目录

[后台-SQL-PHP-AJAX 2](#_Toc491178119)

[ SQL部分 2](#_Toc491178120)

[1、 SQL语句： 2](#_Toc491178121)

[2、 mysql生成的null bug 解决 4](#_Toc491178122)

[3、 SQL-函数 4](#_Toc491178123)

[4、 解决mysql乱码（编码不统一） 4](#_Toc491178124)

[5、 多表查询 5](#_Toc491178125)

[ PHP部分 5](#_Toc491178126)

[6、 唯一打开、运行php的方式：打开浏览器在地址栏，http://127.0.0.1/01.php +回车 5](#_Toc491178127)

[7、 for循环-九九（99）乘法表 6](#_Toc491178128)

[8、 数组array-输出查看和遍历（for和foreach循环遍历） 6](#_Toc491178129)

[9、 使用php连接mysql服务器步骤：（表单添加留言到数据库） 7](#_Toc491178130)

[10、 主键自增长（数据库自动生成id且不重复） 8](#_Toc491178131)

[11、 php与mysql查询 8](#_Toc491178132)

[12、 php与mysql的删除（php生成表格，验证用户名是否存在） 9](#_Toc491178133)

[13、 模糊查找、查找列的个数并专程json的page、多表查询、排序、分页查询 10](#_Toc491178134)

[14、 php常用函数： 11](#_Toc491178135)

[15、 参数传递给php处理 12](#_Toc491178136)

[16、 创建和引用常用公共php代码： 12](#_Toc491178137)

[ HTTP协议部分 12](#_Toc491178138)

[17、 HTTP协议 12](#_Toc491178139)

[18、 请求消息 13](#_Toc491178140)

[19、 响应消息 14](#_Toc491178141)

[20、 面试题：GET和POST区别 15](#_Toc491178142)

[1) AJAX部分 16](#_Toc491178143)

[21、 异步和同步:网络工作方法 16](#_Toc491178144)

[22、 AJAX常用对象属性方法事件 17](#_Toc491178145)

[23、 AJAX处理不同类型数据{text/js/xml/html} 18](#_Toc491178146)

[24、 JSON（die()函数语句的处理） 20](#_Toc491178147)

[25、 AJAX处理JSON数据： 21](#_Toc491178148)

[26、 jQuery中的AJAX 重点&难点 21](#_Toc491178149)

[27、 项目中的几个小功能：加载页头页尾，cookie，跨域 22](#_Toc491178150)

[28、 上传文件到服务器： 25](#_Toc491178151)

[ node.js 26](#_Toc491178152)

[29、 计算程序运行时间：循环计时 27](#_Toc491178153)

[30、 模块-文件模块+目录模块（Module） 27](#_Toc491178154)

[31、 querystring 将查询字符串转换js对象 29](#_Toc491178155)

[32、 url模块（地址字符查询-）将请求url地址转js对象 30](#_Toc491178156)

[33、 Buffer模块 30](#_Toc491178157)

[34、 fs模块-同步和异步读取/写、追加文件 31](#_Toc491178158)

[35、 http-模块（服务器模块）：详情day03(05\_http例子) 32](#_Toc491178159)

[36、 node.js第三方模块：（MySQL）-npm工具下载 32](#_Toc491178160)

[37、 连接池（提高效率）node.js连接mysql 32](#_Toc491178161)

[38、 express模块 33](#_Toc491178162)

# 后台-SQL-PHP-AJAX

## SQL部分

**下载安装数据库软件**

https://www.mysql.com/downloads/

http://mirrors.sohu.com/ 搜狐镜像

xampp软件工具包(mysql/apache/php)

进入数据库：(XAMPP for Windows)mysql -uroot -p

### SQL语句：(SHOW DATABASE/TABLES)

* 1. **#与创建库相关指令**
     1. SHOW DATABASES; 查看当前MYSQL中所库名
     2. USE 库名; 进入到指定库中
  2. **DDL CREATE/DROP （**数据定义语句**）**

CREATE DATABASE 库名 CHARSET=UTF8;

CREATE TABLE 表名(列名 类型)

DROP DATABASE 库名;

DROP TABLE 表名;

* 1. **DML INSERT/UPDATE/DELETE（数据操作语句）**

INSERT INTO 表名 VALUES(新值,'','')

UPDATE 表名 SET 列名=值 WHERE 条件

**如：同时更新多行记录**

UPDATE car SET tid=30 WHERE cid =2 or cid=4 or cid=5 or cid=8;

UPDATE car SET tid=30 WHERE cid in (2,4,5,8);

DELETE FROM 表名 WHERE 条件；

**注：删除为id为NULL的列：查找隔壁列内容为空的列，WHERE msg=’’;**

* 1. **DQL （**数据查询语句**）**

SELECT 列名,列名.. FROM 表名 WHERE 条件

**查询表类型：desc 表名；） （列的格式）**

**模糊查找:** SELECT 列名,列名.. FROM 表名 WHERE 列名 LIKE ‘%关键字%’;

$sql="SELECT bname FROM dd\_book WHERE bname LIKE '%$kw%'";

**查找列的个数并赋值到变量：**

$sql ="SELECT count(\*) as page FROM jd\_product";

**多表查询：**

$sql = "SELECT c.cid,p.pname,p.pic,p.price,c.count FROM

jd\_cart c,jd\_product p WHERE c.pid=p.pid AND uid=$uid";

**（排序）ORDER BY 列名;**

SELECT \* FROM stu ORDER BY ctime; #升序

SELECT \* FROM stu ORDER BY ctime DESC; #降序

**查询指定行与分页：**

$offset=($pageno-1)\*8;

//5:创建SQL语句并且发送SQL

$sql ="SELECT \* FROM jd\_product";

$sql .=" LIMIT $offset,8";

//页码数pageno减1，乘以每页显示产品的个数8等于该页码数开始的产品，即从查找（抽取）($pageno-1)开始，的后八个内容。

* 1. **DCL 数据控制语句**

GRANT/REVOKE

* **列类型**

1. int 整型 表示范围 -21亿~21亿
2. varchar(n) 字符串 n个字符(汉字,字母,数字)
3. datetime 日期时间 年月日时分秒
4. double(m,n) 浮点(小数) 一共m位其中包括n位小数
5. decimal(m,n) 高精度浮点数 同上，货币和薪资等用

**例子：**

DROP DATABASE jd；

CREATE DATABASE jd CHARSET=UTF8;

USE jd;

CREATE TABLE jd (

pid INT,

pname VARCHAR(20),

pic VARCHAR(100),

price DOUBLE(10,2)

);

#SET NAMES GBK;设置当前操作mysql编码

SET NAMES GBK;

INSERT INTO jd VALUES(1,'苹果8','scr=img/1.jpg',6888);

INSERT INTO jd VALUES(2,'三星盖乐世8','scr=img/2.jpg',5888);

INSERT INTO jd VALUES(3,'华为MATE8','scr=img/3.jpg',4888);

