1. Php基础语法
2. 面向对象（记语法）（了解使用）
3. Mysql数据库（主要是数据库的操作，高级查询）
4. 核心扩展（功能）
5. Tp5.1框架，项目

<?php ?>php解析符 所有代码都要放在解析符中

Day2

Echo主要输出字符串，以及能够转化成字符串的数据类型

1. 变量：程序数据会保存在内存中，程序员需要一种机制类修改或访问内存中的数据，这种机制就是变量 例 $a=’包包’；
2. 在一个php解析符中不能出选两个相同的变量名，负责会出现覆盖
3. 命名只能以字母或下划线
4. 四种标量：boolean int float string
5. 标量：不可再分割的数据类型
6. 对于非字符串的类型的数据类型，使用echo输出，会自动转换成字符串
7. var\_dump();可以打印出数据类型和值
8. 每一个字母占三个字节
9. 声明字符串三种方式：1.‘’，2.“”3.$str3=<<<EOT<<< 通过herdoc创建的字符串

// $int=123;

// echo $int;

#数据类型的转换

// $bool=false;

// $newBoo=(int)$bool;

// echo $newBoo;

// echo "<br/>";

// $bool1=false;

// $newBoo1=intval($bool1);

// echo $newBoo1;

// echo "<br/>";

// $bool2=false;

// $three=settype($bool2,'int');

// var\_dump($three);

// $str1='123';

// $str2='123汉字';

// $str2='汉字';

// $str4='汉1字2';

// $newStr1=intval($str1);

// $newStr2=intval($str2);

// $newStr3=intval($str3);

// $newStr4=intval($str4);

// var\_dump($newStr1);

// echo "<br/>";

// var\_dump($newStr2);

// echo "<br/>";

// var\_dump($newStr3);

// echo "<br/>";

// var\_dump($newStr4);

// echo "<br/>";

// $a=123;

// $b=floatval($a);

// var\_dump($b);

// $c=123.123;

// $b=intval($c);

// var\_dump($c);

#传值

// $a=123;

// $b=$a;

// echo $a.'<br/>';

// echo $b.'<br/>';

// #引用赋值

// $a=123;

// $b=&$a;

// $b=456;

// echo $a.'<br/>';

// echo $b.'<br/>';

#可变变量

// $name='xixi';

// $$name='haha';

// echo $name.'<br/>';

// echo $$name.'<br/>';

// echo $xixi.'<br/>';

#isset（）用来判断变量是否被定义

// $age=666;

// $sex='boy';

// $a=isset($age);

// $b=isset($sex);

// var\_dump($a);

// var\_dump($b);

#运算符

// $a=4;

// $b=5;

#三元运算符

// $a=4;

// $b=5;

// echo true?$a:$b;

// echo ($a>$b)?$a:$b;ture的话执行$a false 执行#运算符$b

#复习

// $age=16;

// $pi=3.1415926;

// $bool=false;

// $bool2=true;

// $str3=<<<age

// 通过heredoc声明方式

// age;

// #类型转换

// $a=123;

// $newBoo=(float)$a;

// var\_dump($newBoo);

// $b=1234.12;

// $newBoo=intval($b);

// $c=123456.455;

// $d=settype($c,'int');

// var\_dump($d);

#insert判断变量是否被定义

// $a=123;

// $b-isset($a);

// var\_dump($b);

#三元运算符

#逻辑运算符

// $three=(true || true)?'我是正确的':'我是错误的';

// echo $three;

// $four=(true && true)?'正确的':'错误的';

#定义常量

// define('name','陈才');

// echo name;

#获取常量 同过constant获取常量

// $con1=constant('name');

// echo $con1;

#判断常量是否被定义

// $bool=define('name1');

// var\_dump('name');

# 流程控制

// echo "1".'<br/>';

// echo "2".'<br/>';

// echo "3".'<br/>';

// echo "4".'<br/>';

// if(1==='1'){

// echo "5";

// }

// echo "6".'<br/>';

// echo "7".'<br/>';

// echo "<pre>";

// switch(2){

// case 1:

// echo "1";

// break;

// case 2:

// echo "2";

// break;

// case 3:

// echo "3";

// break; //跳出语句

// }

// echo "6".'<br/>';

// echo "7".'<br/>';

#

// while(true){

// echo "陈才最帅";

// }

#错误也会执行一次

// do{

// echo "1";

// }(false)

// while(false){

// echo "1".'<br/>';

// }

#for

// for($a=1;$a<3;$a++){

// echo "1".'<br/>';

// }

// $a=1;

// for($a;$a<10;$a++){

// echo "1".'<br/>';

// }

// $a=1;

// if($a==2){

// echo "2";

// }else if($a==3){

// echo "default";

// }

#

// $score=88;

// switch($score){

// case $score<60:

// echo "不及格";

// break;

// case $score>60 && $score<90;

// echo "优秀";

// break;

// }

// for($a=0;$a<9;$a++){

// for($i=0;$i<$a;$i++){

// echo '$a\*$i''='$a\*$i.'<br/>';

// }

// }

#作业1

define('PI',3.14);

$area=PI\*10\*10;

echo $area;

echo "<br/>";

$area1=(20+10)\*10/2;

echo $area1;

echo "<br/>";

if($area>$area1){

echo $area;

}else{

echo $area1;

}

echo "<pre>";

#作业2

// for($a=0;$a<=100;$a++){

// if($a/3==0){

// echo $a;

// }else{

// echo "";

// }

// }

#作业3

// for ($i=0; $i <= 100 ; $i++) {

// if(1\*$i=$i){

// echo $i;

// }else

// {

// echo ;

// }

// }

#

<?php

// switch('3月'){

// case'1月':

// echo "1月".'<br/>';

// case '2月':

// echo "2月".'<br/>';

