常量（define/const）

```
define('PI',3.14);
const a = 10;
define  VS  const
    define语法可以在分支结构中定义常量，const不允许的。
    define定义的常量可以自定义是否区分大小写。  define('PI',3.14,true);    true 表示不区分大小写
```

0、0.0、null、’ ’ 、’ 0’ 、空数组　　等效于布尔false的值

```
isset();          判断一个变量的值是否为null，如果是Null返回false
empty()      判断一个变量的值是否为空
```

```
===               同时判断变量的值与类型是否相同，如果相同返回true
```

if标签语法：

标准语法：<?php if(...):?> <?php endif?>

简化语法：<?php if(...){?> <?php }?>

for标签语法 标准语法： <?php for():?> <?php endfor?>

简化语法： <?php for(){?> <?php }?>

while的标签语法

标准语法： <?php while():?> <?php endwhile?>

简化语法 <?php while(){?> <?php }?>

字符串

使用单引号或双引号括起来的0个或多个字符。

单引号：不解析变量的值，能够被转义的 \\ \’

双引号：能解析变量的值，都能被转义

字符串长度 strlen(变量)

查找并截取函数

```
strstr(str,substr);  用于在字符串str中查询子字符串substr首次出现的位置，并截取到最后
strrchr(str,substr);  用于在字符串str中查询子字符串substr最后一次出现的位置，并截取到最后
```

查找

```
strpos(str,substr);  用于在字符串str中查询子字符串substr首次出现的位置
strrpos(str,substr);  用于在字符串str中查询子字符串substr最后次出现的位置
```

分割

```
explode(分隔符,str);  指定指定的分隔符，将字符串str进行分割，并将每一部分组织成数组，并返回
```

替换

```
str_replace(search,rep,str);  在字符串str中，查找search表示的内容，并替换为rep代表的内容
```

大小写转换

```
strtolower();
strtoupper();
```

去除指定字符

```
trim(str【,substr】)
    用于将字符串str两侧的子字符串substr去除。
    substr可以省略，如果省略表示去除空格。
    ltrim(str【,substr】);
    rtrim(str【,substr】);
```

pathinfo

```
pathinfo(path【,option】);
    path是一个文件路径的字符串
    用于获取一个文件的路径信息(文件名、文件夹、文件名、扩展名)
    option参数用于获取路径信息中指定的部分

    D:\wamp64\www\phptest.php:100:
    array (size=4)
    'dirname' => string 'D:\wamp64\www\jxshop' (length=20)
    'basename' => string 'databases.sql' (length=13)
    'extension' => string 'sql' (length=3)
    'filename' => string 'databases' (length=9)
```

md5()

```
md5(str);  用于对str字符串进持加密。对任何长度的字符串进行md5处理得到的都是32位长度的字符串。
```

htmlspecialchars

```
htmlspecialchars(str)
用于将字符串str中的大于号小于号转换为相应的字符实体。  <(&lt;)  >(&gt;)
```

数组：

PHP数组的分类

索引数组:数组的下标是 整数 。这样的数组就是 索引数组

关联数组:数组的下标是 字符串 。这样的数组就是 关联数组

数组创建

索引数组的创建 在php中数组的下标可以不连续。

```
// 显式创建
$arr1 = array(10,20,30,40);
$arr2 = [10,20,30,40];
// 隐式创建
$arr3 = array();
$arr3[0] = 1;
$arr3[2] = 2;
$arr3[3] = 3;
$arr3[5] = 5;
```

关联数组的创建

```
$arr = array(键名=>键值, 键名=>键值,….);
$arr = [键名=>键值, 键名=>键值,….];
```

php中的数组元素由两部分组成，键名(下标)，键值

多维数组

一维数组元素的访问 $数组名[下标/键名]

二维数组元素的访问 $数组名[行下标][列下标]

