

1.php 脚本以<?php 开始，以?>结束；PHP_EOL 为换行符

2.php 注释：//单行注释，/*注释内容*/多行注释

3.PHP 变量规则

变量必须以\$符号开始，后面跟着变量名称

变量名必须以字母或者下划线开始

变量名区分大小写

变量名不能包含空格

没有声明变量的命令，变量在第一次赋值的时候被创建（说明 PHP 是弱类型语言，PHP 会自动根据变量的值将其转换为正确的数据类型）

文本值给变量时，需要给文本加上双引号

4.变量作用域

local

global

static: 当一个函数完成时，其所有变量都会删除。然后某些时候希望该局部变量不被删除，以备下一次调用函数时使用，则可以使用其进行声明

parameter: 参数是通过调用代码将值传递给函数的局部变量。

在所有函数外部定义的变量，拥有全局作用域。除了在函数内，全局变量可以被脚本中的任何部分访问，因此如果需要在函数中访问全局变量的话，需要使用 global 关键字。

（先 global 变量，再使用变量才行，否则会报错）

在 PHP 函数内部声明的变量是局部变量，只能在函数内访问。

5.echo 和 print

echo: 可以输出一个或多个字符串

print: 只允许输出一个字符串，且返回值总为 1

（echo 没有返回值，速度比 print 快）

6.PHP EOF

是一种在命令行 shell（如 sh,bash,powershell）和程序语言（Perl,PHP,Python,Ruby）里定义一个字符串的方法。

6.1 必须后接分号，否则编译不通过

6.2EOF 可以用任意其他字符代替，只需保证开始与结束标识一致

6.3 结束表示必须顶格独自占一行

```
<?php
```

```
$a= <<<EOF
```

```
    "abc:$name
```

```
    "123"
```

```
EOF;
```

```
//结束必需独立一行且前后不能空格  
echo $a;  
?>
```

7.PHP 数据类型包含 PHP 布尔型, PHP 数组, PHP 整型, PHP 浮点型

8.PHP 类型比较

松散比较: 两个等号==, 只比较值, 不比较类型

严格比较: 三个等号===, 值和类型都要比较

9.PHP 常量

PHP 中设置常量, 需要使用 define 函数, 常量为全局变量

define (常量名, 常量值, 常量是否区分大小写)

例如: define("GREETING", "欢迎", true/false);

使用: echo GREETING;

使用时不要加上\$符号, 否则会报错

10.PHP 字符串变量

```
$txt="hello";
```

```
echo $txt;
```

将文本值赋值给变量时, 需要给文本值加上单引号或者双引号。

10.1 字符串连接

字符串连接使用并置运算符一个点(.)

例如:

```
$t1="hello";
```

```
$t2="world";
```

```
echo $t1.$t2;
```

10.2 字符串长度

```
strlen()
```

例如:

```
echo strlen("hello");
```

10.3 字符串内查找

strpos()用于在字符串内查找一个字符或者一段指定的文本的位置。

mb_strpos()按字查找文字, 上面的按字节查找位置。比如说中文就是一个字占两个字节。

例如:

```
echo strpos("hellowrold", "world");
```

11.PHP 判断语句

```
if(条件)
```

```
{
```

```
    条件成立时执行的代码;
```

```

}
else if()
{
}
else
{
}

```

SWITCH 语句:

```

switch(m)
{
case n1:
    如果 m=n1, 则执行
    break;
case n2:
    如果 m=n1, 则执行
    break;
default:
    如果 m 不等于 n1 和 n2, 则执行
}

```

12.PHP 数组

array 函数创建数值数组

```
$cars=array("volvo", "bmw", "toyota");
```

关联数组-----字典

```
$age=array("peter"=>"35", "Ben"=>"27", "Joe"=>"43");
```

foreach 遍历关联数组:

```

foreach($age as $x=$x_value)
{
    echo "Key=".$x."Value=".$x_value;
    echo "<br>";
}

```

13.PHP 超级全局变量

\$GLOBALS --包含全部全局变量的数组

```
<?php
```

```
$x = 75;
```

```
$y = 25;
```

```
function addition()
```

```

{
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}

```

`$_SERVER` --包含头信息(header),路径(path),服务器名(server_name)等信息的关联数组

```
<?php
echo $_SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $_SERVER['HTTP_HOST'];
echo "<br>";
echo $_SERVER['HTTP_REFERER'];
echo "<br>";
echo $_SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>
```

`$_REQUEST` --用于收集 HTML 表单中 input 提交的数据以及 URL 传递的参数，两者都可收集。

`$_POST` --用于收集 HTML 表单中 input 提交的数据，即 POST 方式

`$_GET` --用于收集 URL 中传递过来的数据，即 GET 方式

`$_FILES`

`$_ENV` --默认不开启，php.ini 中 variables_order 设置为 EGPCS 才能使用,显示操作系统环境变量。

`$_COOKIE`

`$_SESSION`

14.PHP 函数

```
<?php
function 函数名(参数)
{
    代码
    return 返回值;
}
?>
```

函数名以字符和下划线开头，名称应该说明函数的功能

函数参数与变量类似

返回值

实例

```
<?php
function add($x,$y)
{
    $total=$x+$y;
    return $total;
}

echo "1 + 16 = " . add(1,16);
?>
```

