PHP基础

[PHP基础 1](#_Toc18668)

[一 PHP基础 2](#_Toc21058)

[1. 在html页面中如何嵌入PHP代码 2](#_Toc32593)

[2. 字符串连接符号（.）；其他语言更多是用+ 2](#_Toc3745)

[二 wampserver——www.wampserver.com 2](#_Toc2086)

[1. Localhost——点击启动浏览器到localhost界面 2](#_Toc9318)

[2. myPhpAdmin——集成的网页版mysql数据库管理工具 2](#_Toc30637)

[3. www目录——网站根目录 2](#_Toc32371)

[4. Apache 2](#_Toc15932)

[5. PHP 2](#_Toc16581)

[6. MYSQL——默认密码为空 2](#_Toc30962)

[7. WebGriend——网站性能分析工具 2](#_Toc16579)

[8. 在线/离线状态 3](#_Toc18745)

[9. 自定义网站根目录 3](#_Toc15689)

[10. 多站点配置 3](#_Toc28977)

[11. 监听端口号修改 3](#_Toc15335)

[三 变量 4](#_Toc9321)

[1. 定义变量 变量名=变量值 4](#_Toc23843)

[2. 变量名命名规则 4](#_Toc14244)

[3. 查看内存消耗memory\_get\_usage() 4](#_Toc18370)

[4. 数据类型——8种数据类型 4](#_Toc28534)

[四 常量 7](#_Toc2915)

[1.常量定义函数 7](#_Toc27890)

[2.系统常量 7](#_Toc7411)

[3. 获取函数名 7](#_Toc28881)

[五 运算符 8](#_Toc25541)

[1. 算数运算符 8](#_Toc12463)

[2. 赋值运算符 9](#_Toc17485)

[4. 比较运算符 9](#_Toc5971)

[5. 三元运算符 10](#_Toc15822)

[6. 逻辑运算符 10](#_Toc20514)

[7. 字符串连接运算符 11](#_Toc6791)

一 PHP基础

1. 在html页面中如何嵌入PHP代码

···

<p>

<?php echo “echo的功能是打印一串字符串”; ?>

</p>

//后面的 ?> 是可以省略的

//echo是PHP中的输出语句，可以把字符串输出

···

1. 字符串连接符号（.）；其他语言更多是用+

···

<p>

<?php echo “hello,”.”php”; ?>

</p>

···

二 wampserver——www.wampserver.com

1. Localhost——点击启动浏览器到localhost界面
2. myPhpAdmin——集成的网页版mysql数据库管理工具
3. www目录——网站根目录
4. Apache
5. PHP
6. MYSQL——默认密码为空
7. WebGriend——网站性能分析工具
8. 在线/离线状态

●在线：局域网可以访问

●离线：仅本机可以访问

9. 自定义网站根目录

1. Httpd.conf文件修改根目录路径

●搜索documentroot

●替换根目录路径

●下拉找到两个路径替换掉<Directory "F:/PHPDemo">

2.wampmanager.ini和wampmanager.tpl文件修改根目录路径

●搜索menu.left

●替换根目录路径

1. 多站点配置
2. httpd-vhosts.conf文件：\wamp\bin\apache\Apache2.2.21\conf\extra

<VirtualHost \*:80>

//邮件地址

ServerAdmin [webmaster@dummy-host2.example.com](mailto:webmaster@dummy-host2.example.com)

//根目录地址

DocumentRoot "c:/Apache2/docs/dummy-host2.example.com"

//服务器名；域名；文件夹名；

ServerName dummy-host2.example.com

//错误日志

ErrorLog "logs/dummy-host2.example.com-error.log"

