

特别说明

此资料来自豆丁网(<http://www.docin.com/>)

您现在所看到的文档是使用下载器所生成的文档

此文档的原件位于

<http://www.docin.com/p-25301186.html>

感谢您的支持

抱米花

<http://blog.sina.com.cn/lotusbaob>

php入门教程:PHP入门

疯狂代码 <http://www.crazycoder.cn/> <http://www.crazycoder.cn/Php/Article4074.html>

2.PHP入门

PHP站点的在线教程已经很棒了。在那里还有一些其他教程的链接。而本文的该部分将让你对PHP熟悉一点。我不可能做到没有任何遗漏，我的目的只在于能让你迅速开始你的PHP编程。

2.1 首要条件

你首先必须要有一个正在工作着的支持PHP的web服务器。我假定在你的服务器上所有PHP文件的扩展名为.php3。

2.2 PHP的安装

生成一个名为test.php3的文件，含有以下内容：

```
<? phpinfo(); ?>
```

然后在你的浏览器中打开此文件。看看这个页面你就知道你的PHP安装使用的选项了。

2.3 语法

就像前面提到的一样，你可以混合编写你的PHP代码和HTML代码。因此你必须有办法将两者区别开来。以下就是你可以采用的几种方法。你可以选用其中一种你最适应的并且就这样坚持这种方法！

从HTML中分离

以下是可以使用的方法：

```
<? ... ?>
```

```
<?php ... ?>
```

```
<script language="php"> ... </script>
```

```
<% ... %>
```

语句

与Perl和C一样，在PHP中用(;)来分隔语句。那些从HTML中分离出来的标志也表示语句的结束。

注释

PHP支持C，C++和Unix风格的注释方式：

```
/* C,C++风格多行注释 */  
// C++风格单行注释  
# Unix风格单行注释
```

Hello,World!

通过我们已经学过的知识，你可以编写一个最简单的程序输出一个也许是程序世界中最有名的词语：

```
<HTML>  
<HEAD>  
<TITLE>  
<?  
echo "Hello World!";  
>  
</TITLE>  
</HEAD>  
<BODY>  
<H1>  
First PHP page  
</H1>  
<HR>  
<?  
// Single line C++ style comment  
/*  
printing the message  
*/  
echo "Hello World!";  
# Unix style single line comment  
>  
</BODY>  
</HTML>
```

2.4 数据类型

PHP支持整数、浮点数、字符串、数组和对象。变量类型通常不由程序员决定而由PHP运行过程决定（真是好的解脱！）。但是类型也可以被函数cast或者settype()明确的设定。

数值

数值类型可以是整数或是浮点数。你可以用以下的语句来为一个数值赋值：

```
$a = 1234; # 十进制数
$a = -123; # 负数
$a = 0123; # 八进制数 (等于十进制数的83)
$a = 0x12; # 十六进制数(等于十进制数的18)
$a = 1.234; # 浮点数"双精度数"
$a = 1.2e3; # 双精度数的指数形式
```

字符串

字符串可以由单引号或双引号引出的字段定义。注意不同的是被单引号引出的字符串是以字面定义的，而双引号引出的字符串可以被扩展。反斜杠（\）可以被用来分割某些特殊字符。举例如下：

```
$first = 'Hello';
$second = "World";
$full1 = "$first $second"; # 产生 Hello World
$full2 = '$first $second';# 产生 $first $second
```

可以将字符和数字利用运算符连接起来。字符被转化成数字，利用其最初位置。在PHP手册中有详细的例子。

数组与哈希表

数组与哈希表以同样的方法被支持。怎样运用取决于你怎样定义它们。你可以用list()或者array()来定义它们，也可以直接为数组赋值。数组的索引从0开始。虽然我在这里没有说明，但是你一样可以轻易的使用多维数组。

```
// 一个包含两个元素的数组
$a[0] = "first";
$a[1] = "second";
$a[] = "third"; // 添加数组元素的简单方法
// 现在$a[2]被赋值为"third"
echo count($a); // 打印出3，因为该是数组有3个元素
// 用一个语句定义一个数组并赋值
```

```
$myphonebook = array (  
    "sbabu" => "5348",  
    "keith" => "4829",  
    "carole" => "4533"  
);  
// 噢，忘了教长吧，让我们添加一个元素  
$myphonebook["dean"] = "5397";  
// 你定义的carale元素错了，让我们更正它  
$myphonebook["carole"] => "4522"  
// 我还没有告诉你怎样使用数组的相似支持方式吗？让我们看一看  
echo "$myphonebook[0]"; // sbabu  
echo "$myphonebook[1]"; // 5348
```

其他一些对数组或哈希表有用的函数包括sort()，next()，prev()和each()。

对象

使用new语句产生一个对象：

```
class foo  
{  
    function do_foo ()  
    {  
        echo "Doing foo.";  
    }  
}  
$bar = new foo;  
$bar->do_foo();
```

