# PhP

第四章变量声明

1.变量:就是一个容器,以$开头,

2.打印变量:echo直接输出,var\_dump打印变量,print\_r打印数组,isset检测变量是否被声明,unset删除变量

第五章 变量的数据类型

1.数据类型分类

1.标量型:布尔型.整形.浮点型,字符串型,

2.复合类型:数组.对象

3特殊类型:资源.null

1.1布尔型:只有真的true,假的false

1.2整形int:整整数.负整数.0

1.3浮点数:float就是小数

1.4字符串型:就是用电引号和双引号包含的字符就是字符串

界定符前后要加表示

5.1.5数组类型array:下标区分:关联数组.索引数组.混合数组/维度区分:一维.二维.多维.

5.2检测变量是否被声明;

isset() 检测变量是否被声明过，如果声明过，返回true，如果没有声明过返回false

empty() 判断变量的值是否为空（零，假，null），如果是返回ture，否则返回false

第六章判断数据类型

is\_bool() 判断是否是布尔型

is\_int() 判断是否是整型

Is\_float() is\_real() 判断是否是浮点型

is\_string() 判断是否是字符串

is\_array() 判断是否是数组

is\_object() 判断是否是对象

is\_resource() 判断是否是资源

is\_null() 判断是否是null

is\_scalar() 判断是否是标量

is\_numeric() 判断是否是任何类型的数字和数字字符串

is\_callable() 判断是否是有效的函数名

第七章 数据类型转换

自动类型转换，五中数据类型，标量中的四个类型和null都可以通过运算自动转换类型。

字符串和数字运算，字符串先转换为数字在运算。

字符串转换为数字，从前开始到第一个不是数字的字符结束(不符合的内容清空)。

整型转换为浮点数 ，精度不改变。

浮点数转换为整型舍去小数点，保留整数部分。

如果浮点数超出整数范围，答案不确定。

null 值转换为字符串，是空字符串。

2.强类型转换

1使用括号加目标类型进行转换

2）使用类型转换函数

第八章 常量

常量，用来表示程序中一些需要经常用的固定值，俗称：常驻内存的量，叫常量。

常量定义格式：define('常量名'，'常量值',false); //false 常量名区分大小写，默认

第九章 系统常量

PHP\_OS ----> php 运行的系统内核名称

PHP\_VERSION -----> php版本号

\_\_LINE\_\_ ----> 返回当前行数

\_\_FILE\_\_ ----> 返回当前文件全部路径包括文件名

第十章 运算符和分类

（1）运算符的功能区分(重点重点重点重点)

算术运算符

+ - \* / % ++ --

前加加(++$a)，先运算后赋值

后加加($a++)，先赋值后运算

（2）按照运算数区分

i. 一元运算符 有一个运算数参与的运算

例如：!$a $a++ $b-- --$b --$a....

ii. 二元运算符 有二个运算数参与的运算

例如：$a>$b $a+$b $a\*$b.......

iii. 三元运算符

例如：$a?$b :$c

10.3链接运算符用米粒 “ . ”

10.6 逻辑算符

逻辑运算符

&& 和(and) ，逻辑与，一边为假 即为假

|| 或者(or) ，逻辑或，一边为真 即为真

！ 逻辑非 真变假，假变真

xor 逻辑异或相同为假，不同为真

第十二章 循环结构

1. 循环结构三要素:

a. 初始值

b. 循环条件

c. 改变条件

2 for循环

声明格式

for(表达式1;表达式2;表达式3){

循环体语句

}

3 while循环

while 格式:

while(判断条件){

循环体语句

}

4 do...while循环

do...while 格式：

do{

循环体语句

}while(判断条件);

6 特殊形成控制

break ----> 跳出整个循环结构，执行下面的代码

continue ----> 跳出本次循环，进入下次循环

exit ----> 终止程序运行

die ----> 也可以停止代码执行

第十三章 函数

1 函数的定义

系统函数：比如：is\_int() is\_float() is\_bool() isset() empty() unset()

2 自定义函数function函数的基本格式：

/\*

function 函数名([参数1，参数2，参数3]){

函数体 (php语句)

[return 返回值]

}

\*/

3 函数的 return 返回值

return 的作用：

（1）如果函数中存在return并且执行该return，那么该函数的将执行结构可以被变量接收。

（2）如果函数的执行过程中执行了return语句，后续的代码将不在执行。

.5 可变参数函数

func\_get\_args() 获取所有实参的值，以数组的形式返回。

func\_num\_args() 获取所有是实参的个数。

func\_get\_arg(index) 通过下标去取出我们数组中的值。

4 变量函数

变量函数（动态调用函数）: 如果一个变量后面有括号: $zhangsan=’demo’;$zhangsan() 就会寻找与变量值同名的

函数进行调用。

5 匿名函数

所谓匿名函数：就是没有名字的函数

声明格式：

$user = function(){};

