

主讲教师：传智.神龙教主

传智播客 PHP学院 <http://php.itcast.cn> 发布

## 简单数据类型（标量类型）

### 整型（见上一讲）

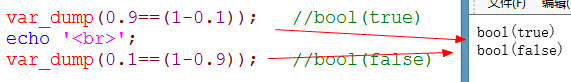
### 浮点数（float）

浮点数就是小数，在PHP中，float（单精度）和double(双精度)是一样的。

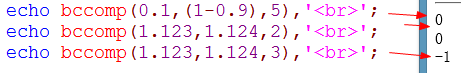
由于浮点数转成二进制是乘2正向去整，可能会一直循环下去。所有浮点数保存的值是近似值。

脚下留心：在C、java、C#语言中浮点数有两种，一种是单精度（float）,还有一种是双精度（double）,双精度的范围比单精度要广。

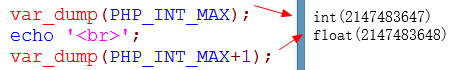
脚下留心：不能用浮点数数做比较



如果要比较浮点数通过bccomp(值1,值2,比较的小数位数)函数去比较，如果两个数相等就返回0，值1>值2返回1，否则返回-1



浮点数的范围比整型要大



理论上，浮点数的最大值是：1.8E+308

浮点数的精度有效位数是14位



### 布尔型（boolean）

布尔型就是true和false。

打印布尔型使用var\_dump().

### 字符串型

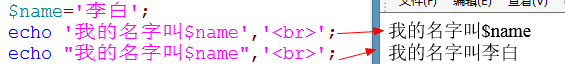
### 单引号和双引号

在javascript中，单引号字符串和双引号字符串一样的。

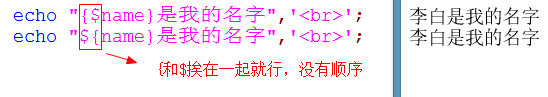
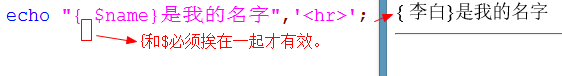
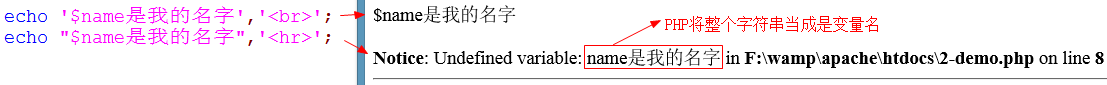
在PHP中，如果字符串内没有变量，显示结果单引号和双引号是一样的。