//日常日志

CustomLog "logs/dummy-host2.example.com-access.log" common

</VirtualHost>

备注：因httpd-vhosts.conf文件是作为扩展文件存在，所以需去开启这个扩展

2.httpd.conf文件

●搜索httpd-vhost

●启动Include conf/extra/httpd-vhosts.conf；去除#号注释

●apache默认不允许别的网络访问本地资源所以还需更改这个配置

●搜索onlineoffline tag - don't remove

●Deny from all 修改为 Allow from all //不允许修改为允许

1. hosts文件：C:\Windows\System32\drivers\etc

●添加

···

127.0.0.1 test001.com

127.0.0.1 test002.com

//如果请求这两个域名优先在127.0.0.1内查找资源

···

1. 监听端口号修改
2. Httpd.conf文件：

···

修改监听端口：listen

修改服务器端口:ServerName

···

三 变量

1. 定义变量 变量名=变量值

在“=”左边是变量名，右边是变量值，完成定义变量后，需要一个；告诉服务器这行指令已经结束。其中变量名必须通过$符号标识；

···

<?php

$var\_name = “变成名”;

$n = 10;

Var\_dump($var\_name);

Var\_dump($n)

//var\_dump这个函数会把变量的数据类型显示出来

?>

···

1. 变量名命名规则

●变量名必须以字母或下划线 “\_”开头，如”$\_name”,”$name” ,”$name2”等， 但是”$9name”是不对的。

●变量名只能由字母、数字、以及“\_”组成，还能包含汉字。如”$\_qq”,”$qq308” ,” $my\_apple”，“比如"$name我等”，但是”$name\*”是不对的。

●变量名不允许包含空格。当变量名由多个单词组成，建议使用“\_”进行分隔（比如 $my\_apple），俗称下划线法，或者以大写字母开头比如 $myApple，俗称骆驼式命名 法（也称驼峰命名法）。

1. 查看内存消耗memory\_get\_usage()
2. 数据类型——8种数据类型
3. 布尔型（boolean）:当我们用”echo”指令输出布尔类型时，如果是“true”则输出的是“1”，“false”则什么也不输出。
4. 整形（int）:十进制就是日常使用的数字；八进制，数字前必须加上“0”(这个0是阿拉伯数字0，可不是英文字母“欧”哦)；十六进制，数字前必须加“0x” (这个0也是阿拉伯数字0，不是“欧”哦)。

···

<?Php

$n\_10 = 10;

$n\_8 = 0123;

$n\_16 = 0xff;

$n\_1 = -10;

echo $var\_dump($n\_10);

Echo $var\_dump($n\_8);

Echo $var\_dump($n\_16);

Echo $var\_dump($n\_1);

?>

···

1. 浮点型（float）：浮点型（浮点数、双精度数或实数），也就是通常说的小数，可以用小数点或者科学计数法表示。科学计数法可以使用小写的e，也可以使用大写的E。

···

<?php

$num\_float = 1.234; //小数点

$num\_float = 1.2e3; //科学计数法，小写e

$num\_float = 7.0E-10; //科学计数法，大写E

?>

···

1. 字符串（string）：字符串型可以用三种方法定义：单引号形式、双引号形式和Heredoc结构形式。

1.引号包含引号

···

<?php

$str\_string1 = '甲问："你在哪里学的PHP？"';

$str\_string2 = "乙毫不犹豫地回答：'当然是慕课网咯！'";

$str\_string3 = '甲问:\'能告诉我网址吗？\'';

$str\_string4 = "乙答道:\"www.imooc.com\"";

echo $str\_string1;

echo "<br />";

echo $str\_string2;

echo "<br />";

echo $str\_string3;

echo "<br />";

echo $str\_string4;

echo "<br />";

?>

···

1. 引号包含变量

●当双引号中包含变量时，变量会与双引号中的内容连接在一起；

●当单引号中包含变量时，变量会被当做字符串输出。

···

<?php

$love =”I love you”

$str\_string1 = “慕课网，$love”;

$str\_string2 = ‘慕课网，$love’;

echo $str\_string1;

echo "<br />";

echo $str\_string2;

echo "<br />";

