

第 3 章 PHP 数据处理

知识点：

- 常量和变量的定义以及变量的作用域
- 数据类型(标准数据类型和复合数据类型)
- 运算符以及运算符的优先级

本章导读：

本章将讲述 PHP 的常量和变量的定义和作用域、各种数据类型和运算符。这些内容是学习 PHP 的基础。

3.1 标准数据类型

PHP 中的标准数据类型包含单个值,通常被称为单一数据类型。常用的标准数据类型有布尔型(boolean)、整型(integer)、浮点型(float 和 double)以及字符串型(string)。

3.1.1 布尔型

在 PHP 中布尔型用于表示事物的真假,比如“婚否”,是否“党员”。布尔值用 true 和 false 来表示。已婚可以使用 true 来表示,未婚可以使用 false 来表示。

【示例 1】 输出刘备的政治面貌和婚姻状态。

步骤 1 eg1-1.php 代码如下。

```
<?php
header("content-type:text/html;charset=utf-8");
$married=false;
$party=true;
echo '刘备的婚姻状态是:'.$married.'<br />';
echo '刘备的政治面貌是:'.$party.'<br />';
?>
```

程序运行结果如图 3-1 所示。我们发现婚姻状态 false 没有显示。在 PHP 中 true 显示为 1,false 显示为 0。要想显示 true 和 false,可以使用 var_dump() 函数。还可以使用强制类型转换为整型,用 1 表示 true,0 表示 false。

步骤 2 eg1-2.php 代码如下。



图 3-1 eg1-1 的程序运行结果

```

<?php
header("content-type:text/html;charset=utf-8");
$married=false;
$party=true;
echo '刘备的婚姻状态是:'.(int)$married.'<br />';
echo '刘备的政治面貌是:'.(int)$party.'<br />';
?>

```

程序运行结果如图 3-2 所示。



图 3-2 将布尔型转换为整型再输出

3.1.2 整型

整型数就是不包含小数部分的数,包含十进制、十六进制、八进制数,整型数包括正数、负数和 0。十六进制数前面要加上 0x,八进制数前面要加 0。

【示例 2】 输出一些整型数。

eg2.php 代码如下。

```

<?php
header("content-type:text/html;charset=utf-8");
$int1=0123;
$int2=0x1ab;
$int3=123;
$int4=1234567890;
$int5=12345678901;
echo $int1.'<br />';
echo $int2.'<br />';
//双引号可以解析引号中的变量,字符串中没有变量时尽量用单引号,代码更优
echo "$int3<br />";
var_dump($int4);
echo '<br />';
var_dump($int5);
?>

```

程序运行结果如图 3-3 所示。



图 3-3 输出整型数

整型数表示的范围是有限度的,如果超过这个限度,则自动转化为浮点型。本例中 1234567890 是整型数,而 12345678901 则超过整型数表示的范围而自动转换为浮点型。整型数表示范围与平台有关,无须死记硬背。

3.1.3 浮点型

浮点型就是实型(real),包括单精度类型(float)和双精度类型(double)。浮点型常用于表示货币、重量、距离等,在整型无法表示的情况下用浮点型。可以使用科学计数法来表示浮点型。

【示例 3】 输出一些浮点型数。

eg3.php 代码如下。

```
<?php
    $f1=1.23;
    $f2=1.0;           //带小数点,是浮点型
    $f3=12345678901; //超过整型数表示的范围,是浮点型
    $f4=1e3;          //表示 1000,用科学计数法表示,是浮点型
    // $f5=1e4.0;     //e 后面不能带小数点,即使小数点后面是 0 也不能带,非法
    $f6=1.23e-2;      //e 后面可以是负数
    var_dump($f2);    //浮点型
    echo '<br />';
    var_dump($f3);    //浮点型
    echo '<br />';
    var_dump($f4);    //浮点型
    echo '<br />';
    var_dump($f6);    //浮点型
?>
```

3.1.4 字符串型

字符串是一个字符序列,可以看成数组。PHP 中可以表示的字符包含 256 种,也就是说使用一个字节来表示一个字符。可以使用单引号或者双引号来表示字符串,字符串中的特殊字符可以使用转义的方法来表示。

1. 单引号

单引号可以用来表示一个字符串。当字符串中仅仅包含单纯的字符,而不包含变量或者转义字符时,使用单引号比较好,代码更加优化。当然使用双引号也是可以的。

【示例 4】 单引号用法。

eg4.php 代码如下。

```
<?php
    $str1='Hello,\nworld!'; //\n 表示两个普通的字符,不表示换行
    $a='Wuhan';
    $b='I like $a!';        //$a 表示两个普通字符,不表示 Wuhan
    echo nl2br($str1);     //nl2br 能把\n 输出为换行
    echo '<br />';
    echo $b;
?>
```

程序运行结果如图 3-4 所示。

2. 双引号

双引号可以解析出字符串中的特殊字符以及变量，普通字符不做任何改变。使用双引号可以使代码非常灵活。

【示例 5】 双引号用法。

```
<?php
    $str1="Hello,\nworld!"; //\n 表示换行
    $a='Wuhan';
    $b="I like $a!";        //$a 表示 Wuhan
    echo nl2br($str1);
    echo '<br />';
    echo $b;
?>
```

程序运行结果如图 3-5 所示。



图 3-4 单引号的程序运行结果



图 3-5 双引号的程序运行结果

【示例 6】 使用大括号来分开字符串中的变量。

eg6.php 代码如下。

```
<?php
    $a='Wuhan';
    $ab='Jingzhou';
    $x="abc$ab$a";        //abcJingzhouWuhan
    $y="abc{$a}b$a";      //abcWuhanbWuhan
    echo $x.'<br />';
    echo $y;
?>
```

3. 转义字符

PHP 可以识别多种特殊字符，例如转义字符，包括换行符等。表 3-1 列出了一些常用的转义字符。

表 3-1 PHP 能够识别的几种转义字符

转义序列	说 明
\n	换行符
\r	回车符
\t	水平制表符
\\	反斜线

转义序列	说 明
\\$	美元符号
\"	双引号
\[0-7]{1,3}	在正则表达式中用于表示八进制
\x[0-9A-Fa-f][1,2]	在正则表达式中用于表示十六进制

【示例 7】 巧用单双引号以及转义字符的输出。

eg7.php 代码如下。

```
<?php
$a="I'm a teacher\n"; //因字符串中带单引号,所以使用双引号
$b='I said:"You are a good student!"; //因字符串中带双引号,所以使用单引号
echo $a.'  


```

说明:

- 字符串"a\\bc\\m\\tcd\\r"前面两个斜线在程序中显示一个斜线。第一个斜线告诉程序后面是特殊的转义字符。
- 因为\\m 不是特殊的转义字符,所以该斜线是普通字符。
- \\t 表示水平制表。
- \\r 表示回车。

【示例 8】 在字符串中如果想要使用 html 的格式标签,可以使用< article> 和</article>。则在<article>和</article>之间的 HTML 标签都可以解释出来。在开发考试系统时,可以用<article>和</article>来定制格式,将格式保存在字符串和数据库中。

eg8.php 代码如下。

```
<?php
$a="
<article>
<h1>Internet Explorer 9</h1>
<p>Windows Internet Explorer 9</p>
<font size='+1' color='#FF0000'>Wuhan</font>
</article>
";
echo $a;
?>
```

程序运行结果如图 3-6 所示。



图 3-6 字符串中使用<article>和</article>