INSERT INTO jd VALUES(4,'小米note8','scr=img/4.jpg',3888);

INSERT INTO jd VALUES(5,'魅族X8','scr=img/5.jpg',2888);

INSERT INTO jd VALUES(6,'OPPO por8','scr=img/6.jpg',1888);

DELETE FROM jd WHERE pid=5;

UPDATE FROM jd SET price=price\*0.9;

SELECT price FROM jd WHERE price>300;

#查询所有记录 条件奖学金大于等于3000并且

SELECT \* FROM stu WHERE bu >=3000 AND bu <=4000;

### mysql生成的null bug 解决

#NOT NULL DEFAULT 是为了解决mysql BUG

#mysql 可以会将错误数据(null)添加到表中

CREATE TABLE jd\_userinfo(

rid INT PRIMARY KEY AUTO\_INCREMENT,

homepage VARCHAR(100) NOT NULL DEFAULT '',

email VARCHAR(50) NOT NULL DEFAULT '',

age INT NOT NULL DEFAULT 0,

rdate DATETIME NOT NULL DEFAULT 0,

uid INT NOT NULL DEFAULT 0

);

### SQL-函数

**函数**

now() 数据库当前时间

max(列) 返回指定列中最大值

min(列) 返回指定列中最小值

sum(列) 返回指定列中总和

count(列) 返回指定列中数据个数

avg(列) 返回指定列中平均数

### 解决mysql乱码（编码不统一）

数据库之所以出现乱码问题:编码不统一

utf-8 优点:支持全世界所有国家文字

**解决方案一:(windows 100%)**

1:创建库指定utf-8

**CREATE DATABASE 库名 CHARSET=UTF8;**

2:如果有中文内容前加一句

**SET NAMES GBK;**

**解决方案二:(linux;韩文)**

1:所有创建表创建库添加记录语句保存文件中

xx.sql

#文件一定要是utf-8

2:不能复制内容

3:打开mysql窗口进入mysql系统

mysql -uroot -p

4:专用指定，导入sql文件

SET NAMES UTF8;

source d:/x.sql

### 多表查询

#### 轻松多表查询

a:将所查询的表名放在FROM后用逗号分隔

FROM emp,dept

b:为每一张表起一个别名

FROM emp e,dept d

c:找一个两张表(某列)有关系 = > <

WHERE e.did = d.didP

**SELECT e.ename,d.dname**

**FROM emp e,dept d**

**WHERE e.did = d.did;**

* PHP部分

php官网提供二套函数库

1:普通版本 mysql\_xxx()

2:增强版本 mysqli\_xxx();

查看php版本：在php文件中添加语句：phpinfo();然后查看php文件

### 唯一打开、运行php的方式：打开浏览器在地址栏，<http://127.0.0.1/01.php> +回车

### for循环-九九（99）乘法表

<?php

//第四题:输出9\*9

for($i=1;$i<=9;$i++){

for($j=1;$j<=$i;$j++){

echo $j."\*".$i."=".($j\*$i)."&nbsp;";

}

echo "<br />";

}

?>

### 数组array-输出查看和遍历（for和foreach循环遍历）

* 1. **创建和输出查看数组：**

**<?php**

$arr2 = [10,20,30];

//echo $arr2;//数组不能直接输出

var\_dump($arr2); // array(2) { ["eid"]=> int(102) ["ename"]=> string(5) "james" } 10

//1:创建一个数组:数组中一共有二个元素,下标 "bid" 图书编号 值 1001,下标 "bname" 图书名称 值 "js大全",//2:遍历数组中每个元素

$arr4 = ["bid"=>1001,"bname"=>"js大全"];

foreach($arr4 as $k=>$v){

echo $v;

}

var\_dump($arr4); // 1001js大全array(2) { ["bid"]=> int(1001) ["bname"]=> string(8) "js大全" }

?>

**2）for循环遍历数组**

$arr2 = [10,20,30];

//php $arr没有.length属性 count($arr)

for($i=0;$i<count($arr2);$i++){

echo $arr2[$i]."<br />";

} //10 20 30

**3）foreach遍历数组**

<?php

$list = [];

$list[] = ["bid"=>101,"title"=>"西游记"];

$list[] = ["bid"=>102,"title"=>"西游记后传"];

$list[] = ["bid"=>103,"title"=>"西游记外传"];

//练习:遍历出数组的每个元素

//bid title $k位下标，$v为值

foreach($list as $k=>$v){

echo $v["bid"];

echo $v["title"];

echo "<br />";

} // 102西游记后传 103西游记外传

?>

### 查看php数组var\_dump($arr4);

### 使用php连接mysql服务器步骤：（表单添加留言到数据库）

(1)创建php到mysql数据库连接

$conn = mysqli\_connect(...);

(2)创建sql语句并且发送SQL语句

$sql = "INSERT....";

$result = mysqli\_query($conn,$sql);

(3)读取mysql服务器返回结果

if($result===false)

(4)关闭连接---可以省略

mysqli\_close($conn);

**例子：**

//**HTML代码：**

<body>

<form action="addmsg.php" method="post">

留言编号 <input type="text" name="id"/>

留言内容 <input type="text" name="content"/>

<input type="submit" />

</form>

</body>

**PHP代码：**

<?php

//0:接收参数 id content

**//php标准语法专门接收参数（从表单获取参数）**

**$id = $\_REQUEST["id"]; $c = $\_REQUEST["content"];**

//1:创建连接

$conn = mysqli\_connect("127.0.0.1","root","","oa",3306);

//修改编码

mysqli\_query($conn,"SET NAMES UTF8");

//2:创建SQL并且发送sql

$sql = "INSERT INTO msg VALUES($id,'$c')";

$result = mysqli\_query($conn,$sql);

//3:判断结果

if($result===true){ echo "添加成功";}else{ echo "添加失败"; }

?>

8、解决二个问题

stu

10 tom1 25 3100 null

11 tom2 24 3000 null

UPDATE stu SET ctime=now();

INSERT INTO stu VALUES(1,'TOM3',26,1000,now());

### 主键自增长（数据库自动生成id且不重复）

**#主键：uid列不能为空不能重复；**

**#自增长：uid列 1 2 3 4 null**

CREATE TABLE user(

**uid INT PRIMARY KEY AUTO\_INCREMENT, //自动生成id**

age INT,

name VARCHAR(10)

);

INSERT INTO user VALUES(null,18,'TOM');

### php与mysql查询

**抓取结果集数据，并且转成php数组 （及判断是否能查找到数据）**

抓一行:

$row = mysqli\_fetch\_row($result); //索引数组

$row =mysqli\_fetch\_assoc($result); //**关联数组 $row==NULL,没有数据；**

抓取多行:

$rows = mysqli\_fetch\_all($result,MYSQLI\_ASSOC);//**关联 count($rows)===0 没数据**

<?php

$u=$\_REQUEST["uname"];

