XAMPP如何下载及安装

XAMPP可通过官网进行下载，下载好后解压最好选在D盘目录下安装，安装好打开xampp-control运行程序
今天将要介绍一款用于建站开发的程序，它的安装和使用都非常简单。接下来将具体为大家介绍xampp的下载与安装，具有一定参考价值，希望对大家学习有所帮助

XAMPP的介绍：

XAMPP是一个集（Apache+MySQL+PHP+PERL）为一体的功能强大的建站集成软件包。它可以在Windows、Linux、Solaris、Mac OS X 等多种操作系统下安装使用。支持多种语言：简体中文、繁体中文、英文、韩文、俄文等等。

这个功能强大的软件包原来的名字是 LAMPP，但是为了避免误解，最新的几个版本就改名为 XAMPP 了。它的使用简便快捷，在性能上易于操作和浏览。正因为它的简单而吸引了大量的人用其建网站，博客。

XAMPP的下载

下载地址：https://www.apachefriends.org/zh_cn/download.html

在这个页面上我们可以选择适合自己操作系统的安装包进行下载（Windows系统、Linux系统、Mac OS X 系统等），在本篇文章将以Windows系统为例

添加系统变量

把PHP.exe所在文件夹路径（“**C:\XAMPP\php**”）添加进**环境变量-系统变量-Path**中（直接搜索框搜索系统变量便可找到）。

在cmd中输入php -v，检查是否配置成功

或者下载PHP

https://www.php.net/distributions/php-7.3.24.tar.gz

```
brew install openssl
brew install gettext
```

```
brew install zlib

tar -zvxf php-7.3.24.tar.gz
cd php-7.3.24

./configure --prefix=/usr/local/php/ \
--with-config-file-path=/usr/local/php/etc \
--with-config-file-scan-dir=/usr/local/php/etc/conf.d \
--enable-fpm \
--with-fpm-user=www \
--with-fpm-group=www \
--with-mysqli \
--with-pdo-mysql \
--with-iconv-dir \
--with-freetype-dir \
--with-zlib=/opt/homebrew/Cellar/zlib/1.2.11 \
--with-jpeg-dir=/opt/homebrew/Cellar/jpeg/9d/ \
--with-png-dir=/opt/homebrew/Cellar/libpng/1.6.37/ \
--with-libxml-dir=/usr/bin/xml2-config \
--enable-xml \
--disable-rpath \
--enable-bcmath \
--enable-shmop \
--enable-sysvsem \
--enable-inline-optimization \
--with-curl=/opt/homebrew/Cellar/curl/7.80.0/ \
--enable-mbregex \
--enable-mbstring \
--with-mcrypt \
--enable-ftp \
--with-gd \
--enable-gd-native-ttf \
--with-openssl=/opt/homebrew/Cellar/openssl@1.1/1.1.1k \
--with-mhash \
--enable-pcntl \
--enable-sockets \
--with-xmlrpc \
--enable-zip \
--enable-soap \
--without-pear \
--with-gettext \
--disable-fileinfo \
--enable-maintainer-zts \
--enable-mysqlnd

make && sudo make install
```

MAC安装PHP

```
# zsh 替换 brew bintray 镜像
echo 'export HOMEBREW_BOTTLE_DOMAIN=https://mirrors.ustc.edu.cn/homebrew-bottles' >>
~/.zshrc
source ~/.zshrc

# bash 替换 brew bintray 镜像
echo 'export HOMEBREW_BOTTLE_DOMAIN=https://mirrors.ustc.edu.cn/homebrew-bottles' >>
~/.bash_profile
source ~/.bash_profile

# 刷新源
brew update

# 搜索PHP
> brew search php

brew-php-switcher   php-cs-fixer        php@7.3             phplint             phpstan
            pup
php ✔               php-cs-fixer@2      php@7.4             phpmd               phpunit
php-code-sniffer    php@7.2             phpbrew             phpmyadmin          pcp

brew install php@7.4
```