数组的长度 count() 用于获取数组的长度

数组的指针：数组指针用于表示当前所关注的元素。

current($arr) 用于当前指针所指向的元素的键值

key($arr) 用于当前指针所指向的元素的键名

next($arr) 用于将数组的指针下移。

prev($arr) 用于将数组的指针上移。

reset($arr); 用于将数组的指针重置(归位，数组的指针默认位于第1个元素)。

end($arr); 用于将数组的指针移到最后一个元素。

数组的遍历

for for循环是使用循环控制变量来模拟下标的方式来遍历数据，只能遍历下连续或有规则

foreach

```
foreach($arr as 【$key=>】$value){
//循环体
}
```

$arr是所要遍历的数组，$key、$value就是一个变量，变量名可以自定义。

数组的长度 count(数组名)

获取数组元素的键名与键值

array_keys() 获取数组元素所有的键名

array_values() 获取数组元素所有的键值

判断键名与键值是否存在

array_key_exists(key,arr) 用于判断某个键名是否存在于数组中，如果存在则返回true，否则返回false

in_array(value,arr); 用于判断某个键值是否存在于数组中，如果存在则返回true，否则返回false

数组的合并 array_merge(数组1,数组2…)

数组的排序

sort($arr); 对数组按键值进行升序排序

rsort($arr); 对数组按键值进行降序排序

asort($arr); 对数组按键值进行升序排序，但原下标不会变

arsort($arr); 对数组按键值进行降序排序，但原下标不会变

extract extract($arr); 用于解压数组，将关联元素转换为以键名为变量名的变量。

数组的相关算法

1、排序算法

①、冒泡排序法

```
$arr = [17,3,14,30,24,10,5];
$count = count($arr);
var_dump($count);
for ($i=0; $i < $count - 1; $i++) {
    for ($j=0; $j < $count - $i -1; $j++) {
        var_dump($i);
```

```
                    if($arr[$j] > $arr[$j+1]){
                        $tmp = $arr[$j];
                        $arr[$j] = $arr[$j+1];
                        $arr[$j+1] = $tmp;
                        var_dump($arr);
                    }
                }
            }
```

文件载入

include 在引入文件时，如果被引入的文件不存在则会报错，但程序还会继续向下执行。

require在引入文件时，如果被引入的文件不存在，则会上断程序的执行。

经验法则：

require一般用于引入php文件。因为php里面一般书写的是功能性的代码。

include一般用于引入html文档。

include、require VS include_conce、require_once的区别

include_once、require_once每次在引入文件时，都会检查所要引入的文件之前有没有被引入过，如果有引入过就不会再引入。

include、require没有这样的检查过程。

__FILE__ 用于获取文件所在的完整文件名。

__DIR__ 用于获取文件所在的路径。

__FILE__与__DIR__并不会被引入后所更改。永远代码__FILE__与__DIR__所在的文件的文件名与文件夹名。

登陆：

mysql -h主机地址 -P端口号 -u用户名 -p密码

数据库操作

增 create database 数据库名 【库选项】;

删 drop database 数据库名;

改

查 show databases; show create database 数据库名;

对表进行操作

增 create table 表名(

字段1 列类型 【列属性】,

字段2 列类型 【列属性】,

….

) 【表选项】 【表选项】 charset 存储字符集，collate 校验集，engine存储引擎

删 drop table 表名;

改

查 show tables; show create table 表名; desc 表名;

对数据进行操作

增 insert into 表名【(字段列表)】 values(值列表)

查 select 字段列表|* from 表名

【where子句】【group by子句】【having子句】【orderby子句】【limit子句】

>、<、>=、<=、<>、=、and、or、not、between m and n、in(值列表)、is null、like

like需要两个占位符

% 代表当前位置及其后的0个或多个字符

_ 代表当前位置1个字符

改 update 表名 set 字段名=值,字段名=值…. 【where子句】

删 delete from 表名 【where 子句】

not null , default , unique , primary key , auto_increment

对于auto_increment属性，必须应用在整型字段并且是primary key 或者unique。但是绝大多数与primary key联合使用。

增加字段 alter table 表名 add 【column】 字段名 列类型 列属性 【first| after 字段名】;

删除字段 alter table 表名 drop 【column】 字段名;

修改字段名 alter table 表名 change 原字段名 新字段名 列类型 列属性;

修改列类型 alter table 表名 modify 字段名 列类型 列属性 【first|after】;

修改表名 alter table 表名 rename to 新名;

