

PHP基础

1课 变量声明

变量名称的命名规范:

[a-zA-Z0-9]和下划线(_)

1)变量名是区分大小写的

2)不能以数字开头

```
$a  
$_  
$_%  
$3c  
$c3
```

什么是变量

变量就是一个盒子,搞清楚变量名与变量值

2课 变量赋值

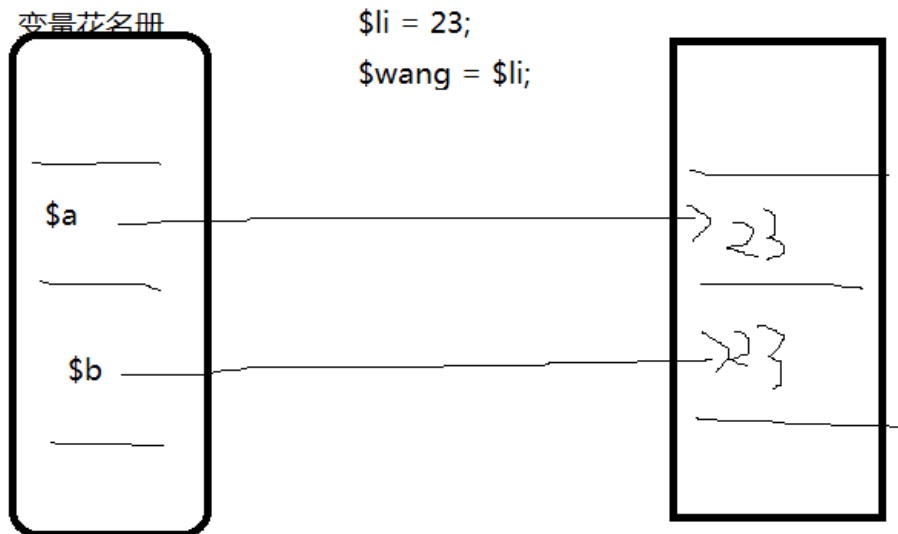
1.传值赋值

变量名其实不是贴在盒子上的,而是有个变量表(像班级的花名册)

变量值和变量类型放在盒子里面;变量表中的变量名指向它所对应的盒子.

```
$li = 23;  
$wang = $li;  
echo $li, '~', $wang;
```

以上代码在内存中发生了什么?



改变\$li的值,\$wang的值会发生变换吗

```
$li = 99;  
echo $li, '~', $wang;
```

这个赋值过程,是把\$li的值,赋给\$wang

3课 数学运算符

有变量没什么意义,还需要有比较
变量需要比较,运算才有意义
幸福感就是比较出来的

```
+, -, *, /, %
```

```
$zhang = 180;  
$lisi = 175;  
echo $zhang-$lisi;
```

+

```
$zhang = 50;  
$lisi = 40;  
echo $zhang + $lisi;
```

*

```
$zhang = 1000;  
$li = $zhang*2;  
echo $li;
```

/

```
$zhang = 1000;  
$li = $zhang/2;  
echo $li;
```

取模 % [求余数]

```
echo 7%3;  
echo 20%7 , '<br >';
```

4课 比较运算符

, >=, <, <=, ==, !=, ===, !==

凡运算,必有运算结果,比较运算符的运算结果是布尔型值

比较运算的运算结果

```
$a = 5;  
$b = 2;  
$c = $a>$b;  
var_dump($c);
```

==和===的区别

```
$c = (3 == '3');  
var_dump($c);  
//== 只验证值是否相等  
  
$c = (3 === '3');  
var_dump($c);  
//要求变量类型相等,且值也要相等;
```

全等于和等于的例子

strpos — 查找字符串首次出现的位置(区分大小写)

如果找到,返回字符串中首次出现的数字位置,如果没找到返回boolean(false)

```
$str = 'abcdef';
$pos = strpos($str, 'a');
var_dump($pos);

//0被if理解为false
if($pos == false) {
    echo '没找到';
} else {
    echo '找到了';
}

if($pos === false) {
    echo '没找到';
} else {
    echo '找到了';
}
```

5课 逻辑运算符

&&(并且)

||(或者)

