第 1 部分

教　程

1 基础教程

1.1 简 介

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，对于像微软 ASP 这样的竞争者来说，PHP 无疑是另一种高效的选择。

1.1.1 什么是 PHP

PHP 是“PHP Hypertext Preprocessor”的首字母缩略词，是一种被广泛使用的开源脚本语言，PHP 脚本在服务器上执行，可供免费下载和使用。PHP 是一门令人惊叹的流行语言，它强大到足以成为在网络上最大的博客系统的核心（WordPress），它深邃到足以运行最大的社交网络（facebook），而它的易用程度足以成为初学者的首选服务器端语言。

PHP 文件能够包含文本、HTML、CSS 以及 PHP 代码，在服务器上执行，而结果以纯文本返回浏览器，其后缀是“.php”。

1.1.2 PHP 能够做什么

（1）PHP 能够生成动态页面内容；
（2）创建、打开、读取、写入、删除以及关闭服务器上的文件；
（3）接收表单数据；
（4）发送并取回 cookies；
（5）添加、删除、修改数据库中的数据；
（6）限制用户访问网站中的某些页面；
（7）对数据进行加密。

通过 PHP，你可以不受限于只输出 HTML，还能够输出图像、PDF 文件，甚至 Flash 影片，也可以输出任何文本，比如 XHTML 和 XML。

1.1.3 为什么使用 PHP

（1）PHP 支持运行于各种平台（Windows，Linux，Unix，Mac OS X 等）；
（2）兼容几乎所有服务器（Apache，IIS 等）；
（3）支持多种数据库；
（4）PHP 是免费的；

（5）易于学习，并可高效地运行在服务器端。

1.2 Wamp Server 安装

Wamp Server 是 Apache、MySQL 和 PHP 集成的开发环境，Wamp 集成了 PHP 以及 phpMyAdmin，集成了管理 MySQL 数据库的图形工具 SQLiteManager 和 phpMyAdmin 两种管理工具，既方便又快捷，可以说，能满足大多数 PHP 开发人员的需求。

1.2.1 安装方法和步骤

（1）下载 wamp5，假设文件名为：wamp5_1.7.4.exe；
（2）下载完毕，双击 wamp5_1.7.4.exe 文件；
（3）出现安装界面后，点击“Next”按钮；
（4）然后选中“I accept the agreement”选项，并点击“Next”按钮；
（5）选择默认安装路径，点击“Next”按钮；
（6）在弹出的对话框中，点击是“（Y）”按钮；
（7）点击“Next”按钮；
（8）点击“Next”按钮；
（9）点击“Install”按钮进行安装；
（10）点击“Next”按钮；
（11）点击“Next”按钮；
（12）点击是“（Y）”按钮；
（13）点击“finish”按钮，完成安装。

1.2.2 使用方法和步骤

（1）点击任务栏右下角图标；
（2）点击 www directory，打开 www 目录；
（3）在空白处右击，选择“新建文本文档”；
（4）修改文本文档文件名为 index.php（为了方便起见，默认主页）；
（5）双击 index.php，打开文件（打开方式，用记事本打开）；
（6）输入 index.php 源程序并保存：

```
<?php

   echo "Hello，world!";

?>
```

点击 Localhost 运行，结果如图 1-1-1 所示。

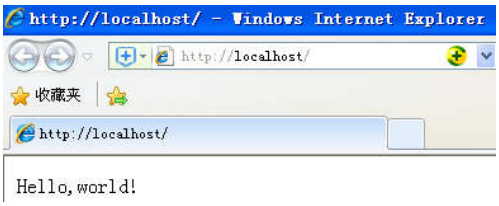

图 1-1-1　运行结果

1.3　PHP 语法

1.3.1　基础语法

PHP 脚本以“<?php”开头，以“?>”结尾。PHP 文件通常包含 HTML 标签以及一些 PHP 脚本代码。PHP 语句以分号“;”结尾，在 PHP 代码块的最后一行不必使用分号。PHP 默认文件扩展名是“.php”。

下面的例子是一个简单的 PHP 文件，其中包含了使用内建 PHP 函数“echo”在网页上输出文本“Hello World!”的一段 PHP 脚本（index.php）：

```
<!DOCTYPE html>
<html>
  <body>

    <h1>我的第一张 PHP 页面</h1>

    <?php
      echo "Hello World!";
    ?>
  </body>
</html>
```