修改表选项 alter table 表名 engine 存储引擎|charset 存储字符集|collate校验集

虽然MySQL提供了修改存储字符集的命令，但是如果一个表中已有数据，那么不要执行修改存储字符集的

命令。
    修改列属性 列属性包含not null、default、unique、auto_increment、primary key
        增加列属性 普通的列属性
            alter table 表名 modify 字段名 not null defult 值 unique、auto_increment
        主键字段的添加
            alter table 表名 add primary key(字段名)

```
create table user(
    id int not null primary key auto_increment comment '主键',
    username varchar(50) unique comment '用户名',
    card_id int not null comment '身份证',
    price varchar(10) not null default 0.00 comment '价格'
    )charset utf8;

insert into user (username,card_id) values ('zhu',410411);
insert into user (username,card_id,price) values ('lin2',410423,'200');
insert into user (username,card_id,price) values ('lin3',410421,'200');
insert into user (username,card_id,price) values ('lin4',410425,'200');
insert into user (username,card_id,price) values ('li15',411424,'300.00');

alter table user charset gbk;
```

复制表结构   create table 表B like 表A;
        使用表A的结构重新创建一个表B
        只是复制结构，不包含表的数据。
备份SQL执行结果   create table 表名 select 语句;
    将一条select语句得到的结果保存到一个新创建的表中。
限制更新   update 表名 set 字段=值 【where子句】【limit子句】
    用于将where子句匹配到记录，仅更改limit子句限制的条数。
限制删除   delete from 表名 【where子句】【limit子句】
        用于将where子句匹配到记录，仅删除limit子句限制的条数。
批量插入 insert into 表A【(字段列表)】 select 字段列表 from 表B;
        将select语句得到的数据，插入到表A中。
清空表   delete from 表名;   只会将表中的数据删除，并不会重建索引。
        truncate 表名;   不但会删除表中的数据，而且还会重建索引。

select 【all|distinct】 字段列表|*|字段名 【as】 别名 from 数据源 【as】 表别名
    【where子句】【group by子句】【having子句】【order by子句】【limit子句】

        all   (默认)在显示结果中包含重复的数据
        distinct   在显示结果中去掉重复的数据。
        select all card_id from user;
        select distinct card_id from user;
        列别名: select username 【as】 用户名 , card_id 【as】 身份证 from user;
        表别名：select * from user as a;

        【group by】
            count(字段名)（起别名）   select username , max(price) 【max】 from user group by price;
            count(*)
            max(字段名) / min(字段名) / avg(字段名) / sum(字段名) /
        多字段分组
            group by 字段1,字段2…   当group by后指定多个字段时，会进行多字段分组
            select * from user group by price , card_id;
            select username , price , count(*) from user group by price , price;
        回溯统计   with rollup
            多字段进行分组，统计函数默认是应用在最小组上的，如果想对包含最小组(最小组之上)的大组进行同样的统计，那么with rollup就可以实现这个需求。

        【having 子句】
            作用：
                where是在将硬盘中的数据读取到内存时进行第1次筛选。
                group by是针对where子句匹配以的记录进行分组统计
                having是针对group by分组统计得到的结果再进行第2次筛选。

        【order by 子句】   order by 字段1【asc|desc】【, 字段2【asc|desc】】…
            order by 对where子句、group by子句、having子句得到的结果进行一个显示顺序上的控制。

asc 缺省的升序
desc 降序
多字段排序

【limit 子句】 limit 【offset,】rows
limit是对面where子句、group by子句、having子句、order by子句得到的结果进行一个显示行数的一个限制。
offset 偏移量，第1条记录偏移量为0，第2条记录偏移量为1，依此类推。如果省略表示0
rows 显示的行数

【数据分页】
分页相关的因素：
每一页显示的记录数： rowsPerPage 人为设定的
当前的页码数： curPage 用户所点击的页码
假设
rowsPerPage=3 表示每一页显示3条记录
curPage 1 2 3 4 5 …
数据分页的公式：

```
select * from user limit (curPage-1)*rowsPerPage,rowsPerPage;
```

联合查询
select语句A
union 【all|distinct】
select语句B

分表存储联合查询。为什么要分表查询，为了解决查询1条记录更快，但是查询单条记录快了，解决解决所有呢，所以才就出现联合查询的语法。
【all|distinct】 union选项
all 表示所有
distinct (默认的)表示去重
在联合查询中的两条select语句，所查询出来的字段的个数必须一致。

```
select * from goodsA
union
select * from goodsB
```

