

第2章 PHP语言

- 经过前面预备知识的学习，从本章开始，正式进入PHP语言的学习。PHP作为一种专门用来开发Web应用的嵌入式语言，大量借用了C、C++和Perl语言的语法，同时加入了一些其它语法特征，使编写Web程序更快更有效。之所以说PHP是嵌入式语言，是因为用PHP开发的Web程序，大多都要在HTML文档中插入PHP代码，或者使用PHP代码生成某些HTML文档，以满足Web应用的需求和特点。
- PHP一般作为HTTP服务器（通常是Apache）的一个模块运行。这意味着，当用户访问到一个含有PHP代码的Web页面时，HTTP服务器就会调用这个模块，通过这个模块来分析并执行该页面的PHP代码，最终将执行结果返回给用户。PHP支持多种数据库，如MySQL、dBase、MS_SQLServer、Oracle等。这对于基于数据库的Web开发来说是有裨益的。
- PHP从上世纪90年代中期间世以来，已经推出了很多版本，到现在已经是PHP5。本书所讲述的PHP语言，将以PHP5为准。本书凡是出现“PHP”的地方，除非特别说明，都将指的是PHP5。



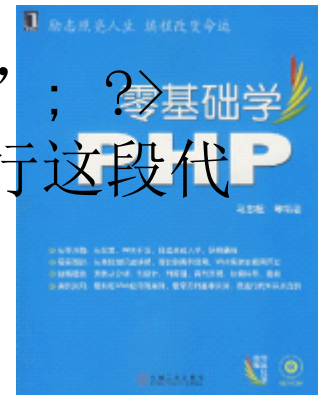
2.1 基本语法

- PHP的语法和C、C++等语言的语法很相似，有C语言基础的读者，可以非常轻松地掌握PHP的基本语法。即便是没有任何语言基础，也是值得庆幸的，那样不会受其它语言的干扰，可以更快速地接受PHP的语法。
- 事实上，PHP的语法并不复杂，再加上PHP提供了大量的预定义函数，使PHP开发事半功倍。只要按本书的讲述、一步步地学习下去，再加上自己的一点信心，相信读者会发现PHP很容易学习掌握，并且应用起来也很快速方便。本节，先简单了解一些PHP的基本语法。



2.1.1 PHP分隔符

- 因为PHP是嵌入式脚本语言，需要使用某种分隔符将PHP代码和HTML的内容区分开来，这里所说的分隔符就是“<?php”和“?>”，它们将PHP代码包含在其中，也就是说，所有的PHP代码都应该写在“<?php”和“?>”之间。如下代码所示：
- <p>一个段落</p>
//这一行是HTML，PHP分析器将会忽略这行代码，不做处理
- <?php echo “这段内容由PHP代码输出” ; ?>
//这一行是PHP代码，PHP分析器将会执行这段代码
- <p>另外一个段落</p>



2.1.2 给PHP程序添加注释

- 程序中的注释是指在一个程序文件中，对一个代码块或一条程序语句所作的文字说明，注释是提供给开发人员看的，因此，程序中的注释会被计算机忽略而不会被执行。PHP中的主要注释风格有：
- 使用符号“//”添加一个单行的注释。
- 使用符号“#”添加一个单行的注释。
- 使用“/*”和“*/”添加一个多行的注释，也可以用来单行注释。



PHP 程序的注释



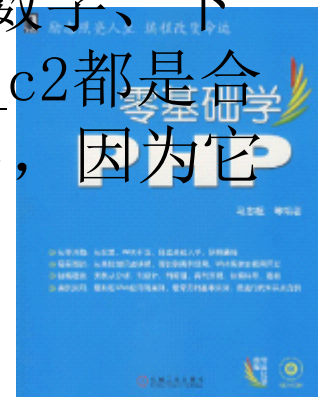
2.2 变量

- 变量是任何程序设计语言中一个基本而且重要的概念。本节的内容将讲述PHP变量基本概念、变量的类型、PHP预定义变量以及如何使用PHP内置函数对变量进行一些处理。