运行结果如图 1-1-2 所示。

图 1-1-2　运行结果

1.3.2　PHP 中的注释

PHP 支持三种注释：

```
<?php
  //这是单行注释

  #这也是单行注释

  /*
   这是多行注释块

   它横跨了

   多行

  */
?>
```

注释用于：

（1）使其他人理解你正在做的工作。注释可以让其他程序员了解你在每个步骤进行的工作；

（2）提醒自己做过什么。大多数程序员都曾经历过一两年后对项目进行返工，然后不得不重新考虑他们做过的事情，注释可以记录你在写代码时的思路；

（3）调试程序时，根据需要注释掉不调试的程序。

通常在文件头注释文件名，修改时间，作者，电子邮件等信息：

```
/*
 * filename:   index.php
 * modify:     2012-05-22 14:15
 * author:     luker
 * e_mail:     service@pcboy.me
 */
```

1.3.3 大小写敏感问题

在 PHP 中，所有用户定义的函数、类和关键词（例如 if、else、echo 等）都对大小写不敏感。在下面的例子中，所有这三行 echo 语句都是合法的（等价的）：

```
<?php
  ECHO "Hello World!<br>";
  echo "Hello World!<br>";
```

```
    EcHo "Hello World!<br>";
?>
```

不过，在 PHP 中，所有变量都对大小写敏感。在下面的例子中，只有第一条语句会显示$color 变量的值（这是因为$color、$COLOR 以及$coLOR 被视作三个不同的变量）：

```
<?php
    $color="红色的";
    echo "我的汽车是$color <br>";
    echo "我的房子是$COLOR <br>";
    echo "我的船是$coLOR <br>";
?>
```

运行结果如图 1-1-3 所示。

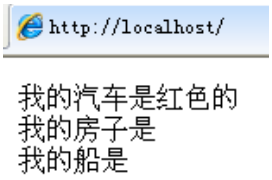

图 1-1-3　运行结果

$color 的值为“红色的”，$COLOR 和$coLOR 的值为 NULL。

1.4　变　量

变量是存储信息的容器。PHP 变量命名规则：

（1）变量以$符号开头，其后是变量的名称。

（2）变量名称对大小写敏感（$Submit 与$submit 是两个不同的变量）。

（3）变量名称必须以字母或下划线开头。

（4）变量名称不能以数字开头。

（5）变量名称只能包含字母数字字符和下划线（A～z、0～9 以及_）。

1.4.1　创建变量

PHP 没有创建变量的命令，变量会在首次为其赋值时被创建：

```
<?php
    $txt="Hello world!";
```

```
    $x=5;
    $y=10.5;
?>
```

以上语句执行后，变量 txt 会保存值“Hello world!”，变量 x 会保存值 5，变量 y 会保存值 10.5。

PHP 是一门类型松散的语言，在上面的例子中，不必告知 PHP 变量的数据类型，PHP 根据它的值，自动把变量转换为正确的数据类型，在诸如 C，C++以及 Java 之类的语言中，程序员必须在使用变量之前声明它的名称和类型。

1.4.2 变量的作用域

在 PHP 中，可以在脚本的任意位置对变量进行声明，变量的作用域指的是变量能够被引用/使用的那部分脚本，PHP 有局部、全局和静态三种不同的变量作用域。

1. 局部和全局作用域

函数之外声明的变量拥有全局作用域，只能在函数以外进行访问，函数内部声明的变量拥有局部作用域，只能在函数内部进行访问，下面的例子测试了带有局部和全局作用域的变量：

```
<?php
    $x=5;//全局作用域
    function myTest()
    {
        $y=10;//局部作用域

        echo "<p>测试函数内部的变量：</p>";

        echo "变量 x 是：$x";
        echo "<br>";
        echo "变量 y 是：$y";
    }
    myTest();
    echo "<p>测试函数之外的变量：</p>";

    echo "变量 x 是：$x";
    echo "<br>";
```

```
    echo "变量 y 是：$y";
?>
```

