* PHP 能够生成动态页面内容
* PHP 能够创建、打开、读取、写入、删除以及关闭服务器上的文件
* PHP 能够接收表单数据
* PHP 能够发送并取回 cookies
* PHP 能够添加、删除、修改数据库中的数据
* PHP 能够限制用户访问网站中的某些页面
* PHP 能够对数据进行加密
* PHP 运行于各种平台（Windows, Linux, Unix, Mac OS X 等等）
* PHP 兼容几乎所有服务器（Apache, IIS 等等）
* PHP 支持多种数据库
* PHP 是免费的。请从官方 PHP 资源下载：[www.php.net](http://www.php.net/" \t "_blank)
* PHP 易于学习，并可高效地运行在服务器端

如需开始使用 PHP，您可以：

* 使用支持 PHP 和 MySQL 的 web 主机
* 在您的 PC 上安装 web 服务器（Apache, IIS 等等），然后安装 PHP 和 MySQL。
* **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

\*/

?>

所有用户定义的函数、类和关键词（例如 if、else、echo 等等）都对大小写不敏感。

所有变量都对大小写敏感。

变量以 $ 符号开头，其后是变量的名称

PHP 有三种不同的变量作用域：

* local（局部）
* global（全局）
* static（静态）

## Local 和 Global 作用域

函数之外声明的变量拥有 Global 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 LOCAL 作用域，只能在函数内部进行访问。

## PHP global 关键词

global 关键词用于访问函数内的全局变量。

要做到这一点，请在（函数内部）变量前面使用 global 关键词：

### 实例

<?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[index] 的数组中存储了所有的全局变量。下标存有变量名。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。

<?php

$x=5;

$y=10;

function myTest() {

$GLOBALS['y']=$GLOBALS['x']+$GLOBALS['y'];

}

myTest();

echo $y; // 输出 15

?>

可以用三种格式规定整数：十进制、十六进制（前缀是 0x）或八进制（前缀是 0）

在 PHP 中，必须明确地声明对象。

首先我们必须声明对象的类。对此，我们使用 class 关键词。类是包含属性和方法的结构。

## PHP strlen() 函数

strlen() 函数返回字符串的长度，以字符计。

下例返回字符串 "Hello world!" 的长度：

<?php

echo strlen("Hello world!");

?>

内置函数很少有对象函数？

如需设置常量，请使用 define() 函数 - 它使用三个参数：

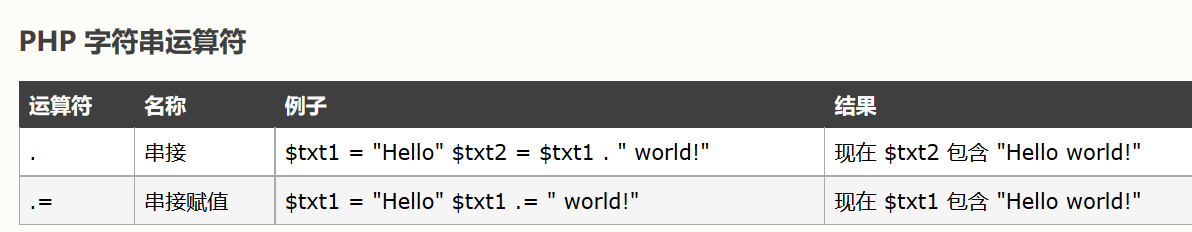
1. 首个参数定义常量的名称
2. 第二个参数定义常量的值
3. 可选的第三个参数规定常量名是否对大小写不敏感。默认是 false。

<?php

define("GREETING", "Welcome to W3School.com.cn!");

echo GREETING;