$p=$\_REQUEST["upwd"];

$conn=mysqli\_connect("127.0.0.1","root","","bbs",3306);

mysqli\_query($conn,"SET NAMES UTF8");

$sql="SELECT\*FROM user WHERE uname='$u' AND upwd='$p'";

$result=mysqli\_query($conn,$sql);

$row=mysqli\_fetch\_assoc($result);

if($row===NULL){

echo"登录失败";

}else{echo"登录成功";}

?>

### php与mysql的删除（php生成表格，验证用户名是否存在）

**例子1：**

<?php

**//如果用户提交的内容（编号）为空，加die（）函数则提示"用户名编号是必须的"**

@$i=$\_REQUEST["uid"] or die("用户名编号是必须的");

//如果不加@函数则会出错会有英文显示：  
//Notice: Undefined index: uid in C:\xampp\htdocs\day03\user\_del.php on line 2

//"用户名编号是必须的"

$conn=mysqli\_connect("127.0.0.1","root","","bbs",3306);

mysqli\_query($conn,"SET NAMES UTF8");

$sql="DELETE FROM user WHERE uid=$i";

$result=mysqli\_query($conn,$sql);

if($result===false){

echo"删除失败";

}else{echo"删除成功";}

?>

**例子2：**

<?php

//功能:查询bbs.user 所有记录

//user[uid/uname/upwd/pic/ctime]

//1:创建连接

$conn = mysqli\_connect("127.0.0.1","root","","bbs",3306);

mysqli\_query($conn,"SET NAMES UTF8");

//2:创建sql语句并且发送sql

$sql = "SELECT \* FROM user";

$result = mysqli\_query($conn,$sql);

//3:获取返回结果

//4:抓取多行记录

$rows = mysqli\_fetch\_all($result,MYSQLI\_ASSOC);

//5:把数据以表格式输出

echo "<table border='1' width='80%'>";

echo "<thead>";

echo "<tr><td>编号</td><td>用户名</td><td>操作</td></tr>";

echo "</thead>";//14:20--14:25

echo "<tbody id='tb1'>";

foreach($rows as $k=>$v){

echo "<tr>";

echo "<td>$v[uid]</td>";

echo "<td>$v[uname]</td>";

echo "<td><a href='$v[uid]' class='btn-del'>删除</a></td>";

echo "</tr>";

}

echo "</tbody>";

echo "</table>";

?>

<script>

var tb1 = document.getElementById("tb1");

//1:获取所有删除按钮 class==btn-del

//2:绑定点击事件 15:55-16:05

//3:阻止事件默认行为

console.log(1);

tb1.onclick = function(e){

console.log(2);

e.preventDefault();

var target = e.target;

console.log(3);

if(target.className=="btn-del"){

//4:获取当前用户编号

var uid = target.getAttribute("href");

//5:创建确认框 window.confirm();

var rs = window.confirm("您确认要删除此记录吗?");

//6:如果用户点击 "是" 16:40--16:45

if(rs===true){

//7:自动跳转

location.href="user\_del.php?uid="+uid;

}

}

}

</script>

### 模糊查找、查找列的个数并专程json的page、多表查询、排序、分页查询

**模糊查找:** SELECT 列名,列名.. FROM 表名 WHERE 列名 LIKE ‘%关键字%’;

$sql="SELECT bname FROM dd\_book WHERE bname LIKE '%$kw%'";

**查找列的个数并赋值到变量：**

$sql ="SELECT count(\*) as page FROM jd\_product";

**多表查询：**

$sql = "SELECT c.cid,p.pname,p.pic,p.price,c.count FROM

jd\_cart c,jd\_product p WHERE c.pid=p.pid AND uid=$uid";

**（排序）ORDER BY 列名;**

SELECT \* FROM stu ORDER BY ctime; #升序

SELECT \* FROM stu ORDER BY ctime DESC; #降序

**查询与分页：**

$offset=($pageno-1)\*8;

//5:创建SQL语句并且发送SQL

$sql ="SELECT \* FROM jd\_product";

$sql .=" LIMIT $offset,8";

//页码数pageno减1，乘以每页显示产品的个数8等于该页码数开始的产品，即从查找（抽取）($pageno-1)开始，的后八个内容。

### php常用函数：

* + die($str) :终止当前php程序执行，并向客户输出一个字

符串.

$i=$\_REQUEST["uid"] or **die("用户名编号是必须的");**

* + @: 禁止当前行代码警告信息

**@**$i=$\_REQUEST["uid"] or die("用户名编号是必须的");

* + $id = mysqli\_insert\_id($conn); 返回连接上刚刚执行insert

语句产生自增编号.

* + ysqli\_affected\_rows($conn);返回刚刚执行增删改

//2:读取所有请求消息头

* + $lists = getallheaders();

//var\_dump($lists);

//3:获取首选语言

* + $lang = $lists["Accept-Language"];

//4:截取字符串 $str = substr($lang,0,2);

* + $str = substr($lang,0,2);//en zh

**例子:**

header("Content-Type:application/javascript;charset=utf-8");

//2:读取所有请求消息头 11:01--11:06

$lists = getallheaders();

//var\_dump($lists);

//3:获取首选语言

$lang = $lists["Accept-Language"];

//4:截取字符串 $str = substr($lang,0,2);

$str = substr($lang,0,2);//en zh

//5:判断 英文 中文 休息一会

if($str==="en"){

echo "alert('hello')";

}else if($str =='zh'){

echo "alert('你好')";

}else{

echo "alert('hello')";

}

### 参数传递给php处理

1. **表单：**
   1. Php(msg.php): $u=$\_REQUEST[“uname”]
   2. Html:<form action=”msg.php”>

<input type=”text” name=”uname”>

<form/>

1. **地址栏直接输入：**//php测试表单用代码：127.0.0.1/x.php?uname=tom1&upwd=123
2. **Js自动跳转**
   1. PHP: $uname=$\_REQUEST[]
   2. JS: location.href=”x.php?uname=tom”+$uid

### 创建和引用常用公共php代码：

require()函数：加载指定的php，功能等于复制

1. 创建init.php文件
2. 引入require(“init.php”)文件

## HTTP协议部分

### HTTP协议

**1、URL**

互联网上任何一个资源必须有一个"URL"才能被找到

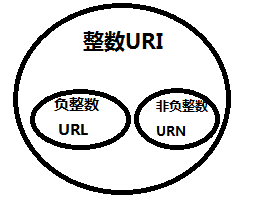
<schema>://<user>:<pwd>@<host>:<port>/<path>:<params>?<query>#<frag>

**URL/URN/URI 分清概念**

URL: 统一资源定位符

URN: 统一资源命名符

URI: 统一资源标识符



**2、URI=URL+URN**

URL: <a href="login.html"></a>

<img src="http://baidu.com/logo.png">

URN:<a href="mailto:admin@163.com">

**3、a: <schema> 方案:指定使用哪种协议从服务器上获取资源/**

方案名不区分大小写

常见方案:http;https;ftp;file;telnet;.....