如果有变量，双引号中的变量要解析。

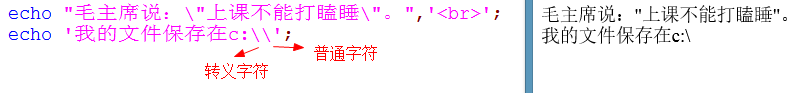


单引号是真正的字符串，双引号要做运算（将字符串中的变量替换成值）。因为双引号要运算，而单引号不需要运算，所以，单引号的执行效率比双引号要高。



### 转义字符（\）

将具有特殊含义的字符转成普通字符。



脚下留心：转义字符后面如果跟的是特殊字符，将特殊字符转成普通字符。如果转义字符后面跟的是普通字符，转义字符不起作用，就是一个普通字符



但是，要注意转义字符和r、n、t的组合，\r组合表示回车，\n表示换行，\t表示制表符。



### ASC||

计算机内部使用二进制计算。键盘上的每个字符都有一个编码和他对应，这个

编码就是ASC||码。

ASC||码用7位的二进制数来表示，可以表示2^7次方个字符（128个）。

1. ASC||是连续的。连续的字符之间ASC码也是连续的。
2. 小写的字母比大写的大 a>A
3. 小写字母和大写字母之间相差32 a-A=32

考试题：已经A=65,求d

d-D=32

d-(A+3)=32

d=32+A+3

d=100

ord()将字符转成十进制ASC||码

chr()将十进制的数字转成字符。

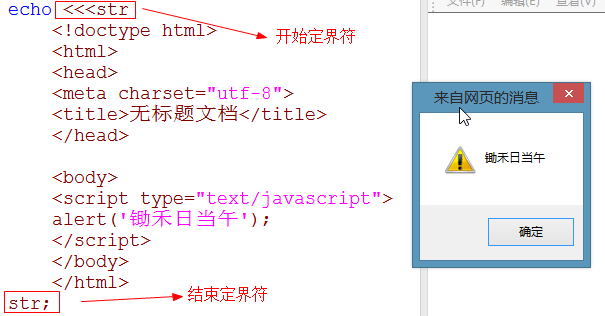


“\“可以转义ASC码。



### 使用字符串定界符

1. 作用：为输出大量的文本并且不用频繁的转义提供了简单方法
2. 开始和结束的定界符必须一样。名字由字符、数字、下划线组成，不能以数字开头。
3. 开启标识符前面有三个小于号“<<<“。
4. 结束标识符必须顶格写，前面不能有空格和其他字符，标识符的后面也不能有空格。



字符串定界符分为heredoc和nowdoc

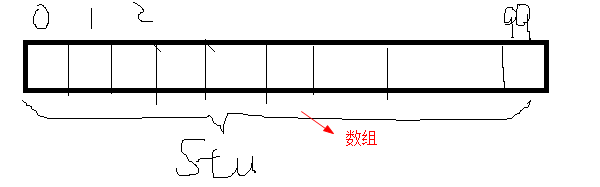
|  |  |
| --- | --- |
| heredoc |  |
| nowdoc |  |

heredoc相当于是双引号字符串

nowdoc相当于是单引号字符串。

## 复合数据类型

### 数组

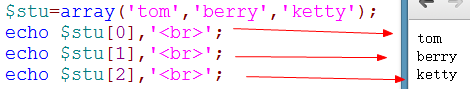


数组：计算机内存中一段连续的空间

数组分为：索引数组和关联数组

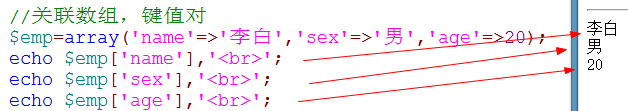
### 索引数组

通过元素的位置作下标。



### 关联数组

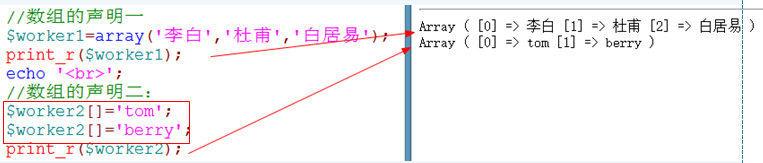
数组的值和一个字符串做关联，形成一个键值对。



### 数组的声明

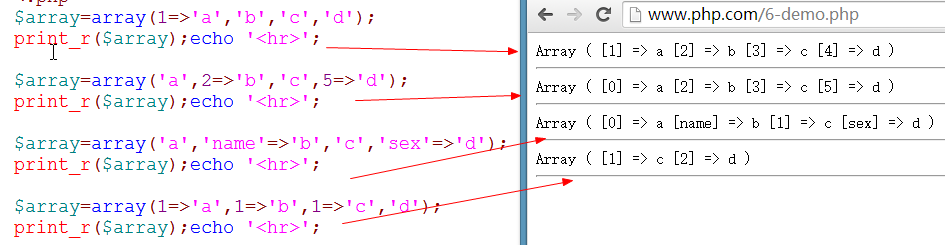
方法一：变量=array();

方法二：通过[]的方式

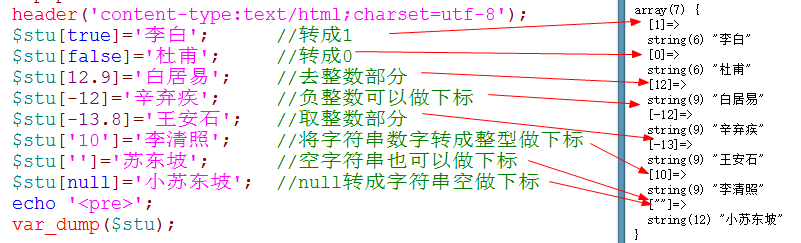


默认情况下：在索引数组中，第一个元素的下标是0，后面依次递增1.