2.2.1 什么是变量

- 在程序中可以改变的数据量叫做变量，变量必须有一个名字，用来代表和存放变量的值。PHP中使用美元符（\$）后跟变量名来表示一个变量，如\$var就是一个变量。PHP中的变量名是区分大小写的，因此\$var和\$Var表示的是不同的两个变量。
- PHP变量的命名需要遵守一定的规则，这个规则是：一个有效的变量名必须由英文字母或下划线开头，后面可以跟任意数量的英文字母、数字、下划线或其组合。如\$abc、\$_ab_c、\$a1b_c2都是合法的变量名，而\$3xyz就不是合法变量名，因为它以数字开头。



2.2.2 变量的数据类型

- PHP属于弱类型语言。这就是说，变量的数据类型一般不用开发人员指定，PHP会在程序执行过程中，根据程序上下文环境决定变量的数据类型。如一串数字“789”，在用echo语句输出时，它作为字符串处理，但是做数学运算时，它就作为整数处理。PHP的变量主要有以下类型：
 - 整数类型。
 - 浮点类型。
 - 字符串类型。
 - 布尔类型。
 - 数组类型。
 - 对象。



2.2.3 变量类型的转换

- 和C、C++等语言不同，PHP在定义变量时，不需要明确指定变量的类型。也就是说，把一个整数指定给变量\$v，那么\$v就是一个整型变量，如果把一个字符串指定给它，那么它就是一个字符串变量。若要转换类型，在PHP程序中也是很自由的，一般不必经过特殊的转换。
- 当然，PHP中也可以对变量做强制转换，这点和C语言相似，在要转换的变量之前加上目标类型，目标类型用括号括起来。
- PHP中允许的强制类型转换有：
 - (int)，(integer)——转换成整型。
 - (bool)，(boolean)——转换成bool型。
 - (float)，(double)，(real)——转换成浮点型。
 - (string)——转换成字符串。
 - (array)——转换成数组。
 - (object)——转换成对象。



2.2.4 可变变量

- 可变变量是PHP中比较特别的一个概念，可变变量是指这样一个变量，它将某个变量的值作为自己的变量名。



可变变量



2.2.5 PHP的预定义变量

- PHP提供了大量的预定义变量，可以在程序或文件的任何地方使用它们。这些变量大多数依赖于服务器的版本及其配置。本书主要讲述 PHP5的预定义变量，这些预定义变量和以前版本的有些不同，有兴趣的读者可自行了解。下面是PHP中一些常用的预定义变量。
- **\$GLOBALS:** 包含指向当前程序中全局范围内有效的变量，它是一个数组，该数组的索引（或键名）就是全局变量的名称。
- **\$SERVER:** 该全局变量是一个包含诸如头信息、路径和脚本位置的数组。常见的\$_SERVER的元素包括PHP_SELF（当前正在执行的脚本的文件名）、SERVER_ADDR（当前执行脚本所在服务器的IP地址）、SERVER_NAME（当前执行脚本所在服务器主机的名称）、DOCUMENT_ROOT（当前脚本所在文档的根目录）、SCRIPT_FILENAME（当前执行脚本的绝对路径）、SCRIPT_NAME（当前脚本的路径）、HTTP_REFERER（链接到当前页面的前一页面的URL）、REQUEST_URI（访问此页面所需的URI）等。
- **\$_GET:** 通过HTTP的GET方法提交至脚本的表单变量。
- **\$_POST:** 通过HTTP的POST方法提交至脚本的表单变量。
- **\$_FILE:** 通过HTTP的POST文件上传提交至脚本的变量。
- **\$_COOKIE:** 通过HTTP的Cookies方法提交至脚本的变量。



2.2.6 判断变量的类型

- 从本小节开始，将介绍一些处理变量的方法。因为在 PHP 中通常通过一些预定义函数来处理变量，所以，需要读者对函数的概念有个大概了解。
- 简单地说，函数是指完成某种特定功能的代码块，可以向函数传入参数，函数对参数进行处理，并且将处理结果返回给用户。本书将在后面详细介绍函数的概念。
- 在 PHP 中，可以通过以下函数对变量的类型做判断。
- 函数 `is_integer` 判断变量是否为整数。
- 函数 `is_string` 判断变量是否为字符串。
- 函数 `is_double` 判断变量是否为浮点数。
- 函数 `is_array` 判断一个变量是否为数组。