http://www.icbc.com.cn/icbc/

https://member.icbc.com.cn/commlog/logon.jsp

**4、b:<host>主机名:资源所在服务器IP地址或者域名**

**5、c:PORT:端口号 每项软件服务器上都对应一个端口号**

ftp 21 文件上传下载

ssh 22 安全远程登录

telnet 23 远程登录

dns 53 域名解析 baidu.com-->61.135.169.125

http 80 超文本传输协议

https 443 加载超文本传输

smtp 25 邮件传输

### 请求消息

1. **请求起始行（的方法）**
   * + GET 表示客户端获得服务器上资源
     + POST 表示客户端想传递数据给服务器==》请求主体
     + PUT 表示客户端想放置一个文件到服务器上==》请求主体
     + DELETE 表示客户端想删除服务器上资源
     + HEAD 表示客户端想获取服务器上资源，但只要响应头信息不要内容
     + CONNECT 连接测试
     + TRACE 追踪请求
     + OPTIONS 选项,保留以后使用
     + 空格
     + 请求URL
     + 空格
     + 协议版本

如：GET /web/index\_new.html?tedu HTTP/1.1

1. **请求头**
   * + User\_Agent: 客户端告诉服务器自己类型
     + Accept-Language:zh-CN 告诉服务器浏览器可以接收自然语句
     + Accept-Encoding:gzip 浏览器在告诉服务器接收响应压缩类型
     + DATE: 请求消息创建时间
     + Accept:text/html; 告诉服务器自己可以接收响应类型
     + Connection:keep-alive 告诉对方启用持久化连接
     + Cache-Control:no-cache 告诉对方如何缓存当前消息主体中的内容==》缓存:目标提交效率
     + Pragma:no-cache HTTP/1.0 控制缓存
     + 请求主体描述头: 主体内容长度 Content-Length:36 字节
     + 请求主体类型描述头:
       - text/plain
       - application/x-www-form-urlencoded 请求主体内容经过编码
       - multipart/form-data 如果主体内容包含上传文件

**如：**User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.102 Safari/537.36

Accept-Language: zh-CN,zh;q=0.8

Accept-Encoding: gzip, deflate, sdch

Accept:text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8

Connection: keep-alive2

Cache-Control: max-age=0

**2、请求主体：POST和PUT 注：GET没有请求主体**

### 响应消息

* 响应起始行

1. 协议版本
2. 空格
3. 响应状态码

100-199 1XX 提示性信息

* + - 100 Continue
    - 101 Swtiching Protocols

2XX 成功的响应

* + - 200 OK
    - 201 Create

3XX 客户端重定向

* + - 301 Moved Permanently
    - 304 Not Modifed(请使用缓存)

4XX 客户端请求错误

* + - 403 Forbiddren 请求资源权限不足
    - 404 Not Found
    - 405 Method Not Allowed

5XX 服务器运行错误

* + - 500 Internal Server Error 服务器内部出错

1. 空格
2. 原因短句ok

* 响应头

1. Date 响应消息创时间
2. Server tarena 服务器介绍
3. Connection: keep-alive 告诉对方启用持久化连接
4. Last-Modified 最后一次修改时间
5. Cache-Control 告诉对方启动缓存
6. Pragma: 告诉对方启动缓存
7. Content-Length:3796 响应主体几个字节
8. Content-Type:

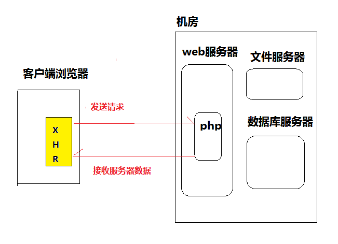
响应什么格式

* + 1. text/plain 纯文本
    2. text/hmlt 文本与网页
    3. text/css
    4. application/javascript
    5. application/xml
    6. application/json
    7. image/jpeg
    8. audio/mpeg3

### 面试题：GET和POST区别

|  |  |  |
| --- | --- | --- |
|  | GET | POST |
| 语义 | 客户端想获取指定资源 | 客户端想传递数据给服务器 |
| 如何发起 | 浏览器地址输入回车  标签 href src  超链接跳转  JS跳转  表单GET提交  AJAX-GET | 表单POST  AJAX POST |
| 安全级别 | 不安全 | 不安全 |
| 数据长度 | 1KB(汉字几个字) | 没有限制 |
| 是否需要编码 | 需要 | 不需要 |
|  |  |  |

## AJAX部分



### 异步和同步:网络工作方法

**同步(表单提交/地址栏/js):电话**

客户端浏览器 服务器程序

1: 发送数据 等待

2: 等待 接收客户端数据

并且处理(delete/insert/update)

3: 接收返回数据 等待

**异步:(AJAX)微信**

1: 发送数据 为其它程序处理数据

2: 输入数据 接收客户端数据

并且处理(delete/insert/update)

3: 接收返回数据 为其它程序处理数据

### AJAX常用对象属性方法事件

* + **对象:AJAX对象:最核心对象**

作用:向服务器程序发送请求，并且接收返回数据

XMLHttpRequest

IE 6 7 8 ActiveXObject("Microsoft.XMLHTTP");

IE 9 10 11 Ch FF Sa... XMLHttpRequest

**兼容IE678代码：**

var xhr = null;

if(window.XMLHttpRequst){

xhr = new XMLHttpRequest();

}else{

xhr = ActiveXObject("

* + **属性**

readyState :表示xhr对象当前状态:不能手工赋值,但是可以获取,其值会随着请求-响应过程不同自动改变! 如：if(xhr.readState===4){}

0 1 2 3 4

0:UNSENT 请求消息未发送

1:OPENED xhr打开连接

2:HEADERS\_RECEVICED xhr 己经收到服务器响应头部数据

3:LOADING xhr正在加载响应主体数据

4:DONE xhr 接收完成响应主体数据

status 服务器响应状态码 200 数据正确返回

404 请求数据不正确

**responseText** 响应文本[服务器程序输出字符（echo）"成功"]

**responseXML** 响应XML数据

* + **方法**

open(method,url,isAsyn) 打开连接

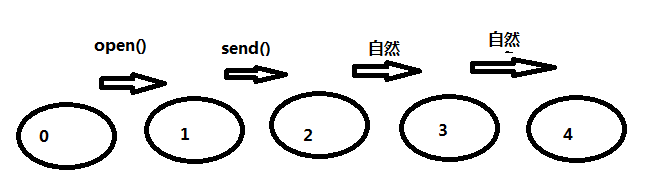
* + - method:请求方法 GET POST DELETE PUT
    - url: 请求程序地址
    - isAysn 是否是异步请求 true/ false 同步

send(body); 把请求消息发送服务器程序

事件

onreadystatechange:xhr.readyState值每次改变自动触

发事件.