// case '3月':

// echo "3月".'<br/>';

// }

//

// for($a= 0;$a<=9;$a++){

// echo "$a".'<br/>';

// }

// for($a=1;$a<9;$a++){

// for($i=1;$i<$a;$i++){

// echo $a.'\*'.$i.'='.$a\*$i.'-----';

// }

// echo '<br/>';

// }echo '<br/>';

// $j=0;

// $j++;

// echo $j;

// $j=0;

// for ($i=100; $i >=1 ; $i=$i/2) {

// $j++;

// if($j==6){

// echo '$i';

// }

// }

// define('chencai', '20');

// echo chencai;

// $i=5;

// $sum=$i++ + ++$i;

// echo $sum;

// $num=1;

// function text($num){

// echo $num+3;

// }

// text($num);

/\*\*

\*

\*/

// class ClassName extends AnotherClass

// {

// function \_\_construct(argument)

// {

// # code...

// }

// }

// $a=1;

// if($a=='1'){

// echo "我是1";

// }

// if($a=='1'):

// echo "我是1";

// endif;

// $a=1;

// switch ($a) {

// case "1":

// echo "1";

// break;

// default:

// echo "0";

// break;

// }

// 接抽象类中写抽象必须加abstract

// 接口中不能加abstract

// 继承是没有多继承的

// 接口中不能写普通方法

// 抽象abstract

// file\_put\_contents将文件传入

// file-get\_contents获取文件

// json\_encodejson编码

// 求面积

// define('PI',3.14);

// function getArea($r){

// $area=PI\*$r\*$r;

// echo "$area";

// }

// getArea(10);

// function getArea($w,$h){

// $area=$w\*$h;

// echo "$area";

// }

// getArea(10,10);

// 求面积

// define('PI', 3.14);

// function getArea($r){

// $area=PI\*$r\*$r;

// echo "$area";

// }

// getArea(10);

// function run(){

// echo "1";

// }

// run();

// 无参有返

// function run(){

// return

// }

// function run(){

// return 3;

// }

// $num=run();

// echo "$num";

// function run($age){ //形参

// $num=1+$age;

// echo "$num";

// }

// run(1);//实参

// 有参有返

// function run($arg1,$arg2){

// $sum=100+$arg1+$arg2;

// return "$sum";

// }

// $sum=run(10,20);

// echo "$sum";

// define('PI', 3.14);

// function getArea($r){

// $area=PI\*$r\*$r;

// echo $area;

// }

// getArea(10);

// function test($age){

// $num=$age+20;

// echo $num;

// }

// test(20);

// $age=20;

// function test(){

// global $age;

// echo $age;

// }

// test();

// $func=function(){

// echo "匿名函数";

// };

// $func();

// function name(){

// return 'chencai';

// }

// $name='name';

// function getName(){

// }

// $age=18;

// function test(){

// global $age;

// echo $age;

// }

// test();

// $arr=array('name'=>'赵云','skill'=>array('七进七出','八进气出'),'feiends'=>array('刘备','关羽'));

// echo $arr['skill'];

// function getMax($num1,$num2){

// return ($num1>$num2)?$num1:$num2;

// }

// $a=4;

// $b=9;

// echo getMax($a,$b);

// $dal=date('y-m-d');

// echo $dal;

// function getMax($num1,$num2){

// return ($num1>$num2)?$num1:$num2;

// }

// $a=3;

// $b=9;

// echo getMax($a,$b);

// $dal=date('y-m-d');

// echo $dal;

// date\_default\_timezone\_set('PRC');

// $dal=date('y-m-d--h:i:s');

// echo $dal;

// $date\_m=date('y-m-d--h:i:s',$time\_m);

// echo $date\_m;

// $time\_m=strtotime('next year');

// $date\_m=date('y-m-d--h:i:s',$time\_m);

// echo $date\_m;

// $int\_1=123456;

// $str\_1=strval($int\_1);

// var\_dump($str\_1);

// $str1='qwertyuiopqwereqqwerr';

// echo str\_replace('q','a',$str1);

// $str1='qwertyuiiop';

// echo strlen($str1);

// $str1=array('q','w','w','e');

// $str\_im=implode('-',$str1);

// echo $str\_im;

// echo rand(100,999);

// function test($name,$age){

// echo '小虎的名字:'.$name.'----'.'小虎的岁数：'.$age.'<br/>';

// }

// test('小虎',18);

// class person{

// public $name='小K';

// public function study(){

// echo '好好学习，天天向上';

// }

// }

// $xk=new person;

// echo $xk->name;

// $just=new person;

// echo $just->study();

// echo "<br/>";

// var\_dump($xk);

// echo "<br/>";

// var\_dump('$just');

// class person{

// public $name='aa';

// protected $age=18;

// private $sex='未知';

// public function study(){

// echo "好好学习，天天向上";

// }

// protected function like(){

// echo "喜欢你";

// }

// private function savemoney(){

// echo "私房钱";

// }

// public function getMassage(){

// // $this->savemoney();

// // echo $this->sex;

// return $this->name;

// }

// }

// $xk=new person;

// echo $xk->name;

// echo $xk-> getMassage();

// $wanglong=new person;

// echo $wanglong->name;

// echo $wanglong->age;

// echo $wanglong->sex;

// $ww=new person;

// echo $ww->study();

// public person{

// public $name;

// }

// public function \_\_construct($name,$value){

// $this->$name=$value;

// }

// $xk=new person('name','老黄');

// echo $xk->name;

// class person{

// public $name='老黄';

// private $sex='未知';

// public function \_\_construct(){

// echo "开始".'<br/>';

// }

// public function study(){

// echo "学习".'<br/>';

// // }

// public function \_\_destruct(){

// echo "结束".'<br/>';