- 简单地说，函数是指完成某种特定功能的代码块，可以向函数传入参数，函数对参数进行处理，并且将处理结果返回给用户。本书将在后面详细介绍函数的概念。

- 在PHP中，可以通过以下函数对变量的类型做判断。

- 函数is_integer判断变量是否为整数。

- 函数is_string判断变量是否为字符串。

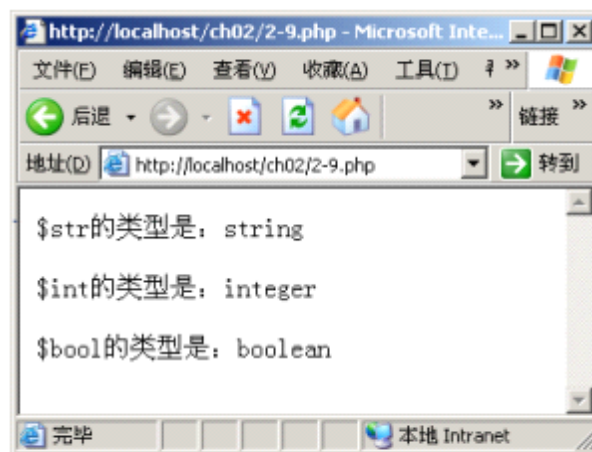
- 函数is_double判断变量是否为浮点数。

- 函数is_array判断一个变量是否为数组。



2.2.7 获取变量的类型

- 在PHP中，可以使用预定义函数`gettype`取得一个变量的类型，它接受一个变量作为参数，返回这个变量的类型。

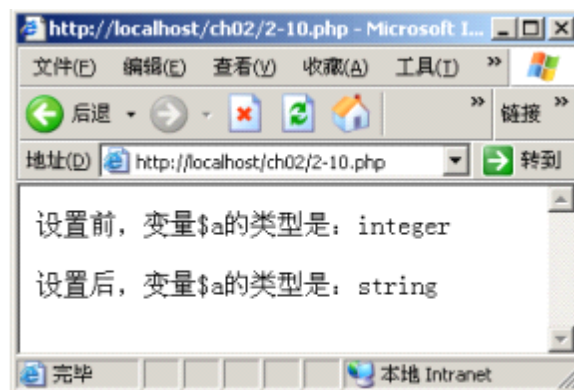


获取变量的类型



2.2.8 设置变量的类型

- 使用预定义函数settype设置一个变量的类型，该函数接受两个参数，第一个参数是变量名，第二个参数是要设置的变量的数据类型。



设置变量的类型



2.2.9 判断一个变量是否已经定义

- 使用预定义函数isset判断一个变量是否已经定义，它接受一个变量作为参数值，返回值如果为TRUE，说明该变量定义过，否则，说明该变量没有被定义。



2.2.10 删除一个变量

- 使用unset语句删除一个变量。从PHP4开始unset不再有返回值，因此，严格意义上讲，它并不是一个函数，而是一个PHP的语言结构。可以用unset一次删除多个PHP变量。



2.3 常量

- 和变量相对应的概念是常量。上节介绍了变量、变量的数据类型及对变量的一些操作、处理，这节将介绍常量的概念及使用。



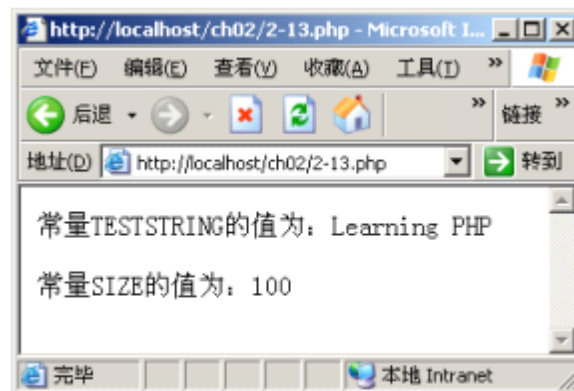
2.3.1 什么是常量

- 在程序执行过程中，其值不能改变的量叫做常量。这就是说，常量不能再被定义成其它的值。常量也可以分为不同的类型，如10、0、-12是整型常量，1.23、-0.45是浮点型常量，常量的类型从形式上就可以判别。
- PHP中有一些定义好的常量，在程序中可以直接使用。开发人员也可以根据程序的需要，自己定义新的常量。



