变量名必须以字母或者下划线字符开始

变量名是区分大小写的

PHP 是一门弱类型语言

不必向 PHP 声明该变量的数据类型。

强类型语言，必须在使用变量前先声明（定义）变量的类型和名称。

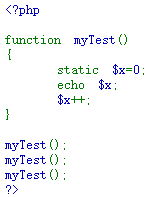
PHP 有四种不同的变量作用域：

* local
* global
* static
* parameter

要在一个函数中访问一个全局变量，需要使用 global 关键字。

全局变量存储在一个名为 $GLOBALS[index] 的数组中。 *index* 保存变量的名称。这个数组可以在函数内部访问，也可以直接用来更新全局变量。

当一个函数完成时，它的所有变量通常都会被删除。希望某个局部变量不要被删除，请在第一次声明变量时使用 static 关键字

C:\Users\林\AppData\Local\Temp\1573961955(1).png似乎第二次执行函数的时候，static $x=0这行代码被忽略了，该变量仍然是函数的局部变量。

* echo - 可以输出一个或多个字符串
* print - 只允许输出一个字符串，返回值总为 1

echo "这是一个", "字符串，", "使用了", "多个", "参数。";



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

常量名不需要加 $ 修饰符

define("GREETING", "欢迎访问 Runoob.com", true); 设置为 TRUE，该常量则大小写不敏感。默认是大小写敏感的。

常量在定义后，默认是全局变量，可以在整个运行的脚本的任何地方使用。

即便常量定义在函数外也可以正常使用常量。

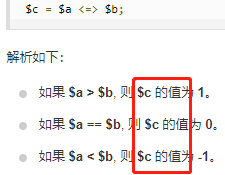
strlen("Hello world!");

strpos("Hello world!","world");该函数会返回第一个匹配的字符位置。如果未找到匹配，则返回 FALSE。

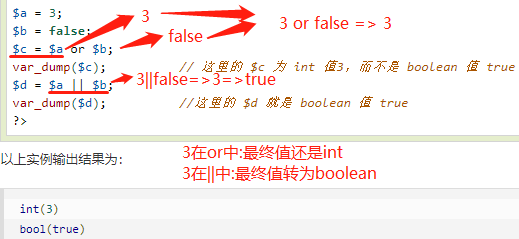
x xor y异或，如果 x 和 y 有且仅有一个为 true，则返回 true

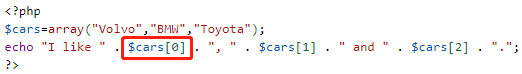


组合比较符（combined comparison operator）也称之为太空船操作符，符号为 **<=>**。组合比较运算符可以轻松实现两个变量的比较，当然不仅限于数值类数据的比较。



*// 优先级： &&  >  =  >  and*  
*// 优先级： ||  >  =  >  or*



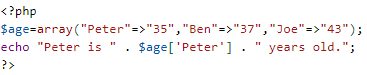


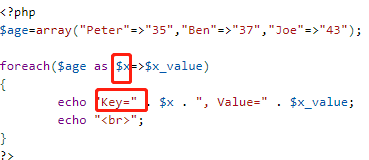
比较一下下面，怎么获取数组的值

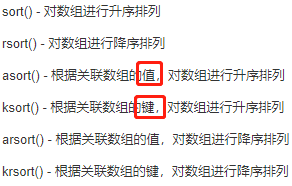


$cars=array("Volvo","BMW","Toyota");  
echo count($cars);

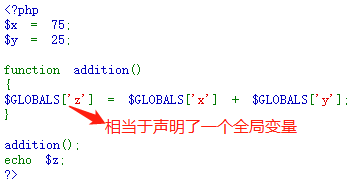
## 关联数组







可以在函数内部访问，也可以直接用来更新全局变量。



$\_SERVER 是一个包含了信息(header)、路径(path)、以及脚本位置(script locations)等的数组。这个数组中的项目由 Web 服务器创建。