// }

// }

// $xk=new person('name','老黄');

// echo $xk->name;

// class person{

// const NAME='等待'.'<br/>';

// public $name1='老黄'.'<br/>';

// public static $name2='zb'.'<br/>';

// }

// $lh=new person;

// echo $lh->name1;

// echo '<br/>';

// echo person::NAME;

// echo person::$name2;

// class login{

// private $user='chencai';

// private $password='123456';

// public function login1($name,$pass){

// if($name==$this->user && $pass==$this->password){

// echo "登陆成功";

// }else{

// echo '登录失败';

// }

// }

// }

// $cc=new login;

// $cc->login1('chencai','123456');

// class father{

// public $name='火影';

// protected $user='chencai';

// private $password='123456';

// public function get1(){

// return $this->user;

// }

// public function get2(){

// return $this->password;

// }

// }

// class child extends father{

// public function getusers(){

// return $this->password;

// }

// }

// $chencai1=new child;

// echo $chencai1->name;

// echo $chencai1->get1();

// echo $chencai1->get2();

// echo $chencai1->passwoed();

// echo $chencai1->user;

// echo $chencai1::user;

// class XH{

// public function study(){

// echo "好好学习，天天向上";

// }

// }

// $xk=new XH;

// $xk->study();

// var\_dump($xk);

// echo "<br/>";

// $cc=clone $xk;

// $cc->study();

// var\_dump($cc);

// abstract class person{

// //

// abstract public function study();

// // public function study(){

// // echo "好好学习，天天向上";

// // }

// }

// class cc extends person{

// public function study(){

// echo "好好学习，天天向上";

// }

// }

// $qwe=new cc;

// $qwe->study();

// interface person{

// public function study();

// public function play();

// }

// class cc implements person{

// public function study(){

// echo "好好学习，天天向上" ;

// }

// public function play(){

// echo "态度决定一切";

// }

// }

// $lh=new cc;

// $lh->study();

// $lh->play();

// $arr=array(1,2,'等待','涛声依旧');

// $str=serialize($arr);

// var\_dump($str);

// class Man{

// public $name='xk';

// public $age='16';

// public $sex='boy';

// public function study(){

// echo "好好学习，天天向上";

// }

// public function \_\_sleep(){

// return array('name','sex');

// }

// // public function \_\_wakeup(){

// // $this->sex=gilr;

// // }

// }

// $cc=new Man;

// $str=serialize($cc);

// var\_dump($str);

// class people{

// public $name='陈才';

// }

// $wanglong=new people;

// echo $wanglong->name;

// $return=serialize($wanglong);

// // echo $wanglong;

// echo "<br/>";

// var\_dump($return);

// echo "<br/>";

// $return1=unserialize($return);

// var\_dump($return1);

// namespace name1;

// class people{

// function \_\_construct(){

// echo "我是name1命名空间的people";

// }

// function getmessage(){

// echo "获取信息";

// }

// }

// $cc=new people;

// $cc->getmessage();

// function study(){

// echo "好好学习，天天向上";

// }

// study();

// namespace LOL\AONY\ZHHSG;

// echo \_\_NAMESPACE\_\_;

// echo "<hr/>";

// const name='晨h'

// echo \lol\cehncai\wanglong\name;

// namespace AONY{

// class Man{

// function \_\_construct(){

// echo "我是爱奥尼亚中的构造方法".'<br/>';

// }

// }

// echo \_\_NAMESPACE\_\_;

// echo "<hr/>";

// // echo \_\_construct();

// }

// namespace PCJB{

// const NAME='亚索'.'<br/>';

// class LOL{

// function \_\_construct(){

// echo "构造方法".'<br/ >';

// }

// }

// class Man{

// function \_\_construct\_\_(){

// echo "我是皮城警备的构造方法";

// }

// }

// function skill(){

// echo "绝招";

// }

// echo \_\_NAMESPACE\_\_;

// }

// // echo Man;

// namespace{

// use const PCJB\NAME.'<br/>';

// echo NAME.'<br/>';

// }

// function study(array $name){

// // echo $name.'好好学习，天天向上';

// var\_dump($name);

// }

// study([1,2,3]);

// class person{

// function \_\_construct(){

// echo "构造函数";

// }

// }

// $p=new person;

// $num=1;

// function test($num){

// echo $num+10;

// }

// test($num);

// namespace cc

// const age=16;

// echo age ;

// $arr=array(1,2,3);

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

// echo '键'.$key;

// echo "<hr/>";

// echo '值：'.$value;

// }

// setcookie('name','hello');

// var\_dump($\_COOKIE);

// $type='mysql';

// $host='localhost';

// $dbname='dbname';

// $user='root';

// $password='root';

// $dsn="$type:host=$host;dbname=$dbname";

// $pdo=new PDO($dsn,$user,$password);

// var\_dump($pdo);

// $sql="create table chencai20(

// id int,

// name varchar(20)

// )";

// // $res=$pdo->exec($sql);//执行一条sql语句

// // var\_dump($res);

// $sql="insert into chencai20(id,name) values(1,'2'),(3,'3')";

// $res=$pdo->exec($sql);//执行一条sql语句

// echo ($res);

// $pdo = new PDO('mysql:host=localhost;dbname=test','root','root');

// $pdo->exec('set names utf8');

// $sql = "insert into test(name) values('张三'),('李四')";

// $res = $pdo->exec($sql);

// var\_dump('$res');

// $sql="select \* from tbname3";

// $statement=$pdo->query($sql);

// $results=$statement->fetch();

// var\_dump($results);

// $sql="select \* from tbname3";

// $statement=$pdo->query($sql);

// $results=$statement->fetch();

// var\_dump($results);

// $sql="insert into tbname3 values(?,'potter')";

// $statement=$pdo->prepare($sql);