2.3.2 定义常量

- 在PHP中通过define()函数定义一个常量。合法的常量名只能以字母和下划线开始，后面可以跟着任意字母、数字或下划线。常量一旦定义就不能再修改或者取消定义。

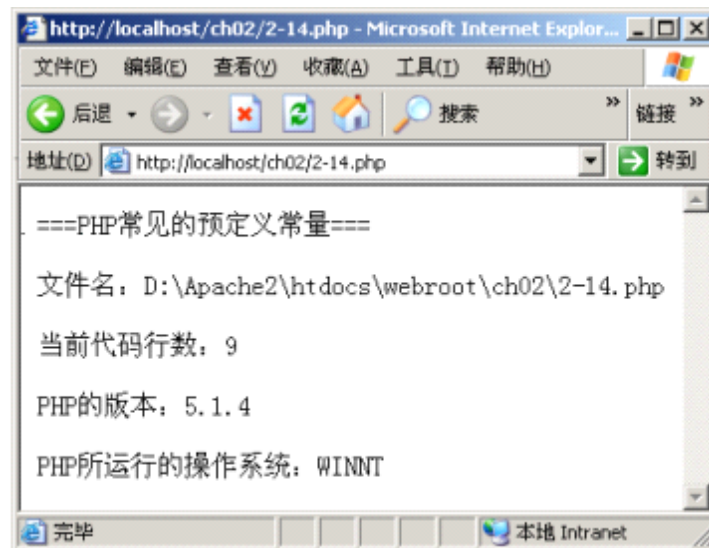


常量的定义和结果输出



2.3.3 使用PHP预定义常量

- PHP提供了一些常量，可以直接在程序中使用。列举了PHP主要的预定义常量及其含义说明。

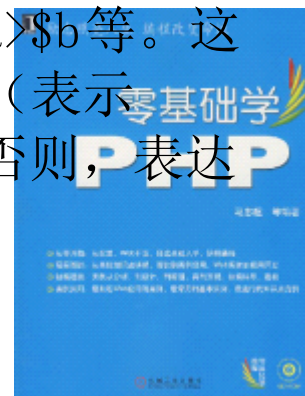


PHP 的预定义常量



2.4 表达式

- 表达式是指程序中任何有值的部分，PHP中几乎所有内容都是表达式。如 `$a=9` 就是一个表达式，这个表达式的含义是：将9指定给变量 `$a`（即赋值操作，将在下一节介绍）。很明显，“9”的值就是9，因此“9”本身就是一个表达式。也就是说，“9”是一个值为9的表达式，只不过在这里，9是一个整型常量。同理，变量 `$a` 也是一个值为9的表达式。从这个例子可以看出两个值：整数常量“9”和被指定值为9的变量 `$a`，但事实上，还有一个值，就是这个 `$a=9` 本身的值，表达式 `$a=9` 的值就是被指定的值——9。
- 另外一类很常见的表达式就是比较表达式，如 `$a>$b` 等。这些表达式的值要么是0（表示FALSE），要么是1（表示TRUE），如果表达式成立，则表达式的值为1，否则，表达式的值为0。



2.5 运算符

- 运算符是指，通过一个或多个表达式，来产生另外一个值的某些符号，如“+”、“%”“.”等都是运算符。在表达式 $2+1$ 中，运算符“+”有两个操作数1和2。具有两个操作数的运算符叫做双目运算符。具有一个操作数的运算符叫做单目运算符，如表达式 -6 ，运算符“-”只有一个操作数6，因此，这里的“-”是单目运算符。
- 运算符有优先级，即表达式中多种运算符同时出现时，哪种运算符应该首先被应用。这节将介绍基本的运算符，最后介绍基本运算符的优先级。