联合查询的注意事项：
联合查询还可以解决，对同一个表的不同部分进行不同的操作。
如果联合查询的select语句中有order by子句，那么必须配合limit使用。

主表 从表 外键字段
创建外键 foreign key(外键字段) references 主表(主键字段)

```
create table goods(
id int not null auto_increment comment '主键',
goodsname varchar(50) not null comment '商品名称',
goodslink varchar(100) not null comment '商品链接',
price double(6,2) not null default 0.00 comment '价格',
primary key(id)
)charset utf8;

insert into goods (goodsname,goodslink,price) values ('Mate 30','https://www.baidu.com','4888.99');
insert into goods (goodsname,goodslink,price) values ('IphoneXs max','https://www.taobao.com','4888.99');
insert into goods (goodsname,goodslink,price) values ('小米','https://xiaomi.com','4888.99');

create table company(
    id int not null primary key auto_increment comment '主键',
    c_id int not null comment '外键字段',
    company varchar(50) not null comment '公司名',
    addr varchar(50) not null comment '公司所在地',
    foreign key(c_id) references goods(id)
)charset utf8;

insert into company (c_id,company,addr) values (1,'华为','中国');
```

```
    insert into company (c_id,company,addr) values (5,'华为','中国');

create table class(
c_id int unsigned primary key auto_increment,
c_name char(10),
c_room char(10)
)charset = utf8;


 create table stu(
 id int unsigned primary key auto_increment,
 s_sum char(10),
 s_name char(10),
 s_sex char(10),
 s_age tinyint unsigned,
 c_id int unsigned comment '外键字段',
 foreign key(c_id) references class(c_id)
 )charset = utf8;

insert into class (c_name,c_room) values ('zhang','三年级一班');
insert into class (c_name,c_room) values ('wang','三年级二班');
insert into class (c_name,c_room) values ('li','三年级三班');
insert into class (c_name,c_room) values ('zhao','三年级二班');
insert into class (c_name,c_room) values ('sun','三年级一班');



get
    数据不安全
    提交的数据量小
    提交的数据类型单一，只能传递文本数据。
    get方式主要用于从服务器端获取数据。

post
    相对安全
    提交的数据量大
    提交的数据类型多样。
    post方式主查用于向服务器端提交数据。

文件上传
    前台部分
        form表单：
            action属性应该指向一个php文件
            method属性必须设置为post
            enctype属性：
                取值：
        application/x-www-url-encoded                    (默认)只能上文本数据
        multipart/form-data                                                  可以上传多种类型的数据(上传
文件)

生成随机文件名
    mt_rand(m,n)          生成m与n之间的随机整数
    chr(code)         将code所表示的整数转换为相应的字符。

文件上传的原理
        当用户上传文件后，文件会被临时保存在临时文件夹中，如果在php脚本执行结束之后，此临时文件就会
被删除，所以我们将位于临时文件夹中的这个文件移动其他位置。
    move_uploaded_file(filename,dest);
        filename就是位于临时文件夹中的文件
        dest是最终所要保存的新路径。



SQL术语/概念        MongoDB术语/概念        解释/说明
database              database              数据库
table              collection             数据库表/集合
row                   document              数据记录行/文档
column             field                    数据字段/域
index              index                    索引
```

table joins		表连接,MongoDB不支持
primary key	primary key	主键,MongoDB自动将_id字段设置为主键

RDBMS	MongoDB
数据库	数据库
表格	集合
行	文档
列	字段
表联合	嵌入文档
主键	主键 (MongoDB 提供了 key 为 _id)

命名空间

命名空间是一种封装事物的方法；例如：函数、类、方法等。
命名空间作用：用来解决类名或应用程序名冲突问题；
使用 namespace 关键字来声明一个命名空间
所有代码都可以存在于命名空间中，但是只有三种代码收到空间影响：类、常量（const）、函数

namespace.php

```
<?php
    namespace App;
    class Student{
        private $name = 'zhang';
        public function __construct ( ) {
            echo "我的名字叫：{$this->name}"<br/>;
        }
    }
    function showInfo(){
        echo 'I Love You';
    }
    const DB_HOST = "localhost";
    $a = 100;
?>
```

```
<?php
    header(Content-Type:text/html;charset=utf8);
    require_one("/namespace.php");
    $obj = new App\student();
    App\showInfo();
    echo "<br>".App\DB_HOST;
    echo "<br/>".$a;
?>
```