改变变量类型

在PHP手册中提到："PHP不支持（也不需要）直接在声明变量时定义变量类型；变量类型将根据其被应用的情况决定。如果你为变量var赋值为一个字符串，那么它变成了一个字符串。如果你又为它赋了整数值，那么它就变成了整数。"

```
$foo = "0"; // $foo是字符串(ASCII 48)  
$foo++; // $foo是字符串"1" (ASCII 49)  
$foo += 1; // $foo现在是整数(2)
```



```
$foo = $foo + 1.3; // $foo是一个双精度数(3.3)
$foo = 5 + "10 Little Piggies"; // $foo是一个整数(15)
$foo = 5 + "10 Small Pigs"; // $foo是一个整数(15)
```

如果想要强行转换变量类型，可以使用与C语言相同的函数settype()。

2.5 变量与常量

可能你已经注意到，变量都有一个美元符号（\$）的前缀。所有变量都是局部变量，为了使得定义的函数中可以使用外部变量，使用global语句。而你要将该变量的作用范围限制在该函数之内，使用static语句。

```
$g_var = 1; // 全局范围
function test()
{
    global $g_var; // 这样就可以声明全局变量了
}
```

更先进一些的是变量的变量表示。请参考PHP手册。这在有时会显得很有用。

PHP内置了许多已定义的变量。你也可以用define函数定义你自己的常量，比如define("CONSTANT","value")。

2.6 运算符

PHP具有C,C++和Java中的通常见到的运算符。这些运算符的优先权也是一致的。赋值同样使用"="。

算术和字符

以下只有一种运算符是有关字符的：

```
$a + $b : 加
$a - $b : 减
$a * $b : 乘
$a / $b : 除
$a % $b : 取模 ( 余数 )
$a . $b : 字符串连接
```

逻辑和比较

逻辑运算符有：

`$a || $b` ：或

`$a or $b` ：或

`$a && $b` ：与

`$a and $b` ：与

`$a xor $b` ：异或（当[\\$a](#)或[\\$b](#)为true时为true，两者一样时为false）

`! $a` ：非

比较运算符有：

`$a == $b` ：相等

`$a != $b` ：不等

`$a < $b` ：小于

`$a <= $b` ：小于等于

`$a > $b` ：大于

`$a >= $b` ：大于等于

与C一样PHP也有三重运算符（?:）。位操作符在PHP同样存在。

优先权

就和C以及Java一样！

2.7 控制流程结构

PHP有着与C一样的流程控制。我将在下面大概介绍。

if, else, elseif, if(): endif

if (表达式一)

{

...

}

elseif (表达式二)

{

...

}

else

```
{  
...  
}  
// 或者像Python一样  
if (表达式一):  
...  
...  
elseif (表达式二):  
...  
else :  
...  
endif;
```

Loops. while, do..while, for

```
while (表达式)  
{  
...  
}  
do  
{  
...  
}  
while (表达式);  
for (表达式一; 表达式二; 表达式三)  
{  
...  
}  
//或者像Python一样  
while (expr) :  
...  
endwhile;
```

switch

switch是对多重if-elseif-else结构的最好的替换：


```
switch ($i)
{
case 0:
print "i equals 0";
case 1:
print "i equals 1";
case 2:
print "i equals 2";
}
```

break, continue

break中断当前的循环控制结构。
continue被用来跳出剩下的当前循环并继续执行下一次循环。

require, include

就像C中的#include预处理一样。你在require中指定的那个文件将替代其在主文件中的位置。在有条件的引用文件时，可以使用include()。这样就使得你可以将复杂的PHP文件分割成多个文件并且在不同需要时分别引用它们。

2.8 函数

你可以像以下的例子一样定义自己的函数。函数的返回值可以是任何数据类型：

```
function foo (变量名一, 变量名二, ..., 变量名n)
{
echo "Example function.\n";
return $retval;
}
```

所有PHP代码都可以出现在函数定义中，甚至包括对其他函数和类的定义。函数必须在引用之前定义。

2.9 类

利用类模型建立类。可以参考PHP手册中对类的详细解释。

```
class Employee
{
```

```
var $empno; // 员工人数
var $empnm; // 员工姓名

function add_employee($in_num, $in_name)
{
    $this->empno = $in_num;
    $this->empnm = $in_name;
}

function show()
{
    echo "$this->empno, $this->empnm";
    return;
}

function changenm($in_name)
{
    $this->empnm = $in_name;
}

$sbabu = new Employee;
$sbabu->add_employee(10,"sbabu");
$sbabu->changenm("babu");

$sbabu->show();
```

2008-9-4 23:48:00

疯狂代码 <http://www.crazycoder.cn/>