注意:一定要加分号一定要加分号 一定要加分号

匿名函数一定要在声明后面调用函数 不能在声明函数前调用函数，因为匿名函数其实就是变量赋值，是把整个函数

赋值给一个变量了，变量是先声明后使用。

.6 递归函数

递归的特点:

1.自身调用自身 (如果一直调用下去肯定不行,死循环)

2.要有一个明确的边界,能够终止调用

7 函数中变量作用域

7.1 局部和全局变量

变量的作用域,就是变量在函数里，和不在函数里。

对于php而言,函数的作用域非常简单，它就区分函数内和函数外。

7.2 global关键字（不推荐）

7.3 static 静态变量

1.普通局部变量

函数调用时--->初始化

函数结束时--->从内存消失

2. static 静态变量

所谓静态变量，是相对内存而言，俗称静止不动的变量。

第十四章文件包含

区别:include 和 require

include include\_ once 文件包含

require require\_once 也是文件包含，但有必须存在的意思

第十五章 数组加强

1. 数组的概念

概念：数组的本质是存储、管理和操作一组变量属于复合类型的一种。

键值对的概念： $arr = array('id'=>1,'name'=>'duke','sex'=>'人妖');

.2 数组的声明方式

/使用array()数组方式的声明：

1. 索引数组

$arr = array(1,'帅气的B哥',18,'男神','182cm','180kg');

1. 关联数组

$arr1 = array('id'=>2,'name'=>'laowang','zhiye'=>'省港澳第一ADC','age'=>'30');

3.混合数组

$arr2 = array('id'=>3,'鹏鹏','sex'=>'雌雄同体','age'=>28,'zhiye'=>'沈阳第一鸭王');

/使用[]数组方式声明（5.6版本以上才能使用）：

3.混合数组

$arr2 = ['id'=>3,'鹏鹏','sex'=>'雌雄同体','age'=>28,'zhiye'=>'沈阳第一鸭王'];

/直接赋值法声明数组：

2.关联数组

$arr2['id']=1;

$arr2['name']='澳洲车神';

$arr2['sex']='女';

$arr2['age']=16;

$arr2['phone']='18610987634';

$arr2['url']='www.dushen.com';

3 读取数组的值

通过变量+[下标]的方法或者某个数组准备的值。

注意：读取准备的某个元素，可以使用 echo ，因为echo可以输出单个非特殊类型的值。

如果要输出整个数组，要知道print\_r或者var\_dump来打印。

4 数组的增删改查

//添加方法1:

//$arr[] = '小星星';

//添加方法2

//$arr['h'] = '大猩猩';

//修改元素

//$arr['h'] = '大猩猩不好惹';

//删除数组元素

//unset($arr['f']);

//unset($arr['d']);

//删除数组

//unset($arr);

//清空数组

$arr = array();

第十六章 数组的遍历

.1 数组的分类

（1）一维数组

数组内部再也没有数组元素

（2）二维数组

数组内还有数组单元

（3）多维数组

数组内还有数组单元 数组单元里面还有数组单元 一层一层的嵌套

.2 数组的遍历

数组的遍历: 使用一种特定的规则来逐个读取数组中的键和值

方式1：使用for循环

for($i=0;$i<count($arr);$i++) {

echo $arr[$i],'<br >';

}

方式2：使用foreach遍历数组（专业遍历数组）

//只遍历值

foreach(要遍历的数组变量 as 值){

循环体;

}

//遍历键和值

foreach(要遍历的数组 as 键=>值){

循环体;

}

3 while list each遍历数组

list()，可以将一组索引数组单元逐个赋值给一组变量，把每个值赋值给list内的变量，而且只能是索引数组，索引数

组必须是从0开始。

each()，每次访问一个数组单元并且将指针下移到下一个将要访问的数组单元中将访问出来的数组单元以混合数组

的形式返回，当达到数组单元最后没有的时候返回false

while(list($key,$value)=each(要遍历的数组)){

循环体;

}

第十七章 预定义数组(超全局数组变量)

1. $\_SERVER --- 服务器和执行环境信息

2. $\_GET --- 获取地址栏上的信息

3. $\_POST --- 用于收集来自 method="post" 的表单中的值

4. $\_FILES --- 打印文件上传是否成功的内容信息

# 留言板

第一步:搭建万年历框架

用html写个表格

第二步:循环本月天数

用while循环循环出每个月的天数记得设一个判断一行只循环一周

第三步:完善本月信息:

设定当月天数;设定当前月份1号是星期几

第四步:根据当前系统时间获取时间:

用data(t,mktime(0,0,0,$month,1,year))用系统时间来获取当月天数,和年.当月1号是星期几是用函数w定义的

第五步:添加按钮可点击上下月

用a标签写链接点击,点击后上传到地址栏,然后再从地址栏上获取信息,记得上一个月和下一个月要加减1 在设一个判断用于防止出现负月个13月的情况.

