WEB服务器

接收客户端请求，查找数据并传输给客户端

MYSQL相当于数据库服务器的直接管理员

静态服务器：任何人任何时间不变

静态内容：HTML/CSS/JS/FLASH

软件：Apache，Httpd,MS IIS.NginX

动态服务器：不同人不同时间不变

动态内容：JSP: HTML+JAVA 静态web+java解释器

PHP:HTML+PHP 静态+PHP解释器

ASP.NET:HTML+C# 静态+C#

Node.js:HTML+Node 静态+Node

搭建PHP开发环境

服务器

静态web服务器

安装PHP解释器，与web整合

编写网页放在web服务器 htdocs

启动web服务器

客服端

访问

PHP

$变量名 = 值; $声明变量名

echo $变量名 输出 == print()

var\_dump($变量)； 输出type加上所赋值

声明常量 const 名字 = 值； 常量名字惯用为大写字符

数据类型：

4标量/值类型

int,

float/double,

strig,“$a”== str($a)

‘$a’== ‘$a’

字符串拼接： .

bool(false输出为空字符)

2复合类型

array:数组

object:对象

2其他类型

null:空

resource:资源

运算符

算数运算符：强行运算字符串只对字符串中的数字相加

+ 不做拼接就是加。

-

\*

/

%

++ 自加一 $a = $n++ $a==$n,$n += 1; $a = ++$n $a== $n==$n +=1

-- 自减一

比较运算符： 对于无数字的字符串会比较unicode码

会包含隐式转换

>

<

==

=== 全等于，会判断类型

！=

！== 不全等，会判断类型

逻辑运算符：&& == and || == or ! == NOT 短路运算可以作为赋值计算

位运算符：<<: 按位左移 左移n位，表示\*2^N

>>：按位右移 右移n位 表示/2^N

赋值运算符：=,+=,-=,\*=./=,%=

拼接运算符：.,

.= 字符串中的+=

三目运算符：？:表达式1？表达式2：表达式3

以表达式1为依据，1位TRUE，返回表达式2，为FALSE，返回表达式3

逻辑结构：

算法的基本结构：顺序执行、选择执行、循环执行

选择执行：if(条件){条件为true的代码};

if(){}else{}

if(){}else if(){}else

选择分支二：switch……case……，指定变量，判定一个条件成立后面条件都会成立，若不想要加break ,而且case语句判定为==不是===

switch(变量名){

case 值1：

语句1；

break

case 值2：

语句2；

break

default:

语句n;

}

循环结构

while(循环条件/TRUE){}

do{}while(循环条件)；判定时间点不同，必定执行一次循环主体

for循环

for( 表达式1:表达式2 :表达式3 ){}表达式1开头只执行一次，2->循环->3->2…..2false退出循环

类型转换

$n = 1;

(string)$n

存在 break;与continue;

复合类型array

创建索引数组：$n = [a,b,c]；就是列表 不能用echo输出整个array

获取列表长度:count($n);,获取其中的一个值$n[… ],$n[ ][ ]修改值 $n[…] = x;

添加值操作python：x.append() php: $n[x] = y;/$n[] = y;

创建关联数组：$n = [‘a’ => 12,’b’=> 14]; python中的字典，添加改变和py一样

foreach循环

foreach($素组名 as $元素变量名){} === py中的 for x in list():

用来便历关联数组

foreach（$数组名 as $下表名字=> $值）{}

预定义数组

预定义数组不用声明就可以用

$\_GET

$\_POST

$\_COOKIE  
 $\_FILES

$\_REQUEST：保存客户端提交给服务器的请求数据，网址后面加?名字=值，都为字符串，都在$\_REQUEST

$\_SESSION

默认都为array(0){},

Function函数

function 函数名(x){……} 其他设置跟py一样

php预定义函数 mysql函数

mysqli\_XXX();

链接服务器

mysqli\_connect(host,name-u,-p, 数据库名称,端口号-P)；

提交sql语句给服务器执行

mysqil\_querry(服务器，sql语句)；

mysqli\_error($conn)

返回值：DML：数据操作语言 INSERT,DELETE,UPDATE 失败false

DCL：数据控制语言 GRANT授权, REVOKE收回

DDL：数据定义语言 CAREATE.DROP,ALTER定义列 TRYNCATE

DQL：数据查询语言 SELECT 失败false 成功返回查询结果及对象

mysqli\_fetch\_row($n);抓取一行数据返回索引列表，没有就为空

mysqli\_fetch\_assoc($n); 抓取一行数据返回关联列表，没有就为空

mysqli\_fetch\_all($n，1/2/3);抓取全部数据返数组的数组，没有就为空

1为返回关联数组，2位索引数组，3位两个都有

查看执行结果

看执行语句的返回值

关闭链接（可以省略）

mysqil\_close(服务器)；

用户注册功能

客户端请求数据

检验用户输入

链接数据库

向数据库提交insert语句

输出结果

@ 符号可以压制语句的报错

die(“。。”)终止页面执行

mysqli\_insert\_id($n);获取刚刚执行insert语句产生的自增编号

mysqli\_affected\_rows($n);获取被影响的行数

导入模块

require(“文件名“)； === python import

binary 二进制输入

php正则和js正则一样

preg\_match(正则，检测的变量){}

$\_SERVER[‘REMOTE\_ADDR’]获得IP地址

上传文件

可指定上传类型

上传方式

表单同步提交

ajax

第三方

获取文件名后缀 strstr($picname, ‘.’);