### 数组下标练习

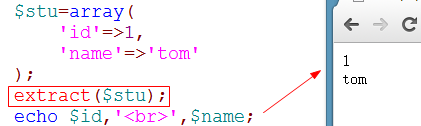


### 数组下标的思考



### extract()

从数组中将变量导入到当前的符号表中。键名做变量名，键值做变量的值。

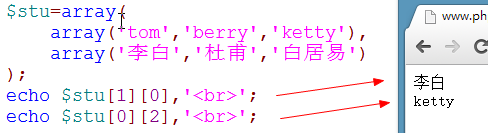


### 数组的分类

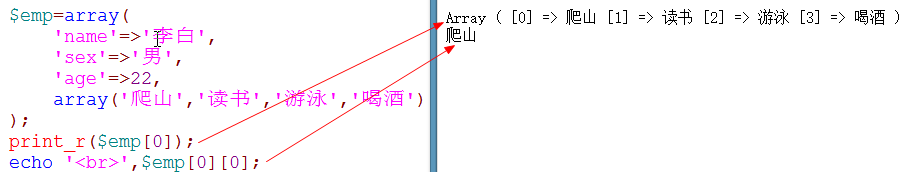
1. 按照元素的键值关系分为：
   1. 索引数组：通过元素的位置坐下标。
   2. 关联数组：通过字符串做下标，键和值存在某种关联。

注意：不能按照键的数据类型来判断是否是索引数组或关联数组，必须看下标和值关联关系，$stu=array(1=>’tom’,2=>’berry’)也属于关联数组。因为下标不是表示值的位置。

1. 按照数组的结构分
   1. 一维数组
   2. 多维数组：二维以上



注意：PHP中的二维数组和强数据类型不一样，它不是一个二维表，PHP数组的分类是应用层面的分类，不是PHP语言本身的分类。



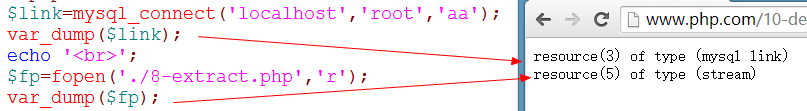
### 对象

在面向对象中详解

## 特殊类型

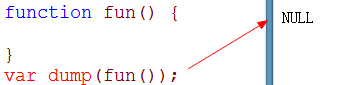
### 资源

1. PHP引用的外部数据称为资源
2. 资源只能获取，不能创建
3. 资源也是由类别的，比如连接资源，数据流资源



### NULL

1. 表示没有定义的变量
2. NULL不区分大小写。
3. 调用一个没有返回值的函数，默认返回NULL

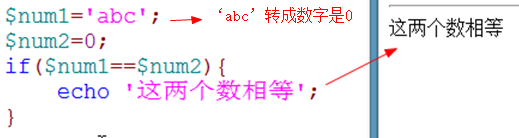
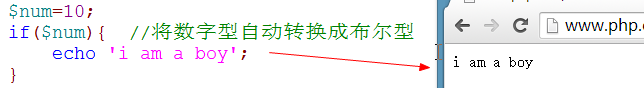