?>



## PHP echo 和 print 语句

echo 和 print 之间的差异：

* echo - 能够输出一个以上的字符串
* print - 只能输出一个字符串，并始终返回 1

提示：echo 比 print 稍快，因为它不返回任何值。

|  |  |  |  |
| --- | --- | --- | --- |
| === | 全等（完全相同） | $x === $y | 如果 $x 等于 $y，且它们类型相同，则返回 true。 |

|  |  |  |  |
| --- | --- | --- | --- |
| === | 全等 | $x === $y | 如果 $x 和 $y 拥有相同的键/值对，且顺序相同类型相同，则返回 true。 |

## PHP foreach 循环

foreach 循环只适用于数组，并用于遍历数组中的每个键/值对。

foreach ($array as $value) {

code to be executed;

}

在 PHP 中，有三种数组类型：

* *索引数组* - 带有数字索引的数组
* *关联数组* - 带有指定键的数组
* *多维数组* - 包含一个或多个数组的数组

有两种创建索引数组的方法：

索引是自动分配的（索引从 0 开始）：

$cars=array("Volvo","BMW","SAAB");

或者也可以手动分配索引：

$cars[0]="Volvo";

$cars[1]="BMW";

$cars[2]="SAAB";

## PHP 关联数组

关联数组是索引不是整数的数组。

关联数组是使用您分配给数组的指定键的数组。

有两种创建关联数组的方法：

$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");

或者：

$age['Peter']="35";

$age['Ben']="37";

$age['Joe']="43";

## 遍历关联数组

如需遍历并输出关联数组的所有值，您可以使用 foreach 循环，就像这样：

$x是键，$x\_value是键对应的值

<?php

$age=array("Bill"=>"35","Steve"=>"37","Peter"=>"43");

foreach($age as $x=>$x\_value) {

echo "Key=" . $x . ", Value=" . $x\_value;

echo "<br>";

}

?>

**超全局变量 在 PHP 4.1.0 中引入，是在全部作用域中始终可用的内置变量。**

这些超全局变量是：

* $GLOBALS
* $\_SERVER
* $\_REQUEST
* $\_POST
* $\_GET
* $\_FILES
* $\_ENV
* $\_COOKIE
* $\_SESSION

## PHP 表单验证

htmlspecialchars() 函数把特殊字符转换为 HTML 实体。这意味着 < 和 > 之类的 HTML 字符会被替换为 &lt; 和 &gt; 。

（通过 PHP trim() 函数）去除用户输入数据中不必要的字符（多余的空格、制表符、换行）

（通过 PHP stripslashes() 函数）删除用户输入数据中的反斜杠（\）

接下来我们创建一个检查函数（相比一遍遍地写代码，这样效率更好）。

我们把函数命名为 test\_input()。

现在，我们能够通过 test\_input() 函数检查每个 $\_POST 变量，脚本是这样的：

<?php

// 定义变量并设置为空值

$name = $email = $gender = $comment = $website = "";

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$name = test\_input($\_POST["name"]);

$email = test\_input($\_POST["email"]);

$website = test\_input($\_POST["website"]);

$comment = test\_input($\_POST["comment"]);

$gender = test\_input($\_POST["gender"]);

}

function test\_input($data) {

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

}

?>

**必填输入字段，并创建需要时所用的错误消息。**

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

if (empty($\_POST["name"])) {

$nameErr = "Name is required";

} else {

$name = test\_input($\_POST["name"]);

}

}

接下来是验证输入数据，即“Name 字段是否只包含字母和空格？”，以及“E-mail 字段是否包含有效的电子邮件地址语法？”，并且如果填写了 Website 字段，“这个字段是否包含了有效的 URL？”。

$name = test\_input($\_POST["name"]);

if (!preg\_match("/^[a-zA-Z ]\*$/",$name)) {

$nameErr = "只允许字母和空格！";

}

**本节展示如何在用户提交表单后保留输入字段中的值**

Name: <input type="text" name="name" value="<?php echo $name;?>">

**服务器端包含 (SSI) 用于创建可在多个页面重复使用的函数、页眉、页脚或元素。**

include 和 require 语句是相同的，除了错误处理方面：

require 会生成致命错误（E\_COMPILE\_ERROR）并停止脚本

include 只生成警告（E\_WARNING），并且脚本会继续

假设我们有一个名为 "vars.php" 的文件，其中定义了一些变量：

<?php

$color='银色的';