运行结果如图 1-1-4 所示。

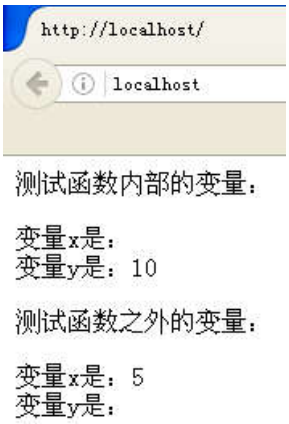

图 1-1-4　运行结果

在上例中，有两个变量$x 和$y 以及一个函数 myTest()，$x 是全局变量，因为它是在函数之外声明的，而$y 是局部变量，因为它是在函数内声明的。如果在 myTest()函数内部输出两个变量的值，$y 会输出在本地声明的值，但是无法输出$x 的值，因为它在函数之外创建，如果在 myTest()函数之外输出两个变量的值，那么会输出$x 的值，但是不会输出$y 的值，因为它是局部变量，在 myTest()内部创建。你可以在不同的函数中创建名称相同的局部变量，因为局部变量只能被在其中创建它的函数识别。

2. global 关键词

global 关键词用于访问函数外的全局变量。要做到这一点，请在（函数内部）变量前面使用 global 关键词。

```
<?php
    $x=5;
    $y=10;
    function myTest()
    {
        global $x , $y;

        $y=$x+$y;
    }
    myTest();

    echo $y; //输出 15
```

```
?>
```

PHP 同时在名为**$GLOBALS** 的数组中存储了所有的全局变量，下标为变量名，这个数组在函数内也可以访问，并能够用于直接更新全局变量，上面的例子可以这样重写：

```
<?php
  $x=5;
  $y=10;
  function myTest()
  {
    $GLOBALS['y']=$GLOBALS['x']+$GLOBALS['y'];
  }
  myTest();
  echo $y; //输出 15
?>
```

或者通过参数传递：

```
<?php
  $x=5;
  $y=10;
  function myTest($x，&$y)
  {
    $y=$x+$y;
  }
  myTest($x，$y);

  echo $y; //输出 15
?>
```

3. static 关键词

通常，当函数完成/执行后，会删除所有变量，如果要保留某个局部变量，声明变量时要使用 **static** 关键词。

```
<?php
  @session_start();
  ini_set("display_errors"，"On");
  static  $DB_HOST="localhost";//数据库服务器
  static  $DB_NAME="acm";//数据名
  static  $DB_USER="root";//数据库用户
```

```
    static  $DB_PASS="root";//数据库密码
    static  $OJ_NAME="SUSE ACM";//系统名
    static  $OJ_HOME="./";//系统根目录
    static  $OJ_ADMIN="service@pcboy.me";//管理员邮箱
    static  $OJ_DATA="/home/judge/data";//测试数据在服务器上的位置
    //static   $OJ_LANG="en";//语言系统  cn->中文  en->英文
    static  $OJ_SIM=false; //相似度检查(此系统暂没有用到)
    static  $OJ_DICT=false;//开启划词翻译插件(仅在中文模式下可用)
    static  $OJ_LANGMASK=1008;//支持编程语言  1->c 2->c++
    static  $OJ_EDITE_AREA=true;//是否开启代码高亮编辑框
    static  $OJ_AUTO_SHARE=false;//是否共享代码，开启后其他人可以查看用户提交
的代码
    static  $OJ_VCODE=true;//是否开启登录注册验证码
    static  $OJ_APPENDCODE=false;//是否显示附加信息
    static  $OJ_MEMCACHE=false;//是否开启 memcache 缓存
    static  $OJ_MEMSERVER="127.0.0.1";//memcache 服务器 ip
    static  $OJ_MEMPORT=11211;//memcache 端口
    static  $OJ_TEMPLATE="default";//模板名，位于./template/下
?>
```

然后，每当函数被调用时，这个变量所存储的信息都是函数最后一次被调用时所包含的信息，但该变量仍然是函数的局部变量。

我们可以利用静态变量求阶乘。从键盘输入数 *n*（确保 *n*>0），阶乘用函数实现，要求函数能“记忆”上一次阶乘的结果，即第一次调用时，函数计算的结果是 1!，第二次调用时，函数记住了上次函数调用的结果（1!），在此基础上计算出 2!。

```
<?php
    function fac($n)
    {
      static $f=1;
      $f=$f*$n;
      return($f);
    }
    for($i=1;$i<=5;$i++)
      printf($i."!=".fac($i)."<br>");
?>
```

1.5 echo 和 print 语句