// $arr=array(10);

// $statement->execute($arr);

// var\_dump($statement);

// $sql="select \*from tbname3 where id=:id";

// $statement=$pdo->prepare($sql);

// $id=1;

// $statement->bindParam(':id',$id);

// // $statement->bindParam(':name',$id);

// // $statement->bindParam(':id',$id);

// $results=$statement->execute();

// var\_dump($results);

// echo "<pre>";

// $res=$statement->fetch(PDO::FETCH\_ASSOC);

// var\_dump($res);

// foreach ($res as $cols => $rows) {

// var\_dump($rows);

// }

// $type='mysql';

// $host='localhost';

// $dbname='dbname';

// $user='root';

// $root='root';

// $dsn="$type:host=$host;dbname=$dbname";

// $pdo=new PDO($dsn,$user,$root);

// // 事务

// $pdo->beginTransAction();

// $sql="insert into tbname3 values(2,2)";

// $res=$pdo->exec($sql);

// var\_dump($res );

// function add($one,$two){

// if($two==0){

// $e=new Exception($message='被除数不能为零');

// throw $e;

// }else{

// $results=$one/$two;

// echo $results;

// }

// }

// add(1,0);

// try{

// }cat ch(PDOException $e){

// }

// $link=mysql\_connect('localhost','root','root');连接数据库

// $sql='set names utf8';

// mysql\_query($sql);设置字符集

// $sql='use tbname3';选择数据库

// mysql\_query($sql);

// $sql='select \* from tbname3';

// $res=mysql\_query($sql);

// $results=mysql\_feach\_assoc($res);

// var\_dump($results);

// mysql只能使用的面向过程的，没有面向对象

// 现在废弃

// musqli及使用面向对象也使用面向对象过程

// pdo面向对象没有过程

// pdo可以跨数据库

// // 增

// setcookie('name','hello');#当浏览器页面执行王城才会存入值不能是符合类型

// // 改

// setcookie('name','world');#覆盖同名

// // 删

// setcookie('name','');#第一种

// setcookie('name','world',time());#第二种数据自动消失 添加时间

// // 查

// echo $\_COOKIE('name');

// var\_dump($\_COOKIE);

// #必须的第一步开启session

// session\_start();

// 增

// $\_SESSION['name']='harry';#键=>值

// #session不能小写因为session打印的是数组

// 改

// $\_SESSION['name']='world';

// 删

// unset($\_SESSION['name']);

// session\_destroy();#销毁session文件

// 查

// echo $\_SESSION['name'];

// var\_dump($\_SESSION);

// sesion\_start();

// if(!isset($\_SESSION['car'])){

// $\_SESSION['car']=array();

// }

// $goods=$\_REQUEST['car'];

// $\_SESSION['test']=$goods;

// 安装reids

// reads

// 非关系型数据库nosql，操作在内存上进行，比关系型数据库快

// 数据都是在内存上操作，如果数据超过内存会产生内存溢出，数据相对关系型数据库不安全

// 获取redis扩展

// $redis=new Redis();

// $redis->connect('127.0.0.1,6379');

// var\_dump($redis);

// #字符串类型

// #增

// $redis->set('age',6);

// #批量增加

// $arr=array('age=>6','name'=>'hello');

// $redis->mSet($arr);

// #查

// echo $redis->get('age');

// #批量查询

// var\_dump($redis->mGet(array('age','name')));

// #改

// echo $redis->get('age');

// $redis->set('age',7) ;

// echo $redis->get('age');

// #删

// $redis->delete('age');

// echo $redis->get('age');

// #删除多个

// $redis->delete('name','age');

// $redis->get('name');

// $redis->get('age');

// hash类型

// #增

// $redis->hset('db','key1','vall');

// #整个像一个数组名键值

// $arr1=array('set'=>1,2,30);

// redis->hMset('db',$arr1);

// #查

// echo $redis->hGet('db','key1');

// #删

// $redis->hDel('db','key1');

// list

// #增 l表示left左查

// $redis->lPush('list1','lVal');

// $redis->rPush('list1','lVal');

// $redis->lInsert('list1',redis::before,';lVall','test');

// #查

// echo $redis->lPop('list1');

// echo $redis->rPop('list1');

// echo $redis->lGet('list1',0);-1 表示尾部0表示头部

// $redis->lrange('list',0,-1);全部

// #删

// $redis->lrem('list','lVal',2);

// var\_dump($redis->;lrage('list1'0,-1));

// set

// #增

// $redis->sAdd('set1','sVall');

// #查

// var\_dump($redis->sMembers('set1');

// echo $redis->sCard('set1'); #查找个数

// #删除

// $redis->sRem('set1','sVall');

// echo $redis->SCaed('set1');

// $redis->sMove('set1','set2','sVal1');

// #移动到另一个set中

// $var\_int=1;

// $var\_float=1.23;

// $var\_str='hello';

// define('name','hello');

// const AGE=16;

// echo name,AGE;

// echo \_FILE\_精确到文件

// echo —DIR\_精确到文件夹

// if(true && false){

// echo "true";

// }else{

// echo "false";

// }

// if(true || false){

// echo "true";

// }else{

// echo "false";

// }

// $i=0;

// $a=0;

// ++$i + $i++ + $i++

// $a=0;

// while($a<10){

// echo $a;

// $a++;

// }

// if(1){

// echo "";

// }else if{

// }else{

// }

// switch(1){

// case '':

// # code...