?>

···

1. heredoc结构:在赋值符号后，输入定界符“<<<”,接着是标识符，你可以用你的女神作为标识符“GOD”，如第2行，也可以使用你喜欢的狗狗，“DOG”作为标识符，但是，结尾处的标识符也必须是一样的。此外，在结尾的一行，如第7行，一定要另起一行，并且此行除了“GOD”，并以“；”号结束之外，不能有任何其他字符，前后都不能有，包括空格，否则会出现错误的哦。

···

<?php

$string1 = <<<GOD

我有一只小毛驴，我从来也不骑。

有一天我心血来潮，骑着去赶集。

我手里拿着小皮鞭，我心里正得意。

不知怎么哗啦啦啦啦，我摔了一身泥.

GOD;

echo $string1;

?>

···

1. 特殊变量——资源（resource）：资源是由专门的函数来建立和使用的，例如打开文件、数据连接、图形画布。我们可以对资源进行操作（创建、使用和释放）。任何资源，在不需要的时候应该被及时释放。如果我们忘记了释放资源，系统自动启用垃圾回收机制，在页面执行完毕后回收资源，以避免内存被消耗殆尽。

···

<?php

$file=fopen("f.txt","r"); //打开文件

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

$img=imagecreate(100,100);//图形画布

?>

···

6.特殊变量——空类型（null）:

···

<?php

error\_reporting(0); //禁止显示PHP警告提示

$var;

var\_dump($var);

$var1 = null;

var\_dump($var1);

$var2 = NULL;

var\_dump( $var2);

$var3 = "节日快乐！";

unset($var3);

var\_dump($var3);

?>

···

四 常量

什么是常量？常量可以理解为值不变的量（如圆周率）；或者是常量值被定义后，在脚本的其他任何地方都不可以被改变。PHP中的常量分为自定义常量和系统常量

1.常量定义函数

bool define(string $constant\_name, mixed $value[, $case\_sensitive = true])

第一个参数“constant\_name”为必选参数，常量名称,即标志符，常量的命名规则与变量的一致，但是要注意哦，它可不带美元符号哦。第二个参数“value”为必选参数，它是常量的值。第三个参数“case\_sensitive”为可选参数，指定是否大小写敏感，设定为true表示不敏感，一般不指定第三个参数的情况下，默认第三个参数的值为false。

（注： string表示参数类型为字符串类型，mixed表示参数类型可以接受为多种不同的类型，case\_sensitive = true表示默认为布尔类型TRUE）

···

<?php

$p = "PII";

define("PI",3.14);

define($p,3.14);

echo PI;

echo "<br />";

echo PII;

?>

···

2.系统常量

●\_\_FILE\_\_ :php程序文件名。它可以帮助我们获取当前文件在服务器的物理位置。

●\_\_LINE\_\_ :PHP程序文件行数。它可以告诉我们，当前代码在第几行。

●PHP\_VERSION:当前解析器的版本号。它可以告诉我们当前PHP解析器的版本号，我们可以提前知道我们的PHP代码是否可被该PHP解析器解析。

●PHP\_OS：执行当前PHP版本的操作系统名称。它可以告诉我们服务器所用的操作系统名称，我们可以根据该操作系统优化我们的代码。

1. 获取函数名

●直接使用常量名

●使用constant函数：mixed constant(string constant\_name)

第一个参数constant\_name为要获取常量的名称，也可为存储常量名的变量。如果成功则返回常量的值，失败则提示错误信息常量没有被定义。（注：mixed表示函数返回值类型为多种不同的类型，string表示参数类型为字符串类型）

···

<?php

$p="";

//定义圆周率的两种取值

define("PI1",3.14);

define("PI2",3.142);

//定义值的精度

$height = "低";

//根据精度返回常量名，将常量变成了一个可变的常量

if($height == "中"){

$p = "PI1";

}else if($height == "低"){

$p = "PI2";