安装位置

```
/usr/bin/php

# 最新版Mac系统
/opt/homebrew/opt/php@8.0/bin/
# 配置文件位置
/opt/homebrew/etc/php/8.0/php.ini
```

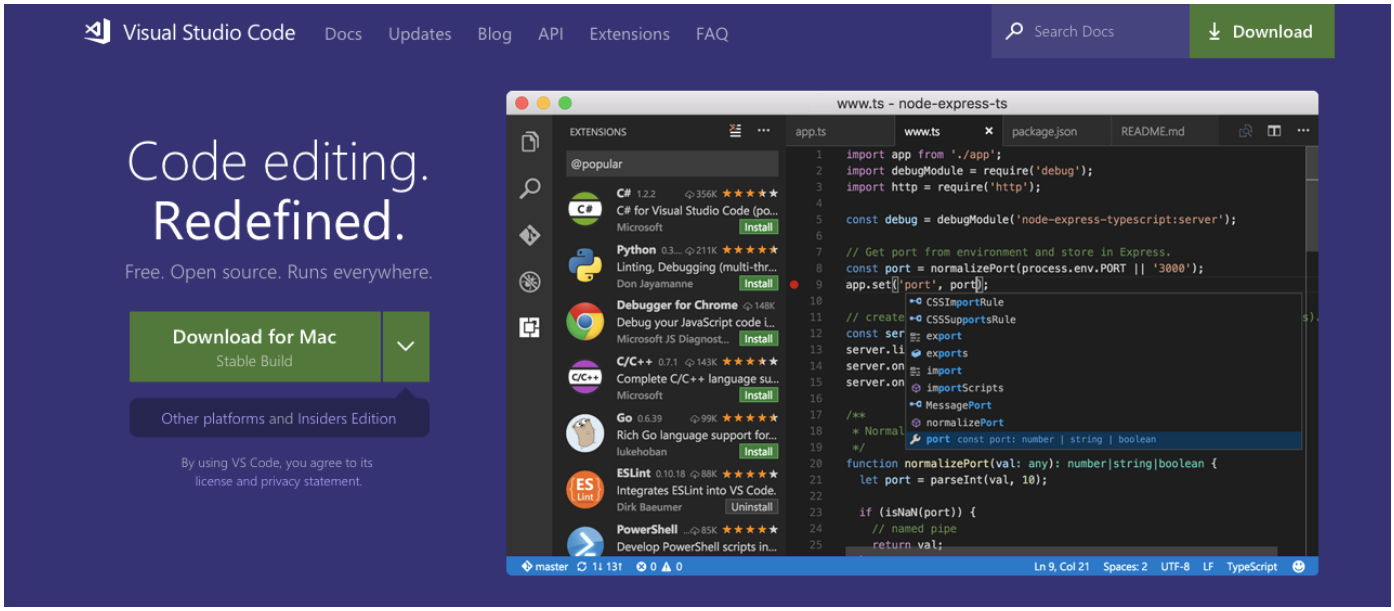
配置文件位置

```
vim /opt/homebrew/etc/php/8.0/php.ini
```

