Server Http

一、服务器相关

1、WEB架构

C/S架构 -- Client(客户端)/Server(服务器端)

优点：运行稳定、对宽带要求相对低、用户体验更好

缺点：更新过于复杂(同时更新客户端+服务器端)、占硬盘空间

比如：QQ、大型网游

B/S架构 -- Browser(浏览器)/Server

优点：更新简单(只更新服务器端)、实现更容易

缺点：用户体验相对差、对带宽要求相对高

比如：轻应用(PC端)、流应用(移动端)、网页游戏

2、XAMPP软件

(1)Apache，用于运行PHP的服务

①错误日志：Error:Apache shutdown unexpectedly.

原因 -- Apache服务使用端口号被占用

解决 -- 开始菜单→cmd→输入命令netstat -ano(查看当前使用的端口号)→查看80端口号被占用的PID是多少(如果为4)→打开任务管理器→进程→查看(选择列→PID勾选)→找到PID为4的选项

如果PID不为4，直接结束进程

如果PID也是4，说明Windows操作系统占用80端口，不能直接结束

解决 -- 修改Apache的端口号，让它不为80即可

在XAMPP中点击Apache后面的config→<Browser>[Apache]→在文件夹中找到conf文件夹→找到httpd.conf文件→使用EditPlus打开→查找listen(58行)→修改80为任意数(最多为四位，比如8888)→重新启动Apache服务

②如何打开：在浏览器地址栏输入 http://127.0.0.1:80

或者 http://localhost:80

③搭建本地WEB应用：

在XAMPP安装根目录下，找到htdocs文件夹打开→把原有的所有文件，存放在一个新文件夹中(或者全部删除)→在htdocs文件夹下创建index.html文件→重新访问Apache的服务页面，就可看到刚刚创建的主页信息(index.html内容)

④配置顶级域名：

系统盘(C盘)→Windows→System32→drivers→etc→用EditPlus打开hosts文件→找到最后一行写入 127.0.0.0 www.baidu.com→另存为...在桌面上→将桌面的新文件，替换旧文件即可→在浏览器地址栏输入配置的域名进行访问

(2)MySQL

①默认使用端口号为3306，一般不修改

②访问数据库

图形化界面

启动条件：Apache和MySQL同时启动服务

访问地址：http://localhost:8888/phpmyadmin

phpmyadmin，php提供的一种服务，访问MySQL数据库

命令行方式

报错 -- 直接在cmd中输入"mysql"命令，会显示"mysql"不是内部命令

原因：mysql的环境变量没有配置

解决：打开mysql的安装根目录→bin文件夹→mysql.exe→直接可以进入mysql

或者在cmd命令行中输入cd c:\xampp\mysql\bin→输入"mysql"命令

③登陆MySQL：

在cmd先启动MySQL→输入"mysql -u用户名 -p"

④退出 -- 输入命令"exit"

⑤每句SQL语句结束时，都要增加结束符";"

(3)Tomcat，用于运行Java的服务

二、数据库

1、关系型数据库

以表格(行和列)为主

Oracle，主要用于企业级开发市场

MySQL，主要用于互联网开发市场

2、非关系型数据库

是一种运动 -- 反关系型数据库

主流产品：mangoDB

3、SQL语言(语句) -- 所有数据库的通用操作语言

SQL官方建议：所有关键字全部大写

(1)DCL -- Date Controle Language(数据控制语言)

(2)DDL -- Date Define Language(数据定义语言)

①数据库

创建数据库

CREATE DATABASE 数据库名称;

创建并设置编码

CREATE DATABASE 数据库名称 CHARACTER SET utf8;

创建时增加判断(防止数据库重复)

CREATE DATABASE IF NOT EXISTS 数据库名称 CHARACTER SET utf8;

查看数据库

SHOW DATABASES;

切换数据库

USE 数据库名称;

修改数据库

ALTER DATABASE 数据库名称 要修改的内容;

删除数据库

DROP DATABASE 数据库名称;

如果存在，删除数据库

DROP DATABASE IF EXISTS 数据库名称;

②数据表 -- 具有行(记录)和列(字段)的表格

数据类型

数值(Number)数据类型

INT，整型(整数)

FLOAT或DOUBLE，浮点型(小数)

DECIMAL，精确值(金额)

\*注：取整数

日期时间数据类型

DATE，YYYY-MM-DD标准日期格式

DATETIME，YYYY-MM-DD HH:MM:SS

TIMESTAMP，时间戳(标识，唯一不可重复)

字符串数据类型

CHAR，长度固定的字符串

VARCHAR，长度可变的字符串

创建数据表

CREATE TABLE(

字段名称1 数据类型,

字段名称2 数据类型,

...