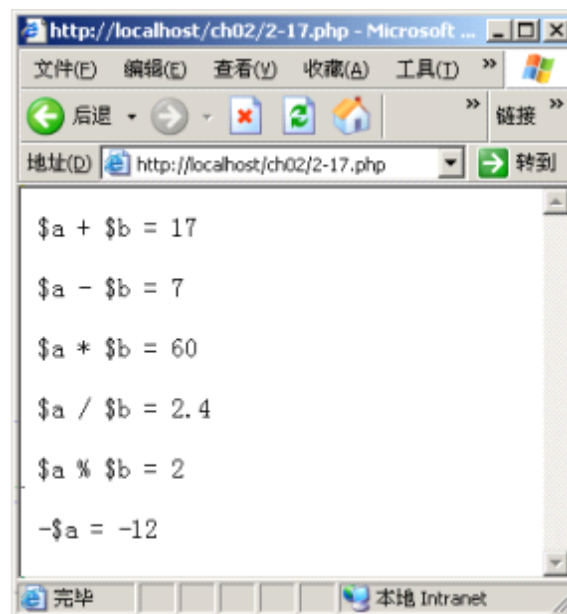
2.5.1 赋值运算符

- 在PHP中，符号“=”不表示相等，而表示赋值。它的含义是将一个值指定给一个变量，更专业一点地说，它表示把“=”右边的表达式的值赋给左边的变量，如“\$a=5”表示将5赋给\$a。赋值表达式的值也就是所赋的值，这就是说，“\$a=5”的值是5。除了上述最基本的赋值方式外，还有一种所谓的“组合赋值”，通过例子来说明它的含义和用法。



2.5.2 算术运算符

- PHP的算术运算符有加（+）、减（-）、乘（*）、除（/）和取模（%）、取反（-，即取负值）。这些运算符的用法和学校里学到的数学知识一样，



A screenshot of a Microsoft Internet Explorer browser window. The title bar reads "http://localhost/ch02/2-17.php - Microsoft ...". The address bar shows "http://localhost/ch02/2-17.php". The main content area displays the following PHP arithmetic results:

```
$a + $b = 17  
$a - $b = 7  
$a * $b = 60  
$a / $b = 2.4  
$a % $b = 2  
-$a = -12
```

The status bar at the bottom shows "完毕" (Completed) and "本地 Intranet" (Local Intranet).

算术运算符的使用



2.5.3 递增/递减运算符

- PHP有和C语言风格相同的递增/递减运算符。递增是指对当前表达式的值增加1，递减正相反，对表达式的值减1。本书仅讲述整数表达式的递增/递减运算，下面分别介绍四种风格的递增/递减运算。
- `$a++`: 先返回\$a的值，然后将\$a的值加1。
- `++$a`: 先将\$a的值加1，然后将\$a返回。
- `$a--`: 先返回\$a的值，然后将\$a的值减1。
- `--$a`: 先将\$a的值减1，然后返回\$a的值。



2.5.4 字符串运算符

- 字符串运算符只有一个，即字符串的连接运算符“.”。这个运算符将两个字符串连接成一个新的字符串。在2.2.2小节介绍字符串变量时，简单提到过这个运算符。
- 其实在此之前，已经在很多示例程序中使用过这个运算符。比如程序要显示一个执行结果，会用“.”将一些内容连接起来，然后输出。



2.5.5 逻辑运算符

- 逻辑运算符

运算符	名称	举例
!	逻辑非	!\$a
&& (或and)	逻辑与	\$a && \$b
(或or)	逻辑或	\$a \$b
xor	逻辑异或	\$a xor \$b

结果
如果\$a为FALSE, 则!\$a为TRUE。反之, !\$a为FALSE。
如果\$a和\$b都为TRUE, 则\$a && \$b为TRUE。 如果\$a或\$b任意一个为FALSE, 则\$a && \$b为FALSE。
如果\$a或\$b任意一个为TRUE, 则\$a \$b为TRUE。 如果\$a和\$b都为FALSE, 则\$a \$b为FALSE
如果\$a或\$b任意一个为TRUE, 但不同时为TRUE, 则\$a xor \$b为TRUE。



