

# Web应用系统安全开发

## PHP 的语言基础



# PHP基本语法

# 1. 概述

- PHP 是一种创建动态交互性站点的强有力的服务器端脚本语言，它由服务器负责解释执行，具有自己的语法结构。
- PHP可以用于管理动态内容、支持数据库、处理会话跟踪，甚至构建整个电子商务站点。
- PHP支持多种数据库，包括 MySQL、 Oracle和 SQL Server等。
- 本节主要介绍如何在Windows系统平台上开发PHP程序以及如何在HTML网页中加入合法的 PHP 程序代码。

## 2. PHP标记符

- 标记符，就是为了便于与其他内容区分所使用的一种特殊符号
- PHP代码可以嵌入到HTML、JavaScript等代码中使用，因此就需要使用PHP标记符
- PHP标记符将PHP代码与HTML内容进行识别，当服务器读取该段代码时，就会调用PHP编译程序进行编译处理

```
<?php  
echo "Hello, world!" ;  
?>
```

## 3. PHP注释

- 注释可以理解为代码中的解释和说明，是程序中不可缺少的重要元素。
- 使用注释不仅能够提高程序的可读性，而且还有利于程序的后期维护工作。
- 注释不会影响程序的执行，因为在执行时，注释部分的内容不会被解释器执行。
- 在PHP程序中添加注释的方法有3种，可以混合使用
  - (1) `“//”`：C++语言风格的单行注释。
  - (2) `“/* ..... */”`：C语言风格的多行注释。
  - (3) `“#”`：UNIX的Shell语言风格的单行注释。



## 4. 标识符

- 在系统的开发过程中，需要在程序中定义一些符号来标记一些名称，如变量名、函数名、类名等，这些符号被称为标识符
- 在PHP中，定义标识符要遵循一定的规则
  - (1)标识符只能由字母、数字和下划线组成。
  - (2)标识符可以由一个或多个字符组成，且必须以字母或下划线开头。
  - (3)当标识符用用变量时候，区分大小写。
  - (4)当标识符由多个单词组成，那么应使用下划线进行分隔，如user\_name。

# 5. 关键字

- 关键字就是编程语言里事先定义好并赋予了特殊含义的单词，也称为保留字。如echo用于输出数据，function用于定义函数。见下表所示，该表列举了PHP中所有的关键字。
- 注意：(1)关键字不能作为常量、函数名或类名使用。(2)关键字虽然可作为变量名使用，但是容易导导致混淆，不建议使用。

表 3-1 PHP 中的关键字

and	or	xor	__FILE__	exception
__LINE__	array()	as	break	case
class	const	continue	declare	default
die()	do	echo	else	elseif
empty()	enddeclare	endfor	endforeach	endif
endswitch	endwhile	eval()	exit()	extends
for	foreach	function	global	if
include	include_once	isset()	list()	new
print	require	require_once	return	static
switch	unset()	use	var	while
__FUNCTION__	__CLASS__	__METHOD__	final	php_user_filter
interface	implements	extends	public	private
protected	abstract	clone	try	catch
throw	this			



# PHP数据类型



# 1. 概述

- 数据是计算机程序的核心，计算机程序运行时需要操作各种数据，这些数据在程序运行时临时存储在计算机内存中。
- 定义变量时，系统在计算机内存中开辟了一块空间用于存放这些数据，空间名就是变量，空间大小则取决于所定义的数据类型。
- 因此就应当根据程序的不同需要来使用各种类型的数据，以避免浪费内存空间。
- PHP支持的数据类型分为3类，分别是标量数据类型，复合数据类型和特殊数据类型。

表 3-2 PHP 数据类型

分类	数据类型	说明
标量数据类型	integer(整型)	取值范围为整数：正整数、负整数和 0
	float/double(浮点型)	用来存储数字，和整型不同的是它有小数位
	string(字符串型)	连续的字符序列，可以是计算机所能表示的一切字符的集合
	boolean(布尔型)	取值真 (true) 和假 (false)
复合数据类型	array(数组)	数组是一组数据的集合
	object(对象)	对象是存储数据和有关如何处理数据的信息的数据类型
特殊数据类型	resource (资源)	资源是由专门的函数来建立和使用的
	null(空值)	null 或 NULL(不区分大小写)