$car='奔驰轿车';

?>

然后，如果我们引用这个 "vars.php" 文件，就可以在调用文件中使用这些变量：

<html>

<body>

<h1>欢迎访问我的首页！</h1>

<?php

include 'vars.php';

echo "我有一辆" . $color . $car "。";

?>

</body>

</html>

20180529

<http://php.net/manual/zh/>

PHP（“PHP: Hypertext Preprocessor”，超文本预处理器的字母缩写

PHP 脚本主要用于以下三个领域：

1服务端脚本。这是 PHP 最传统，也是最主要的目标领域。开展这项工作需要具备以下三点：PHP 解析器（CGI 或者服务器模块）、web 服务器(IIS, Apache 等)和 web 浏览器。

2命令行脚本。可以编写一段 PHP 脚本，并且不需要任何服务器或者浏览器来运行它。通过这种方式，仅仅只需要 PHP 解析器来执行。 这些脚本也可以用来处理简单的文本

3编写桌面应用程序。以利用 PHP-GTK 来编写这些程序。用这种方法，还可以编写跨平台的应用程序

PHP 能够在所有的主流操作系统上[使用](http://php.net/manual/zh/install.php)，PHP已经支持了大多数的 web 服务器。对于大多数的服务器，PHP 提供了一个模块；还有一些 PHP 支持 CGI 标准，使得 PHP 能够作为 CGI 处理器来工作。

综上所述，使用 PHP，可以自由地选择操作系统和 web 服务器。同时，还可以在开发时选择使用面对过程和面对对象，或者两者混和的方式来开发

PHP 最强大最显著的特性之一，是它支持[很大范围的数据库](http://php.net/manual/zh/refs.database.php)。

PHP 还支持利用诸如 LDAP、IMAP、SNMP、NNTP、POP3、HTTP、COM（Windows 环境）等不计其数的协议的服务

PHP 具有极其有效的[文本处理](http://php.net/manual/zh/refs.basic.text.php)特性

Windows下配置php

### Microsoft IIS 7.0 及更高版本

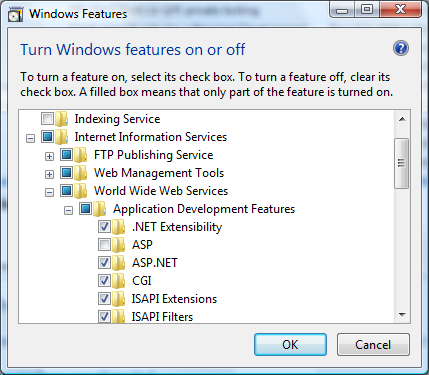
#### 在 IIS 中激活 FastCGI 支持

选中“CGI”的选择框

默认安装的 IIS 中 FastCGI 模块被关闭。要激活其的步骤在不同版本的 Windows 下不同。

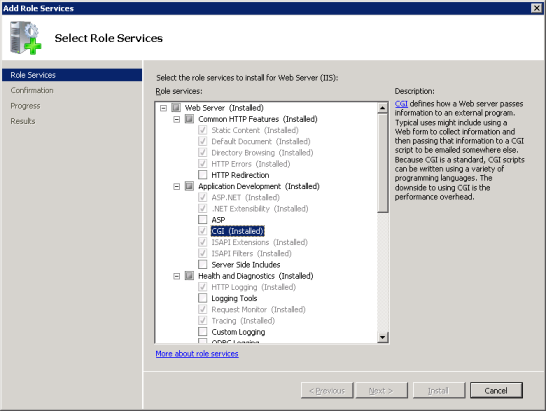
要在 Windows Vista SP1 和 Windows 7 中激活 FastCGI 支持：