某美女要求,有房且有车,才嫁

```
$house = true;
$car = false;

if( $house && $car ) {
    echo 'marry';
} else {
    echo 'sorry';
}
```

某女要求,有房或有车,就嫁

```
$house = true;
$car = false;

if( $house || $car ) {
    echo 'marry';
} else {
    echo 'sorry';
}
```

或者就是有一个为真就行,而并且需要都为真才行

6课 字符串运算符

. 拼接运算

```
$a = 'hello';
$b = 'word';
$c = $a . $b;
echo $c;

$c = $c . '123456';
echo $c;
```

下面哪个运行速度更快?

```
echo $a . $b;  
echo $a,$b;
```

字符串并不是可以用逗号拼接的

只是echo可以输出多个变量,用逗号隔开即可

而点是先拼接再输出

所以逗号更快一些,不需要拼接字符串再输出

比如:

我想拿出来两个橡皮泥给你,是两个直接给你更快,

还是将两个橡皮泥揉成一块给你快

7课 while循环

循环

1.while

先判断再执行

条件为真就执行,直到条件不为真才截至

如果上来条件就不为真,那一次都不会执行

```
while(条件为真){  
    //执行体  
}  
  
$i = 1;  
while ($i < 10) {  
    echo $i , '<br >';  
    $i = $i + 1;  
}
```

8课 for循环

for和while都是一种循环

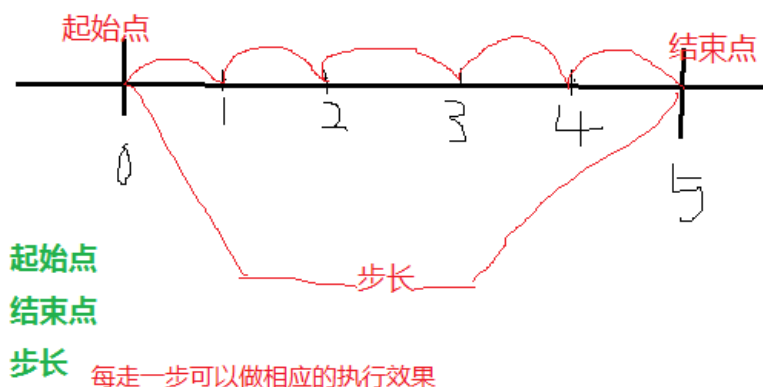
解释:所有循环的共同点

画图详解

在一条数轴上,循环就是有起始点($i = 0$);

有结束点($i \leq 10$);有步长($i++$);

每走一个相应的步长,就执行一次代码(执行语句)



```
循环的一般要素: 初始化(1),判断(2),执行体(3),修改变量(4)  
for(1[起始点];2[结束点(到结束了没有)];4[步长]) {  
    3[执行语句];  
}
```

```
}
```

1.for循环打印0-9

```
for($i=0; $i<=9; $i=$i+1) {  
    echo $i , '<br >';  
}
```

2.步长可以不为1

```
for($i=0; $i<=9; $i=$i+3) {  
    echo $i , '<br >';  
}
```

3.初始变量可以为多个

```
for($i=1,$j=5 ; $i<=5; $i=$i+1,$j=$j-1){  
    echo $i,'~~~~~',$j , '<br />';  
}
```

4.for循环的改造

可以有其他的写法[类似while循环]

```
for($i=0; $i<=9; $i=$i+1) {  
    echo $i , '<br >';  
}
```

改造之后

```
$i=0;  
for( ; $i<=9 ; ) {  
    echo $i;  
    $i=$i+1;  
}
```

for循环很常用,要数量掌握

9课 break/continue

1.continue 下一个 (用next更形象一点)

整体的循环没有破坏掉,而是跳到下一个循环单位中

```
for($i=1; $i<=10; $i = $i+1){  
    if($i == 4) {  
        continue;  
    }  
    echo $i,<br />;  
}
```

2.break 破坏(将后面的执行语句破坏掉)

整体的循环都破坏掉了,循环到此结束掉了

```
for($i=1; $i<=10; $i+=1){  
    if($i == 4) {  
        break;  
    }  
    echo $i,<br />;  
}
```

在c语言,java中都有类似与continue和break的这个东西

3.生活中的例子,相亲(只讲解)