VS Code 的安装

- VS Code 官网：https://code.visualstudio.com

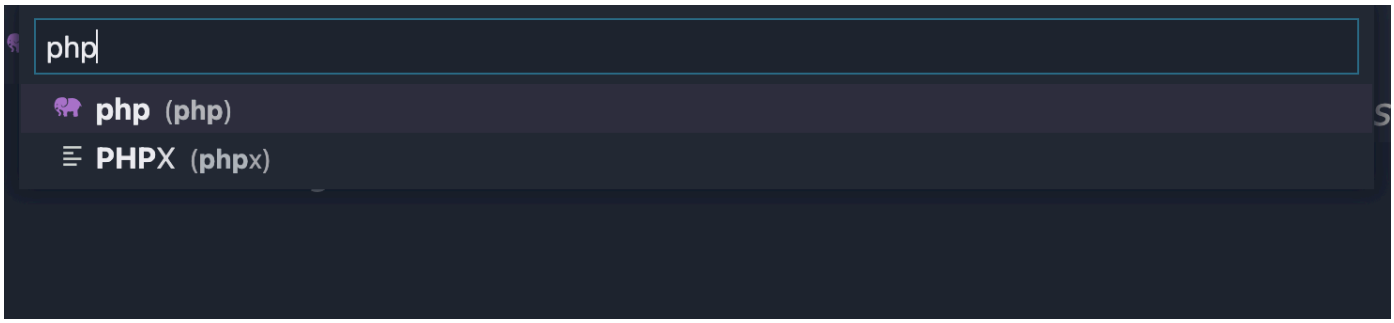
VS Code 的安装很简单，直接去官网下载安装包，然后双击安装即可。



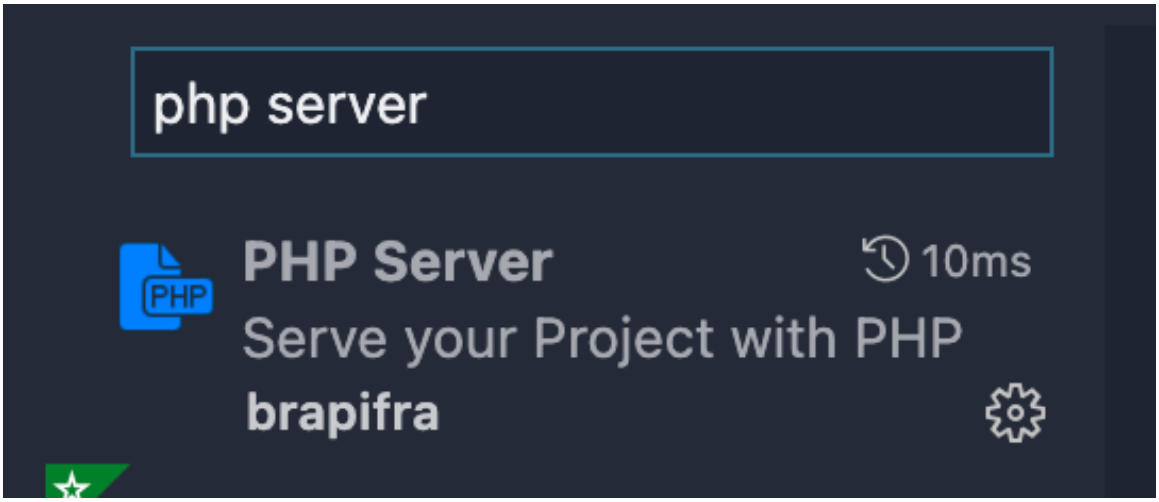
上图中，直接点击 download，一键下载安装即可。

私人订制：VS Code 的常见配置

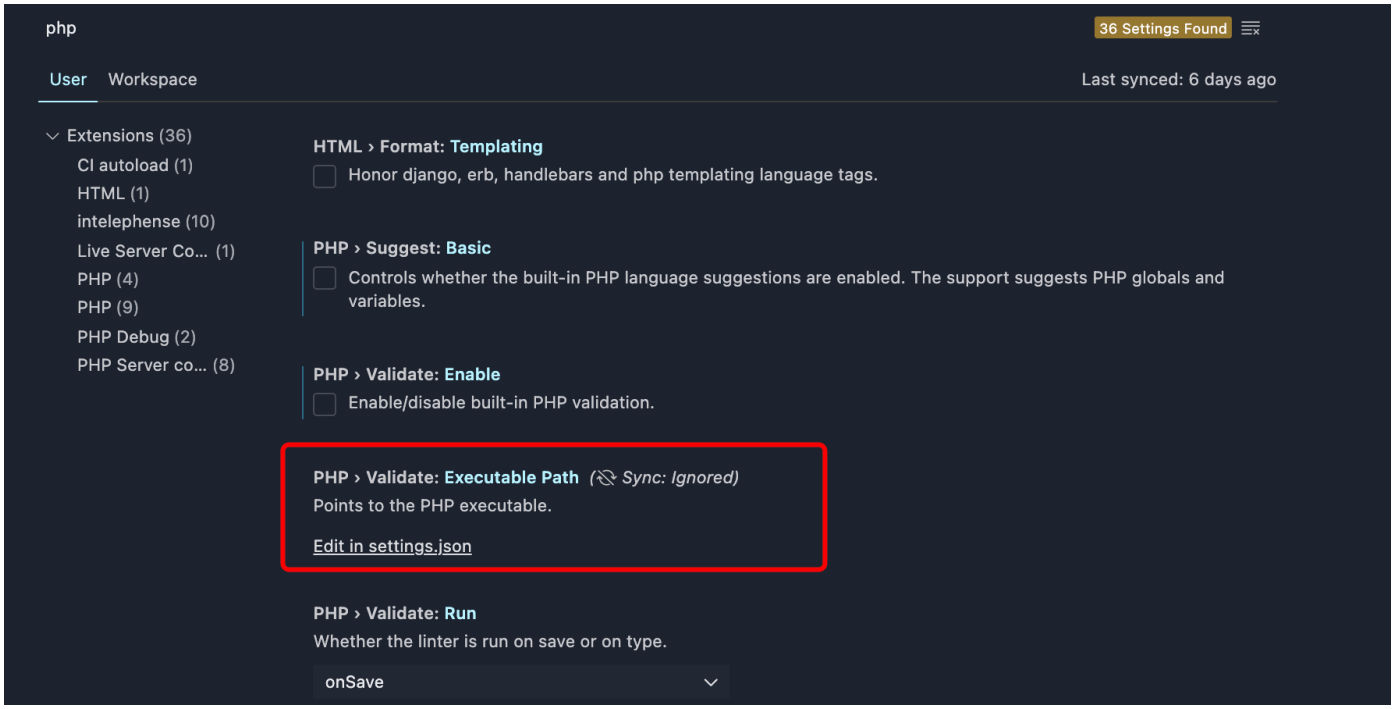
打开文件，选择PHP



还需要下载一个PHP Server



指定php的路径：文件->首选项->设置




```
{
    "php.validate.executablePath": "/usr/bin/php",
    "phpserver.phpConfigPath": "/etc/php.ini",
}

{
    "phpserver.phpConfigPath": "/opt/homebrew/etc/php/8.0/php.ini",
    "php.validate.executablePath": "/opt/homebrew/Cellar/php/8.0.13/bin/php",
}
```

2、open in browser

安装`open in browser`插件后，在 HTML 文件中「右键选择 --> Open in Default Browser」，即可在浏览器中预览网页。

3、安装PHP Server

安装`PHP Server`,可以选择右键 PHP Server：Serve project，直接跳转到浏览器

PHP 是什么?

PHP（“`PHP: Hypertext Preprocessor`”，超文本预处理器的字母缩写）是一种被广泛应用的开放源代码的多用途脚本语言，它可嵌入到 HTML中，尤其适合 web 开发。

以上是一个简单的回答，不过这是什么意思呢？请看如下例子：

示例 #1 一个介绍性的范例

```
<html>

  <head>
    <title>Example</title>
  </head>
  <body>

      <?php
      echo "Hi, I'm a PHP script!";
      ?>

    </body>

</html>
```

与用大量的命令来编写程序以输出 HTML 不同的是，PHP 页面就是 HTML，只不过在其中嵌入了一些代码来做一些事情（在本例中输出了 "Hi, I'm a PHP script!"）。