1. 在 Windows 开始菜单中选择“运行...”（或在搜索框内），输入“optionalfeatures.exe”并按“确定”（或敲回车键）；
2. 在“Windows 功能”对话框中展开“Internet 信息服务”，“万维网服务”，“应用程序开发功能”，并选中“CGI”的选择框；
3. 点击确定按钮并等待安装完成。



要在 Windows Server 2008 和 Windows Server 2008 R2 中激活 FastCGI 支持：

1. 在 Windows 开始菜单中选择 "Run:"，输入 "CompMgmtLauncher" 并点击 "Ok"；
2. 如果在 "Roles" 节点下没有 "Web Server (IIS)" role，则点击 "Add Roles" 添加之；
3. 如果存在 "Web Server (IIS)" role，则点击 "Add Role Services" 并激活 "Application Development" 组之下的 "CGI" 选择框；
4. 点击 "Next" 及 "Install"，等待安装完成。



#### 配置 IIS 以处理 PHP 请求

按以下示例在 *php.ini* 文件中配置 针对 CGI- 和 FastCGI- 的指令：

**Example #8 *php.ini* 中的 CGI 和 FastCGI 设定**

fastcgi.impersonate = 1

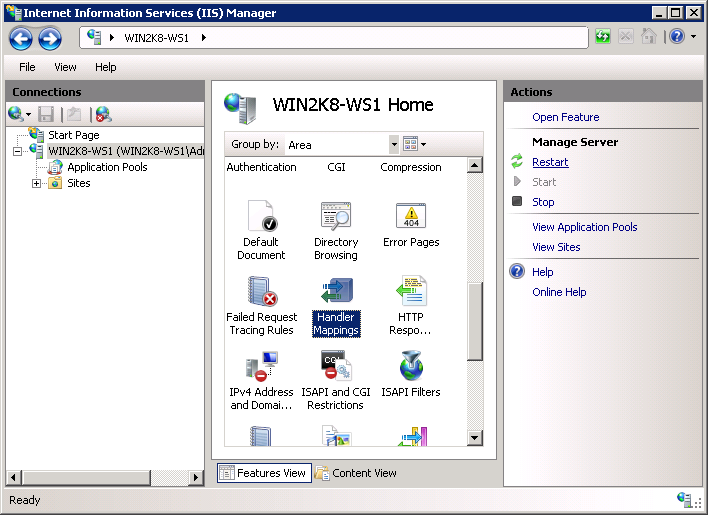
fastcgi.logging = 0

cgi.fix\_pathinfo=1

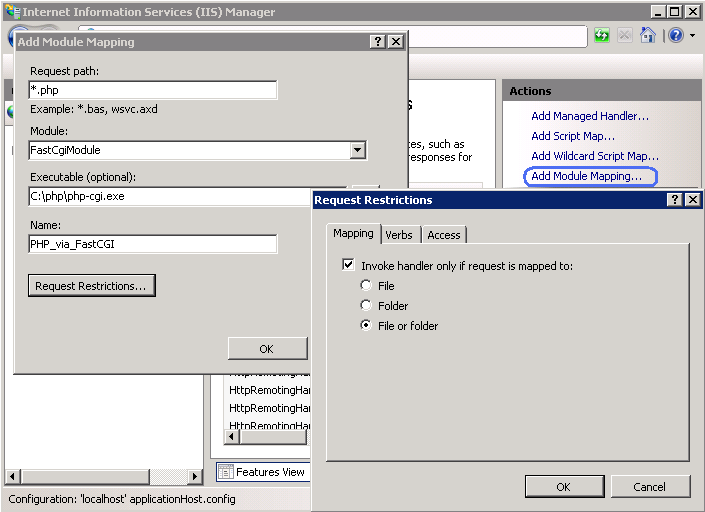
cgi.force\_redirect = 0

按照以下步骤在 IIS 管理界面中创建 PHP 的程序映射：

1. 在 Windows 开始菜单中选择“运行...”，输入“inetmgr”并点击“确定”；
2. 在 IIS 管理器中左边面板“连接”下面的树状图中选择该服务器的节点；
3. 在中间面板下方的“功能视图”页面打开“处理程序映射”功能；