## 数据类型转换

数据类型是数据在内存中存储的结构，和建立在该数据结构上的操作算法的集合。这就意味着不同的数据类型运算操作可能不一样。不同类型的数据在运算过程中要注意类型的转换。

### 自动转换

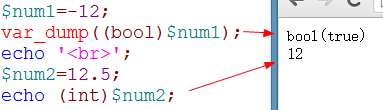
运算过程中需要的数据类型和提供的数据类型不一致。



$num1和$num2要比较，必须转成相同的数据类型才能比较，一个是字符串一个是数字它们都转成数字型再比较。

### 强制转换

语法：(数据类型)数据



### 其他数据类型和bool型之间的转换规律

1. 字符串转换规律：非0非空为true，0或空为false
2. 数字转换规则：0为假，非0为真。

//字符串和bool的转换规律

var\_dump((bool)'abcd'); //true

var\_dump((bool)''); //false

var\_dump((bool)'0'); //false

var\_dump((bool)'0.0'); //true

var\_dump((bool)'00'); //true

var\_dump((bool)'false'); //true

var\_dump((bool)'null'); //true

//数字和bool的转换规律

var\_dump((bool)1); //true

var\_dump((bool)0); //false

var\_dump((bool)-10); //true

var\_dump((bool)1.1); //true

var\_dump((bool)0.0); //false

var\_dump((bool)-1.1); //true

//数组和bool的转换规律

var\_dump((bool)array()); //false

var\_dump((bool)array(1)); //true

var\_dump((bool)array(false));//true

//数组和null的转换规律

var\_dump((bool)null); //false

## 运算符

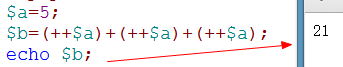
### 算术运算符

|  |  |
| --- | --- |
| 一元运算符 | 二元运算符 |
| 负号（一元减） | + |
| ++ | - |
| -- | \* |
|  | / |
|  | % |

脚下留心：在js中，+可以做数字和字符串相加。在PHP中，+只能做数字相加。

++是递增1，--是递减1，要注意++前置和++后置

++前置是先自增，++后置是后自增



### 关系运算符

>

>=

<

<=

==

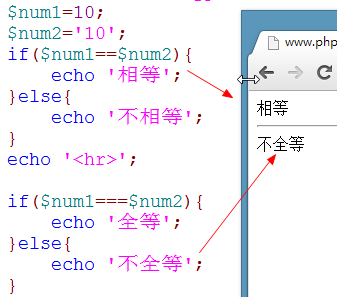
!=

===

!==

比较运算符的结果不是true就是false。

注意==和===的区别：==只做值的比较，===做值和数据类型的比较



### 逻辑元素符

&& 与

|| 或

！ 非

逻辑运算符的作用是用来连接比较运算符的。

true && true true

true && false false

false && true false

false && false false

true || true true

true || false true

false || true true

false || false false

!true false

!false true

练习1：写出如下表达式

评选三号学生的条件是:语文($ch)和数学（$math）必须都要超过90分

$ch>90 && $math>90

!($ch<=90 || $math<=90)

练习2：写出如下表达式

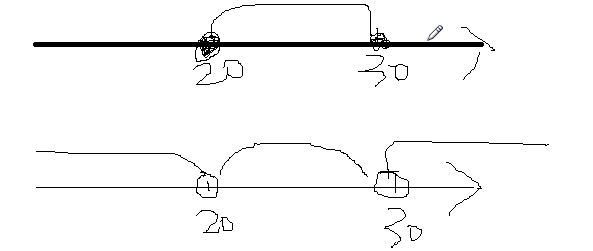
评选三号学生的条件是:语文($ch)或数学（$math）不低于90分

$ch>=90 || $math>=90

!($ch<90 && $math<90)

练习3：写出如下表达式

年龄($age)在20到30之间



$age>=20 && $age<=30

!($age<20 || $age>30)

多学一招：非运算符去括号，每个表达式取非，或变与，与变或。

!($age<20 || $age>30)

= !($age<20) && !($age>30)

= $age>=20 && $age<=30

!($ch<90 && $math<90)

= !($ch<90) || !($math<90)

