PHP主要用在服务器端做网站后台开发，有些功能用PHP来实现有点费劲或者无法实现，现在在学习[python](http://www.zongk.com/s/python/" \t "https://www.cnblogs.com/feng18/p/_blank)，同样是脚本语言，感觉python能做的事情PHP不一定能胜任。但是现在大部分的网站后台也是用PHP写的，虽然python可以作为WEB开发（flask，这个看了一段时间感觉既然网站用PHP开发好了，python可以做PHP做不好的事情，岂不是更好，逐暂缓），现在有在思考一个问题：有没有什么办法可以让PHP与python结合进来，PHP与python各做他们擅长的事情，毕竟工作中快速开发运行见效果还是占到了很大的需求比例。

　　记得PHP中有一个函数可以运行第三方的脚本：exec(),来看看exec的函数结构:

**string exec ( string $command [, array &$output [, int &$return\_var ]] )**

　　参数解释（来自PHP手册）

　　command

　　要执行的命令。

　　output

　　如果提供了 output 参数， 那么会用命令执行的输出填充此数组， 每行输出填充数组中的一个元素。 数组中的数据不包含行尾的空白字符，例如 \n 字符。 请注意，如果数组中已经包含了部分元素， exec() 函数会在数组末尾追加内容。如果你不想在数组末尾进行追加， 请在传入 exec() 函数之前 对数组使用 unset() 函数进行重置。

　　return\_var

　　如果同时提供 output 和 return\_var 参数， 命令执行后的返回状态会被写入到此变量。

　　返回值：

　　命令执行结果的最后一行内容。 如果你需要获取未经处理的全部输出数据， 请使用 passthru() 函数。

　　如果想要获取命令的输出内容， 请确保使用 output 参数。

**PHP执行外部程序延伸：**

**1、system — 执行外部程序，并且显示输出**

　　说明：

　　string system ( string $command [, int &$return\_var ] )

　　同 C 版本的 system() 函数一样， 本函数执行 command 参数所指定的命令， 并且输出执行结果。

　　如果 PHP 运行在服务器模块中， system() 函数还会尝试在每行输出完毕之后， 自动刷新 web 服务器的输出缓存。

　　如果要获取一个命令未经任何处理的 原始输出， 请使用 passthru() 函数。

　　参数：

　　command

　　要执行的命令。

　　return\_var

　　如果提供 return\_var 参数， 则外部命令执行后的返回状态将会被设置到此变量中。

　　返回值：

　　成功则返回命令输出的最后一行， 失败则返回 FALSE

**2、passthru — 执行外部程序并且显示原始输出**

　　void passthru ( string $command [, int &$return\_var ] )

　　同 exec() 函数类似， passthru() 函数 也是用来执行外部命令（command）的。 当所执行的 Unix 命令输出二进制数据， 并且需要直接传送到浏览器的时候， 需要用此函数来替代 exec() 或 system() 函数。 常用来执行诸如 pbmplus 之类的可以直接输出图像流的命令。 通过设置 Content-type 为 image/gif， 然后调用 pbmplus 程序输出 gif 文件， 就可以从 PHP 脚本中直接输出图像到浏览器。

　　参数：

　　command

　　要执行的命令。

　　return\_var

　　如果提供 return\_var 参数， Unix 命令的返回状态会被记录到此参数。

　　返回值：

　　没有返回值。

　　这次使用的执行外部程序的函数为exec函数：

**PHP测试代码：**

header("Content-type:text/html;charset=utf-8");

　　// 测试PHP执行python代码

　　$a = 5;

　　$b = 8;

　　$c = 'Davidszhou的PHP操作带参数的python脚本并返回结果';

　　$d = urlencode($c);

　　unset($out);

　　$c = exec("C:\python35\python plug/index.py {$a} {$b} {$d}",$out,$res);

　　print\_r(urldecode($out[0]));

　　echo "<br>";

　　echo '外部程序运行是否成功:'.$res."(0代表成功,1代表失败)";

**python测试代码：**

[IMG_258](https://www.cnblogs.com/feng18/p/javascript:void(0);)

#-\*- coding:utf-8 -\*-import sysdef ceshi(a,b,c):

d = int(a)+int(b)

r = int(d)+c

return rif \_\_name\_\_ == "\_\_main\_\_":

res = ceshi(a=sys.argv[1],b=sys.argv[2],c = sys.argv[3])

print(res)

[IMG_259](https://www.cnblogs.com/feng18/p/javascript:void(0);)

　解释：

　　1、exec问题：

　　exec("C:\python35\python plug/index.py {$a} {$b} {$d}",$out,$res)："C:\python35\python"：这是本地环境中python安装的绝对环境，‘plug/index.py’：这个是python脚本的目录，后面{$a} {$b} {$d}是紧随的参数注意它与python脚本目录之间以及每个参数之间的空格，

　　2、python如何接收PHP给他传的参数

　　python接收PHP传参数主要通过sys这个模块：

　　解释（来自python手册）：

　　sys.argv

　　传递给Python脚本的命令行参数列表。argv[0]是脚本的名称（是否是完整的路径名这要取决于操作系统）。如果使用解释器的-c命令行选项执行命令，则argv[0]设置为字符串'-c'如果无脚本名称传递给 Python 解释器， argv[0]是空字符串。

　　若要遍历标准输入或命令行上给出的文件列表，请参阅fileinput模块。

分类: [python](https://www.cnblogs.com/feng18/category/837354.html" \t "https://www.cnblogs.com/feng18/p/_blank)