1. 在右边“操作”面板中点击“添加模块映射...”；
2. 在“添加模块映射”对话框中输入以下内容：
   * 请求路径(P)：\*.php
   * 模块(M)：FastCgiModule
   * 可执行文件(可选)(E)：C:\[Path to PHP installation]\php-cgi.exe
   * 名称(N)：PHP\_via\_FastCGI
3. 点击“请求限制(R)...”按钮并选中“仅当请求映射至以下内容时才调用处理程序(I)：”然后选择“文件或文件夹(O)”；
4. 在所有对话框中点击确定以保存配置。



#### 角色扮演及文件系统访问

在 IIS 中使用 PHP 建议激活 FastCGI 的角色扮演功能。此功能在 php.ini 中由 fastcgi.impersonate 指令控制。激活角色扮演后，PHP 将以 IIS 所认证的用户帐号身份进行所有的文件系统操作。这将确保即使在（同一个主机）不同的 IIS 网站下使用了同一个 PHP 进程，只要每个网站使用了不同的用户帐号作为 IIS 身份认证，则这些网站的 PHP 脚本将不能访问彼此的文件。

例如在 IIS 7 中，默认配置下的匿名认证将使用内置的用户帐号 IUSR 作为默认身份。这意味着要使得 IIS 能够运行 PHP 脚本，至少要将这些脚本的读取权限授予 IUSR 帐号。如果 PHP 程序需要对某些文件或文件夹进行写入操作，那 IUSER 帐号也需要有相对应的写入权限。

在 IIS 7 中要查看哪个用户帐号被用作匿名认证的身份，使用以下命令。将其中的 "Default Web Site" 替换为自己使用的网站名。在输出的 XML 配置单元中查找 userName 属性。（注意：要以管理员身份运行此命令行）

**Example #11 确定用于 IIS 匿名认证的用户帐号**

%windir%\system32\inetsrv\appcmd.exe list config "Default Web Site" ^

/section:anonymousAuthentication

<system.webServer>

<security>

<authentication>

<anonymousAuthentication enabled="true" userName="IUSR" />

</authentication>

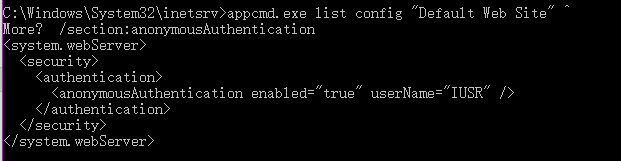
</security>

</system.webServer>

由于xml文件太长。

先输入appcmd.exe list config "Default Web Site" ^ 回车。命令行提示 More?

然后输入/section:anonymousAuthentication，只显示这一节点



**Note**:

如果在 anonymousAuthentication 单元中没有显示 userName 属性，或者其值为一个空字符串，那意味着应用程序池的身份被用于该网站的匿名身份。

要修改文件及文件夹的权限，使用 Windows 资源管理器或者 icacls 命令行。

**Example #12 配置文件访问权限**

icacls C:\inetpub\wwwroot\upload /grant IUSR:(OI)(CI)(M)

### Microsoft Windows 下的 Apache 1.3.x

有两种方式让 PHP 工作在 Windows 下的 Apache 1.3.x 中。首先是使用 CGI 可执行程序（PHP 4 下为 *php.exe*，PHP 5 下为 *php-cgi.exe*），另外一种方式是使用 Apache 模块 DLL。无论是那种方式，都需要修改 *httpd.conf* 来配置 Apache，使 PHP 能够在其上运行，然后都需要重启服务。

