PHP经典格式：

<?php

echo “Hello World!”

?>

PHP变量要以**$**开始，如变量X必须写成$X

变量首字母不能以数字开头，且变量名字区分大小写

PHP变量作用域：**local global static parameter**

要在函数中访问全局变量，需要使用**global**关键字

echo输出的字符串可以带HTML的标签

**var\_dump($x)** 可以输出变量x的类型、长度和值

在PHP语言中声明字符串有三种方式：

1.    用单引号声明

2.    用双引号声明

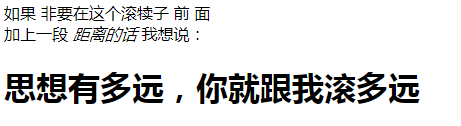
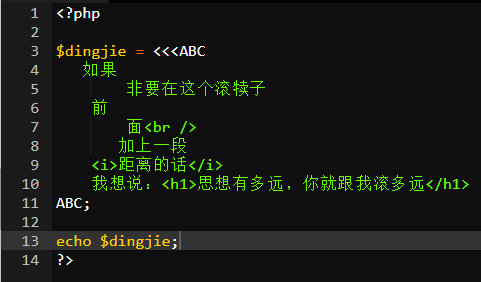
3.    用字界符声明（需要输入非常大段的字符串时使用）

**字界符声明**  
1).在变量后面的等号写三个小于号（<<<）。

2).然后在<<<后面写上字符（建议英文大写字符）。如下例中的：ABC

3).然后换行写上任意想写的字符

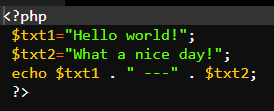
4).写完后，顶行。在行最开始处，再写上<<<后面的字符和分号。



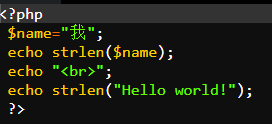
数组(**Array**)声明：

$cars=array("Volvo","BMW","SAAB");

在PHP中，只有一个字符串运算符。叫做**并置运算符**（.点号），用于把两个字符串值连接起来。



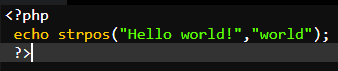
**strlen()** 函数返回字符串的长度（字符数）。



注意：在UTF-8下 strlen 把中文字符算成3个字节，英文，空格，符号占 1 个字节。

**提示**：strlen() 常常用在循环和其他函数中，因为那时确定字符串何时结束是很重要的。（例如，在循环中，我们需要在字符串中的最后一个字符之后结束循环。）

**strpos()** 函数用于在字符串内查找一个字符或一段指定的文本。

输出6

注意：第一个字符的位置是0。

PHP**常量**的值被定义后，在任何地方都不能被改变；

有效的常量名以字符或下划线开头（前面没有$）；

常量名一般大写；

常量贯穿整个脚本是**自动全局**的。

如需设置常量，请使用 **define()** 函数 - 它使用三个参数：

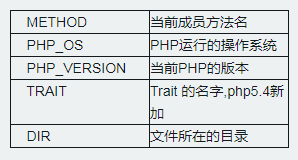
1.   首个参数定义常量的名称

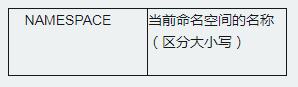
2.   第二个参数定义常量的值

3.   可选的第三个参数规定常量名是否对大小写不敏感。默认为 false ，即对大小写敏感。

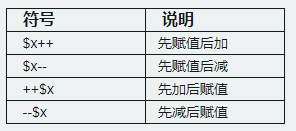
系统内置常量：





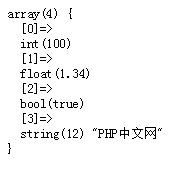
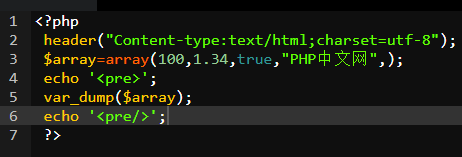