2.5.6 比较运算符

- 比较运算符用来对两个值进行比较。列举了主要的比较运算符及其可能的运算结果。

运算符及名称	举例
<code>==</code> （等于）	<code>\$a == \$b</code>
<code>!=</code> （不等于）	<code>\$a != \$b</code>
<code>===</code> （全等于）	<code>\$a === \$b</code>
<code>></code> （大于）	<code>\$a > \$b</code>
<code>>=</code> （大于等于）	<code>\$a >= \$b</code>
<code><</code> （小于）	<code>\$a < \$b</code>
<code><=</code> （小于等于）	<code>\$a <= \$b</code>

结果
如果 <code>\$a</code> 等于 <code>\$b</code> ，则 <code>\$a == \$b</code> 的值为 <code>TRUE</code> 。
如果 <code>\$a</code> 不等于 <code>\$b</code> ，则 <code>\$a != \$b</code> 的值为 <code>TRUE</code> 。
如果 <code>\$a</code> 等于 <code>\$b</code> ，并且它们的类型也相同，则 <code>\$a === \$b</code> 的值为 <code>TRUE</code> 。
如果 <code>\$a</code> 大于 <code>\$b</code> ，则 <code>\$a > \$b</code> 的值为 <code>TRUE</code> 。
如果 <code>\$a</code> 大于或者等于 <code>\$b</code> ，则 <code>\$a >= \$b</code> 的值为 <code>TRUE</code> 。
如果 <code>\$a</code> 小于 <code>\$b</code> ，则 <code>\$a < \$b</code> 的值为 <code>TRUE</code> 。
如果 <code>\$a</code> 小于或者等于 <code>\$b</code> ，则 <code>\$a <= \$b</code> 的值为 <code>TRUE</code> 。



2.5.7 运算符的优先级

- 事实上，在小学的数学知识中，就已经学习过运算符的优先级。比如 $1+2\times 3$ 的结果是7，不是9。因为 \times 号的优先级高于 $+$ 号的优先级。只不过在PHP中，运算符不仅限于加减乘除。下面列举一些常见的 PHP运算符的优先级，最上面的优先级最高。
- new（new运算符，将在后面讲述）
- ++、--（递增、递减运算符）
- *、/、%
- +、-、.
- &&
- ||
- ? :（条件运算符，将在后面讲述）
- =（赋值运算符，包含+=、*=、.=等）
- and
- xor
- or



2.6 PHP程序的流程控制

- 所有的PHP程序都由语句构成，程序就是一系列语句的序列。计算机通过执行这些语句可以完成特定的功能。一般情况下，程序都是从第一条语句开始执行，按顺序执行到最后一句。但有时因为某种情况，需要改变程序的执行顺序，这就需要对程序的流程进行控制。本节将讲述PHP程序的各种流程控制结构。



2.6.1 程序流程控制概述

- 计算机程序的执行方式有3种：顺序执行、选择执行、循环执行，通过使用这3种控制结构，可以改变程序的执行顺序，以满足开发人员解决问题的需求。
- 顺序结构使程序从第一条语句开始，按顺序执行到最后一句。在选择结构中，程序可以根据某个条件是否成立，选择执行不同的语句。在循环结构中，可以使程序根据某种条件和指定的次数，使某些语句执行多次。
- PHP程序都是由一系列语句组成，语句通常以分号结尾。此外，可以使用一对花括号“{”和“}”将一组语句组成一个语句组。例如：
 - {
 - \$i = 123;
 - \$s = “This is a string” ;
 - }



2.6.2 条件控制语句：if和if...else语句

- PHP通过一系列条件控制语句完成程序的选择执行流程。PHP中使用if、if...else (elseif) 语句构建选择程序结构。
- 1. if语句：f条件语句的结构如下所示。
 - if(expr)
 - statement
- 2. if...else语句：f...else语句的结构如下所示。
 - if(expr)
 - statement1
 - else
 - statement2
- 3. if...elseif语句：f...elseif语句的结构如下所示。
 - if(expr1)
 - statement1
 - elseif(expr2)
 - statement2
 - ...
 - else
 - statement