* + ajax用到的js函数：
  1. eval() : 括号里面放一段js代码，并解析js代码

xhr.onreadystatechange=function(){

if(xhr.readyState==4){ //两个判断

if(xhr.status==200){ //两个判断

eval(xhr.responseText);}}};

* 1. header("Content-Type:application/javascript;charset=utf-8");
  2. $lists=getallheaders(); 读取所有请求消息头；
  3. $lang = $lists["Accept-Language"]; 获取首选语言
  4. $str = substr($lang,0,2);//en zh 截取字符串

**例子：**

<script>

//1:获取用户名绑定事件 onblur

uname.onblur = function(){

//2:获取当前用户名

var u = this.value;

//3:创建AJAX对象

var xhr = new XMLHttpRequest();

console.log(3);

//4:绑定事件

xhr.onreadystatechange = function(){

//4.1:接收返回数据

if(xhr.readyState===4){

unameTip.innerHTML = xhr.responseText;

}

}

//5:打开连接

xhr.open("GET",

"check\_uname.php?uname="+u,

true);

//6:发送请求

xhr.send(null);

}

</script>

### AJAX处理不同类型数据{text/js/xml/html}

**a:txt 纯文本[标准写法]**

php:

#header() 修改响应头中数据格式

header("Content-Type:text/plain;charset=utf-8");

echo "hello";

ajax

xhr.responseText;

b:js

php:

header("Content-Type:application/javascript;charset=utf-8");

echo "var msg='hello';alert(msg);";

ajax:

eval(xhr.responseText);

c:html

php:

header("Content-Type:text/html;charset=utf-8");

echo "<li>tom</li><li>tom2</li><li>tom3</li>";

ajax:

xhr.responseText;

d:xml

**XML语法:**

HTML:超文本标记语言，所有标签预定义好的用于

描述一个网页

**XML**:可扩展的标签语言，所有标签都是**自定义**的，

用于描述一段数据

**XML语法要求**

1:XML文档声明如下

<?xml version="1.0" encoding="utf-8"?>

2:整篇文档有且只有一个根元素

3:标签有开始必须有结束，开始结束必须一致

4:标签可嵌套不能交叉

5:标签可以有任意属性，属性值必须用""双引号括起来

php:

header("Content-Type:application/xml;charset=utf-8");

echo "<books><book>js大全</book></books>";

ajax:

xhr.responseXML;

e:json

php:

header("Content-Type:application/json;charset=utf-8");

echo json\_encode($row);

ajax:

xhr.responseText;

|  |
| --- |
| 服务器端返回的数据是XML字符串格式  XML字符串=>JS对象  方法1:使用js提供DOMParse对字符串解析  方法2:使用xhr. responseXML |
| W3C DOM 标准分三种  1:核心 DOM 提供方法要以操作任间DOM树,如  querySelector(), getAttribute();  2:HTML DOM  3:XML DOM |

|  |
| --- |
| GET-->POST 修改三个位置  1:xhr.open("POST","x.php",true);  2:xhr.setRequestHeader("Content-Type","application/x-www-form-urlencoded");  3:xhr.send(`uname=tom&age=10`); |

e:json

php:

header("Content-Type:application/json;charset=utf-8");

echo json\_encode($row);

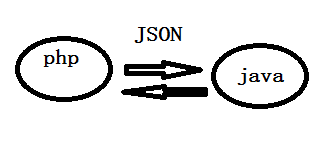
ajax:

xhr.responseText;

### JSON（die()函数语句的处理）

###### 工业标准[JSON字符串格式]

JSON[JavaScript Object Notation]是一种轻量级的数据交换格式(标准信封)，易于阅读和编写，语法其于JS语言，但目前己经被各种语言所有支持，成为一种"异构交互数据标准格式"

****

###### JSON字符串语法要求

(1)一个JSON字符串有且只且个根元素

可以是[ ]--表示数组

可以是{ }--表示对象

(2)JSON中可以表示数字;bool;null;字符串--必须双引号

(3)[]数组中可以包含多个值,使用逗号分隔

(4){}对象可以包含多个键和值,使用逗号分隔，键和值之间用冒号分隔，键必须双引号

###### 服务器端php中如何把数组转换json字符串

$str = json\_encode($rows);

echo $str;

//关联数组==>JSON

###### die()函数语句的处理

如果只有一个对象则用{“键”:“值”, “键”:“值”}

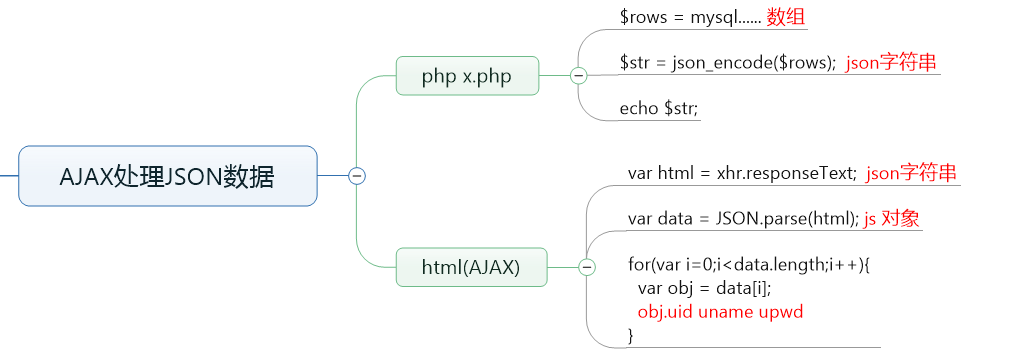
@$tid = $\_REQUEST["tid"]or die('{"code":-2,"msg":"编号是必须的"}');

###### 客户端JS如何把json字符串转换js对象

var obj = JSON.parse(str);

JSON-->JSON Obj

### AJAX处理JSON数据：



### jQuery中的AJAX 重点&难点

$("选择器").load(url);

1. **$.get();**
2. **$.post();**
3. **$.getJSON();**
4. **$.getScript();**

**详情查看:** **\正课\11-AJAX\day08\src\01.html**

1. **$.ajax()----最重要**

万能的ajax封装函数!提供非常多的**可选项**,可以处理各种情形。前面的五个函数都是**$.ajax() 简化版本.**

$.ajax({

url:"x.php", ///请求程序地址(必选项)

type:"GET", //请求方式POST/HEAD/DELETE/..

data:***{}*** //提交给服务器参数

beforeSend:fn //请求消息发送之前调用函数

success:fn(data)//响应完成并且成功调用函数

error:fn //响应完成并且失败调用函数

complete:fn //响应完成后调用函数(无认成败)

});

**例子：**  $.ajax({

url:"cate\_getall.php",

data:{tid:id2},

success:function(data){

var html = "";

for(var i=0;i<data.length;i++){

var obj = data[i];//17:00-17:10

html += `

<tr>

<td><img src="img/${obj.pic}" width="120" height="100"></td>

<td>${obj.cname}</td>

<td>${obj.price}</td>

</tr>`;