可变参数：使用..实现

```
<?php
header("charset=utf-8");
function test(...$args) //定义可变参数函数，使用...实现
{
    $num=count($args);//统计参数个数
    echo "函数调用参数个数：" . $num . PHP_EOL;
    echo "函数参数详情：" . PHP_EOL;
    foreach($args as $arg)
    {
        echo $arg . " ";
    };//遍历打印出参数
}

test("a");//一个参数
test("a","b");//两个参数
test("a","b","c");//三个参数
?>
```

15.PHP 魔术常量

LINE：文件中当前行号
FILE：当前文件在操作系统中的绝对路径
DIR：当前文件所在目录的绝对路径
FUNCTION：当前函数名
CLASS：当前当前类名
NAMESPACE：当前命名空间名称
METHOD：类的方法名
TRAIT：类覆盖方法

16.PHP 表单

17.@符号：错误控制运算符

PHP 中的@称为错误控制运算符，加上@时，对应行的代码在运行出错时不再报错，也不输出错误。如果不加@，代码运行出错会报错。

18.&符号：取内存地址符号

在变量前加上&符号，表示取该变量地址
在函数前加上&符号，表示取该函数的返回值的地址

form.html 文件代码:

```
<html>
<head>
<meta charset="utf-8">
<title>菜鸟教程(runoob.com)</title>
</head>
<body>

<form action="welcome.php" method="post">
名字: <input type="text" name="fname">
年龄: <input type="text" name="age">
<input type="submit" value="提交">
</form>

</body>
</html>
```

当用户填写完上面的表单并点击提交按钮时, 表单的数据会被送往名为 "welcome.php" 的 PHP 文件:

welcome.php 文件代码:

```
欢迎<?php echo $_POST["fname"]; ?>!<br>
你的年龄是 <?php echo $_POST["age"]; ?> 岁。
```

PHP 获取下拉菜单的数据

PHP 下拉菜单单选

以下实例我们设置了下拉菜单三个选项, 表单使用 GET 方式获取数据, action 属性值为空表示提交到当前脚本, 我们可以通过 select 的 name 属性获取下拉菜单的值:

php_form_select.php 文件代码:

```
<?php
$q = isset($_GET['q'])? htmlspecialchars($_GET['q']) : '';
if($q) {
    if($q == 'RUNOOB') {
        echo '菜鸟教程<br>http://www.runoob.com';
    } else if($q == 'GOOGLE') {
        echo 'Google 搜索<br>http://www.google.com';
    } else if($q == 'TAOBAO') {
        echo '淘宝<br>http://www.taobao.com';
    }
} else {
    ?>
<form action="" method="get">
    <select name="q">
        <option value="">选择一个站点:</option>
        <option value="RUNOOB">Runoob</option>
        <option value="GOOGLE">Google</option>
        <option value="TAOBAO">Taobao</option>
    </select>
    <input type="submit" value="提交">
</form>
<?php
}
?>
```

单选按钮表单

PHP 单选按钮表单中 name 属性的值是一致的，value 值是不同的，代码如下所示：

php_form_radio.php 文件代码：

```
<?php
$q = isset($_GET['q'])? htmlspecialchars($_GET['q']) : '';
if($q) {
    if($q == 'RUNOOB') {
        echo '菜鸟教程<br>http://www.runoob.com';
    } else if($q == 'GOOGLE') {
        echo 'Google 搜索<br>http://www.google.com';
    } else if($q == 'TAOBAO') {
        echo '淘宝<br>http://www.taobao.com';
    }
} else {
    ?><form action="" method="get">
        <input type="radio" name="q" value="RUNOOB" />Runoob
        <input type="radio" name="q" value="GOOGLE" />Google
        <input type="radio" name="q" value="TAOBAO" />Taobao
        <input type="submit" value="提交">
    </form>
<?php
}
?>
```

checkbox 复选框

PHP checkbox 复选框可以选择多个值：

php_form_select_checkbox.php 文件代码：

```
<?php
$q = isset($_POST['q'])? $_POST['q'] : '';
if(is_array($q)) {
    $sites = array(
        'RUNOOB' => '菜鸟教程： http://www.runoob.com',
        'GOOGLE' => 'Google 搜索： http://www.google.com',
        'TAOBAO' => '淘宝： http://www.taobao.com',
    );
    foreach($q as $val) {
        // PHP_EOL 为常量，用于换行
        echo $sites[$val] . PHP_EOL;
    }
} else {
    ?><form action="" method="post">
        <input type="checkbox" name="q[]" value="RUNOOB"> Runoob<br>
        <input type="checkbox" name="q[]" value="GOOGLE"> Google<br>
        <input type="checkbox" name="q[]" value="TAOBAO"> Taobao<br>
        <input type="submit" value="提交">
    </form>
<?php
}
?>
```

默认情况下，POST 方法的发送信息的量最大值为 8 MB（可通过设置 php.ini 文件中的 post_max_size 进行更改）

序列化：一种数据对象传递手段，在传递数据值的同时保留了数据的结构属性，但是，如果在传递过程中处理不当，导致用户可控序列数据，在数据反序列化过程中就有可能造成命令执行或者越权行为。