遇到长得丑的直接continue掉,接着相亲下一个

遇到白马王子,直接break,后面的相亲对象就不见面了

1)continue

```
for($i=1; $i<=10; $i+=1) {  
    if ($i == 4) {  
        echo '照片丑,不约';  
        continue;  
    }  
    echo $i,'<br />';  
}
```

2.break

```
for($i=1; $i<=10; $i+=1) {  
    if ($i == 6) {  
        echo $i,'是白马王子';  
        break;  
    }  
    echo $i,'<br />';  
}
```

10课 过桥问题

有了前面学的变量,表达式,控制结构,我们实际做一些编程的小题目.小题目在于锻炼你的编程思维

为什么遇到实际题目无法写出来,因为你学习的是语法,但是你们缺少编程思维,让自己学会像计算机一样思考.

那是不是计算机太聪明了,我们太笨呢?

不是的,恰恰相反,是我们太聪明了,计算机太笨跟不上我们的思想.

变笨需要我们慢慢锻炼!

【程序 6】

```
/*  
初始 $money=100000  
判断条件 $money>=5000  
步长2种选择 if($i>50000) else if($i<=50000)  
*/  
  
for($i=1,$money=100000; $money>=5000; $i +=1) {  
    if($money>50000){  
        $money -= $money*0.05;  
    }else if($money<=50000) {  
        $money -=5000;  
    }  
    echo '第',$i,'次过路口,剩下',$money,'元<br />';  
}
```

【程序 6】

假设某人有 100,000 现金。 每经过一次路口需要进行一次交费。 交费规则为当他现金大于 50,000 时每次需要交 5%如果现金小于等于 50,000 时每次交 5,000。 请写一程序计算此人可以经过多少次这个路口。

11课 九九乘法表

编程时,遇到不会的问题,要将它拆解,一步一步的来

你不要想一下子就能做出来

九九乘法表									
1x1=1									
2x1=2	2x2=4								
3x1=3	3x2=6	3x3=9							
4x1=4	4x2=8	4x3=12	4x4=16						
5x1=5	5x2=10	5x3=15	5x4=20	5x5=25					
6x1=6	6x2=12	6x3=18	6x4=24	6x5=30	6x6=36				
7x1=7	7x2=14	7x3=21	7x4=28	7x5=35	7x6=42	7x7=49			
8x1=8	8x2=16	8x3=24	8x4=32	8x5=40	8x6=48	8x7=56	8x8=64		
9x1=9	9x2=18	9x3=27	9x4=36	9x5=45	9x6=54	9x7=63	9x8=72	9x9=81	

1.先for循环1-9

```
//for循环打印1-9
for($i=1;$i<=9;$i++){
    echo $i,'<br />';
}
1
2
3
4
5
6
7
8
9
```

2.

```
for($i=1;$i<=9;$i++){
    echo $i,'~';
    for($j=1;$j<=$i;$j++) {
        echo $j;
    }
    echo '<br />';
}

1~1
2~12
3~123
4~1234
5~12345
6~123456
7~1234567
8~12345678
9~123456789
```

3.

```
for($i=1;$i<=9;$i++){
    for($j=1;$j<=$i;$j++) {
        echo $j,'*',$i,'=', $j*$i,'&nbsp;';
    }
    echo '<br >';
}

1*1=1
2*1=2 2*2=4
3*1=3 3*2=6 3*3=9
4*1=4 4*2=8 4*3=12 4*4=16
5*1=5 5*2=10 5*3=15 5*4=20 5*5=25
6*1=6 6*2=12 6*3=18 6*4=24 6*5=30 6*6=36
7*1=7 7*2=14 7*3=21 7*4=28 7*5=35 7*6=42 7*7=49
8*1=8 8*2=16 8*3=24 8*4=32 8*5=40 8*6=48 8*7=56 8*8=64
9*1=9 9*2=18 9*3=27 9*4=36 9*5=45 9*6=54 9*7=63 9*8=72 9*9=81
```

12课 百钱买百鸡

1.

```
for($m=1;$m<=100;$m++){
    for($g=1;$g<=100;$g++){
        for($s=1;$s<=100;$s++){
            if(($m+$g+$s==100) && ($m*3+$g*5+$s/3)==100) {
                echo '公鸡',$g,'只,母鸡',$m,'只,小鸡',$s,'只<br />';
            }
        }
    }
}
```

