PHP将所有全局变量存储在一个名为$GLOBAL[index]数组中。

echo PHP\_EOL; // 换行符

## PHP echo 和 print 语句

echo 和 print 区别:

* 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

**提示：**echo 输出的速度比 print 快， echo 没有返回值，print有返回值1。

Var\_dump()函数返回变量的数据类型和值。

如： $x=100;var\_dump($x);

## PHP 并置运算符

PHP中，并置运算符(.)运用将两个字符串连接起来。

如 $text2=$text1.” “.$text2

strlen()函数返回字符串长度。

strplos()函数用于在字符串内查找一个字符或者一段指定文本，如果找到，返回第一个字符匹配的位置，否则返回false.

如 strpos(“hello world”,”world”);

如果查找中文，用mb\_strops()

三元运算符

(expr1) ? (expr2) : (expr3)

也支持写为expr1?:expr3,expr1 求值为 TRUE 时返回 expr1，否则返回 expr3。

Php 没有指针的概念，但可以用引用。￥c=&￥a。

组合比较运算符<=>,也叫太空船操作符，可以容易的实现两个变量的比较，但不仅限于数值的比较。

$c=$a<=>$b

解析如下：

* 如果**$a > $b**, 则 **$c** 的值为 **1**。
* 如果 **$a == $b**, 则 **$c** 的值为 **0**。
* 如果 **$a < $b**, 则 **$c** 的值为 **-1**。

用array()创建一个数组。

用函数count()获取数组的长度。如count($cars)。

PHP关联数组

关联数组是使用你分配给数组的指定键的数组，相似与字典。创建关联数组的两个方法

$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");

or:

$age['Peter']="35";

$age['Ben']="37";

$age['Joe']="43";

数组的遍历可以用普通的for循环遍历，也可以用foreach。

如 foreach(arrays as value){

echo value;}

其中arrays是数组。

PHP 数组排序函数

sort() -对数组升序排列

rsort() -对数组进行降序排列

asort() arsort() 根据关联数组的值，对数组进行升序(降序)排列

arsort() krsort() 根据关联数组的键，对数组进行升序(降序)排列

php魔术常量

\_\_LINE\_\_ 文件中的当前行号

\_\_FILE\_\_ 文件的完整路径和文件名

\_\_DIR\_\_ 文件所在目录

\_\_FUNCTION\_\_ 返回所在函数的函数名

\_\_CLASS\_\_ 类名

\_\_NAMESPACE\_\_命名空间

\_\_METHOD\_\_ 返回类名和函数名

\_\_TRAIT\_\_ 代码复用，比如，前面带trait 的类，后面的代码可以通过关键字use复用其代码。

<?php

trait Hello {

public function sayHello() {

echo 'Hello ';

}

}

trait World {

public function sayWorld() {

echo 'World';

}

}

class MyHelloWorld {

use Hello, World;

public function sayExclamationMark() {

echo '!';

}

}

$o = new MyHelloWorld();

$o->sayHello();

$o->sayWorld();

$o->sayExclamationMark();

?>

Php 中类里面的变量用var声明。

类成员的访问控制权限

* **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。

如果类方法中没有成员函数前面没有这些关键字，默认是public;

PHP中用关键字extends表示继承，如 class son extends father{…}

用interface定义接口，接口中的函数全部不用实现，只需要声明即可，由继承类实现，用关键字implements表示继承。

## 抽象类

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。

子类实现父类中的抽象函数的时候，可以增加函数参数，但不能减少函数参数。

## Final 关键字

PHP 5 新增了一个 final 关键字。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。

## 调用父类构造方法

PHP 不会在子类的构造方法中自动的调用父类的构造方法。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()** 。

声明常量可以用const,也可以用define.如define(PI,3.14)。但const只能在类中使用。

用date()函数获得当前日期，这里列出一些可用字符

* d - 代表月中的天 (01 - 31)
* m - 代表月 (01 - 12)
* Y - 代表年 (四位数)
* L – 代表星期几
* h – 小时
* i – 分钟
* s – 秒
* 可以在字母之间插入其他字符，比如 "/"、"." 或者 "-"，这样就可以增加附加格式了：
* <?php
* echo date("Y/m/d") . "<br>";
* echo date("Y.m.d") . "<br>";
* echo date("Y-m-d");
* ?>