}

$r=1;

$area= constant($p)\*$r\*$r;

echo $area;

?>

···

1. 判断常量是否重复定义

Bool defined（string contants\_name）;

它只有参数constant\_name，指的是要获取常量的名称，若存在则返回布尔类型true，否则返回布尔类型false; （注：bool表示函数返回值类型为布尔类型）

···

<?php

define("PI1",3.14);

$p = "PI1";

$is1 = defined($p);

$is2 = defined("PI2");

var\_dump($is1);

var\_dump($is2);

?>

···

五 运算符

1. 算数运算符



1. 赋值运算符

●“=”：把右边表达式的值赋给左边的运算数。它将右边表达式值复制一份，交给左边的运算数。换而言之，首先给左边的运算数申请了一块内存，然后把复制的值放到这个内存中。

●“&”：引用赋值，意味着两个变量都指向同一个数据。它将使两个变量共享一块内存，如果这个内存存储的数据变了，那么两个变量的值都会发生变化。

···

<?php

$a = "我在慕课网学习PHP！";

$b = $a;

$c = &$a;

$a = "我天天在慕课网学习PHP！";

echo $b."<br />";

echo $c."<br />";

?>

···

3.比较运算符



4. 三元运算符

(“?:”)三元运算符也是一个比较运算符，对于表达式(expr1)?(expr2):(expr3)，如果expr1的值为true，则此表达式的值为expr2，否则为expr3。

···

<?php

$a = 78;//成绩

$b = $a>60?"及格":"不及格";

echo $b;

?>

···

1. 逻辑运算符



1、逻辑与：要求所有人都投票同意，才会通过某协议；

2、逻辑或：只要求一个人投票同意就行；

3、逻辑异或：只能有且只能有一个人投票同意；

1. 逻辑非：某个人要反对，但是通过逻辑非，使其反对无效；
2. 字符串连接运算符

（1）连接运算符(“.”)：它返回将右参数附加到左参数后面所得的字符串。

（2）连接赋值运算符(“.=”)：它将右边参数附加到左边的参数后。

···

<?php

$a = "张先生";

$tip = $a.",欢迎您在慕课网学习PHP！";

$b = "东边日出西边雨";

$b .= ",道是无晴却有晴";

$c = "东边日出西边雨";

$c = $c.",道是无晴却有晴";

echo $tip."<br />";

echo $b."<br />";

echo $c."<br />";

?>

···

1. 错误控制运算符

PHP中提供了一个错误控制运算符“@”，对于一些可能会在运行过程中出错的表达式时，我们不希望出错的时候给客户显示错误信息，这样对用户不友好。于是，可以将@放置在一个PHP表达式之前，该表达式可能产生的任何错误信息都被忽略掉；

如果激活了track\_error（这个玩意在php.ini中设置）特性，表达式所产生的任何错误信息都被存放在变量$php\_errormsg中，此变量在每次出错时都会被覆盖，所以如果想用它的话必须尽早检查。

需要注意的是：错误控制前缀“@”不会屏蔽解析错误的信息，不能把它放在函数或类的定义之前，也不能用于条件结构例如if和foreach等

···

<?php

$conn = @mysql\_connect("localhost","username","password");

echo "出错了，错误原因是：".$php\_errormsg;

?>

···

五 语言结构语句

1. 条件结构之if…else…

···

<?php

date\_default\_timezone\_set('Asia/ShangHai');

$today = date('m-d',time());//获取当天日期

$birthday = "02-14";//生日

$money = 238;//消费金额

$discount = 0.8;//八折优惠

if($today == $birthday){

$money = $money \* $discount;

}else{

$money = $money \*1;

}

echo $money;

?>

···

1. 条件结构之switch…case…

首先判断条件，若条件的返回值为条件值一，则执行任务一，若条件返回的值为条件值二，则执行任务二，若条件的返回值既不是条件值一也不是条件值二，则执行默认任务。break的作用是结束switch（后面会有专门举例说明），使用 switch 语句可以避免冗长的 “if..else if..else”代码块。