}

$("#tb1").html(html);

},

error:function(){

alert("网络出现故障，请检查");

}

});

### 项目中的几个小功能：加载页头页尾，cookie，跨域

* 1. **加载页头页尾**

（<header id="header"></header>）

$(“#header”).load(“header.html”); $(“select”).load(“footer.html”)

$("父元素选择器").load("x.php");

* 1. **cookie跨页面传递参数**

其它跨页传递方法：1、超链接 location.href="2.php?uid=3"**临时；2、“**c:sessionStorage

**Cookie传递参数标准语法**

1:将数据保存cookie

js: document.cookie = 'uname=tom'

document.cookie = 'age=19'

**例子：（登录页面）**

document.cookie = 'uname='+u; // u和loginUid为登录用户信息变量

document.cookie = 'uid='+loginUid

2:从cookie读取数据(操作繁琐)

js:document.cookie

**例子：（购物车页面）**

**//功能点十:从cookie中读取数据并且转obj**

var arr = document.cookie.split("; ");

var obj = {};

for(var i=0;i<arr.length;i++){

var pair = arr[i].split("=");

var key = pair[0];

var val = pair[1];

obj[key]=val;

}

$.ajax({

url:"data/cart.php",

data:{uid:obj.uid},

success:function(data){

var html = "";

$.each(data,function(idx,obj){…}

* 1. **跨域请求--->重点&难点(百度有多种方法)**

Cross Domain Request:(跨域请求)从一个域名下资源访问另一个域名下的资源。两个域名不同，称为跨域请求

www.facha.com/index.js -> www.fafa.com/index.php

跨域其它方式

:跨协议名

http://facha.com/index.js -> https://facha.com/index.php

:不同主机名和ip地址

http://127.0.0.1/index.js -> http://localhost/index.php

:端口不同

http://127.0.0.1:81/index.js -> http://127.0.0.1:82/index.php

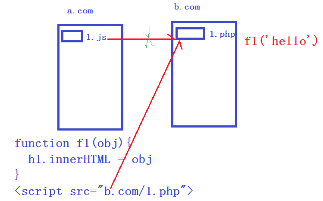
浏览器允许跨请求的情形

IMG/LINK/SCRIPT/IFRAME 等标签

浏览器允许跨请求的情形

浏览器默认禁止XHR跨域请求--跨域访问可能导致外来数据对前面页造成**安全**风险.

如何解决浏览器默认禁止XHR问题



解决方案:JSONP和JSON关系

JSON:是一种字符串数据格式，用于描述数据

JSONP: JSON with Padding 填充式JSON，是一种使用JSON数据的方法，用于解浏览器XHR跨域限制.

JSONP的思路

发起异步请求，不使用XHR使用一个动态创建SCRIPT标签代替换XHR <script async src="跨域地址/x.php"></script>

要求服务器端必须使用一段JS片段，包含服务器端数据

doResponse({"uname":"tom","age":20})

客户端特意提供一个名为 doResponse(data)

pm\_01\_jsonp.html

pm\_01\_jsonp.php

**例子：pm\_json.php部分：（绿色字体为自定义函数名f1替换doResponse）**

**<?php**

**header("Content-Type:application/jacascript;charset=utf-8");**

**//$cb=$\_REQUEST["callback"]; //doResponse改为f1时用（callback=f1）**

**$json='{"ename":"tom","tom":25}'; //相应格式必须是JavaScript**

**echo 'doResponse('.$json.')'; //输出数据,给script的doResponse函数调用**

**//echo $cb.'('.$json.')'; // f1({“ename”:”tom”,”tom”:25})**

**?>**

**例子1：js方法：**

**<script type="text/javascript">**

**bt1.onclick=function(){**

**var s=document.createElement("script");**

**s.async=true;**

**s.src="http://localhost/nodjs/day01/pm\_json.php";**

**//s.src="http://localhost/nodjs/day01/pm\_json2.php?callback=f1";**

**document.head.appendChild(s);**

**}**

**function doResponse(data){ //function f1(data){ //doResponse=f1**

**console.log("客户端开始处理响应数据");**

**console.log(data);**

**}**

**</script>**

**例子2：jq方法：**

**<script type="text/javascript">**

**bt1.onclick=function(){**

**$.getJSON("http://localhost/nodjs/day01/pm\_json3.php?callback=?",**

**function(data){**

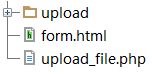
**console.log(data);**

**})**

**}**

**</script>**

### 上传文件到服务器：

* **原生html上传：**（服务器文件目录关系）

1、创建接收上传文件的文件夹（上传文件存储的位置）

2、创建form上传表单：

<**form action="upload\_file.php" method="post" enctype="multipart/form-data"**>  
 <**label for="file"**>文件名：</**label**>  
 <**input type="file" name="file" id="file"**><**br**>  
 <**input type="submit" name="submit" value="提交"**>  
</**form**>

1. **创建处理表单提交的文件的php处理文件**

<?php  
 // 允许上传的图片后缀  
 $allowedExts = array("gif", "jpeg", "jpg", "png");  
 $temp = explode(".", $\_FILES["file"]["name"]);  
 echo $\_FILES["file"]["size"];  
 $extension = end($temp); // 获取文件后缀名  
 if ((($\_FILES["file"]["type"] == "image/gif")  
 || ($\_FILES["file"]["type"] == "image/jpeg")  
 || ($\_FILES["file"]["type"] == "image/jpg")  
 || ($\_FILES["file"]["type"] == "image/pjpeg")  
 || ($\_FILES["file"]["type"] == "image/x-png")  
 || ($\_FILES["file"]["type"] == "image/png"))  
 && ($\_FILES["file"]["size"] < 204800000) // 小于 200 kb  
 && in\_array($extension, $allowedExts)){  
 if ($\_FILES["file"]["error"] > 0){  
 echo "错误：: " . $\_FILES["file"]["error"] . "<br>";  
 }  
 else{  
 echo "上传文件名: " . $\_FILES["file"]["name"] . "<br>";  
 echo "文件类型: " . $\_FILES["file"]["type"] . "<br>";  
 echo "文件大小: " . ($\_FILES["file"]["size"] / 1024) . " kB<br>";  
 echo "文件临时存储的位置: " . $\_FILES["file"]["tmp\_name"] . "<br>";  
 // 判断当期目录下的 upload 目录是否存在该文件  
 // 如果没有 upload 目录，你需要创建它，upload 目录权限为 777  
 if (file\_exists("upload/" . $\_FILES["file"]["name"])){  
 echo $\_FILES["file"]["name"] . " 文件已经存在。 ";  
 }  
 else{  
 // 如果 upload 目录不存在该文件则将文件上传到 upload 目录下  
 move\_uploaded\_file($\_FILES["file"]["tmp\_name"], "upload/" . $\_FILES["file"]["name"]);  
 echo "文件存储在: " . "upload/" . $\_FILES["file"]["name"];  
 }  
 }  
 }  
 else{  
 echo "非法的文件格式";  
 }  
?>