= $ch>=90 || $math>=90

### 字符串连接符（.）



### 赋值运算符

=

a+=b a=a+b

-=

\*=

/=

%=

a.=b a=a.b

### 三元运算符

语法：表达式?值1:值2



## 注释

1. HTML注释： <!-- HTML注释 -->
2. CSS注释： /\* CSS注释 \*/
3. Javascript注释
   1. 单行注释： //单行注释
   2. 多行注释： /\*多行注释\*/
4. PHP注释
   1. 单行注释: //单行注释 #单行注释
   2. 多行注释： /\* 多行注释 \*/
5. MySQL注释：
   1. 单行注释： --单行注释
   2. 多行注释： /\* 多行注释 \*/

## 判断语句

### 单分支

语法：

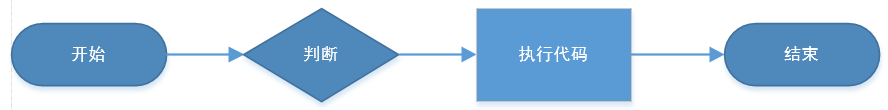
if(条件){

//代码块

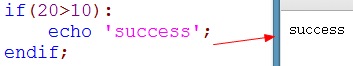
}

如果条件成立就执行代码块，条件不成立就不执行。

流程图：



替代语法：



多学一招：如果if块中就一句代码，则大括号是可以省略的。

### 双分支

语法：

if(条件){

//代码1

}

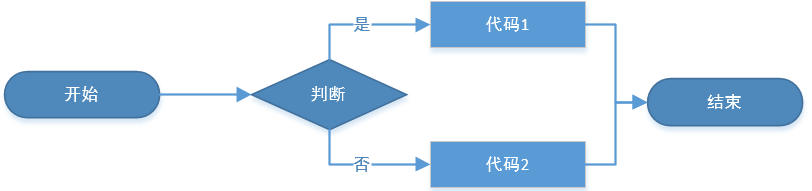
else{

//代码2

}

如果条件成立就执行if一段，否则执行else一段。

流程图：



替代语法：

if(条件):

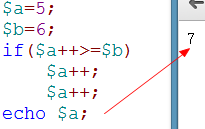
//代码1

else:

//代码2

endif;

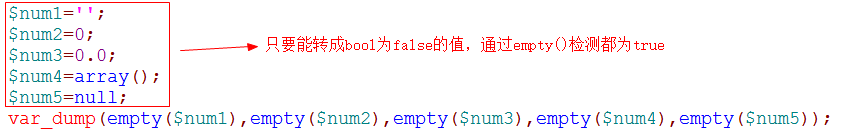
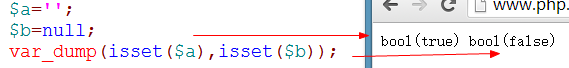
练习



## 判断变量是否为空

在处理一个没有定义的变量时候会报错，所以在处理变量之前，我们先要判断一下变量，判断的函数常用如下两个

|  |  |
| --- | --- |
| 函数 | 说明 |
| isset() | 判断变量是否有值，并且值不为null |
| empty() | 判断变量是否为空 |



## 判断闰年

1. 能被4整除，但不能被100整除
2. 能被400整除

### 知识点：

1. trim()：去除数据两边的空格
2. is\_numeric()：判断是否是数字或数字字符串
3. is\_int()：是否是整型

<?php

//判断是否按了提交按钮，方法一

/\*

if(isset($\_POST['button'])){

//按了提交按钮

}

\*/

//判断是否按了提交按钮，方法二

if(!empty($\_POST)){

$year=trim($\_POST['year']); //获取年份并去除两边的空格

if($year=='')

echo '您没有输入年份';

else {

if(is\_numeric($year)){ //判断是否是数字或数字字符串

$year+=0; //将字符串转成数字型

if(is\_int($year)){

if($year>=0){

if($year%4==0 && $year%100!=0 || $year%400==0)

echo "{$year}是闰年";

else

echo "{$year}是平年";

}else{

echo '年份必须是正整数';

}

}else {

echo '您输入的是小数';

}

}else {

echo '您输入的不是数字';

}

}

echo '<hr>';

}

?>

<form method="post" action="">

请输入年份： <input type="text" name="year"> <br />

<input type="submit" name="button" value="判断闰年">

</form>