C:\Users\林\AppData\Local\Temp\1573979611(1).png

$\_SERVER['SERVER\_ADDR'] 当前运行脚本所在的服务器的 IP 地址。

$\_SERVER['SCRIPT\_NAME'] 包含当前脚本的路径。

$\_SERVER['SERVER\_PORT'] Web 服务器使用的端口。

$\_SERVER['SCRIPT\_FILENAME'] 当前执行脚本的绝对路径。

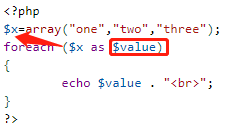
$\_SERVER['REMOTE\_PORT'] 用户机器上连接到 Web 服务器所使用的端口号。

$\_SERVER['REMOTE\_HOST'] 浏览当前页面的用户的主机名。

$\_SERVER['REQUEST\_TIME'] 请求开始时的时间戳。

$\_SERVER['REQUEST\_METHOD'] 访问页面使用的请求方法

$\_REQUEST 用于收集HTML表单提交的数据。



# 魔术常量

**\_\_LINE\_\_** 行

该文件位于**\_\_FILE\_\_** E:\wamp\www\test\index.php

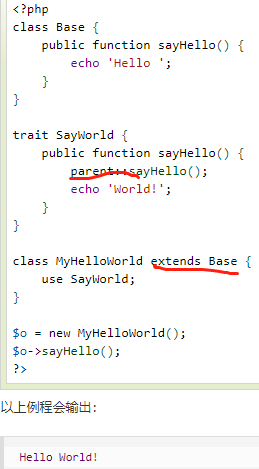
\_\_DIR\_\_

该文件位于 “ E:\wamp\www\test ”

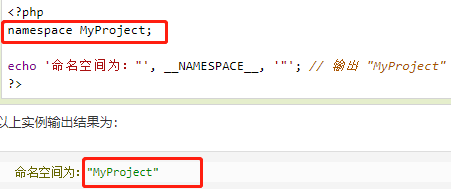
函数名为： **\_\_FUNCTION\_\_**

类名为：'  . **\_\_CLASS\_\_**

实现了代码复用的一个方法，称为 traits



函数名为：' . **\_\_METHOD\_\_**



命名空间：1用户编写的代码与PHP内部的类/函数/常量或第三方类/函数/常量之间的名字冲突。

2为很长的标识符名称(通常是为了缓解第一类问题而定义的)创建一个别名（或简短）的名称，提高源代码的可读性。

在同一个文件中定义不同的命名空间代码

在声明命名空间之前唯一合法的代码是用于定义源文件编码方式的 declare 语句。所有非 PHP 代码包括空白符都不能出现在命名空间的声明之前。

declare(encoding='UTF-8');

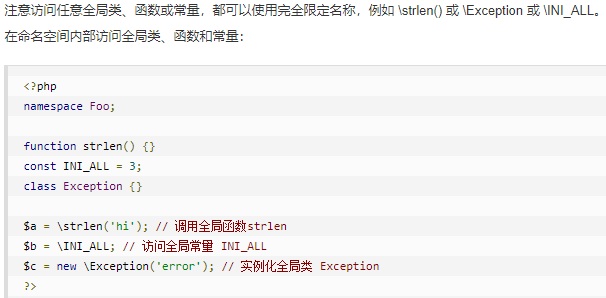
命名空间必须是程序脚本的第一条语句

命名空间的名字可以使用分层次的方式定义：

namespace MyProject\Sub\Level; //声明分层次的单个命名空间

**非限定名称，或不包含前缀的类名称，**例如 $a=new foo(); 或 foo::staticmethod();。如果当前命名空间是 currentnamespace，foo 将被解析为 currentnamespace\foo。如果使用 foo 的代码是全局的，不包含在任何命名空间中的代码，则 foo 会被解析为foo。 警告：如果命名空间中的函数或常量未定义，则该非限定的函数名称或常量名称会被解析为全局函数名称或常量名称。

**限定名称,或包含前缀的名称，**例如 $a = new subnamespace\foo(); 或 subnamespace\foo::staticmethod();。如果当前的命名空间是 currentnamespace，则 foo 会被解析为 currentnamespace\subnamespace\foo。如果使用 foo 的代码是全局的，不包含在任何命名空间中的代码，foo 会被解析为subnamespace\foo。