);

例如 CREATE TABLE jd\_user(

user\_id INT,

user\_name VARCHAR(30),

user\_gender VARCHAR(30),

user\_email VARCHAR(30)

);

创建时增加判断(防止数据表重复)

CREATE TABLE IF NOT EXISTS xxx();

删除数据表

DROP TABLE 数据表名称;

如果存在，删除数据表

DROP TABLE IF EXISTS 数据表名称;

修改数据表

ALTER TABLE 数据表名称 要修改的内容;

查看数据表

DESC 数据表名称;

③约束 -- 防止新创建的数据类型id重复

主键约束，PRIMARY KEY

主键自增，AUTO\_INCREMENT

比如 CREATE TABLE jd\_order(

order\_id INT PRIMARY KEY AUTO\_INCREMENT,

order\_number VARCHAR(30),

order\_price DECIMAL,

order\_time DATETIME

);

(3)DML -- Date Manipulate Language(数据操作语言)

①新增

INSERT INTO 数据表名 VALUES(字段值1,字段值2,...);

VALUES()中的内容为该表的所有字段值，且类型必须一一对应

例如 INSERT INTO jd\_user VALUES(

0,

'张无忌',

'男',

'xxx@qq.com'

);

如果是主键自增，INSERT INTO jd\_order VALUES(

NULL,

'123456789',

6288.09,

'2017-1-11 17:50:11'

);

INSERT INTO 数据表名(字段1,字段2,...) VALUES(字段值1,字段值2,...);

VALUES()中的内容与表名()的内容必须一一对应

②修改(更新)

UPDATE 数据表名 SET 字段名=字段值;

默认修改所有记录的当前字段

UPDATE 数据表名 SET 字段名=字段值, WHERE 字段名=字段值;

WHERE前面是新值，后面是要修改的旧值的信息

例如 UPDATE jd\_user SET user\_name='张无忌' WHERE user\_id=0;

UPDATE 数据表名 SET 字段名1=字段值1,字段名2=字段值2,... WHERE 字段名1=字段值1,字段名2=字段值2,...;

③删除

DELETE FROM 数据表名;

默认删除当前表中所有数据

DELETE FROM 数据表名 WHERE 条件;

删除符合条件的所有数据内容

例如 DELETE FROM jd\_user WHERE user\_id=2;

\*注：DELETE语句一旦执行，删除的内容无法恢复

物理删除：永久删除，无法恢复

逻辑删除：从逻辑上删除，并没有真正删除(并不执行DELETE语句)

为数据表增加一个字段，该字段专门用于表示当前这条记录的状态

比如删除状态和正常状态，用户操作后，可将两者互相切换

(4)DQL -- Date Query Language(数据查询语言)

SELECT \* FROM 数据表名;

默认查询表中所有记录(字段)

SELECT 字段名1,字段名2,... FROM 数据表名;

查询指定字段名

SELECT 字段名1,字段名2,... FROM 数据表名 WHERE 条件;

例如 SELECT \* FROM jd\_user WHERE user\_id=0;

4、SQL错误

(1)ERROR 1007：Can't create database 'xxx';database exists

表示数据库已存在，名称重复

(2)ERROR 1050(42S01):Table 'xxx' already exists

表示数据表已存在

(3)ERROR 1008(HY000):Can't drop database 'xxx' ;database doesn't exists

表示数据库不存在，无法执行删除

三、PHP语言

1、基本内容

(1)LAMP -- Linux Apache MySQL PHP

(2)PHP文件的扩展名为".php"