【程序 8】

《张丘建算经》成书于公元 5 世纪，作者是北魏人。书中最后一道题堪称亮点，通常也被称为“百钱买百鸡”问题，民间则流传着县令考问神童的佳话书中原文如下：

今有鸡翁一，值钱五；鸡母一，值钱三；鸡雏三，值钱一；百钱买鸡百只，问鸡翁、母、雏各几何？

题目的意思是，公鸡 5 文钱 1 只，母鸡 3 文钱 1 只，小鸡 1 文钱买 3 只，现在用 100 文钱共买了 100 只鸡，问：在这 100 只鸡中，公鸡、母鸡和小鸡各是多少只？（设每种至少一只）

```

    }
    }
}

公鸡12只,母鸡4只,小鸡84只
公鸡8只,母鸡11只,小鸡81只
公鸡4只,母鸡18只,小鸡78只

```

```
for($m=1;$m<=100;$m++){
    for($g=1;$g<=100;$g++){
        $s=100-$m-$g;
        if(($m*3+$g*5+$s/3)==100) {
            echo "公鸡",$g,"只,母鸡",$m,"只,小鸡",$s,"只<br />";
        }
    }
}
```

公鸡12只,母鸡4只,小鸡84只
 公鸡8只,母鸡11只,小鸡81只
 公鸡4只,母鸡18只,小鸡78只

```
for($m=1;$m<=31;$m++){
    for($g=1;$g<=18;$g++){
        $s=100-$m-$g;
        if(($m*3+$g*5+$s/3)==100) {
            echo '公鸡',$g,'只,母鸡',$m,'只,小鸡',$s,'只<br />';
        }
    }
}
```

13课 字符串定义

字符串的操作在php中十分丰富

1.如何定义字符串:

字符串的操作在php中十分丰富

1.如何定义字符串:

单引号,双引号

```
$str1 = 'hello';  
$str2 = "word";  
echo $str1, ' ', $str2;
```

2.如何定义大段的字符串

有两种方式,heredoc和nowdoc

1)heredoc 对应大段的""(双引号)字符串

要求非常的严格

开头前后不能有空格

中间不能插注释,前后保持一致,大小写均可

```
$str = <<<here
第一段
第二段
第三段
here;
```



```
echo $str;
```

2)nowdoc 对应大段的"(单引号)字符串

```
$str2 = <<<'NOW'  
第一段  
第二段  
第三段  
NOW;  
  
echo $str2;
```

14课 字符串函数

字符串对于做web开发而言,非常之常用;(每天都离不开它,计算它的长度,切割,反转等)

字符串函数非常丰富且强大,一定要认真练习!

【字符串长度函数】

int strlen(\$str) 计算字符长度

int mb_strlen (string \$str [, string \$encoding])

【查找字符串位置函数】

strpos(\$str,search,[int]):查找 search 在\$str 中的第一次位置从 int 开始;

stripos(\$str,search,[int]):函数返回字符串在另一个字符串中第一次出现的位置。该函数对大小写不敏感

strrpos(\$str,search,[int]):查找 search 在\$str 中的最后一次出现的位置从 int

【字符串替换函数】

str_replace(search,replace,\$str):从\$str 中查找 search 用 replace 来替换

str_ireplace(search,replace,\$str):

strtr(\$str,search,replace):这个函数中 replace 不能为"";

substr_replace(\$Str,\$rep,\$start[,length])\$str 原始字符串,\$rep 替换后的新字符串,\$start 起始位置,\$length 替换的长度, 该项可选

【截取子字符串函数】

substr(\$str,int start[,int length]):从\$str中start位置开始提取[length 长度的字符串]。

strstr(\$str1,\$str2): 从\$str1(第一个的位置)搜索\$str2 并从它开始截取到结束字符串;若没有则返回 FALSE。

stristr() 功能同 strstr, 只是不区分大小写。

strrchr() 从最后一次搜索到的字符处返回;