// break;

// }

// for($i=0,$i<10,$i++){

// }

// $a=10;

// do{

// echo "1";

// }while($a<10);

// $name='c++';

// function test($name){

// $res=我喜欢'.$name;

// return $res;

// }

// test(''php);

// class Person{

// public $name='普通方法';

// public static $name\_s='静态方法';

// public function study(){

// echo "好好学习，天天向上";

// public static function study1(){

// echo "静态学习";

// }

// }

// }

// trait Man{

// public function study(){

// }

// }

// echo 'helloWorld';

#变量

// $a='包包';

// $apple='苹果';

// $\_apple='苹果';

// $apple='苹郭';

// echo $apple;

// echo "<br/>";

// echo $\_apple;

// $bool=ture;

// $int=123;

// $float=1213.2;

// $str='我是字符串';

// echo $bool;

// echo '<br/>';

// echo $int;

// echo '<br/>';

// echo $float;

// echo '<br/>';

// echo $str;

// echo '<br/>';

#var\_dump();可以打印出类型和值

// $sum=(0.7+0.8)\*10;

// echo $sum;

// $str1='字符串';

// $str2='字符串';

// $str3=<<<WAVN字符串

// WAVN;

?>

<?php

// $int=123;

// echo $int;

#数据类型的转换

// $bool=false;

// $newBoo=(int)$bool;

// echo $newBoo;

// echo "<br/>";

// $bool1=false;

// $newBoo1=intval($bool1);

// echo $newBoo1;

// echo "<br/>";

// $bool2=false;

// $three=settype($bool2,'int');

// var\_dump($three);

// $str1='123';

// $str2='123汉字';

// $str2='汉字';

// $str4='汉1字2';

// $newStr1=intval($str1);

// $newStr2=intval($str2);

// $newStr3=intval($str3);

// $newStr4=intval($str4);

// var\_dump($newStr1);

// echo "<br/>";

// var\_dump($newStr2);

// echo "<br/>";

// var\_dump($newStr3);

// echo "<br/>";

// var\_dump($newStr4);

// echo "<br/>";

// $a=123;

// $b=floatval($a);

// var\_dump($b);

// $c=123.123;

// $b=intval($c);

// var\_dump($c);

#传值

// $a=123;

// $b=$a;

// echo $a.'<br/>';

// echo $b.'<br/>';

// #引用赋值

// $a=123;

// $b=&$a;

// $b=456;

// echo $a.'<br/>';

// echo $b.'<br/>';

#可变变量

// $name='xixi';

// $$name='haha';

// echo $name.'<br/>';

// echo $$name.'<br/>';

// echo $xixi.'<br/>';

#isset（）用来判断变量是否被定义

// $age=666;

// $sex='boy';

// $a=isset($age);

// $b=isset($sex);

// var\_dump($a);

// var\_dump($b);

#运算符

// $a=4;

// $b=5;

#三元运算符

// $a=4;

// $b=5;

// echo true?$a:$b;

// echo ($a>$b)?$a:$b;ture的话执行$a false 执行#运算符$b

#复习

// $age=16;

// $pi=3.1415926;

// $bool=false;

// $bool2=true;

// $str3=<<<age

// 通过heredoc声明方式

// age;

// #类型转换

// $a=123;

// $newBoo=(float)$a;

// var\_dump($newBoo);

// $b=1234.12;

// $newBoo=intval($b);

// $c=123456.455;

// $d=settype($c,'int');

// var\_dump($d);

#insert判断变量是否被定义

// $a=123;

// $b-isset($a);

// var\_dump($b);

#三元运算符

#逻辑运算符

// $three=(true || true)?'我是正确的':'我是错误的';

// echo $three;

// $four=(true && true)?'正确的':'错误的';

#定义常量

// define('name','陈才');

// echo name;

#获取常量 同过constant获取常量

// $con1=constant('name');

// echo $con1;

#判断常量是否被定义

// $bool=define('name1');

// var\_dump('name');

# 流程控制

// echo "1".'<br/>';

// echo "2".'<br/>';

// echo "3".'<br/>';

// echo "4".'<br/>';

// if(1==='1'){

// echo "5";

// }

// echo "6".'<br/>';

// echo "7".'<br/>';

// echo "<pre>";

// switch(2){

// case 1:

// echo "1";

// break;

// case 2:

// echo "2";

// break;

// case 3:

// echo "3";

// break; //跳出语句

// }

// echo "6".'<br/>';

// echo "7".'<br/>';

#

// while(true){

// echo "陈才最帅";

// }

#错误也会执行一次

// do{

// echo "1";

// }(false)

// while(false){

// echo "1".'<br/>';

// }

#for

// for($a=1;$a<3;$a++){

// echo "1".'<br/>';

// }

// $a=1;

// for($a;$a<10;$a++){

// echo "1".'<br/>';

// }

// $a=1;

// if($a==2){

// echo "2";

// }else if($a==3){

// echo "default";

// }

#

// $score=88;

// switch($score){

// case $score<60:

// echo "不及格";

// break;

// case $score>60 && $score<90;

// echo "优秀";

// break;

// }

// for($a=0;$a<9;$a++){

// for($i=0;$i<$a;$i++){

// echo '$a\*$i''='$a\*$i.'<br/>';

// }

// }

#作业1

// define('PI',3.14);

// $area=PI\*10\*10;

// echo $area;

// echo "<br/>";

// $area1=(20+10)\*10/2;

// echo $area1;

// echo "<br/>";

// if($area>$area1){

// echo $area;

// }else{

// echo $area1;

// }

// echo "<pre>";

#作业2

// for($a=0;$a<=100;$a++){

// if($a/3==0){

// echo $a;

// }else{

// echo "";

// }

// }

#作业3

// for ($i=0; $i <= 100 ; $i++) {

// if(1\*$i=$i){

// echo $i;

// }else

// {

// echo ;

// }

// }

// $a=1;

// if($a=='1'){

// echo "我是一";

// }

#为什么需要函数

#什么是GD库：输出验证码

#解析图片文件

// header("content-type:image/png");

