异步非等待 同步等待

模块

文件模块（JS文件），文件作用域 module.exports require（）

核心模块

第三方模块

带路径文件模块

#### 不带路径的文件模块

带路径的目录模块

#### 不带路径的目录模块

|  |  |  |
| --- | --- | --- |
|  | 带路径的 | 不带路径的 |
| 文件模块 | require(./circle.js)  通常用于用户自定义的模块 | Require（’querystring’）  用于官方定义的核心模块 |
| 目录模块 | Require（./2）  在2这个目录下存在一个index.js  或者使用package.json声明main属性 | Require(‘3’)  要求把3目录放在node\_modules中  通常用于第三方模块。 |

包和npm

npm -> Node Package Manage 包管路工具

包：一个目录模块，含有任意的JS文件，含有一个命名为package.json包说明文件。

淘宝NPM镜像 使用CNPM命令，需要单独安装 npm install cnpm –global

下载包

npm install 包名称

#### querystring 模块

查询字符串模块

parse 将查询字符串解析为对象

stringify 将对象转换为查询字符串

querystring.parse( )

querystring.stringify( )

#### url模块

parse 将url解析为对象 url.parse( )

format 将对象解析为url url.format

#### fs模块

文件系统模块，可以实现对文件（目录）的操作（创建，写入，删除，读取）

异步函数：非阻塞模式，通过回调函数来获取结果

同步函数：阻塞模式，通过函数的返回值来获取结果

state/stateSync

isFile()

isDirectory（）查看状态是否为目录 返回结果为布尔型

mkdir 创建目录

rmdir 删除目录

练习：检查文件myDir的状态，如果不存在，创建目录。

readdir 读取目录，读取结果是数组

readFile 读取文件，读取结果是buffer数据，使用toString()转为普通的字符串

writeFile 写入文件，如果文件不存在，则创建。如果文件已经存在，则清空文件里内容，然后写入

appendFile 追加写入文件，如果文件不存在则创建，如果文件已经存在，则会在末尾写入

unlink 删除文件

练习：实现计数器功能，要求没运行一次数字加1.

①如果文件不存在则创建，num.txt. 写入1

②读取文件，把结果+1，并打印出来

③把+1的结果放入文件

(2)课后作业：

获取url中查询字符串的值，并且把这些值写入str.txt中(如果不存在创建)

http://www.codebody.com/list.html? lid=10&pname=apple&price=5000

复习

模块：文件模块 核心模块 第三方模块

buffer 数据形式

buffer.alloc(5,’abcde’) 得到的数据是Unicode码

url模块 parse format

fs模块

异步：通过回调函数来获取结果——回调地狱

异步函数会放到程序最后执行，通常用于一些耗费时间，耗费内存的代码，对于一些执行速度快的代码不需要使用异步。

同步：通过返回值获取结果 fs.existsSync( );结果是布尔型

stat statSync

mkdir readdir rmdir

writeFile appendFile readFile unlink

#### 回调地狱解决方法：promise

2018.8.21

### http协议

web服务器和浏览器之间通信遵循的协议

包含请求 和 响应

请求方法 ：GET POST

get:从服务器端获取内容（默认）

post 向服务器端发送内容（注册，登录）

请求状态码：

1\*\*：正在响应

2\*\*：成功响应

3\*\*：响应的重定向到其他URL

4\*\*：客户端请求错误

5\*\*:服务器端运行错误

### http模块

构建web服务器

createServer（） 创建web服务器

listen（）监听端口的变化

server.on(‘request’(req,res)=>{

req.method

req.url

req.headers

})

writeHead 响应头信息

write 设置响应到浏览器页面的内容

end 结束响应

### express框架

express用于构建web服务器的框架

中文官网 [www.expressjs.com.cn](http://www.expressjs.com.cn)

--save –div

### 路由

http客户端发来消息，服务器端根据请求的不同的URL，来显示不同的内容

路由组成：请求的方法（get post），请求的url，响应的方法（回调函数）

#### 请求的对象（req）

url method

req.query 获取请求的url中查询字符串——返回一个对象

req.headers 获取头信息

#### 响应对象res

res.send( )发送一段文本到浏览器

res.sendFile( ) 发送文件到浏览器，文件使用绝对路径（\_\_dirname）

res.redirect( )响应重定向

课后任务

思维导图

使用express创建路由， get /index 文本

get ‘/shopping’ 文件

get ‘/’ 重定向到index

复习：

HTTP协议：浏览器和web服务器通信的协议；

浏览器发出请求，web服务器作出响应。

请求方法：get post

请求状态码：

nodejs http 模块 模拟浏览器发出请求

http模块 构建web服务器

createServer listen(3000)

server.on(‘request’,(req,res)=>{

req.method/url/headers.params

res.write/writeHead/end

})

nodejs 框架 express

const express=require(‘express’)