【分割,连接,反转函数】

str_split(\$str,len):把\$str 按 len 长度进行分割返回数组

explode(search,\$str[,int])

implode —— 将数组用特定的分割符转变为字符串

【空白处理函数】

string trim (string \$str [, string \$charlist])

string ltrim (string \$str [, string \$charlist])

string rtrim (string \$str [, string \$charlist])

chunk_split(\$str,2);向\$str 字符里面按 2 个字符就加入一个空格;

str_pad —— 对字符串进行两侧的补白

【字符转义函数】

addslashes (string \$str)
stripslashes (string \$str)
get_magic_quotes_gpc()
htmlspecialchars —— 将字符串中一些字符转换为 HTML 实体
htmlspecialchars_decode —— htmlspecialchars()函数的反函数，将 HTML 实体转换为字符
html_entity_decode —— htmlentities ()函数的反函数，将 HTML 实体转换为字符
htmlentities —— 将字符串中所有可转换字符转换为 HTML 实体

【字符串比较函数】

int strcmp(\$str1,\$str2): (字符串比较)
strcasecmp() 同上 (不分大小写)

计算字符串长度

strlen

mb_strlen

strlen — 获取字符串长度 (计算的是字节数)

int strlen (string \$string)

```
$str1 = 'abc d';  
$str2 = '中国人';  
echo strlen($str1);  
echo strlen($str2);
```

用记事本分别保存:中国,一个gbk,一个utf8,分别观察占几个字节

有些文档了为了区分编码,到底按照哪种编码解析它

计算机都是存的0,1

有些文档就会在utf8编码的文档中加上3个人眼无法看到的字节

这个字节称为BOM头,正好占3个字节

如果我们用记事本建utf8的文档,就会带来这个问题;

为什么去掉BOM头?

学习session和cookie的时候,有BOM头会报错

因为在session或者cookie启用前有了输出

如何去掉BOM头?

用editplus[右下角编码带utf8+ 表示有BOM]和sublime(默认无BOM),都可以去掉BOM头]

gbk一个中文占2个字节,utf8一个中文占3个字节

mb_strlen — 获取字符串的长度 (计算的是字符数)

mb 宽字节--国际编码支持(英文,法文中文等都支持)

注意,第二参数要声明字符编码

```
$str2 = '中国人';  
echo mb_strlen($str2,'utf-8');
```

字符串位置

strpos—查找字符串首次出现的位置

查找一个子串在一个大串中首次出现的位置,计算机中,位置从0记数

```
$str = 'abcde';
```

```
echo strpos($str,'c');
```

注意:

此函数可能返回布尔值 FALSE，但也可能返回等同于 FALSE 的非布尔值，因为字符串位置是从0开始，而不是从1开始的。

```
$str = 'abcde';
if(strpos($str,'a') == false) {
    echo '不存在';
} else {
    echo '存在';
}

if(strpos($str,$str2) === false) {
    echo '不存在';
} else {
    echo '存在';
}
```

替换字符串

str_replace

strtr

str_replace — 子字符串替换

```
$reply = 'fuck you';
echo str_replace('fuck', 'f**k', $reply);
```

网站公开发表或者评论,应该转换为'文明用语'是很必要的.涉及非法发表言论是不允许的.

strtr — 转换指定字符

替换一批字符串,批量替换

第二个参数为数组,键是待替换的,值是替换后的

1)第一种写法

```
$str = '男人,女人,男孩,女孩';
echo strtr($str,array('男'=>'女','女'=>'男'));
//这个是将字符串看成一个整体,不会出现乱码
```

2)第二种写法

```
$str = '男人,女人,男孩,女孩';
echo strtr($str,'男','屁'), '<br >';
echo strtr($str,'人','屁');
//此行是按字节
```

替换,中文易乱码,推荐上一种用法

截取子字符串

substr

substr — 返回字符串的子串

substr('大字符串','从哪个位置开始截取',[截取几个])

从0开始截取

(第三个参数不写默认截取到大串最后)

1)

```
$str = 'helloworld';  
echo substr($str,5), '<br >';  
echo substr($str,0,5), '<br >';  
echo substr($str,3,5);
```

2)始终保持截取字符串后三位,倒着来,用负数

```
echo substr($str,-3), '<br >';  
echo substr($str,-3,1);
```