## node.js

静态网页和动态网页

**静态**:网页内容任何人在任何时间访问都不变的

HTML/CSS/JS/FLASH/音频/视频

**动态:** 网页内容任何人在不同的时间访问可能不同的

JSP/PHP/ASP/NODEJS

JSP =HTML+JAVA 功能强大可靠，适合大型企业级项目

PHP =HTML+PHP 简单易用，适合互联网项目

ASP.NET =HTML+C# 易用，windows平台

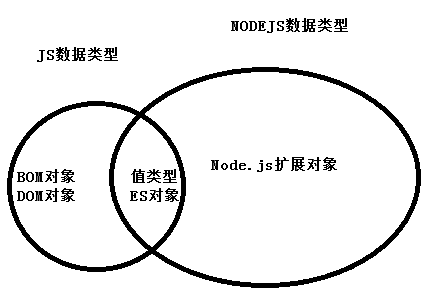
Node.JS =性能好，适合于IO密集型项目，不适合

CPU密集型项目

1亿=8小时\*3600=28800/10= 347/s

\*IO密集型项目:(input output)查询、删除、更新

\*CPU密集型项目:天气预报,滴滴打车

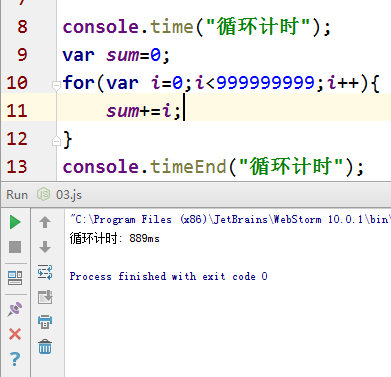


### 计算程序运行时间：循环计时

console.time(“循环计时”)； //开始计时

function; //…..运行的程序；

console.timeEnd(“循环计时”)； //结束计时，循环计时的内容要一致



### 模块-文件模块+目录模块（Module）

* Modal:模态框
* Model:模型
* Module:模块

Node.JS按照功能不同，可以把函数，对象分别放到不同文件、目录下、这些**文件/目录**NodeJS中就称为模块"Module"（如商品管理;用户管理模块;支付模块;）

Node.js中每个模块都是一个独立构造函数，解释器会为每个.js文件Node.js添加如下代码.

(function(exports,require,module,\_\_filename,\_\_dirname){

exports:{} 用于声明向外导出自己的成员

require:fn 用于导入其它的模块，并且创建指定模块对象

module: 当前模块对象

\_\_filename 文件绝对路径

\_\_dirname 文件所属目录绝对路径

//自己编写的文件内容

});

* **自定义文件模块：**

创建一个js文件,如m3.js 导出需要公共数据,其它模块可以require("./m3") 模块

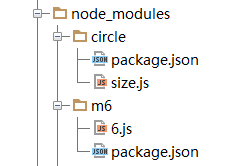
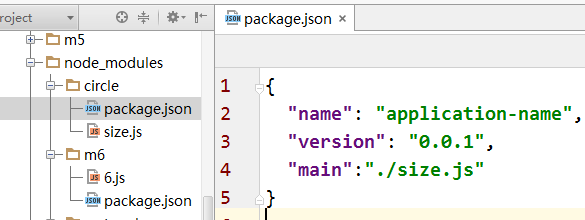
1.db.js:

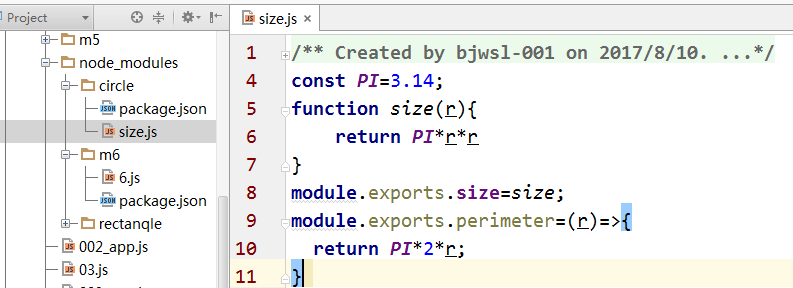
**var *uname***=**"james"**;  
**function** *run*(){  
 **console**.log(**"奔跑的"**+***uname***)  
}  
*//对象向外公开变量和函数*exports.**run**=*run*;

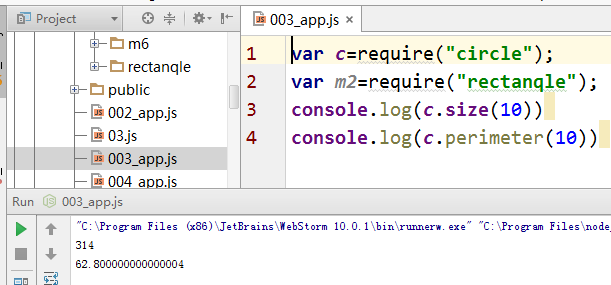
2.user

**var *m***=require(**"./db"**);  
***m***.**run**();  
*//console.log(m.a);*

* **自定义目录模块：**





### querystring 将查询字符串转换js对象

*//功能:处理查询字符串  
//qs.parse(str); 将查询字符串转换js对象  
//qs.stringify(obj);js对象转换查询字符串  
//加载其它模块:通常不需要修改只使用***const *qs*** = require(**"querystring"**);  
**var *str*** = **"uname=tom&upwd=123&age=19"**;  
*//把客户端传递给 node.js服务器查询字符串解析  
//容易使用JS对象***var *obj*** = ***qs***.parse(***str***);  
**console**.log(***obj***);  
  
*//把一个JS对象转抡为查询字符串格式***var *obj2*** = {**ename**:**"强东"**,**age**:20,**addr**:**"bj"**};  
***str*** = ***qs***.stringify(***obj2***);  
**console**.log(***str***);  
*//结果：ename=%E5%BC%BA%E4%B8%9C&age=20&addr=bj*

### url模块（地址字符查询-）将请求url地址转js对象

* 1. **const *url*** = *require*(**"url"**);  
     **var *u*** = **"** **http://127.0.0.1:8081/add.do?sname=dsaf&scope=98"**;  
     **var *obj*** = ***url***.parse(***u***); //转为一个对象  
     *//console.log(obj);  
     //true:指定对URL中的查询字符串进行进一步解析,  
     //解析查询字符串:*

Url {

protocol: null,

slashes: null,

auth: null,

host: null,

port: null,

hostname: null,

hash: null,

search: '?sname=dsaf&scope=98',

**query: { sname: 'dsaf', scope: '98' },**

pathname: '/add.do',

path: '/add.do?sname=dsaf&scope=98',

href: '/add.do?sname=dsaf&scope=98' }

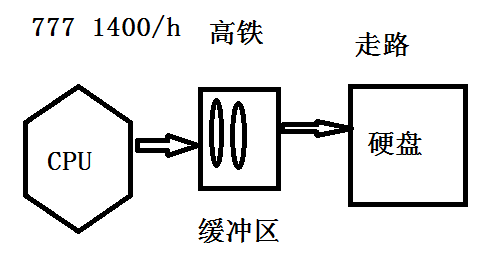
**var** path=obj.**pathname**; *//请求路径不带查询字符串*