# 时间和日期

Unix时间戳:

相关函数:

Time函数返回一个当前系统的时间戳

Mktime取得一个日期的unix时间戳

格式:int mktime(时,[分,[秒,[月,[日[,年[,is\_dst区]]]]]])

Strtoinme 将任何英文文本的日期时间描述解析为unix时间戳

在PHP中获取日期和时间

Getdata 获取时间/日期信息

格式array getdata([int timestamo])

date -- 格式化一个本地时间／日期

格式：string date ( string format [, int timestamp] )

返回将整数 timestamp 按照给定的格式字串而产生的字符串。如果没有给出时间戳则使用本地当前时间。换句话说，timestamp 是可选的，默认值为 time()（当前时间戳）。

例如： echo date(“Y年m月d日 H:i:s”); //2010年10月28日 14:22:28

常用参数:

Y：四位数年 m：月01-12 n:月1-12 d：天01-31 j：天1-31

H：时24时制 h：小时12制 i：分钟00-59 s：秒00-59 w：星期几0-6

A：上午AM或下午PM a：上午am或下午pm

echo date("Y-m-d H:i:s");echo date("Y年m月d"，time());

echo date("Y/m/d"，time());

注意:设定中国时区date\_default\_timezone\_set('PRC');

使用微秒计算PHP脚本执行时间

1ms = 0.001s; 1us = 0.001ms ; 1ns = 0.001us

Microtime 返回时间戳和微秒数

格式mixed microtime([bool get\_as\_float])

# 文件系统管理

文件系统概述

Windows 闭源系列

Linux 开源系列

RIM 黑莓系统用的人少,所以安全

文件类型

Filetype 过去文件具体类型

**i**s\_dir( ) -- 判断给定文件名是否是一个目录。

is\_executable( ) -- 判断给定文件是否可执行。

is\_file( ) -- 判断给定文件名是否为一个正常的文件。

is\_writable( ) -- 判断给定的文件名是否可写。

文件属性

有好多函数……在4.常用模块之文件系统管理.md 里的1.3自己看

目录的基本操作

解析目录路径

常见的文件目录路径格式:绝对路径,,相对路径

推荐使用相对路径

Basename 返回路径中的文件名部分

Dirname 返回路径中的目录部分

Pathinfo 返回文件路径的信息

可选参数2:

PATHINFO\_DIRNAME 只返回路径

PATHINFO\_BASEAME 只返回带后缀的文件名

PATHINFO\_EXTENSION 只返回后缀名

遍历目录

Opendir 打开目录

Readdir 从目录中读取条目

Closedir 关闭目录

$handle = opendir(‘./xxx’)

While( $filename = readdir($handle));{

echo $filename;

}

Closedir;

遍历目录到表格中

就是将上面的遍历目录写入到一个表格中.

统计磁盘的大小

Disk\_free\_space 返回磁盘中的可用空间.单位是字节

Disk\_total\_space 返回一个目录的磁盘总大小.单位是字节

建立和删除目录

Mkdir 新建目录

Rmdir删除目录

Touch 设定文件的访问个修改时间

Unlink 删除文件

文件的基本操作

文件的打开与关闭

Fopen 打开文件货这url

Fclose 关闭一个已打开的文件

写入文件

Fwrite 写入文件

File\_put\_contents 将一个字符串写入文件

格式1.要写入的文件名称 2.要写入的内容 3.一般写FILE\_APPEND，与a+效果一样

读取文件内容

Fread 读取文件

File\_get\_content 将文件内容读取出来.并读取到字符串返回

移动文件指针

ftell -- 返回文件指针读/写的位置

rewind -- 倒回文件指针的位置

文件的一些基本操作函数

rename -- 重命名一个文件或目录

copy -- 拷贝/复制 文件

unlink -- 删除文件

# 文件上传和下载

PHP配置文件中与文件上传

$\_FILES 多维数组

1、$\_FILES["myfile"]["name"]中的值是:客户端文件系统的文件的名称。

2、$FILES["myfile"]["type"]中的值是:客户端传递的文件的类型。

3、$\_FILES["myfile"]["size"]中的值是:文件的字节的大小。

4、$\_FILES["myfile"]["tmp\_name"]中的值是：文件被上传后在服务器存储的临时全路径。

5、$\_FILES["myfile"]["error"]中的值是:文件上传的错误代码－php 4.2以后增加的功能。

Error文件上传的错误代码

Upload\_err\_ok

其值为0.没有错误发生.文件上传成功

Upload\_err\_ini\_size

其值为1.上传的文件超过了php.ini