创建文件名称，时间戳，随机数

time().rand(1,999).后缀名

移动文件

move\_uploaded\_file(文件名，新目录);

上传大于50MB文件

apache服务器 php.ini专用于配置php程序

phpinfo()获取所有的php配置文件信息

修改配置

file\_uploads = on/off 是否允许上传

upload\_max\_filesize = xxx 允许上传的最大文件

post\_max\_size = xxx post上传文件的大小

max\_execution\_time = xxxs 执行php时间长度

max\_input\_time = 600s input自大的时长

memory\_limit = xxxM 指定单个php执行时内存使用的最大值

memory\_limit>post\_max\_size>upload\_max\_filesize

删除数据

软件项目中有一个极特殊用户（超级管理员）可以删除数据

修改header运行跨域

header(‘Access-Control-Allow-Orgin:\*’);\*为允许所有跨域

$\_SERVER的内容

$\_SERVER['HTTP\_HOST'] 请求头信息中的Host内容，获取当前域名。

$\_SERVER["SERVER\_NAME"] 输出配置文件httpd.conf中的ServerName，一般情况下与HTTP\_HOST值相同，但如果服务器端口不是默认的80端口，或者协议规范不是HTTP/1.1时，HTTP\_HOST会包含这些信息，而SERVER\_NAME不一定包含。（主要看配置文件的设置）。

$\_SERVER["HTTP\_USER\_AGENT"] 获取用户相关信息，包括用户浏览器、操作系统等信息。

$\_SERVER['HTTP\_ACCEPT'] 当前请求的ACCEPT头部信息。

$\_SERVER["HTTP\_ACCEPT\_LANGUAGE"] 这个值是由浏览器发送，表明用户默认的语言设置，后面的q值表示用户对该语言的喜好程度。

$\_SERVER["HTTP\_ACCEPT\_ENCODING"] 大部分的现代浏览器都支持gzip压缩，并会把这一信息报告给服务器。这时服务器就会压缩过的HTML发送给浏览器。这可以减少近80%的文件大小，以节省下载时间和带宽。

$\_SERVER["HTTP\_COOKIE"] 浏览器的cookie信息。

$\_SERVER["HTTP\_CONNECTION"] 当前请求的连接情况。

$\_SERVER["HTTP\_UPGRADE\_INSECURE\_REQUESTS"] 表示浏览器可读懂服务器发过来的请求，

$\_SERVER["HTTP\_CACHE\_CONTROL"] 表示浏览器是否会缓存这个页面信息。

$\_SERVER["PATH"] 当前脚本所在文件系统。

$\_SERVER["SystemRoot"] 当前服务器的操作系统。

$\_SERVER["COMSPEC"] 指向cmd.exe的路径。

$\_SERVER["PATHEXT"] 环境变量设置。

$\_SERVER["WINDIR"] 脚本指向的系统目录。

$\_SERVER["SERVER\_SIGNATURE"] 包含服务器版本和虚拟主机名的字符串。

$\_SERVER["SERVER\_SOFTWARE"] 服务器软件配置信息。

$\_SERVER["SERVER\_ADDR"] 当前运行脚本的服务器的ip地址。

$\_SERVER["SERVER\_PORT"] 服务器端口。

$\_SERVER["REMOTE\_ADDR"] 浏览网页的用户ip。

$\_SERVER["DOCUMENT\_ROOT"] 当前运行脚本所在的根目录。

$\_SERVER["REQUEST\_SCHEME"] 服务器通信协议，是http或https。

$\_SERVER["CONTEXT\_PREFIX"] 前缀。

$\_SERVER["CONTEXT\_DOCUMENT\_ROOT"] 当前脚本所在的文档根目录。

$\_SERVER["SERVER\_ADMIN"] 服务器管理员信息。

$\_SERVER["SCRIPT\_FILENAME"] 当前执行脚本的绝对路径。

$\_SERVER ["REMOTE\_PORT"] 用户连接到服务器时所使用的端口。

$\_SERVER["GATEWAY\_INTERFACE"] 服务器使用的CGI规范的版本。

$\_SERVER["SERVER\_PROTOCOL"] 请求页面时通信协议的名称和版本。

$\_SERVER["REQUEST\_METHOD"] 请求提交数据的方式。

$\_SERVER["QUERY\_STRING"] 服务器请求时？后面的参数。

$\_SERVER["REQUEST\_URI"] 当前脚本路径，根目录之后的目录。

$\_SERVER["SCRIPT\_NAME"] 当前脚本的路径。这在页面需要指向自己时非常有用。

$\_SERVER["PHP\_SELF"] 当前正在执行脚本的文件名。

$\_SERVER["REQUEST\_TIME"] 得到请求开始时的时间戳。