**多态** − 多态性是指相同的函数或方法可作用于多种类型的对象上并获得不同的结果。不同的对象，收到同一消息可以产生不同的结果，这种现象称为多态性。

**重载** −函数或者方法有同样的名称，但是参数列表不相同的情形，这样的同名不同参数的函数或者方法之间，互相称之为重载函数或者方法。

**抽象性** − 抽象性是指将具有一致的数据结构（属性）和行为（操作）的对象抽象成类。一个类就是这样一种抽象。

**析构函数** − 析构函数(destructor) 与构造函数相反，当对象结束其生命周期时（例如对象所在的函数已调用完毕），系统自动执行析构函数。析构函数往往用来做"清理善后" 的工作（例如在建立对象时用new开辟了一片内存空间，应在退出前在析构函数中用delete释放）。

类的变量使用 **var** 来声明

$obj->getUrl();其中：->相当于js中的点（.），即调用函数或者获取属性

\_\_construct构造函数

\_\_destruct析构函数

PHP 不支持多继承

* **public（公有）：**公有的类成员可以在任何地方被访问。
* **protected（受保护）：**受保护的类成员则可以被其自身以及其子类和父类访问。
* **private（私有）：**私有的类成员则只能被其定义所在的类访问。

类属性必须定义为公有，受保护，私有之一。如果用 var 定义，则被视为公有。

<https://www.runoob.com/php/php-oop.html>

类中必须实现接口中定义的所有方法

任何一个类，如果它里面至少有一个方法是被声明为抽象的，那么这个类就必须被声明为抽象的。

定义为抽象的类不能被实例化。

被定义为抽象的方法只是声明了其调用方式（参数），不能定义其具体的功能实现。

继承一个抽象类的时候，子类必须定义父类中的所有抽象方法；另外，这些方法的访问控制必须和父类中一样（或者更为宽松）。例如某个抽象方法被声明为受保护的，那么子类中实现的方法就应该声明为受保护的或者公有的，而不能定义为私有的。

声明类属性或方法为 static(静态)，就可以不实例化类而直接访问。

静态属性不能通过一个类已实例化的对象来访问（但静态方法可以）。

由于静态方法不需要通过对象即可调用，所以伪变量 $this 在静态方法中不可用。

静态属性不可以由对象通过 -> 操作符来访问。

如果父类中的方法被声明为 final，则子类无法覆盖该方法。如果一个类被声明为 final，则不能被继承。

要执行父类的构造方法，需要在子类的构造方法中调用 **parent::\_\_construct()** 。

function \_\_construct() {

parent::\_\_construct(); // 子类构造方法不能自动调用父类的构造方法

print "SubClass 类中构造方法" . PHP\_EOL;