PHP 代码被包含在特殊的起始符和结束符 中，使得可以进出“PHP 模式”。

和客户端的 JavaScript 不同的是，PHP 代码是运行在服务端的。如果在服务器上建立了如上例类似的代码，则在运行该脚本后，客户端就能接收到其结果，但他们无法得知其背后的代码是如何运作的。

甚至可以将 web 服务器设置成让 PHP 来处理所有的 HTML 文件，这么一来，用户就无法得知服务端到底做了什么。

使用 PHP 的一大好处是它对于初学者来说极其简单，同时也给专业的程序员提供了各种高级的特性。当看到 PHP 长长的特性列表时，请不要害怕。可以很快的入门，只需几个小时就可以自己写一些简单的脚本。

尽管 PHP 的开发是以服务端脚本为目的，但事实上其功能远不局限与此。

PHP 能做什么?

PHP 能做任何事。PHP 主要是用于服务端的脚本程序，因此可以用 PHP 来完成任何其它的程序能够完成的工作，例如收集表单数据，生成动态网页，或者发送 / 接收 Cookies。但 PHP 的功能远不局限于此。

PHP 脚本主要用于以下三个领域：

- 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、web 服务器和 web 浏览器。需要在运行 web 服务器时，安装并配置 PHP，然后，可以用 web 浏览器来访问 PHP 程序的输出，即浏览服务端的 PHP 页面。
- 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。这种用法对于依赖 cron（Unix 或者 Linux 环境）或者 Task Scheduler（Windows 环境）的日常运行的脚本来说是理想的选择。这些脚本也可以用来处理简单的文本。

PHP 能够在所有的主流操作系统上使用，包括 Linux、Unix 的各种变种（包括 HP-UX、Solaris 和 OpenBSD）、Microsoft Windows、Mac OS X、RISC OS 等。

今天，PHP已经支持了大多数的 web 服务器，包括 Apache、Microsoft Internet Information Server（IIS）、Personal Web Server（PWS）、Netscape 以及 iPlant server、Oreilly Website Pro Server、Caudium、Xitami、OmniHTTPd 等。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。

使用 PHP，并不局限于输出 HTML。PHP 还能被用来动态输出图像、PDF 文件甚至 Flash 动画（使用 libswf 和 Ming）。还能够非常简便的输出文本，例如 XHTML 以及任何其它形式的 XML 文件。PHP 能够自动生成这些文件，在服务端开辟出一块动态内容的缓存，可以直接把它们打印出来，或者将它们存储到文件系统中。

PHP 最强大最显著的特性之一，是它支持很大范围的数据库。使用任何针对某数据库的扩展（例如 mysql）编写数据库支持的网页非常简单，或者使用抽象层如 PDO，或者通过 ODBC 扩展连接到任何支持 ODBC 标准的数据库。

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。还可以开放原始网络端口，使得任何其它的协议能够协同工作。PHP 支持和所有 web 开发语言之间的 WDDX 复杂数据交换。

第一个 PHP 页面

在 web 服务器根目录（DOCUMENT_ROOT）下建立一个文件名为 hello.php，然后完成如下内容：

示例 #1 第一个 PHP 脚本：hello.php

```
<html>
    <head>
        <title>PHP 测试</title>
</head>
<body>
<?php echo '<p>Hello World</p>'; ?>
</body>
</html>
```

在浏览器的地址栏里输入 web 服务器的 URL 访问这个文件，在结尾加上“/hello.php”。如果本地开发，那么这个 URL 一般是 `http://localhost/hello.php` 或者 `http://127.0.0.1/hello.php`，当然这取决于 web 服务器的设置。如果所有的设置都正确，那么这个文件将被 PHP 解析，浏览器中将会输出如下结果：

```
<html>
 <head>
  <title>PHP 测试</title>
 </head>
 <body>
 <p>Hello World</p>
 </body>
</html>
```