目录的分割符号正斜杠(/)，空间路径的分割符是反斜杠(\)

```
mysql_fetch_row(result);                读取出来的是索引数组
mysql_fetch_assoc(result);      读取出来的是关联数组
mysql_fetch_array(result);      读取出来的是混合数组
```

```
<?php
    $link = mysqli_connect("127.0.0.1","root","root");
    $sql = "set names utf8";
    $result = mysqli_query($link,$sql);
    mysqli_select_db($link,"jxshop");
    $sql = "select * from jx_role";
    $result = mysqli_query($link,$sql);
    // var_dump($result);
    // while($row = mysqli_fetch_assoc($result)){
    //     print_r($row);
    //     echo "<br/>";
    // }
    // while($row1 = mysqli_fetch_row($result)){
    //     print_r($row1);
    //     echo "<br/>";
    // }
    while($row1 = mysqli_fetch_array($result)){
        print_r($row1);
```

```
        echo "<br/>";
    }
?>
```

文件相关信息

```
<?php
    date_default_timezone_set('PRC');
    $fileName = "file.php";
    $result1 = file_exists($fileName);
    $result2 = filemtime($fileName);
    $result3 = filesize($fileName);
    $result4 = basename($fileName);
    $result5 = realpath($fileName);

    var_dump($result1);
    var_dump(date('Y-m-d H:i:s',$result2));
    var_dump($result3);
    var_dump($result4);
    var_dump($result5);

    // D:\wamp64\www\0000project\file.php:12:boolean true
    // D:\wamp64\www\0000project\file.php:13:string '2020-03-31 21:49:14' (length=19)
    // D:\wamp64\www\0000project\file.php:14:int 421
    // D:\wamp64\www\0000project\file.php:15:string 'file.php' (length=8)
    // D:\wamp64\www\0000project\file.php:16:string 'D:\wamp64\www\0000project\file.php' (length=34)
?>
```

文件操作系统：

对文件

```
        fopen(filename,mode)    返回一个资源类型        stream
        fgetc、fgets、fread、file_get_contents
        readfile、file
        fwrite()
```

对目录

```
        opendir()       返回一个资源类型
        readdir()
        scandir()
```

遍历文件夹

http协议

客户端与服务器建立的一种通讯规则。规定了客户端与服务器端通讯的格式以及格式的含义。

请求：

http协议的请求，是由浏览器自己组织的
请求头：
请求报头
空行
数据

响应：
http协议的响应，是由服务器自己组织的
响应头：
响应报头：
空行
数组

header()用于设置http协议的响应头
header(‘content-type:text/html;charset=utf-8’)
header(‘location:url’);
header(‘content-type:application/octet-stream’);
header(‘content-disposition:attachment;filename=文件名’);
echo file_get_contents()

curl
主要用于数据采集，代码版的浏览器。
curl_init();

curl_setopt(ci,选项名,值);

curl_exec(ci)

会话技术
由于http协议是 无连接 无状态，所以http协议 无法 记住客户端的信息。为了弥补http协议这两点的“不足”，所以出现了会话技术。

设置cookie setcookie(name,value【,expire【,path【,domain【,secure【,httponly】】】】】);

setcookie(name,value,expire,path,domain,secure)
secure 取值为true或false
如果设置为true，那么只要当客户端使用的协议是https时，则会将cookie携带给服务器端。
https = http + ssl

setcookie(name,value,expire,path,domain,secure,httponly)
secure 取值为true或false
如果设置为true，那么只能由php访问cookie，js就不能访问了。