(3)如何运行PHP页面

将创建好的PHP页面拷贝到Xampp安装目录中的htdocs目录中；启动Apache服务，在浏览器中进行访问

(4)PHP允许编写HTML代码+PHP代码

PHP内容必须写在<?php ... ?>这样的格式中

(5)PHP与Js的区别

PHP -- 运行在服务器端的脚本语言

Js -- 运行在客户端页面中的脚本语言

(6)面向对象与面向过程

①面向对象

基于原型Prototype的面向对象 -- Js

基于类Class的面向对象 -- PHP/Java

②面向过程

例如 一个人的一生，面向对象是吃饭、睡觉、等等；面向过程是出生、成长、结婚、去世

2、变量和常量

变量，值允许改变

定义

$变量名=值;

测试

var\_dump($变量名);

修改

$变量名=新值;(新值可以是不同类型，比如字符串改为数字)

PHP中的变量与Js的变量类似，都是弱类型

常量，值不允许改变

定义

const 常量名=值;

define(常量名,值);

例如 define("username","zhang");

var\_dump(username);

PHP中的常量一旦定义，不能重新赋值，否则会报错(Parse error，解析错误)

3、数据类型

(1)四种标量类型

Integer，Int的全称，表示数字类型

Float或Double，浮点型

String

单引号，直接定义字符串，如果包含变量名，不能识别；效率高

双引号，如果包含变量名，会识别并自动替换为变量值；效率低

Boolean

(2)两种复合类型

Array

①直接量方式

例如 $arr1=[1,2,3];

②内置对象方式

例如 $arr2=array(

'0'=>'zhangwuji',

'1'=>'zhouzhiruo'

);

或$arr3=array(

'p1'=>'zhangwuji',

'p2'=>'zhouzhiruo'

);

Object

定义类(Class)

class 类名{

$属性名=值;

function 方法名(){...}

}

基于类创建对象

$对象名=new 类名;

调用对象的属性或方法

$对象名->方法名();

$对象名->属性名;

(3)两种特殊类型

Resource，资源

Null，释放资源

Null和""的区别

null，不存在

""，存在，但值为空

4、运算符

字符串的连接符为"."

其他的跟Js一样

5、循环结构

while，先判断后执行

do...while，先执行后判断

for

foreach，类似于Js中的forin

foreach(array as value){...}

遍历后的value，其实是PHP的变量

例如 $arr=[1,2,3,4];

foreach($arr as $value){

echo $value;

}

foreach($arr as $key=>$value){

echo "[$key]=>$value";

}

6、分支结构

if...else if...else

switch...case

break，停止循环

continue，停止本次循环，后面的继续

7、PHP预定义

变量

$\_GET，客户端的GET请求方式

定义HTML页面中的表单的method属性值为get

定义HTML页面中的表单元素时，定义name属性值

在PHP页面中使用$\_GET[name属性值]

$\_POST，客户端的POST请求方式

定义HTML页面中的表单的method属性值为post

定义HTML页面中的表单元素时，定义name属性值

在PHP页面中使用$\_POST[name属性值]

$\_REQUEST，客户端的GET/POST请求方式

$\_COOKIE，客户端的COOKIE请求

$\_FILES，专门处理文件上传

函数 -- MySQL数据库的扩展

mysql，原生MySQL API

mysqli，MySQL增强版扩展

四、PHP链接MySQL

1、面向过程风格

(1)DML(增删改)

①PHP与MySQL之间建立连接

mysqli\_connect(host,username,passwd,dbname,port);

host，MySQL数据库所在计算机的IP地址

本机的MySQL -- 127.0.0.1或localhost

username，登陆MySQL数据库的用户名

默认为"root"

passwd，登陆MySQL数据库的用户密码

默认为空，但是必须占位(空字符串)

dbname，操作的MySQL数据库的数据库名称

port，MySQL数据库使用的端口号

默认为3306

该方法具有返回值，返回一个连接对象

例如 $conn=mysqli\_connect('127.0.0.1','root','','jd','3306');

②定义SQL语句

例如 $sql="INSERT INTO jd\_user VALUES(2,'周芷若','女','zzz@qq.com')";

如何测试SQL语句编写是正确的

使用var\_dump()函数将其打印出来；然后将SQL语句放在cmd命令行中进行运行查找错误所在

③PHP向MySQL数据库发送SQL语句

mysqli\_query(link,query);

link，与MySQL数据库建立的连接对象

query，向MySQL数据库发送的SQL语句

该方法具有返回值，返回boolean值

true表示SQL语句执行成功

例如 $result=mysqli\_query($conn,$sql);