2.6.3 switch结构

- 上一小节讲述了使用if或者if...else语句的选择控制结构，if语句通常需要计算逻辑表达式的值。这小节将介绍另一种选择控制结构——switch结构，它不需要计算逻辑表达式的值。
- 注意：在PHP中，switch、case、break和default都是保留关键字。
- switch结构首先计算表达式expr的值，如果expr的值与某个case的值匹配，则从case后面的语句开始执行，直到遇到break语句（该语句将在后面详细介绍）或整个switch结构结束。比如，如果expr的值是value2，那么语句statement2将会被执行。



2.6.4 循环控制语句：for循环语句

- 前两小节讲述的是程序的选择控制流程，这节开始讲述程序的循环控制流程。循环控制流程可以控制程序，在满足某些条件的时候，某些语句被循环执行多次。
- PHP的循环控制主要有：for语句、while语句和do...while语句。本节先介绍for循环语句。
- for循环语句的结构如下：
- for(初始化语句；循环条件表达式；更新语句)
- statement



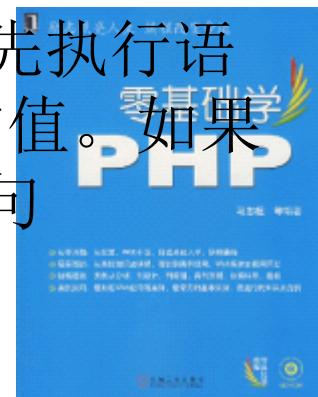
2.6.5 while循环语句

- 除了for循环语句之外，还可以使用while语句控制程序循环执行。while循环语句的结构如下：
- while(expr)
- statement
- 这里的语句statement可以是单条语句，也可以是语句组。该结构的执行流程是：当表达式expr的值为真时，就执行循环体——语句statement，然后再次计算表达式expr的值，直到expr的值为假，程序中断循环，跳出while循环结构。



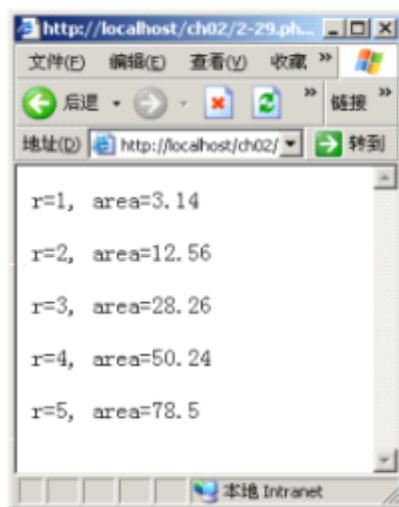
2.6.6 do...while循环语句

- 这小节讲述第三种循环控制结构：do...while语句，它也可以控制程序循环执行。do...while循环语句的结构如下：
- do
- statement
- while(expr);
- 这里的语句statement可以是单条语句，也可以是语句组。该结构的执行流程是：程序首先执行语句statement，然后再计算表达式expr的值。如果表达式expr的值为TRUE，就再次执行语句statement。

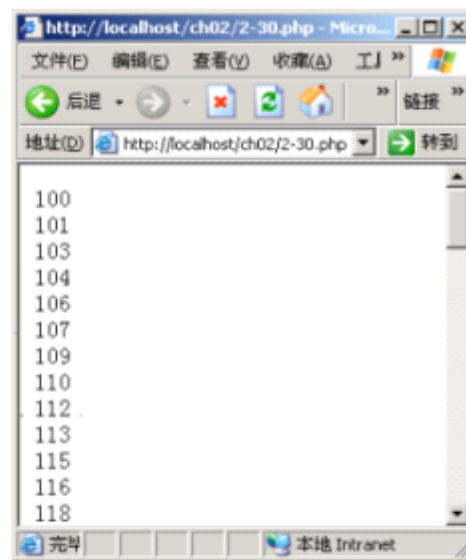


2.6.7 break和continue语句

- 在2.6.3小节中介绍switch语句时，提及过break语句，它可以使程序流程跳出switch结构。除此之外，break语句还可以在for、while和do...while语句中使用，这样可以使程序立即跳出该循环结构。