该程序非常的简单，它仅仅只是利用了 PHP 的 echo 语句显示了 `Hello World`。用户一定不会满足与此。请注意该文件*无需被执行*或以任何方式指定。服务器会找到该文件并提供给 PHP 进行解释，因为使用了“.php”的扩展名，服务器已被配置成自动传递有着“.php”扩展名的文件给 PHP。一个普通的 HTML 文件，加上了几个特别的标签，就可以做很多非常有趣的事情！

如果试过了这个例子，但是没有得到任何输出，或者浏览器弹出了下载框，或者浏览器以文本方式显示了源文件，可能的原因是服务器还没有支持 PHP，或者没有正确配置。

还要确认通过浏览器访问的 URL 确实指向了服务器上的这个文件。如果只是从本地文件系统调用这个文件，它不会被 PHP 解析。

以上例子的目的是为了显示 PHP 特殊标识符的格式。在这个例子中，用 `<?php` 来表示 PHP 标识符的起始，然后放入 PHP 语句并通过加上一个终止标识符 `?>` 来退出 PHP 模式。可以根据自己的需要在 HTML 文件中像这样开启或关闭 PHP 模式。

> **注意: 关于换行**
>
> 尽管换行在 HTML 中的实际意义不是很大，但适当地使用换行可以使 HTML 代码易读且美观。PHP 会在输出时自动删除其结束符 `?>` 后的一个换行。该功能主要是针对在一个页面中嵌入多段 PHP 代码或者包含了无实质性输出的 PHP 文件而设计，与此同时也造成了一些疑惑。如果需要在 PHP 结束符 `?>` 之后输出换行的话，可以在其后加一个空格，或者在最后的一个 echo/print 语句中加入一个换行。

现在已经成功建立了一个简单的 PHP 脚本，那么再来建立一个最著名的 PHP 脚本！调用函数 phpinfo()，将会看到很多有关自己系统的有用信息，例如预定义变量、已经加载的 PHP 模块和配置信息。请花一些时间来查看这些重要的信息。

示例 #2 从 PHP 获取系统信息

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

实用的脚本

现在来编写一些更实用的脚本，比如检查浏览页面的访问者在用什么浏览器。要达到这个目的，需要检查用户的 agent 字符串，它是浏览器发送的 HTTP 请求的一部分。该信息被存储在一个变量中。在 PHP 中，变量总是以一个美元符开头。我们现在感兴趣的变量是 [$_SERVER'HTTP_USER_AGENT']。

> **注意**:
>
> $_SERVER 是一个特殊的 PHP 保留变量，它包含了 web 服务器提供的所有信息，被称为超全局变量。请查阅本手册“超全局变量”中的有关内容以获取更多信息。这些特殊的变量是在 PHP » 4.1.0 版本引入的。在这之前使用 $HTTP*VARS 数组，如 $HTTP_SERVER_VARS。自 PHP 5.4.0 起，这些旧变量已经移除了。（参见“旧代码”一节中的注解）。

要显示该变量，只需简单地进行如下操作：

示例 #1 打印一个变量（数组元素）

```
<?php
echo $_SERVER['HTTP_USER_AGENT'];
?>
```

该脚本的输出可能是：

Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)

PHP 有很多种不同类型的变量。在以上例子中我们打印了一个数组的单元。数组是一类非常有用的变量。

$_SERVER 只是 PHP 自动全局化的变量之一。可以查阅“预定义变量”一节来查看这些变量的列表，或者也可以通过上节例子中 phpinfo() 函数的输出来查看。

示例 #2 流程控制与函数的使用

```
<?php
if (strpos($_SERVER['HTTP_USER_AGENT'], 'Firefox') !== FALSE) {
    echo '正在使用 Firefox<br />';
}
```

该脚本的输出可能是：

```
正在使用 正在使用 Firefox
```

这里要介绍一些新的原理。上面用了一个 if 语句。

需要介绍的第二个原理，是对 strpos() 函数的调用。strpos() 是 PHP 的一个内置函数，其功能是在一个字符串中搜索另外一个字符串。例如我们现在需要在 [$_SERVER'HTTP_USER_AGENT'] 变量中寻找 `'MSIE'`。如果在这个中该字符串被找到，则函数返回字符串相对于开头的位置；如果没有，则返回 **`false`**。