// #创建画布

// $image=imagecreatetruecolor(100, 100);

// // var\_dump($image);

// #创建颜色

// $red=imagecolorallocate(image,255,0,0);

// #水平画出一个字符

// imagechar($image,15,0,0,'h',$red);

// #水平滑出一个字符串

// imagestring($image,15,12,0,'hello',$red);

// #垂直画出一个字符

// imagecharup($image,15,50,50,'w',$red);

// #垂直画出字符串

// imagestringup($image,15,50,37,'orld',$red);

// #背景颜色填充

// imagefill($image,0,0,bg);

// #添加字体

// imagefttext($image,15,0,50,50,$red,\_DIR\_."/stxingka.ttf",'你好');

// #加入干扰元素

// for($i=0;$i<20;$i++){

// imagesetpixel(￥image,15,15,$red);

// }

// #画直线

// imageline($image,0,50,50,50,$red);

// #画弧线

// $imagearc($image,70,,70,10,20,0,90,$red);

#产生随机数

// $a=rand(1,9);

// for () {

// # code...

// }

// <!DOCTYPE html>

// <html>

// <head>

// <title></title>

// </head>

// <body>

// 欢迎登陆

// <from action="index.php" method="post">

// <input id='yz' type="text" name="yzm">

// </from>

// </body>

// </html>

// #在浏览器显示图片

// imagepng($image);

// # if switch

// switch('3'){

// case '1':

// echo "一月";

// break;

// case '2':

// echo "2月";

// break;

// case '3月':

// echo '3月';

// break;

// default:

// echo "默认的";

// }

// $a='1月';

// if($a=='1月'){

// echo '1月';

// }else{

// echo '默认的';

// }

// $a=1;

// if($a=='1'){

// echo "我是1";

// }

// if($a==1):

// echo "我是1";

// endif;

// $a==1;

// switch($a){

// case 1:

// echo "1";

// break;

// case 2:

// echo "2";

// break;

// default:

// echo "没有";

// break;

// }

// define()

// function getArea(){

// }

// getArea()

// define('PI',3.14);

// function getArea($r){

// $area=PI\*$r\*$r;

// echo "$area";

// }

// getArea(10);

// echo "1";

// function run(){

// echo "2";

// }

// // run();

// echo "3";

// run();

#把run放到上面就执行123

#放到下面就执行132

#无参无返

// function run(){

// echo "1";

// }

// run();

// #无参有返

// function run(){

// return 3;

// }

// $num=run();

// echo $num;

// #有参无返

// function run($arg,$arg1){//形参

// $sum=1+$arg+$arg1;

// echo $sum;

// }

// run(3,2);//实参，有实际意义

// #行参有几个实参就必须有几个

// #有参有返

// function run($arg){

// $sum=1+$arg;

// return $sum;

// }

// $num= run(1);

// echo $num;

// #运行的结果还可以进行其他操作

// echo run(3);

// function test(){

// $age=16;

// // echo $age;

// return $age;

// }

// // test();

// echo test();

#局部变量只能在函数本身中使用

#在外部使用必须使用return返回

#全局变量不能在函数中直接使用

#

// 1.通过传参可以获取

// 2.通过global可以将全局变量在局部变量中使用

// 3. $age=$GLOBAL['age'];

#全局变量

// $age=1;

// $sum=18;

// function test(){

// global $age,$sum;

// # $age=$GLOBALS['age'];

// #一般家引号就不加$ $就不加引号

// echo $age;

// echo $sum;

// }

// test();

// #可变函数

// function eat(){

// echo "吃东西";

// }

// eat();

// $getEat='eat';

// $getEat();

// #匿名函数

// $func=function(){echo "匿名函数";};

// #必须加一个分号表示这句话写完

// $func();

#回调函数

// function name(){

// return 'chencai';

// }

// $name='name';

// function getName($name){

// $getName=call\_user\_func($name);

// echo "姓名：".$getName;

// }

// getName($name);

// #递归函数

// for($i=1;$i<7;$i++){

// echo ;

// }

// function getI(){

// static $i=1;//静态变量被static关键字修饰的变量

// #只会被初始化一次

// $i++;

// if($i<6){

// echo $i;

// getI();

// }

// }

// getI();

#导入

#include

// require 'index1.php';

#repuire

#获取外层文件，往前跳../

#函数复习

// function test($r){

// $sum=$r+1;

// return $sum;

// // echo "函数的功能";

// }

// // $a=1;

// $get=test(12);

// echo $get;

#数组：一系列数据的集合

// $xk=88;

// $wave=99;

// $cc=100;

// $class=array(88,99,100);

#第一种

// $arr[]=1;

// $arr[]=2;

// echo "<pre>";

// var\_dump($arr);

#第二种\*

// $arr=array(1,2,3);

// echo "<pre>";

// var\_dump($arr);

// #<pre>格式化

// #第三种

// $arr=[1,2,3,4];

// echo "<pre>";

// var\_dump($arr);

// $arr=array(1,'cc',1.23,false);

// echo "<pre>";

// var\_dump($arr);

#索引数组

// $arr=array(1,2,3);

// var\_dump($arr);

// //数组的键默认从0开始

// //认为改变键

// $arr=array(1=>1,2=>2,3=>3);

// var\_dump($arr);

// $arr=array(1,2,1=>3);

// var\_dump($arr);

#如果认为改变键那么后面的键会根据前面的键增加

#关联数组

// $arr=array('h'=>1,'e'=>2,'1'=>3);

// var\_dump($arr);

// echo "<pre>";

// print\_r($arr);

// $arr=array(1,2,3,4);

// var\_dump($arr);

// echo "<br/>";

// var\_dump($arr[2]);

// echo "<br/>";

// echo ($arr[2]);;

// echo "<br/>";