}

测试

PHP（＂PHP: Hypertext Preprocessor＂，超文本预处理器的字母缩写）

函数体内引用全局变量，必须在函数内部使用“global”关键字

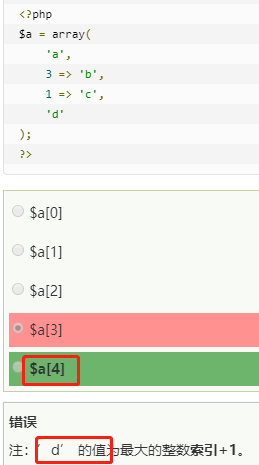
用户定义的函数和语言关键字对大小写不敏感。

注释方式//、/\*\*/、#

不区分大小写的关键字 NULL 用于没有定义值的情况。

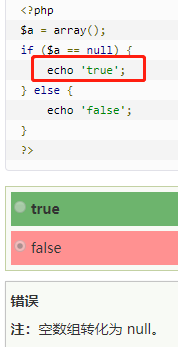
false==null

1 / 2值为0.5



以只读模式打开文件 “time.txt” 的正确方法是**fopen("time.txt","r");**

魔术常量: \_\_LINE\_\_\_\_FILE\_\_ \_\_CLASS\_\_\_\_METHOD\_\_\_\_FUNCTION\_\_



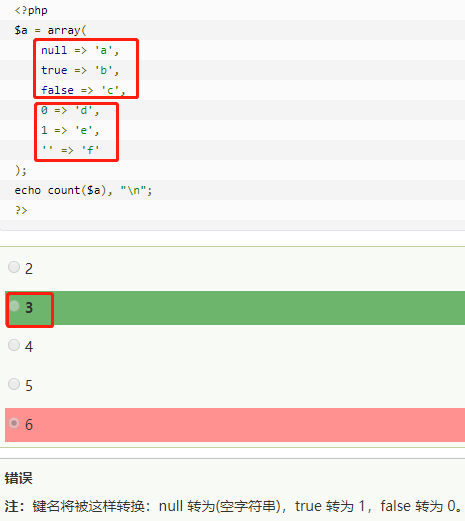
C:\Users\林\AppData\Local\Temp\1574004093(1).png

C:\Users\林\AppData\Local\Temp\1574004102(1).png

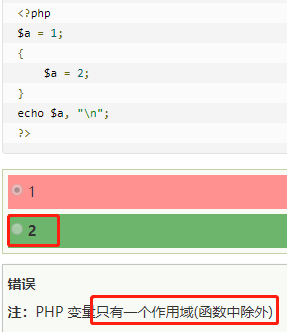
'2' == '02'//true

数值字符串作为整数进行比较。

变量名区分大小写







headers\_list() 函数返回已发送的（或待发送的）响应头部的一个列表，如需确定是否已发送报头，使用 headers\_sent() 函数。

如何在HTML <form> 中创建PHP数组? **<input name=”MyArray[]”>** 请注意在变量名后面的中括号，它就用来生成数组。

upload\_max\_filesize 默认值是 2MB。

 Joomla！是一个开源 CMS 软件

set\_time\_limit — 设置脚本最大执行时间，默认为 30 秒。

支持 php 多版本同时运行

trim() 移除字符串两侧的字符

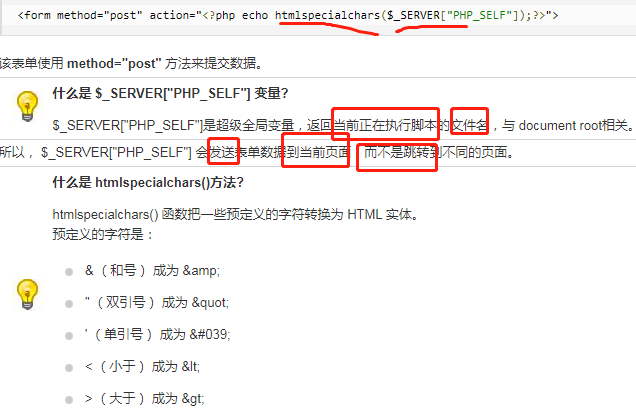
测试四完成





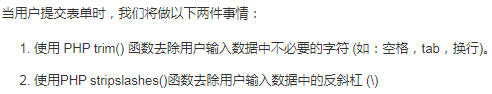






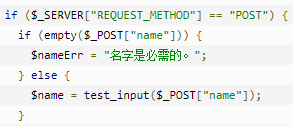
避免 $\_SERVER["PHP\_SELF"] 被利用?

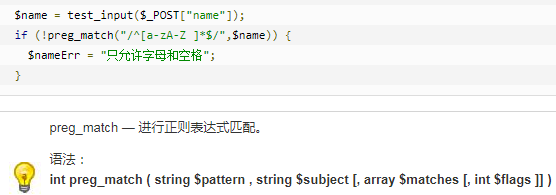
htmlspecialchars() 把一些预定义的字符转换为 HTML 实体。





php的 empty() 函数





PHP date() 函数用于格式化时间/日期。把时间戳格式化为可读性更好的日期和时间。可以在字母之间插入其他字符，比如 "/"、"." 或者 "-"可以在字母之间插入其他字符，比如 "/"、"." 或者 "-"date("Y/m/d") . "<br>";

* require 生成一个致命错误（E\_COMPILE\_ERROR），在错误发生后脚本会停止执行。
* include 生成一个警告（E\_WARNING），在错误发生后脚本会继续执行。