PHP递增/递减运算符





PHP数组可以存入多个不同类型的数据，是一个**复合数据类型**。



注：上例中echo pre这个标签的主要目的是为原样输出，格式展示的更好看，更清晰。

1. array(size = 4) 说明里面有4个元素

2. 0 => int 100我们知道int是整型的意思，100是一个整型的数值。那前面的0,1,2,3和=>代表什么意思呢？

3. 最新前的0，1，2，3代表的是值的读取标识号，我们称之为下标或者键（英文：key）

4. => 是一个符号标准叫法叫作：**键值对应符**。因此，以后再看到 0=> int 1 可以这样来说。 下标访问符0对应整型的1。

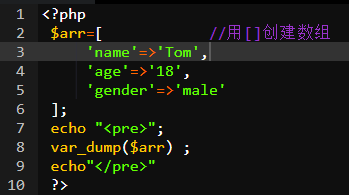
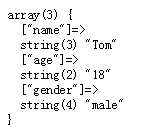
5. 我们还称数组里面的键值对为**元素**，元素就是键值对的组合。

通过上例，我们就完成了数组其中的一种声明方式：索引数组的声明。所谓索引数组：就是下标全为整型的数组。

1. 索引数组若不强制声明他的下标，他的下标是从0开始的。（我们的第一个数组的值：只有不断努力才能博得未来。这个值的下标为0）。

2. 如果我指定过下标他的下标就为我指定的值。如下标为5,10和下标为19的，都是我指定过的值。

3. 若某个值（如php.cn），强制指定了下标（下标为10）。在它后面加上的值（PHP学院），不指定下标的话。他们的下标增长规律为最大值+1。

**count(**$变量) 函数用于返回数组的长度（元素数**）；**

**strlen(**$变量) 函数用于检测字符串实际长度。

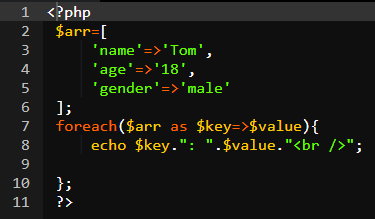
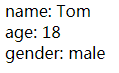
**foreach的基本语法**

有一个布尔型循环是专门用来循环数组的。这个循环的基本语法就是**foreach**基本语法

foreach( 要循环的数组变量 as [键变量 =>] 值变量){

    //循环的内容

}

通过上面的运行结果我们得出下面的结果：

1.    每次循环的时候，把下标赋值给了变量$key，把值的变量赋值给了变量$value

2.    循环一次读一次键和值。读到最后，发现没有可以读取的数组元素后，停止循环遍历数据。

注意：$key 和$value并不是变量名必须得为这两个名字。你命名为其他的也可以，如 $kai => $wen是一样的。 你要清楚键赋值给了哪个变量，值赋值给了另外的哪个变量。

**数组排序函数**

· sort() - 对数组进行升序排列

注：sort大多是用来排序数字索引数组的，如果把一个关联数组放到sort里排序，那么数组的键会丢失

· rsort() - 对数组进行降序排列

· asort() - 根据关联数组的**值**，对数组进行升序排列

· ksort() - 根据关联数组的**键**，对数组进行升序排列

· arsort() - 根据关联数组的**值**，对数组进行降序排列

· krsort() - 根据关联数组的**键**，对数组进行降序排列

**PHP 超级全局变量列表:**

· $GLOBALS

$GLOBALS 是PHP的一个超级全局变量组，在一个PHP脚本的全部作用域中都可以访问。

$GLOBALS 是一个包含了全部变量的全局组合数组。变量的名字就是数组的键。

· $\_SERVER

$\_SERVER 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。这个数组中的项目由 Web 服务器创建。不能保证每个服务器都提供全部项目；服务器可能会忽略一些，或者提供一些没有在这里列举出来的项目。

· $\_REQUEST

PHP $\_REQUEST 用于收集HTML表单提交的数据。

· $\_POST

 $\_POST 变量用于收集来自 method="post" 的表单中的值

· $\_GET

PHP $\_GET 同样被广泛应用于收集表单数据，在HTML form标签的指定该属性："method="get"。

 从带有 GET 方法的表单发送的信息，对任何人都是可见的(会显示在浏览器的地址栏)，并且对发送的信息量也有限制(最多 100

· $\_FILES

· $\_ENV

· $\_COOKIE

· $\_SESSION

**PHP八个魔术变量：**

· \_\_LINE\_\_

· \_\_FILE\_\_ 文件的完整路径和文件名。

· \_\_DIR\_\_ 文件所在的目录。

· \_\_FUNCTION\_\_

· \_\_CLASS\_\_

· \_\_TRAIT\_\_

· \_\_METHOD\_\_

· \_\_NAMESPACE\_\_