···

<?php

switch (条件)

{

case 条件值一:

//任务一

break;

case 条件值二:

//任务二

break;

default:

//默认任务

}

?>

<?php

$num = rand(1,50);//获取1至50的随机数

$info = "";//提示信息

switch($num){

case 1:

$info = "恭喜你！中了一等奖！";

break;

case 2:

$info = "恭喜你！中了二等奖！";

break;

case 3:

$info = "恭喜你！中了三等奖！";

break;

default:

$info = "很遗憾！你没有中奖！";

}

echo $num;

echo $info; //输出是否中奖

?>

···

1. PHP中循环结构之while循环语句

首先判断某个条件是否符合（条件返回值是否为TRUE），若符合则执行任务，执行完毕任务，再判断条件是否满足要求，符合则重复执行此任务，否则结束任务。

···

<?php

while(条件){

//执行任务

}

?>

<?php

$sum = 12;//小宠物当前的饥饿程度

echo "我饿啦:-(";

echo "<br />";

while($sum<100){//小宠物的饥饿程度到100，表示小宠物吃饱啦,不用继续喂了，没吃饱继续喂食

echo $sum;

$num = rand(1,20);//随机数，模拟喂食小宠物的小面包

$sum = $sum + $num; //小宠物吃小面包

echo "我还没吃饱呢！";

echo "<br />";

}

echo "终于吃饱啦^\_^";

?>

···

1. PHP中循环结构之do while循环语句

首先执行任务（上一节的while语句是先判断条件是否成立，再执行任务），执行任务完毕，判断某个条件是否符合（条件返回值是否为TRUE），若符合则再次执行任务，执行完毕任务，继续判定条件。

···

<?php

do{

//执行任务

}while(条件)

?>

<?php

$i = 1 ; //从第1圈开始跑

do{ //跑10圈

echo $i;

echo "在跑第".$i."圈。";

echo "<br/>";

$i++;

}while($i<=10)

?>

···

1. PHP中循环结构之for循环语句

for 语句中，“初始化”在循环开始前无条件求值一次，“循环条件”在每次循环开始前求值。如果值为 TRUE，则继续循环，执行循环体语句（执行任务）。如果值为 FALSE，则终止循环。“递增项”在每次循环之后被求值（执行）。其常用于循环执行代码块指定的次数。

···

<?php

for(初始化;循环条件;递增项){

//执行任务

}

?>

<?php

for($i=1,$sum=0;$i<=100;$i++)

{

$sum = $sum + $i; // 累加求和

}

echo "for语句的运行结果：".$sum."<br />" ;

//while语句写法

$i = 1 ; // 从1开始累加

$sum = 0; //初始化和为0

while($i<=100){ //判断是否小于100

$sum = $sum + $i; // 累加求和

$i++; //递增1

}

echo "while语句的运行结果：".$sum."<br />" ;

?>

···

1. PHP中循环结构之foreach循环语句(任务一)
2. 只取值，不取下标

···

<?php

foreach (数组 as 值){

//执行的任务

}

?>

···

1. 同时取下标和值

···

<?php

foreach (数组 as 下标 => 值){

//执行的任务

}

?>

<?php

$students = array(

'2010'=>'令狐冲',

'2011'=>'林平之',

'2012'=>'曲洋',

'2013'=>'任盈盈',

'2014'=>'向问天',

'2015'=>'任我行',

'2016'=>'冲虚',

'2017'=>'方正',

'2018'=>'岳不群',

'2019'=>'宁中则',

);

//10个学生的学号和姓名，用数组存储

//使用循环结构遍历数组,获取学号和姓名

foreach($students as $s => $v)

{

echo $s;

echo $v;//输出（打印）姓名

echo "<br />";

}

?>

···