3)第3个 为整数: 代表截取的长度

如果为负 代表结束位置,从后往前数.

```
echo substr($str,-5,-2), '<br >';
```

拆分字符串

explode

explode — 使用一个字符串分割另一个字符串,返回一个数组

拆分字符串,按某个指定的标志,把字符串拆的数组

1)发表博文的时候会发布标签

```
$str='php,mysql,study';
```

在写入数据库的时候我们需要将这几个标签拆成独立的标签

```
$str='php,mysql,study';  
print_r(explode(',', $str));
```

拼接字符串

implode

implode — 将一个一维数组的值转化为字符串

1)将一个数组的值,拼接为字符串

```
$arr = array('name'=>'zhangsan', 'age'=>23, 'gender'=>'man');  
echo implode('/', $arr);
```

分析文件后缀名

strpos 字符串位置

substr 截取字符串

```
$file = 'a.jpg';  
echo substr($file, strpos($file, '.'), '<br >');  
echo substr($file, strpos($file, '.')+1);
```

15课 数组定义

数组是php中,非常常用非常重要的一个数据结构,

php本身的数组非常的强大

如何来定义一个数组?

数组是一种复合数据,可以装下多个值,每个值用不同的键来区分.(键值对应)

键 -> 箱子上的编号

值 -> 箱子里面的内容

```
$arr = array('name'=>'zhangsan','age'=>19);  
print_r($arr);
```

16课 操作数组单元

数组是一个复合数据,里面放了很多数据

print_r()将里面所有的数据都打印了出来

需要对数组的具体单元进行操作

数组单元的 增,删,改,查

1.单独取出某一个单元的值,用键来取 \$数组名[键]

```
$arr = array('a','b','c');  
echo $arr[1], '<br >';
```

2.取出二维数组的值

数组的层级顶多到3层,否则人无法直观的理解

```
$arr2 = array('name'=>'lisi','hobby'=>array('basketball','football','pingpang'));  
echo $arr2['hobby'][0];
```

3.更改一个数组单元的值

```
$arr2['hobby'][0] = 'swiming';  
print_r($arr2);
```

4.增加一个数组单元

```
$arr2['area'] = 'beijing';  
print_r($arr2);  
  
$arr2['area'] = 'shanghai';  
print_r($arr2);
```

5.删除数组单元

```
unset($arr2['hobby']);  
print_r($arr2);
```

17课 遍历数组

1.取出数组的每个单元

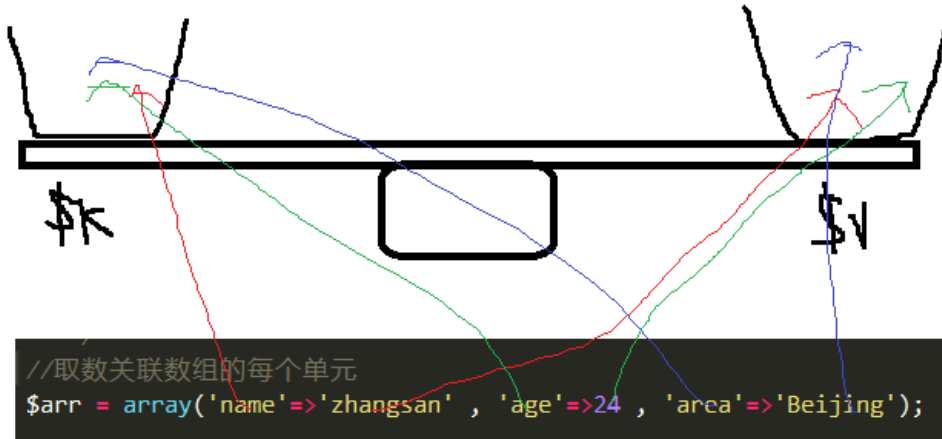
```
$arr = array('a','b','c','d','e');  
echo $arr[0], '<br >';  
echo $arr[1], '<br >';  
echo $arr[2], '<br >';  
echo $arr[3], '<br >';  
echo $arr[4], '<br >';
```

索引数组可以这样

```
for($i=0;$i<count($arr);$i++) {  
    echo $arr[$i], '<br >';  
}
```

2.关联数组如何取出每个数组单元?