如果该函数没有返回 **`false`**，则 if 会将条件判断为 **`true`** 并运行其花括号 {} 内的代码；否则，则不运行这些代码。

可以自己尝试利用 if，else 以及其它的函数如 strtoupper() 和 strlen() 来建立类似的脚本。在本手册中相关的页面也包含有范例。如果对如何使用函数不是很确定，可以阅读手册中有关“如何阅读函数的定义”和“函数”的有关章节。

以下我们进一步显示如何进出 PHP 模式，甚至是在一个 PHP 代码块的中间：

示例 #3 混和 HTML 和 PHP 模式

```
<?php
if (strpos($_SERVER['HTTP_USER_AGENT'], 'MSIE') !== FALSE) {
?>
<h3>strpos() 肯定没有返回假 (FALSE)</h3>
<p>正在使用 Internet Explorer</p>
<?php
                                                    } else {
?>
<h3>strpos() 肯定返回假 (FALSE)</h3>
<center><b>没有使用 Internet Explorer</b></center>
<?php
                                                     }
?>
```

该脚本的输出可能是：

```
<h3>strpos() 肯定没有返回假 (FALSE)</h3>
<p>正在使用 Internet Explorer</p>
```

和以上我们用一个 PHP 的 echo 语句来输出不同的是，我们跳出了 PHP 模式来直接写 HTML 代码。这里很值得注意的一点是，对于这两种情况而言，脚本的逻辑效率是相同的。在判断了 strpos() 函数的返回值是 **`true`** 或是 **`false`**，也就是判断了字符串 `'MSIE'` 是否被找到之后，最终只有一个 HTML 块被发送给浏览者。

处理表单

PHP 一个很有用的特点体现在它处理 PHP 表单的方式。需要理解的非常重要的原理，是表单的任何元素都在 PHP 脚本中自动生效。请参阅本手册中“PHP 的外部变量”以获取关于在 PHP 中使用表单的详细信息及范例。以下是 HTML 表单的范例：

示例 #1 一个简单的 HTML 表单

```
<form action="action.php" method="post">
 <p>姓名：<input type="text" name="name" /></p>
 <p>年龄：<input type="text" name="age" /></p>
 <p><input type="submit" /></p>
</form>
```

该表单中并没有什么特殊的地方，其中没有使用任何特殊的标识符。当用户填写了该表单并点击了提交按钮，页面 action.php 将被调用。在该文件中，可以加入如下内容：

示例 #2 打印来自表单的数据

```
你好，<?php echo htmlspecialchars($_POST['name']); ?>。你 <?php echo (int)$_POST['age']; ?> 岁了。
```

该脚本的输出可能是：

```
你好，马哥。你 22 岁了。
```

除了htmlspecialchars() 部分，这段程序做什么用显而易见。

htmlspecialchars() 使得 HTML 之中的特殊字符被正确的编码，从而不会被使用者在页面注入 HTML 标签或者 Javascript 代码。

例如 age 字段，我们明确知道他是一个数值，因此我们将它转换为一个int来自动的消除任何不必要的字符。也可以使用 PHP 的 filter 扩展来自动完成该工作。

PHP 将自动设置 [$*POST*'name'] *和 [*$POST'age'] 变量。

在这之前我们使用了超全局变量 $_SERVER，现在我们引入了超全局变量 $_POST，它包含了所有的 POST 数据。请注意我们的表单提交数据的*方法*（method）。

如果使用了 *GET* 方法，那么表单中的信息将被储存到超全局变量 $_GET 中。

如果并不关心请求数据的来源，也可以用超全局变量 $_REQUEST，它包含了所有 GET、POST、COOKIE 和 FILE 的数据。