路由功能

2018.8.22

#### 1.请求的方法

post 只能用表单提交的方式（现阶段）

get 可以通过提交表单提交，也可以在浏览器直接输入url

两者的区别

post不会在地址栏暴露输入的数据，相对安全，get会在地址栏显示输入的数据，相对不安全。

post 适用于安全性较高的 如：注册，登陆

get 适用于安全性较低的，例如;搜索关键字

get的执行速度比较快

#### 路由的参数

服务器端配置路由参数

app.get(‘/list/:lid/:uname’,(req,res)=>{

//req.params 获取传递参数对象

})

### 路由器

把相关的路由归纳到一起

路由器的目的是便于管理路由，最终要用于web服务器

### require引入第三方机制

先查看当前目录是否有node\_modules,如果没有就往上一级找，直至根目录

### 中间件

中间件就是express函数，作用是为主逻辑服务，中间件不能单独使用，需要配合其他的中间件或者路由，来实现一个完整的业务逻辑。

中间件拦截路由

app.use(fn) 拦截所有的 路由

app.use(‘/list’ fn)拦截特定的路由

#### 中间件分类

应用级中间件 app.use(fn)

路由级中间件：路由器使用

app.use(‘./user’,user)

内置中间件：

第三方中间件：

错误级中间件：（自学）

①内置中间件——static 托管静态资源

sendFile只有在请求的时候才会发送到客户端

app.use(express.static(‘./public’))

注意：如果两个目录有两个相同的文件名称，网页显示第一个目录下文件

②body-parser

引入中间件：const bodyParser=require(‘body-parser’)

调用中间件

app.use(bodyPaser.urlencoded({extended:false}))

req.body

课后练习，

注册文件 register.html(1234),把这个文件作为静态资源托管，用post请求将数据发送给服务器，用中间件body-parser 来获取数据

复习sql语句

express 中的路由

请求方法 get post 请求的url 响应的方法回调函数

路由器

按照模块，

创建路由器文件：

var router=express.Router()

router.get(‘/list’)

内置中间件

app.use(express.static(‘.public’));

第三方中间件

body-parser 对post请求的数据进行获取

#### mysql模块

mysql.exe –h127.0.0.1 –P3306 –uroot –p

mysql –uroot

mysql –uroot < 路径

SHOW DATABASES

USE xz

SHOW TABLES

DESC xz\_user

SET NAMES utf8;

DROP DATABASE IF EXISTS xz;

CREATE DATABASES xz CHARSET=utf8;

USE xz;

CREATE TABLE xz\_user(

uid int primary key;

uname varchar(16);

)

INSERT INTO xz\_user VALUERS(1,’Tom’),(2,’Tony’);

SELECT \*FROM xz\_user;

UPDATE xz-user SET uname=’zhangsan’ WHERE uid=1;

DELETE FROM xz-user WHERE uid=1;

1. 普通连接

createConnection（{ }）

connect( )

query()

（2）连接池连接（较高效）

createPool({

connectionLimit:20 //设置连接池的数量20；

});

练习：

学子商城目录结构

node\_modules………………项目依赖包

public……………………………项目中静态资源文件（css,js,html，图片）

user\_register.html…………用户注册html

routers………………………..项目中的路由器

user.js………………………….用户模块路由器

app.js……………………………项目入口文件

xz.sql…………………………….项目数据库文件

pool.js……………………………连接mysql数据库文件

### npm包管理命令

npm init –y 初始化package.json,用于记录包信息

npm install 包名称 --save 把包以运行依赖安装

npm install 包名称 --save-dev 把包以开发依赖安装

npm install 自动安装package.json 文件中所保存的包信息

### 项目开发流程

1. 安装包
2. 创建web服务器
3. 静态托管资源
4. 创建、导出路由器
5. 导入路由器，挂载到url下
6. 导入body-parser,配置中间件
7. 创建mysql连接池

分页查询sql语句

SELECT \* FROM xz\_user limit start,count;

start=(页码-1)\*count