它的键是字符串,没有任何规律
托盘称图讲解foreach循环遍历数组



foreach是专门用来循环数组的,速度非常快
foreach 里面的键值的变量名(\$k,\$v) 为任意合法的变量名

```
$arr = array('name'=>'zhangsan' , 'age'=>24 , 'area'=>'Beijing');

foreach($arr as $k=>$v) {
    echo $k.':',$v,'<br >';
}
```

3.只循环值

```
foreach($arr as $v) {
    echo $v,'<br >';
}
```

只循环键

```
foreach($arr as $k) { //错误的
    echo $v,'<br >';
}

foreach($arr as $k=>){ //语法错误
}

```

4.array_keys — 返回数组中所有的键名

foreach 没有办法单循环取出键

5.把下面这个的数组的每个单元值变成原来的 2 倍

```
//
$stu = array('lisi'=>3,'wang'=>5,'zhao'=>6);

$stu = array('lisi'=>3,'wang'=>5,'zhao'=>6);
print_r($stu);

foreach({}){

print_r($stu); //array('lisi'=>6,'wang'=>10,'zhao'=>12);
}
```

10分钟时间完成这道题

1)常见错误1:

```
foreach($stu as $k=>$v){
    echo $v*2;
}
print_r($stu);
```

2)常见错误2:

```
foreach($stu as $k=>$v){
    $v = $v*2;
}
print_r($stu);
```

3)正确:

```
foreach($stu as $k=>$v) {
    $stu[$k] = $v*2;
}
print_r($stu);
```

18课 函数概念

函数就是一段封装起来的代码,可以随时调用.

1.如果有三个人来北京

```
echo 'welcome ' ;
echo 'to ' ;
echo 'Beijing!<br >';

echo 'welcome ' ;
echo 'to ' ;
echo 'Beijing!<br >';

echo 'welcome ' ;
echo 'to ' ;
echo 'Beijing!<br >';
```

我们需要重复三次,那再来一个人,能否用简单的办法招呼人

函数 -> 几行函数封装起来,再起个名字

```
function wel(){//复合语句
    echo 'welcome ' ;
    echo 'to ' ;
    echo 'Beijing!<br >';
}
wel();
wel();
wel();
```

2.函数可以是一个计算器(通过调用函数,算出两个数的和)

```
function add($a,$b) {
    echo $a+$b;
}
add(2,3);
```

3.函数可以把运算结果返回给调用者

```
function add($a,$b) {
    return $a+$b;
}
echo add(3,5);
```

此计算器的运算过程如下

```
function add($a,$b) {  
    //赋值过程,虽然人眼看不到,但是内部确实如此  
    $a = 3;  
    $b = 5;  
    return $a+$b;  
}  
echo add(3,5);
```

函数,如果没有任何参数

它就是纯粹的一段封装代码,外界调用这个函数的时候,就会将此函数中的代码执行一下.

如果有参数,那就相当于一个加工机器

将外部传来的参数,就是加工机的原料,将参数带到函数内部进行加工.再返回一个结果出来;

19课 函数定义

定义格式是死记硬背的,没有什么技巧

注意:

无论函数有没有参数[可以没有参数],都需要在函数名后面加上小括号

```
function 函数名([参数1],[参数2],...[参数n]) {  
    //函数体,就是php语句  
    return 某值/表达式//  
}
```

return的作用:

函数就好比一个豆浆机器,参数就像黄豆和水

我们将黄豆和水放进去, 如果没有产出豆浆,那就没有任何作用;

return返回一个结果

谁能捕捉到这个调用结果呢?调用结束后,调用行可以捕捉return的结果

函数的命名规范:

对于函数,命名规则和变量是一样的,但是函数不区分大小写.

注意:

虽然不区分大小写,但是声明时和调用时大小写要保持一致

```
function jia() {  
    return $a = 3;  
}  
echo JIA();
```

20课 时间戳格式化

date()

注意:

如果我们格式化后时间跟现在有时差

```
找到php.ini  
将时区更改为以下即可  
date.timezone = PRC
```

date — 格式化一个本地时间 / 日期

时间戳好记录,但是对人眼直观上没有什么意义,不够直观

我们在前台显示出来需要将记录下来的时间戳格式化之后再显示.

