复习：

Node.js中的数据类型

(1)基本值类型

number、boolean、string、null、undefined

(2)引用/对象类型

1)ES原生类型

2)Node.js提供的类型

3)用户自定义的类型

Node.js中模块(Module)系统

(1)Node.js官方提供的模块

(2)第三方扩展模块——npm

(3)自定义模块：

文件模块

circle.js：exports.size = fn;

app.js： require('./circle');

目录模块

node\_modules/mysql/package.json: { "main":"./index.js" }

node\_modules/mysql/index.js: exports.conn = fn

app.js: require('mysql')

Node.js官方提供的模块：

(1)global

global.console、global.parseInt、global.setInterval

(2)util

const util = require('util')

var str = util.inspect(obj)

(3)querystring

const qs = require('querystring')

var obj = qs.parse('uname=tom&age=20')

(4)url

const url = require('url');

var obj = url.parse('http://tmooc.cn:80/st/index.html?k=v#c1', true)

(5)Buffer

缓冲区就是一块内存，用于暂存数据

var buf = Buffer.alloc(1024);

var buf = Buffer.from('abcde');

(6)fs

const fs = require('fs')

//同步IO操作

var buf = fs.readFileSync('1.html')

fs.writeFileSync('2.log', buf)

fs.appendFileSync('3.log', buf)

//异步IO操作

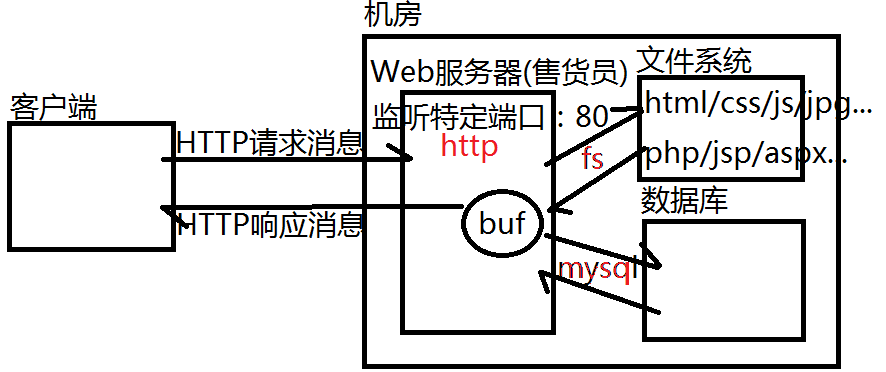
fs.readFile('1.html', (err, buf)=>{...})

fs.writeFile('2.log', buf, (err)=>{})

fs.appendFile('3.log', buf, (err)=>{})

console.log('JS执行完成')

(7)http



const http = require('http')

var server = http.createServer()

server.listen(80)

server.on('request', (req, res)=>{

//读取req中的请求信息

//输出响应消息

})

练习：静态Web服务器：创建Web服务器，接收客户端请求，一律向客户端输出一句<h1>Hello</h1>

练习：动态Web服务器：创建Web服务器，接收客户端请求，若客户端请求地址是/register，则向客户端输出public/register.html中的内容：

<form action="register.do ">

<input name="uname">

<input name="upwd">

<input type="submit">

</form>

若客户端请求的URL是/register.do，则解析URL中的查询字符串，把客户端提交uname和upwd追加写出到public/user.log的文件中，并向客户端输出“注册成功”

1.MySQL中的SQL

SQL语句的分类：

**DDL:(Data Define Language) —— 定义数据列**

CREATE、DROP、ALTER、TRUNCATE

**DML:(Data Manipulate Language) —— 操作数据行**

INSERT、DELETE、UPDATE

**DQL:(Data Query Lanaguage) —— 数据查询**

SELECT(最难)

**DCL:(Data Control Language) —— 控制用户权限**

GRANT、REVOKE

2.使用Node.js访问MySQL服务器

为了精简Node.js解释器，官方没有提供访问任何数据库相关模块，必须使用npm工具下载第三方模块。

在www.npmjs.org上搜索关键字mysql，可以得到很多相关的模块，每个模块都有使用说明。

使用npm工具下载mysql模块：

**npm i mysql**

mysql模块的使用步骤：

(1)创建到数据库服务器的连接

const mysql = require('mysql');

var conn = mysql.createConnection({...})

(2)发送SQL语句给数据库服务器来执行

conn.query('SQL...', function(err, result){....})

(3)关闭连接

conn.end()

练习：仿写上述代码，使用Node.js，向jd/emp表添加一行新的记录

练习：仿写上述代码，使用Node.js，将jd/emp表中编号为1的员工ename修改为汤米,工资修改为6000，入职时间修改为1331234567890 —— 注意？占位符的使用

课后练习： 使用http 和 mysql模块

创建动态Web服务器，接收如下的请求URL：

/register： 服务器向客户端返回public/register.html内容。

/register.do： 接收客户端提交的uname和upwd，保存入MySQL数据库，返回“注册成功”。

/login：服务器向客户端返回public/login.html内容。

/login.do：接收客户端提交的uname和upwd，查询数据库，返回“登录成功”或“用户名或密码错误”。

/userlist：服务器查询出数据库中所有的用户信息，在一个TABLE像客户端输出。