读取cookie
$_COOKIE 就是一个关联数组，主要是用于存储客户端通过cookie的技术，提交的数据。

删除cookie
方法1:，在设置一次cookie，将其有效期设置为过期的时间
方法2：将cookie的值设置为 ‘ ’

cookie值的类型
由于cookie的值最终是保存在客户端的文本文件中，由于文本文件中只能存储字符，所以cookie是不能存储数组。

session
session也是会话技术中的一种，session是以cookie为基础，将重要的数据保存在服务器端，同时将能够唯一标识这份数据的数据以cookie的保存客户端。

开启session session_start();

设置session session的操作就是向$_SESSION这个变量中写数据，或读数据。
设置session session的操作就是向$_SESSION这个变量中写数据，或读数据。

$_SESSION与session文件之间的交互的过程。
首次：
请求时，服务器首先在硬盘中创建一个唯一的文件，将文件名以cookie的形式返回客户端。

在php脚本执行结束时，将$_SESSION内存变量中的数据写入到session文件中。

其后各次：
请求时，服务器会收到客户端的cookie中保存的session文件名，同时会在服务器端硬盘中找到对应的session文件。
会打开这个session文件，将文件中的数据读取到$_SESSION变量中。
脚本执行结束时，会再将$_SESSION中的保存的数据写回session文件中。

session_id();　用于获取当前sessionid值
session数据　session可以存储任何类型的数据。但是$_SESSION变量本身的下标必须是字符串。
销毁session　session_destroy()
销毁session中的部分数据　unset($_SESSION[下标]);

创建画布
imagecreatetruecolor(w,h);
w　表示的是所要创建的画布宽
h　表示所要所要创建的画布高
如果创建成功返回的一个gd资源

分配颜色
imagecolorallocate(img,r,g,b);
img　画布资源(gd资源)
r,g,b　十进制的颜色表示方式
此函数仅是创建了一个颜色资源，但并没有使用。

填充颜色
imagefill(img,x,y,color);
img　画布资源
x,y　画布上的某个点
color　所要填充的颜色

绘制矩形
imagerectangle(img,x1,y1,x2,y2,color)
rectangle矩形
img　画布资源
x1,y1　左上角顶点坐标
x2,y2　右下角顶点坐标
color　所要绘制的矩形边线的颜色

```
$w = 500;
$h = 400;
$img = imagecreatetruecolor($w,$h);
$bg = imagecolorallocate($img,0,0,255);
imagefill($img,0,0,$bg);
$border = imagecolorallocate($img,0,255,0);
imagerectangle($img,50,30,200,250,$border);
header('content-type:image/jpeg');
imagejpeg($img);
```

绘制直线
imageline(img,x1,y1,x2,y2,color);
img　画布资源
x1,y1　所要绘制的直线的起点坐标
x2,y2　所要绘制的直线的终点坐标
color　直线的颜色

绘制直线
imageline(img,x1,y1,x2,y2,color);
img　画布资源
x1,y1　所要绘制的直线的起点坐标
x2,y2　所要绘制的直线的终点坐标
color　直线的颜色

绘制字母
imagestring(img,size,x,y,string,color);
img　画布资源
size　文字大小，0至5个等级

x,y 所要绘制内容的左上角坐标
string 所要绘制的内容
color 所要绘制的内容的颜色

绘制汉字

imagettftext(img,size,angle,x,y,color,font,text);
img 画布资源
size 大小，单位像素
angle 角度
x,y 所要绘制的内容的左下角坐标
color 所要绘制的内容的颜色
font 所要绘制的内容的字体
text 所要绘制的内容

输出画布

imagejpeg(img【,filename】)
imagepng(img【,filename】)
imagegif(img【,filename】);
说明：
img 画布资源
filename 表示是所要保存的文件名，可以省略，如果省略表示输出到浏览器
在输出到浏览器之前必须设置header(‘content-type:image/gif’)
在将画布输出到浏览器时，不要有任何多余的输出，如果无法显示图片，需要将header()注释掉才可以看到错误信息。

从图片创建画布

imagecreatefromjpeg(file)
imagecreatefrompng(file)
imagecreatefromgif(file)
file是所要读取的图片文件，
以上函数会使用file图片的宽高来创建一个画布，并将图片的原内容读取到画布中。
在由一个图创建一下画布时，什么样类型的图片文件就必须使用相应的创建函数。如何获取图片的格式。
getimagesize()函数

验证码