1)前一天的时间

```
$lastday = time()-24*3600;  
echo date('Y年m月d日 H:i:s' , $lastday),'<br >';
```

2)不写第二个参数,默认是time();当前时间戳

```
echo date('Y年m月d日 H:i:s'),'<br >';  
  
echo date('Y-m/d H:i:s' , $lastday),'<br >';
```

3)国外写法,将年份放在后面也是可以的

```
echo date('m/d Y H:i:s' , $lastday),'<br >';
```

21课 超全局变量

特殊的数组,是系统提供给我们的.

特点:在页面的任意部分,无论是函数里,还是正常的页面里,都能随时获取到这几个变量,它不受作用域的干扰.这意味着它们在一个脚本的全部作用域中都可用,这就是超全局变量

超级变局变量

```
$_GET // 地址栏上获得的值  
$_POST // POST 表单发送的数据  
$_REQUEST // 既有 GET,也有 POST 的内容(GET和POST的并集)  
  
//后面会学到以下三个  
$_SESSION  
$_COOKIE  
$_FILES  
  
$_SERVER //web服务器的环境  
$_ENV // 服务器操作系统的环境变量,如操作系统型,linux,win,mac,环境变量等等  
  
$GLOBALS //引用全局作用域中可用的全部变量
```

面试题:<http://www.zixue.it/thread-266-1-1.html>

1.函数调用,写一个html,有post和get

\$_GET \$_POST \$_REQUEST

```
$arr = array(1,2);  
  
function t1() {  
    t2();  
}  
  
function t2() {  
    print_r($arr);  
    print_r($_GET);  
    print_r($_POST);  
    echo '<hr>';  
    print_r($_REQUEST);  
}  
  
t1();
```

2.GET和POST都有id,那么\$_REQUEST怎样显示

```
<!DOCTYPE html>  
<html lang="en">  
<head>
```

```

<meta charset="UTF-8">
<title>Document</title>
</head>
<body>
  <form action="1.php?id=3" method="post">
    <p>姓名:<input type="text" name="name" id=""></p>
    <p>email:<input type="text" name="emai" id=""></p>
    <p><input type="text" name="id" id="" value="999"></p>
    <input type="submit" value="submit">
  </form>
</body>
</html>

```

POST的优先级似乎要比GET的要高,这决定于php的配置文件,php.ini

request_order = "GP"

request有合并的意思,先合并GET后POST,后者覆盖前者

3. \$_SERVER — 服务器和执行环境信息

是一个包含了诸如头信息(header)、路径(path)、以及脚本位置(script locations)等等信息的数组

对于我们做网站而言,服务器指的是web服务器

web服务器就是给我们提供网页服务的这种环境的软件

在win下是apache,在linux下nginx

这里的环境指的是apache的一个运行环境

```

echo '<pre>';
print_r($_SERVER);

```

SERVER_SOFTWARE	服务用到的软件
SERVER_NAME	域名
SERVER_ADDR	服务器地址
SERVER_PORT	服务器端口
REMOTE_ADDR	远程地址(来访者的IP)
DOCUMENT_ROOT	文件根目录
SCRIPT_FILENAME	当前运行的文件

访问: <http://192.168.1.100/0903php/1.php>

4. \$_ENV (enviroment 环境) 操作系统的环境

打印为空

操作系统的环境和安全密切相关,别人上传个.php就可以轻易查看到我们系统的环境,或者黑客利用漏洞,运行个脚本,看到了操作系统环境,很容易给他的破解带来方便.

所以在php.ini里面禁用了这个超全局变量

variables_order = "GPCSE"

即使禁止使用ENV,我们也可以获取环境变量

getenv — 获取一个环境变量的值

```
echo getenv('USERNAME');
```

在实际使用中,是禁用掉ENV的,如果想使用ENV的值,则可以用getenv获取

5. \$GLOBALS 它就像是全局变量的一个花名册

//分析过程

```

$a = 1;
$b = 2;
function t1() {
    print_r($GLOBALS);
    $GLOBALS['a'] = 99;
}

```

```
t1();  
echo $a; //99
```