值得注意的是，现在 Windows 下的 SAPI 模块已经稳定得多，我们建议首先考虑使用 SAPI 而不要使用 CGI 可执行程序。因为 SAPI 更加透明和安全。

在修改完配置文件后，请记得重启 Apache 服务。例如，如果把 Apache 作为 Windows 的一个服务来运行，那么在命令提示行下使用 **NET STOP APACHE**命令然后再使用 **NET START APACHE**命令便可重启服务。也可以使用重启 Apache 服务的快捷方式来重启。

**Note**: 记住当在 Windows 环境下的 Apache 配置文件中添加路径值时，所有的反斜线，如 *c:\directory\file.ext*，应转换为正斜线： *c:/directory/file.ext*。对目录来说，也必须由斜线结尾。

#### 作为 Apache 的模块安装

应该将下面几行加入 Apache 的 httpd.conf 文件：

这里假设 PHP 安装在 c:\php。如果不是这样请根据情况修改路径。

对于 PHP 5：

# 在 LoadModule 一节的末尾添加

LoadModule php5\_module "C:/php/php5apache.dll"

# 在 AddModule 一节的末尾添加

AddModule mod\_php5.c

两个 PHP 版本都需要添加的内容：

# 将下面这行添加到 <IfModule mod\_mime.c> 条件块中

AddType application/x-httpd-php .php

# 如果要使用语法高亮的 .phps 文件，需要添加

AddType application/x-httpd-php-source .phps

#### 作为 CGI 可执行文件的安装

如果按照[手工安装步骤](http://php.net/manual/zh/install.windows.manual.php)将 PHP 解压到 C:\php\，需要在 Apache 的配置文件中添加如下内容以使 PHP 按照 CGI 方式运行:

**Example #18 PHP 以 CGI 方式运行在 Apache 1.3.x**

ScriptAlias /php/ "c:/php/"

AddType application/x-httpd-php .php

# 对于 PHP 4

Action application/x-httpd-php "/php/php.exe"

# 对于 PHP 5

Action application/x-httpd-php "/php/php-cgi.exe"

# 指定 php.ini 所在目录

SetEnv PHPRC C:/php

请注意第二行的配置可以在默认的 httpd.conf 中找到，但是是被注释掉的。也请记得将 c:/php/替换为 PHP 所在的真实路径。

**Warning**

服务器使用 CGI 方式进行部署可能存在几个公开的缺陷。请阅读 [CGI 安全](http://php.net/manual/zh/security.cgi-bin.php)一章 以学习 如何抵御这些攻击。

如果想发布语法高亮的 php 文件，没有类似于模块方式下 PHP 那种方便的方法。如果选择了 CGI 方式运行 PHP，需要使用 [highlight\_file()](http://php.net/manual/zh/function.highlight-file.php) 函数来进行语法高亮。创建一个 PHP 文件，加入下述代码即可：<?php highlight\_file('some\_php\_script.php'); ?>。

### Microsoft Windows 下的 Apache 2.x

本节包括在 Microsoft Windows 系统中针对 Apache 2.x 安装 PHP 的指导与说明。在其它页面也有 [Apache 1.3.x 用户指导与说明](http://php.net/manual/zh/install.windows.legacy.index.php#install.windows.legacy.apache1)。

**Note**:

应该先阅读[手工安装步骤](http://php.net/manual/zh/install.windows.legacy.index.php" \l "install.windows.legacy.manual)！

**Note**: **Apache 2.2 支持**

Apache 2.2 用户应留意对于 Apache 2.2 的 DLL 文件名是 php5apache2\_2.dll 而不是 php5apache2.dll，并且只在 PHP 5.2.0 以及更高版本中出现。参见 [broken link]。

强烈建议阅读 [» Apache 文档](http://httpd.apache.org/docs/current/)来加深对 Apache 2.x 服务器的基本理解。此外在继续下去之前考虑先阅读一下 Apache 2.x 的 [» Windows 下使用说明](http://httpd.apache.org/docs/current/platform/windows.html)。