## 2. 标量数据类型

- 标量数据类型是数据结构中最基本的单元，只能存储一种数据。

### (1) 整型 (integer)

- 整型数据类型取值只能是整数，包括正整数、负整数和0。
- 整型数据可以用十进制、八进制和十六进制表示。八进制整数前面必须加0；十六进制整数前面必须加0x。
- 字长与操作系统有关，在32位的操作系统中的有效范围是-2147483648 ~ +2147483647。

```
$a=666; //十进制
```

```
$b= -666; //负整数
```

```
$c=0666; //八进制
```

```
$d=0x666; //十六进制
```

## (2) 浮点型 (float/ double)

- 浮点数据类型可以存储整数和小数。
- 字长与操作系统有关，在32位的操作系统中的有效范围是 $1.7E^{-308} \sim 1.7E^{+308}$ 。
- 浮点型数据有两种书写格式，分别是标准格式和科学计数法格式。

5.1286, 0.88, -18.9 //标准格式

8.31E2, 32.64E-2 //科学计数法格式

## (3) 布尔型 (boolean)

- 布尔型也称逻辑型数据。取值范围为真值(true) 或假值(false)。

```
$a = true;
```

```
$b = false;
```

## (4) 字符串型 (string)

- 字符串是字符序列，比如 “Hello world!” ，字符串可以是引号内的任何文本
- 可以使用单引号或双引号，两者的不同之处是：单引号中包含的变量名称或者任何其他的文本都会不经修改地按普通字符串输出，而在双引号中所包含的变量会自动被替换成实际变量值。

```
$x=666;  
$a= '深圳欢迎你$x';  
$b ="深圳欢迎你$x";  
echo $a; //输出：深圳欢迎你$x  
echo $b; //输出：深圳欢迎你666
```



# 3. 复合数据类型

## (1) 数组 ( array)

- 数组在一个变量中存储多个值

```
<?php
    $cars=array("Volvo","BMW","SAAB");
    print_f($cars);
?>
```

输出如下结果:

```
Array([0]=> Volvo [1]=> BMW [2]=> SAAB)
```

## (2)对象 ( object)

- 对象是存储数据和有关如何处理数据的信息的数据类型。
- 与C++、Java等面向对象编程语言类似，在PHP中声明一个对象之前，必须先使用class关键字来定义一个类，然后再使用new运算符来建立这个类的一个实例(instance)，对象就是类的一个实例。
- 类是包含属性和方法的结构。在类中定义数据类型，然后在该类的对象中使用此数据类型。

```
class Student
{
    var $name;
    var $id;
    var $sex;
    var $grade_chinese;
    var $grade_maths;
    var $grade_english;
    function grade_sum()
    {
        return $this -> grade_chinese + $this ->
grade_maths + $this -> grade_english;
    }
}
```

```
$stu_zhang = new Student();

$stu_zhang -> name = "张三";
$stu_zhang -> id = "2022010501";
$stu_zhang -> sex = "男";
$stu_zhang -> grade_chinese = 70;
$stu_zhang -> grade_maths = 90;
$stu_zhang -> grade_english = 80;

$stu_zhang -> grade_sum(); //调用对象$stu_zhang 中
的grade_sum()函数
```



# 4. 特殊数据类型

## (1) 资源 (resource)

- 资源是一种特殊的数据类型，用于表示一个PHP 的外部资源，由特定的函数来建立和使用。
- 任何资源在不需要使用时应及时释放。如果程序员忘记了释放资源， PHP 垃圾回收机制将自动回收资源。

## (2) 空值 (null)

- 空值表示没有为该变量设置任何值
- 由于null不区分大小写，所以null 和NULL 是等效的
- 下列三种情况都表示空值
  - 1)尚未赋值
  - 2)被赋值为null
  - 3)被unset ()函数销毁的变量

# 5. 数据类型检测函数

➤ PHP为变量或常量提供了很多检测数据类型的函数，有了这些函数用户就可以对不同类型的数据进行检测。

函数名	功能说明	示例
is_bool()	检测变量或常量是否为布尔类型	bool is_bool(\$a);
is_string()	检测变量或常量是否为字符串类型	bool is_string(\$a);
is_float/is_double()	检测变量或常量是否为浮点类型	bool is_float(\$a); bool is_double(\$a);
is_integer/is_int()	检测变量或常量是否为整型	bool is_integer(\$a); bool is_int(\$a);
is_numeric()	检测变量或常量是否为数字或数字字符串	bool is_numeric(\$a);
is_null()	检测变量或常量是否为空值	bool is_null(\$a);
is_array()	检测变量是否为数组类型	bool is_array(\$a);
is_object()	检测变量是否为对象类型	bool is_object(\$a);