④关闭PHP与MySQL之间的连接

mysqli\_close(link);

link，与MySQL数据库建立的连接对象

(2)DQL(查询)

①PHP与MySQL之间建立连接

②定义SQL语句

③PHP向MySQL数据库发送SQL语句

mysqli\_query(link,query);

该方法具有返回值，返回boolean值

如果SQL语句执行成功，返回结果集对象(mysqli\_result)

查询回来的结果(二维表格)被封装在结果集对象中

false，表示SQL语句执行失败

④解析结果集对象

mysqli\_result结果集对象

field\_count，得到当前数据表的字段数量

num\_rows，得到当前数据表的记录(行)数

mysqli\_fetch\_array(result,resulttype);

作用：将结果集转换为数组

result，结果集对象

resulttype，设置当前转换后的数组类型

MYSQLI\_BOTH，默认值

MYSQLI\_ASSOC，表示转换为关联数组

MYSQLI\_NUM，表示转换为索引数组

mysqli\_fetch\_assoc();

将结果集对象转换为关联数组

\*解析结果集对象的所有记录

while($arr=mysqli\_fetch\_array($result)){

得到查询后的所有记录

}

⑤关闭PHP与MySQL之间的连接

(3)解决中文乱码问题

在向MySQL数据库发送SQL语句之前

mysqli\_query($conn,'SET NAMES utf8');

2、面向对象风格

①通过构造器创建mysqli对象，与MySQL数据库建立连接

$mysqli=new mysqli(host,suername,passwd,dbname,port);

②定义SQL语句

③PHP向MySQL数据库发送SQL语句

通过mysqli对象调用query(query)方法

$result=$mysqli->query($sql);

query，发送的SQL语句

解决乱码

$mysqli->query('SET NAMES utf8');

④解析结果集对象

$arr=$result->fetch\_array();

⑤关闭PHP与MySQL之间的连接

$mysqli->close();

四、HTTP

它是一个网络协议(客户端与服务器端)

1、分类：

(1)请求协议

GET和POST两种请求类型是最常用的

请求类型至少有7种

标准请求API

PUT，新增

POST，修改

DELETE，删除

GET，查询

①GET请求方式

请求行

http协议的版本信息，比如1.1

请求地址，URL?key=value&key=value

请求方式，GET

状态码

1xx，获取信息

2xx，请求成功

200，表示请求成功

3xx，重定向

302，配合location

304，(服务器端)没有修改，访问缓存

305，使用代理

4xx，客户端错误

400，请求失败

403，请求被拒绝

404，网页未找到

405，请求类型未被允许

5xx，服务器端错误

500，服务器端内部错误

502，路径错误

504，请求超时

505，http版本不支持

请求头

格式

key:value

key:value,value,value,...

选项

Accept，服务器端允许接收的MIME类型

MIME类型，用于表示当前文件的格式

例如 html -- text/html

xhtml -- application/xhtml+xml

xml -- text/xml，application/xml

Js -- text/javascript，application/javascript

CSS -- text/css

text -- text/plain

jpg -- image/jpeg

png -- image/png

mp4 -- video/mp4

ogv -- video/ogg

mp3 -- audio/mpeg

Accept-Encodiong，是否压缩数据

使用gzip压缩格式(Linux系统的压缩格式)

Accept-Language，表示接受的语言

中文：zh\_CN和zh

Connection，表示是否保持连接

keep-alive，表示保持连接

Host，服务器端的地址

Referer，表示当前请求是从哪里来的

防盗链功能

User-Agent，用户浏览器的相关信息

Cookie，服务器端向客户端的缓存数据(自动携带)

Content-Length，请求数据的字符个数

请求体 -- 空(并不是不存在)