#单独获取之后不再是数组，而是键所对赢得存储数据类型

#数组的遍历

// $arr=array(1,2,3,array(4,5,6),array(7,8,9));

// var\_dump($arr);

// $arr=array('name'=>'赵云','skill'=>array('七进七出','千军万马'),'firend'='');

// // var\_dump($arr);

// // echo arr['name'];

// //$arr['skill'];

// $arr1=$arr['skill'];

// echo $arr1[0];

//遍历

#1.索引键必须有规律

#2.通过count只能获取数组的个数，不获取下表

#3.不能输出有关联的数组

// $arr=array(1,2,3,4,5,6);

// // $con=count($arr);

// // for($i=0;$i<$con;$i++){

// // echo $arr[$i].'<br/>';

// // }

// #二维数组的遍历

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

// // foreach ($value as $k => $v) {

// // echo $v;

// // }

// // // echo $key.'<br/>';

// // // echo $value.'<br/>';

// // }

// $arr[]=1;

// $arr[]=2;

// $arr=array(1,2,3);

// $arr=[1,2,3];

#索引数组

// function getMax($num1,$num2){

// return ($num1>$num2)?$num1:$num2;

// }

// $a=1;

// $b=2;

// echo getMax($a,$b);

// max();

#时间函数

// date\_default\_timezone\_set('PRC');

// $dal=date('y-m-d--h:i:s');

// echo $dal;

// $time\_mmktime(1,2,3,4,5,6);

// echo $time\_m;

// echo $timestrtotime();

// $int1=456;

// $str\_i=strval($int1);

// var\_dump($str\_i);

#查找字符串位置

// $str1='qwertyuiopq';

// echo strpos($str1,'q');

// // var\_dump(strpos);

// // echo srepos;

// #查找字符串首次出现的位置

// echo strstr($str1,'t');

// #从出现字母的位置开始出现的字母

#替换字符串

// echo str\_replace('q','a',$str1);

// $str1=' qwer tyuiopq';

#获取字符串长度

// echo strlen($str1);

#去除空格

// var\_dump($str1);

// echo "<br/>";

// $str\_t=trim($str1);

// var\_dump($str\_t);

# 连接字符串implode

// $str2=array('d','d','d');

// $str\_im=implode('\_',$str2);

// var\_dump($str\_im);

// echo "<br>";

// #大小写转换

// $str1=' qwer tyuiopq';

// echo strtoupper($str1);

// echo str3;

#数学函数

#将其他类型转化为整形

// $str='asdf';

// $int\_i=intval($str);

// echo $int\_i;

// echo '<br/>'

// #求绝对值

// $int=-123;

// echo abs($int);

// #进一法取整

// ceil();

// #四舍五入

// round();

#产生随机数

// echo rand(1,9);

// #产生一个更好的随机数

// mt\_rand();

#格式化数字

$int1=147852963.369852147;

echo number\_format($int1,2,'/','-');

echo '<br/>';

echo intval($int1);

echo '<br/>';

#count();数组元素的个数

$arr=array(1,2,3);

echo count($arr);

echo "<br/>";

#回调函数，过滤不符合的元素

function test(){

return ;

}

$arr\_f=array\_filter($arr,'test');

var\_dump($arr\_f);

#将数组里的数当作test当中的而参数使用

echo "<br/>";

#交换字符串的键和值

$arr\_af=array\_flip($arr);

var\_dump($arr\_af);

#扩展和补充

#引用传值

function test1($b){

echo "$b";

}

$a=1;

test1($a);

echo "<br/>";

echo $a;

function test2($name,$age){

echo '小虎的名字：'.$name.'-----'.'小虎的年龄：'.$age.'<br/>';

}

test2('陈才',16);

echo '<br/>';

// class person{

// public $name='小k';#public访问修饰符，属性不加不会默认为public，protected，private

// function study(){#不加默认public，方法

// echo "好好学习";

// }

// // public function getMessage(){

// // echo $this->name;

// // }

// }

#通过$this可以调用protected和private

#实例化就是创建对象的过程，需要通过new来实现

// $xk=new person;

// #获取类中的元素，。通过对象名-》来实现，但是不能加$ 否则报错

// // echo $xk->name;

// // echo "$xk";

// echo "<br/>";

// #获取方法

// $just=new person;

// $just->study();

// echo "<br/";

// class child extends Person{

// }

// $a=new child;

// echo $a->name;

// class Person{

// public $name='小k';

// protected $age=16;

// private $sex='未知';

// public function study(){

// echo "学习";

// }

// public function like(){

// echo "喜欢你";

// }

// public function saveMoney(){

// echo "钱";

// }

// public function getMessage(){

// return $this->age;

// }

// }

// $xk=new Person;

// echo $xk->name;

// echo "<br/>";

// echo $xk->getMessage();

// echo "<br/>";

#魔术方法

#1.构造方法

// class Person{

// public $name='小k';

// public $age;

// private $sex='未知';

// #实例化对象是被调用

// public function \_\_construct($name,$value){

// // echo "我是构造方法";

// $this->$name=$value;

// }

// public function study(){

// echo "学习";

// }

// public function like(){

// echo "喜欢你";

// }

// public function saveMoney(){

// echo "钱";

// }

// public function getMessage(){

// return $this->age;

// }

// }

// echo "<br/>";

// $xk=new Person('name','老黄');

// // echo $xk->name;

// // echo "<br/>";

// // echo $xk->getMessage();

// // echo "<br/>";

// echo $xk->name;

#析构函数

// class Person{

// public $name='小k';

// public $age;

// private $sex='未知';

// #实例化对象是被调用

// public function \_\_construct($name,$value){

// // echo "我是构造方法";

// // $this->$name=$value;

// echo "开始".'<br/>';