# 6. 数据类型的转换

- PHP 变量属于松散的数据类型，在定义PHP 变量时不需要指定数据类型，数据类型是由赋给变量或常量的值自动确定的。
- 当不同数据类型的变量或常量之间进行运算时，需要先将变量或常量转换成相同的数据类型，再进行运算。
- PHP 数据类型转换分为自动类型转换和强制类型转换。
  - 自动类型转换：是指PHP 预处理器根据运算需要，自动将变量转换成合适的数据类型再进行运算。例如，浮点数在与整数进行算术运算时，PHP 预处理器会先将整数转换成浮点数，然后再进行算术运算。
  - 强制类型转换：是指程序员通过编程手段强制将某变量或常量的数据类型转换成指定的数据类型。  
PHP 提供了3种强制类型转换的方法：
    - 1) 在变量前面加上一个小括号，然后把目标数据类型写在小括号中。  
例如：`$b=(int)$a;`
    - 2) 使用通用类型转换函数settype()。  
例如：`settype($a,"int");`
    - 3) 使用类型转换函数intval()、strval()、floatval()。  
例如：`$b = intval($a);`



# PHP常量

# 1. 声明和使用

- 常量是指在程序运行过程中始终保持不变的数据。
- 常量的值被定义后，在程序的整个执行期间，这个值都有效，不需要也不可以再次对该常量进行赋值。
- PHP 提供两种常量，分别是系统预定义常量和自定义常量。
- 常量是单个值的标识符(名称)，在脚本中无法改变该值。
- 有效的常量名以字符或下划线开头(常量名称前面没有 \$ 符号)。
- 与变量不同，常量贯穿整个脚本，是自动全局的。

➤ 程序员在开发过程中不仅可以使用PHP 预定义常量，也可以自己定义和使用常量。

(1)使用define ()函数定义常量

```
define( "常量名称", "常量值", 大小写是否敏感) ;
```

## 大小写敏感

```
<?php // 定义对大小写敏感的常量  
define("LOGO", "Welcome to Shenzhen!");  
echo LOGO;  
echo "<br>";  
echo logo; // 不会输出常量的值  
?>
```

运行结果如下：

```
Welcome to Shenzhen!  
logo
```

## 大小写不敏感

```
<?php // 定义对大小写不敏感的常量  
define("LOGO", "Welcome to Shenzhen!", true);  
echo LOGO;  
echo "<br>";  
echo logo; // 不会输出常量的值  
?>
```

运行结果如下：

```
Welcome to Shenzhen!  
Welcome to Shenzhen!
```

(2)使用defined ( )函数判断常量是否已经被定义

bool defined (常量名称);

(3)常量是全局的

常量是自动全局的，而且可以贯穿整个脚本使用。下面的例子在函数内使用了一个常量，即使它在函数外定义

```
<?php  
define("LOGO", "Welcome to Shenzhen!");  
function myTest() {  
    echo LOGO;  
}  
myTest();  
?>
```

运行结果如下：

Welcome to Shenzhen!

## 2. 预定义常量

- PHP 中提供了大量预定义常量，用于获取PHP 中相关系统参数信息，但不能任意更改这些常量的值。
- 有些常量是由扩展库所定义的，只有加载了相关扩展库才能使用。

表 3-7 常用 PHP 预定义常量

常量名称	功能
<code>__FILE__</code>	返回当前文件所在的完整路径和文件名
<code>__LINE__</code>	返回代码当前所在行数
<code>PHP_VERSION</code>	返回当前 PHP 程序的版本
<code>PHP_OS</code>	返回 PHP 解释器所在操作系统名称
<code>TRUE</code>	真值 <code>true</code>
<code>FALSE</code>	假值 <code>false</code>
<code>NULL</code>	空值 <code>null</code>
<code>E_ERROR</code>	指到最近的错误处
<code>E_WARNING</code>	指到最近的警告处
<code>E_PARSE</code>	指到解释语法有潜在问题处
<code>E_NOTICE</code>	提示发生不寻常，但不一定是错误处



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：  
<https://d.book118.com/988046016110006143>