**include 和 require 除了处理错误的方式不同之外，在其他方面都是相同的：**

* 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。

Php 的文件读写函数和其他语言差不多，

$f=fopen(file,method) 打开文件，method是打开方式

fclose($f) 关闭文件

feof($f) 检查是否到文件尾（EOF）

fgets() 逐行读取文件

fgetc() 逐字符读取文件

下面给出一个例子

<?php  
$file =fopen(“file.txt”,”r”);

while(!feof(file)){

echo fgets($file).”<br>”;

}

fclose(file);  
?>

如果打开一个文件，文件名为中文，如果系统编程和当前文档编码不一致，会导致文件找不到无法打开，解决方法是用 **iconv** 函数编码转换成一致，如下：

$fpath=iconv(‘utf-8’,’GB2312’,’题库.csv’);

$file=fopen($path,”r”) or exit(“无法打开”);

fcolse($file);

网页文件上传

1.构建表单

<html>

<head>

<meta charset="utf-8">

<title>菜鸟教程(runoob.com)</title>

</head>

<body>

<form action=”upload\_file.php” method=”post” enctype=”multipart/form-data”>

<label for=”file”>文件名:</label>

<input type=”file” name=”file” id=”file”><br>

<input type=”submit” name=”提交” value=”提交”>

</form>

</body>

</html>

有关上面的 HTML 表单的一些注意项列举如下：

* **<form>** 标签的 **enctype** 属性规定了在提交表单时要使用哪种内容类型。在表单需要二进制数据时，比如文件内容，请使用 "**multipart/form-data**"。
* **<input>** 标签的 **type="file"** 属性规定了应该把输入作为文件来处理。举例来说，当在浏览器中预览时，会看到输入框旁边有一个浏览按钮。

通过使用 PHP 的全局数组 $\_FILES，你可以从客户计算机向远程服务器上传文件。

第一个参数是表单的 input name，第二个下标可以是 "name"、"type"、"size"、"tmp\_name" 或 "error"。如下所示：

* $\_FILES["file"]["name"] - 上传文件的名称
* $\_FILES["file"]["type"] - 上传文件的类型
* $\_FILES["file"]["size"] - 上传文件的大小，以字节计
* $\_FILES["file"]["tmp\_name"] - 存储在服务器的文件的临时副本的名称
* $\_FILES["file"]["error"] - 由文件上传导致的错误代码

这是一种非常简单文件上传方式。基于安全方面的考虑，您应当增加有关允许哪些用户上传文件的限制。

保存上传的文件用move\_uploaded\_file()函数

move\_uploaded\_file($\_FILES["file"]["tmp\_name"], "upload/" . $\_FILES["file"]["name"]);

explode(sep,str)按照分隔符sep将str划分成一个数组。

json\_encode($value)对变量进行json编码

json\_decode($json\_string,assoc=true/false)对json字符串进行解码转换成php变量

当为true时返回数组，false时返回对象

php常用函数

strlen($str) //返回字符串长度

strtolower($str) //字母转小写

strtoupper($str) //字母转大写

ucwords($str) //第一个单词的首字母大写

ucfirst(‘$str) //首字母转大写

str\_replace(‘a’,’b’,$str) //用b替换str中的a，区分大小写

htmlsepcialchars($str,ENT\_NOQUOTES);

//字符串转换为html 实体 ENT\_COMPT(默认只编译双引号)ENT\_QUOTES单引号双引号都编译,ENT\_NOQUOTES不编译任何引号

trim($str) //删除字符串前后空格

ltrim($str) //删除字符串左端空格

rtrim($str) //删除字符串右边空格

strops($str,’a’) //返回‘a’在str中第一次出现的位置

stripos($str,’a’) //同上，但是不区分大小写

strrpos($str,’a’) //字符串最后一次出现的位置

strripos($str,’a’) //同上，但是不区分大小写

substr($str,0,3) //截取str的第一个字符起长度为3的子串

strrchr($str,’a’) //截取str中最后一个“a’及后面的字符

strstr($str,’a’) //截取str中第一个’a’及后面的字符

strrev($str) //字符串反转

str\_shuffle($str) //随机打乱字符串

explode(‘-‘,$str) //按指定分隔符分隔字符串，返回数组

implode(‘-‘,$str) //按分隔符拼接数组

$\_POST[]可以方便的收集来自表单的数据

如$\_POST[‘password’]