break 语句的使用



continue 语句的使用



2.6.8 条件运算符：？：

- 条件运算符（？：）的用法是：
- `expr1 ? expr2 : expr2`
- 可以看出，条件运算符有3个操作数，所以它是三目运算符。它的计算规则是：如果表达式`expr1`的值为TRUE，那么整个表达式的值就取`expr2`的值，否则，就取`expr3`的值。下面的代码使用条件运算符来获取两个数中最大的那个数。
- `$max = ($a>=$b) ? $a : $b`



2.7 函数

- 在很多编程语言中都有函数这个概念。函数将为解决某一问题而编写的代码组织在一起，使得在解决同一个问题时，可以重复这些代码。本节将介绍PHP中函数的概念、构建和调用函数等内容。



2.7.1 PHP中函数的概念

- 在数学知识里，函数是由参数的定义域和在这个参数定义域上的某种规则组成的。当选定某一参数时，函数的值也是惟一确定的。例如，有这样一个数学函数： $f(x)=2x+3$ ，那么就有 $f(1)=5$ ， $f(3)=9$ 。这里的1、3都是函数 f 的参数，而5、9都是这些参数对应的函数 f 的值。
- PHP语言中的函数和数学中函数的概念很相似，只不过 PHP 中的函数不仅仅是做一些数学运算，而是要完成更多、更复杂的功能。
- 在程序设计中，经常将一些常用的功能模块编写成函数，放在公用函数库中，供程序或其它文件使用。函数就像一些小程序，用它们可以组成更大的程序。函数之间也可以相互调用，完成更复杂的功能，但它们之间是相互独立的，互不隶属。



2.7.2 定义函数和调用函数

- PHP使用下面的语法定义一个函数：
- `function func_name(param_list)`
- `{statement}`
- 其中function是PHP的保留关键字，表示开始定义一个函数。func_name是函数名，由开发人员自行指定，函数名以字母或下划线开始，后跟任意字母、数字或下划线。函数名后的一对括号，用来存放函数的参数param_list，如果所定义的函数不需要传入参数，括号内留空，但不能没有括号。最后花括号括住的语句statement叫做函数体，它可以是单条语句，也可以是多条语句，这些语句完成函数所要实现的功能。下面的代码演示了如何定义了一个函数。



2.7.4 PHP函数的传值与传址

- 上小节讲述的向函数传入参数，是按传值方式传入的。传值的含义是指，在函数体内，会生成一个传入值的拷贝，在函数内部对参数的修改，不会影响到传入的值。有时，因为开发的需要，希望在函数内部能够修改传入的值，这就需要我们对函数参数使用传址方式。传址的含义是指，在函数体内，真实引用传入的值，这意味着，在函数体内使用的函数参数，和传入的值完全是同一个，而不单单是传入值的一个拷贝。这时，在函数内部修改了参数的值，同时也就修改了传入的值。
- 在PHP中，要想在函数参数中传址，需要在定义函数时，在参数前加上符号：&。



2.7.5 函数和变量作用域

- 变量的作用域就是变量的有效范围。对于大多数 PHP 变量，作用域只能有一个。但是，在用户自定义函数中，存在一个单独的局部函数范围。在一个函数内部定义的变量是局部变量，它只在本函数内有效，它的作用域就是当前的函数之内。这就是说，一个在函数外部定义的变量，不会在函数内部起作用，反之亦然。



在程序中使用 global 关键字



在程序中未使用 global 关键字



2.8 小结

- 本章主要介绍了PHP的基本语法，包括变量类型、常量、表达式、运算符、程序控制流程和函数的基本概念。其中变量、表达式、程序控制流程和函数是本章的重点知识，读者应该熟练掌握这些内容。
- 流程控制是程序设计中非常关键的地方，虽然这些控制语句看起来很简单，但使用过程中，如果因为逻辑错误，将可能导致死循环发生。