**console.log(path) // 结果：**//add.do

**var *obj2*** = ***url***.parse(***u***,**true**);

***console***.log(***obj2***.**query**.**uname**); //结果：dsaf

### Buffer模块



分配一个指定大小的缓冲区

var buff1 = Buffer.alloc(1024)

**var *buf1***=Buffer.alloc(10);  
**console**.log(***buf1***);

使用一个数字数组创建一个缓冲区

var buff2 = Buffer.from([1,2,3]);

使用一个字符中创一个缓冲区

var buff3 =Buffer.form("abc");

把一个缓冲区中的数据转换字符串

var str = buffe3.toString();

### fs模块-同步和异步读取/写、追加文件

fs(file system)模块提供了对文件系统中文件/目录进行增删改查，读写的功能.

原生模块--fs =》const fs=require("fs");

**同步：**

* + 读取文件内容：**readFileSync(file)**

**const** fs = require(**"fs"**);  
console.log(**"开始文件写开始。。。"**);  
**var** str=**new** Date().toString();  
**var** file=**"./public/2.log"**;  
fs.writeFileSync(file,str);  
console.log(**"开始文件写结束..."**);

* + 写入文件：**writeFileSync(file,str/buff)；**
  + 向文件添加内容：**appendFileSync(file,str/buff);**

**异步：**

* + 异步读取文件中的内容

**fs.readFile(file,function(err,data){}) //file☞文件地址**

* + 异步向文件写内容

**fs.writeFile(file,str/buf,function(err){})**

*//异步（asynchronize）读取文件内容***const *fs***=require(**"fs"**);  
**console**.log(**"1、文件读取开始"**);  
**var *src***=**"./public/4.css"**;  
**var *des***=**"./public/44.backup.css"**;  
***fs***.readFile(**"./public/4.css"**,**function**(err,data){

//文件读取完成调用该函数  
 **if**(err)**throw** err;  
 **console**.log(**"文件读取完成"**);  
 ***fs***.writeFile(***des***,data,**function**(err){  
 **if**(err)**throw** err;  
 **console**.log(**"文件复制完成"**)  
 })  
});  
**console**.log(**"3:js执行完成"**);

### http-模块（服务器模块）：详情day03(05\_http例子)

网站正在升级：（指定输出编码utf-8）

**const *http***=require(**"http"**);  
**const *fs***=require(**"fs"**);  
**var *server***=***http***.createServer();  
***server***.listen(8081);  
***server***.on(**"request"**,(req,res)=>{  
 res.setHeader(**"Content-type"**,**"text/html;charset=utf-8"**);  
 res.write(**"<h1>您好！网站正在升级中请5秒后重试</h1>"**)  
 res.end(); //通知客户端响应消息结束；  
})

### node.js第三方模块：（MySQL）-npm工具下载

**const *mysql***=*require*(**"mysql"**);  
**var *conn***=***mysql***.createConnection({  
 **host**:**"127.0.0.1"**,  
 **user**:**"root"**,  
 **password**:**""**,  
 **database**:**"jd"**,  
 **port**:3306  
});

* ***//使用占位符解决安全问题 //防var u = "' drop database mysql#";*var *sql***=**"INSERT INTO jd\_user VALUES(null,?,?)"**;  
  ***conn***.query(***sql***,[**"tom3"**,**"123"**],(err,result)=>{  
   **if**(err) **throw** err;  
   **console**.log(result);  
   **if**(result.**affectedRows**>0){  
   **console**.log(**"添加成功"**)  
   }**else**{  
   **console**.log(**"添加失败"**)  
   }  
   ***conn***.end(); //关闭连接  
  })

### 连接池（提高效率）node.js连接mysql

**const *mysql***=*require*(**"mysql"**);  
**var *pool***=***mysql***.createPool({  
 **host**:**"127.0.0.1"**,  
 **user**:**"root"**,  
 **password**:**""**,  
 **database**:**"jd"**,  
 **port**:3306,

* **//连接池**  
   **connectionLimit**:10 *//连接池大小；*})  
  ***pool***.getConnection((err,conn)=>{  
   **if**(err)**throw** err;  
   **var** sql=**"UPDATE jd\_user SET upwd=? WHERE uid=?"**;  
   conn.query(sql,[**"111"**,2],(err,result)=>{  
   **if**(err)**throw** err;  
   **if**(result.**affectedRows**>0){  
   **console**.log(**"更新成功"**)  
   }**else**{  
   **console**.log(**"更新失败"**)  
   }  
   });  
   conn.release(); *//归还连接*  
  });

### express模块

该模块还有自己专用官网: http://expressjs.com

该模块中文镜像网站:http://www.expressjs.com.cn

也可以npmjs.org 上下载模块

npm install express

npm i express

*//1:加载相应的模块***const *http*** = *require*(**"http"**);  
**const *express*** = *require*(**"express"**);  
**const *mysql*** = *require*(**"mysql"**); *//insert***const *qs*** = *require*(**"querystring"**);*//post 参数  
//2:创建express对象***var *app*** = ***express***();  
*//3:创建服务器对象***var *server*** = ***http***.createServer(***app***);  
*//4:绑定监听端口****server***.listen(8081);  
*//5:创建连接池***var *pool*** = ***mysql***.createPool({  
 **host**:**"127.0.0.1"**,  
 **user**:**"root"**,  
 **password**:**""**,  
 **database**:**"dangdang"**,  
 **port**:3306,  
 **connectionLimit**:10  
});  
*//6:处理二个应用  
//get /book\_add.html****app***.get(**"/book\_add.html"**,(req,res)=>{  
 res.sendFile(***\_\_dirname***+**"/public/book\_add.html"**);  
});  
*//post /bookadd.do****app***.post(**"/bookadd.do"**,(req,res)=>{  
 req.on(**"data"**,(data)=>{  
 **var** str = data.toString();  
 **var** obj = ***qs***.parse(str);  
 ***console***.log(obj.**title**+obj.**price**);  
  
 ***pool***.getConnection((err,conn)=>{  
 **if**(err)**throw** err;  
 **var** sql = **"INSERT INTO book VALUES(null,?,?,now())"**;  
 conn.query(sql,[obj.**title**,obj.**price**],(err,result)=>{  
 **if**(err)**throw** err;  
 **if**(result.**affectedRows**>0){  
 res.json({**code**:1,**msg**:**"添加成功"**});  
 }**else**{  
 res.json({**code**:-1,**msg**:**"添加失败"**});  
 }  
 conn.***release***();  
 })  
 });  
 *//res.send("ok");* });  
});