// }

// #对像的生命结束自动调用

// public function \_\_destruct(){

// // echo "学习".'<br/>';

// unset($this->name);

// }

// public function study(){

// echo "结束";

// }

// public function like(){

// echo "喜欢你";

// }

// public function saveMoney(){

// echo "钱";

// }

// public function getMessage(){

// return $this->age;

// }

// }

// echo "<br/>";

// $xk=new Person('name','老黄');

// // echo $xk->name;

// // echo "<br/>";

// // echo $xk->getMessage();

// // echo "<br/>";

// // echo $xk->study();

// // unset($xk);#销毁变量

// $xk->study();

// class Person{

// public $name='老黄';

// private $sex='未知';

// #当我们在类外面掉用不可方法的属性是就用到这个方发

// #\_\_get只能掉用不可访问的属性

// public function \_\_get($sex){

// return $this->$sex;

// }

// private function study($name1){

// echo "学习".'<br/>';

// }

// #调用不可见的方法

// public function \_\_call($name,$arg){

// // $this->study();

// echo "调用了";

// }

// }

// $xk=new Person;

// echo $xk->sex;

// echo '<br/>';

// $lh=new Person;

// echo $lh->study($name1);

// // set

// class Person{

// public $xingming='老黄';

// public $sex='未知';

// private $age='26';

// // public function \_\_get(){

// // return $this->$name;

// // }

// public function \_\_set($name,$values){

// // return $this->$name;

// echo "被改制调用";

// }

// }

// $xk=new Person();

// echo $xk->sex='boy'.'<br/>';

// // echo $xk->xingming;

// $lh=new Person();

// echo $lh->xingming;

// class Person{

// public $xingming='老黄';

// public $sex='未知';

// private $age='26';

// // public function \_\_get(){

// // return $this->$name;

// // }

// public function \_\_isset($name){

// echo "判断被调用";

// }

// }

// $xk=new Person();

// // $lh=$xk->xingming.'<br/>';

// // var\_dump(isset($lh));

// // // echo $xk->xingming;

// // // $lh=new Person();

// // // echo $lh->xingming;

// $sex=$xk->sex;

// var\_dump(isset($sex));

// // unset()

// class Person{

// public $xingming='老黄';

// public $sex='未知';

// private $age='26';

// // public function \_\_get(){

// // return $this->$name;

// // }

// public function \_\_unset($name){

// // echo "判断被调用";

// $this

// }

// }

// class Person{

// #当创建对象实例化的时候触发

// public function \_\_construct(){

// }

// #对生命周期结束是触发

// public function \_\_destruct(){

// }

// #获取不可见得属性

// public function \_\_get(){

// }

// #获取不可见的方法

// public function \_\_call(){

// }

// #存在

// public function \_\_isset(){

// }

// #销毁不可见属性

// public function \_\_unset(){

// }

// #把对象当做字符串使用时触发

// public function \_\_toString(){

// }

// }

#1.静态:属性对于这个类是唯一的

// class Person{

// public $name='老黄';

// #创建静态的属性

// public static $name1='zb';

// #静态方法

// public static function study(){

// echo "好好学习，天天向上";

// }

// }

// $lh=new Person;

// $zb=new Person;

// echo $lh->name;

// // echo $lh->name1;

// echo $zb::$name1;

// echo "<br/>";

// echo $zb->name;

// // echo $zb->name1;

// #获取静态 ::范围解析符必须加$才可以

// echo $zb::$name1;

// echo "<br/>";

// #通过类名::属性名但是属性必须加$才可以

// echo Person::$name1;

// echo "<br/>";

// #获取静态方法

// Person::study();

// echo "<br/>";

#1.普通成员:使用对象->成员(属性和方法)不加

#2.静态成员:使用对象名::成员;成员需要加$

#静态成员属于整个类

#类常量:定义在类内部的常量

define('PI',3.14);#在任何地方都可以获取

class Person{

#创建类常量,必须有const关键字

const NAME='dd';

public $name='老黄';

// define('PI',3.14);#不能在类中定义

#创建静态的属性

public static $name1='zb';

#静态方法

public function study(){

echo PI;

echo '<br/>';

echo "好好学习，天天向上";

}

}

$lh=new Person;

echo $lh->name;

echo "<br/>";

echo Person::$name1;

echo "<br/>";

echo Person::NAME;#类常量不能直接拿出来使用会报错有两种获取方式

echo "<br/>";

echo PI;

echo "<br/>";

$lh->study();

echo "<br/>";

#封装:把对象的属性和方法结合成一个独立的单元

// class login{

// private $user='wave';

// private $password='123456';

// public function login1($name,$password){

// if($name==$this->user && $password==$this->password){

// echo "登陆成功";

// }else{

// echo "登录失败";

// }

// }

// }

// $cc=new login;

// $cc->login1('wave',12345);

// echo "<pre>";

#继承

// class Father{

// public $user1='火影';

// private $user='wave';

// private $password='123456';

// public function getMessage(){

// return $this->user;

// }

// public function get1(){

// return $this->user;

// }

// private function login1($name,$password){

// if($name==$this->user && $password==$this->password){

// echo "登陆成功";

// }else{

// echo "登录失败";

// }

// }

// }

// $cc=new login;

// $cc->login1('wave',12345);

// class Child extends Father{

// }

// $xz=new Child;

// echo $xz->$user1;

#多重继承

// class Father{

// public $user1='火影';

// private $user='wave';

// private $password='123456';

// public function getMessage(){

// return $this->user;

// }

class Person{

public $name1='老黄'.'<br/>';

public static $name2='zb'.'<br/>';

public function study1(){

echo "非晶态调用静态".'<br/>';

// echo self::$name2;

// self::study();

}

public static function study(){

echo "静态调用非静态";

}

}

$lh=new Person;

$lh->study1();

?>