GET方式的请求，请求数据增加在URL?key=value

②POST请求方式

请求行

http协议的版本为1.1

请求地址URL后没有请求数据

请求方式为POST

状态码 -- 200 OK

请求头

Cache-Control，表示是否包含缓存内容

max-age=0，设置最大缓存周期为0，即没有缓存

Content-Type，表示表单的enctype属性值

application/x-www-form-urlencoded

如果没有它，POST方式是无法向服务器端发送数据的

其他选项与GET方式一样

请求体

请求数据

格式 -- key:value

③GET与POST方式的区别

GET

请求行中的请求类型 -- GET

请求体为空

请求数据增加在URL(请求地址)后

不安全

容量有限

可能会出现中文乱码

POST

请求行中的请求类型 -- POST

请求体为请求数据

请求数据并没有增加在URL之后

相对安全

没有容量限制

依旧可能会出现中文乱码

(2)响应协议

响应行

http协议的版本1.1

状态码 -- 200 OK

响应头

格式

key:value

key:value,value,...

选项

Content-Length，响应数据的字符个数

Content-Type，设置响应数据的MIME类型和编码格式

html -- text/html;charset=utf-8

Date，返回当前响应的时间

标准英文格式 -- Thu, 19 Jan 2017 08:01:06 GMT

Keep-Alive，设置保持连接的时间

timeout=5，设置超时时间

max=100，设置最大有效时间

Server，返回服务器端的相关信息

X-Powered-By，服务器端所使用的语言

Accept-Ranges设置当前响应数据的计量单位

bytes，表示字节

Last-Modified，返回服务器端最后一次修改的时间

响应体

所有的响应数据

设置响应头

①在服务器端(PHP)

使用header()函数

header('key:value')

②在HTML页面

使用HTML页面的<meta>元元素

<meta http-equiv="" content="">

http-equiv，设置响应头的key

content，设置响应头的value

特殊响应头

①设置客户端是否允许缓存

Cache-Control

no-cache，表示不缓存

Pragma

no-cache，表示不缓存

Expires

0，表示不缓存

\*这3个兼容各种浏览器的缓存机制

②设置重定向

Location，设置重定向到的地址

Status，302

2、HTTP与HTTPS

HTTP，普通协议

HTTPS，加密协议

3、cookie与session

cookie，客户端浏览器的缓存(存储在硬盘)

cookie中的数据是以明文(未加密)存储的

cookie有限制

单个cookie的大小(4KB)

每个网站最多只能存储200多cookie文件

安全性并不高

session，服务器端的缓存(存储在硬盘)

4、浏览器内核

IE -- IE6/7/8 IE9/10/11

其他浏览器 -- Webkit

chrome -- V8引擎(Node.js)

firefox -- Gecko

safari -- Webkit

国内浏览器

360 -- IE或Chrome

猎豹 -- Chrome

搜狗 -- IE+Webkit

百度 -- 自主内核(V5)

QQ -- 自主内核(V5、X5)

遨游 -- 自主内核

5、中文乱码

客户端与服务器端之间的交互

客户端页面中文乱码问题

定义<meta charset="UTF-8">元元素

设置HTML页面文件的编码也是utf-8

设置浏览器的编码格式也是utf-8

客户端向服务器端发送请求数据时

先将中文转换为Unicode码

再将Unicode码转换为中文

服务器端向客户端响应数据时

设置响应头"Content-Type"为"charset=utf-8"

服务器端与数据库之间的交互

mysqli\_query($conn,"SET NAMES utf8")

6、中文编码

UTF-8 -- Unicode

UTF-16、UTF-32

GBK

GB2312

7、静态资源与动态资源

静态资源

HTML、CSS、Js

动态资源

PHP -- HTML+PHP

JSP(Java Server Page) -- Java+HTML

ASP(.net)

Node.js

8、性能优化

尽量减少对外部资源的引用

页面中的图片

页面中的Js脚本

页面中的CSS样式

页面中的视频或音频

尽量减少创建连接的次数

客户端能完成的功能，就不麻烦服务器端

优化服务器端和数据库

尽量减小请求和响应的数据内容

9、主流协议

HTTP、FTP、POP3、SSH

五、URL(访问地址)

URL，统一资源定位符

完整URL格式

网络协议:IP地址:端口号/路径;参数?数据#锚点

例如 https://www.baidu.com

http://127.0.0.1:80

URI，统一资源标识符

\*如果是URL，一定是URI；如果是URI，不一定是URL