Apache 2.x 被设计运行于 Windows 版的服务器平台下，例如 Windows NT 4.0，Windows 2000，Windows XP 或 Windows 7。虽然 Apache 2.x 可以在 Windows 9x 下勉强运行，但对此平台的支持尚未完成，某些功能无法正确工作。对此并无补救计划。

下载最新版本的 [» Apache 2.x](http://httpd.apache.org/) 以及适合的 PHP 版本。先完成[手工安装步骤](http://php.net/manual/zh/install.windows.manual.php)后再回来继续将 PHP 集成入 Apache。

Windows 下有三种方法使 PHP 工作于 Apache 2.x 之中。可以以 handler，CGI，或者 FastCGI 方式运行 PHP。

**Note**: 记住当在 Windows 环境下的 Apache 配置文件中添加路径值时，所有的反斜线，如 c:\directory\file.ext，应转换为正斜线： c:/directory/file.ext。对目录来说，也必须由斜线结尾。

#### 以 Apache handler 方式安装

需要将以下几行加入到 Apache 的 httpd.conf 配置文件中以加载 Apache 2.x 的 PHP 模块：

**Example #19 PHP 在 Apache 2.x 中作为 handler**

#

LoadModule php5\_module "c:/php/php5apache2.dll"

AddHandler application/x-httpd-php .php

# 配置 php.ini 的路径

PHPIniDir "C:/php"

**Note**: 记得用自己 PHP 实际所在的路径替换掉上例中的 c:/php/。要留意在 LoadModule 指令中使用了 php5apache2.dll 或者 php5apache2\_2.dll，并且该文件确实位于所指定的位置。

以上配置将使 PHP 处理任何具有 .php 后缀的文件，即使该文件还有其它的文件后缀。例如一个名为 example.php.txt 的文件将被作为 PHP 文件运行。要确保只有以 .php 结尾的文件才被执行，则用以下配置替换上面的：

<FilesMatch \.php$>

SetHandler application/x-httpd-php

</FilesMatch>

#### 以 CGI 方式运行 PHP

要更好地理解在 Apache 下运行 CGI，请参阅 [» Apache CGI 文档](http://httpd.apache.org/docs/current/howto/cgi.html)。

要将 PHP 以 CGI 方式运行，需要将 php-cgi 文件放入到用 ScriptAlias 指令所指定的 CGI 目录中。

然后需要给 PHP 文件中添加一 #! 的行来指明 PHP 可执行文件的位置：

**Example #20 Apache 2.x 下 CGI 方式的 PHP**

#!C:/php/php.exe

<?php

phpinfo();

?>

**Warning**

服务器使用 CGI 方式进行部署可能存在几个公开的缺陷。请阅读 [CGI 安全](http://php.net/manual/zh/security.cgi-bin.php)一章 以学习 如何抵御这些攻击。

#### 以 FastCGI 方式运行 PHP

以 FastCGI 方式运行 PHP 比起 CGI 方式有很多优点。设定的方式很直接：

从 [» http://httpd.apache.org/mod\_fcgid/](http://httpd.apache.org/mod_fcgid/) 取得 mod\_fcgid，该站点有 Win32 可执行文件的下载。按照下载文件中的指示安装此模块。

按以下方法配置 web 服务器，注意用自己系统上的路径替换其中相应的内容：

**Example #21 配置 Apache 以 FastCGI 方式运行 PHP**

LoadModule fcgid\_module modules/mod\_fcgid.so

# Where is your php.ini file?

FcgidInitialEnv PHPRC "c:/php"

AddHandler fcgid-script .php

FcgidWrapper "c:/php/php-cgi.exe" .php

此时具有 .php 后缀的文件将被 PHP FastCGI 所解析执行。

<https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/hh994592%28v%3dws.11%29>