|  |
| --- |
| 初识模块 |
| **作者：李昂**  **归档：python之路day1**  **日期：2018/2/1** |
| **快捷键：**  Ctrl + 1 标题1  Ctrl + 2 标题2  Ctrl + 3 标题3  Ctrl + 4 实例  Ctrl + 5 程序代码  Ctrl + 6 正文  Ctrl + q 正文注释 |
| **格式说明：**  蓝色字体：注释  黄色背景：了解  绿色背景：重要  红色背景：特别重要 |

目 录

[初识模块 I](#_Toc505284835)

[第1章 模块介绍 1-3](#_Toc505284836)

[第2章 标准库-getpass 2-3](#_Toc505284837)

[第3章 标准库-sys模块 3-4](#_Toc505284838)

[3.1 sys.path ---当前系统的环境变量 3-5](#_Toc505284839)

[3.2 sys.argv—打印相对路径 3-5](#_Toc505284840)

[3.2.1 提取传入的参数 3-6](#_Toc505284841)

[第4章 标准库-os模块 4-6](#_Toc505284842)

[4.1 os.system—调用系统命令，只是打印结果，并不保存结果，保存的是状态码 import os print(os.system("dir")) #打印当前目录 4-7](#_Toc505284843)

[4.2 os.popen – 4-7](#_Toc505284844)

# 模块介绍

模块是写好的一堆程序功能的集合。也成为库

库分两种：

标准库： 常用的直接导入就可以用的库。

第三方库：需要下载安装的库

# 标准库-getpass

密码加密：

*#Author：LiAng*

*#encoding=utf8*

username = input(**"Username:"**)

password = input(**"Password:"**)

print(username,password)

1235 
123s 

 此时输入的密码是明文的，可以导入模块将输入变成密文。

引入getpass 模块

*#Author：LiAng*

*#encoding=utf8*

**import** getpass #导入getpass 模块

username = input(**"Username:"**)

password = getpass.getpass(**"Password:"**)

print(username,password)

计算机生成了可选文字:
\Users\Admini strator 〉 python E ： \python\python 之 路 \dayl \ ． i dea\passwd\passwd. PY 
sername 。 yyttt 
yyttt 12345 
\Users\Administrator 〉 

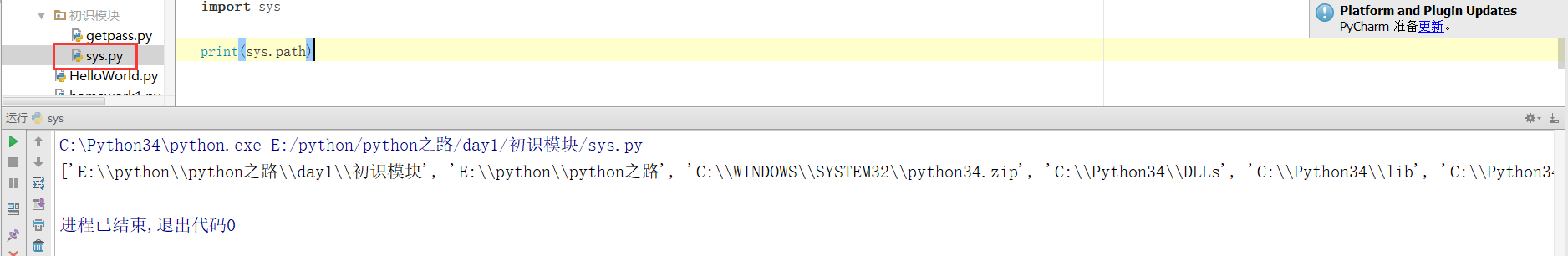
# 标准库-sys模块

sys 是一个模块，而他的功能通过sys. 调用。

例如：

sys.path 调用sys模块的path功能,也称为path方法。

## sys.path ---当前系统的环境变量



当导入一个模块时，python首先会在当前路径下搜索，然后才会去系统中搜索。--可以理解为linux的环境变量

所以文件名不要和模块名冲突。

import 导入时会按照顺序去以下路径寻找。

['E:\\python\\python之路\\day1\\初识模块',

'E:\\python\\python之路',

'C:\\WINDOWS\\SYSTEM32\\python34.zip',

'C:\\Python34\\DLLs',

'C:\\Python34\\lib',

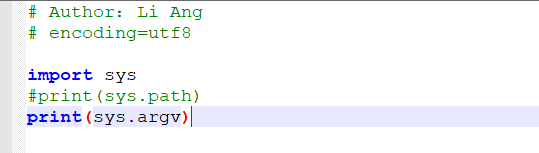
'C:\\Python34',

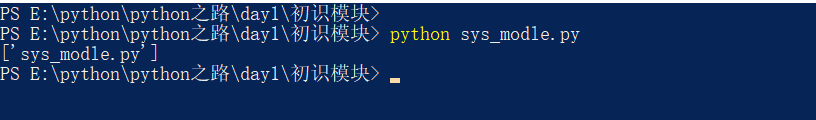
'C:\\Python34\\lib\\site-packages']

pytho安装的第三方库一般会存放在Python34\\lib\\site-packages 下。

标准库则放在第三方库的上级目录中

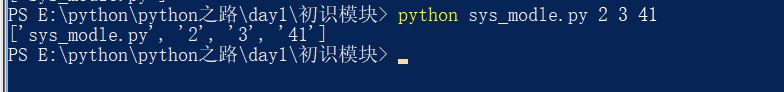
## sys.argv—打印相对路径





sys.argv 打印的是脚本的相对路径即脚本名称，

也可以通过argv向脚本中传入参数



### 提取传入的参数

# Author：Li Ang

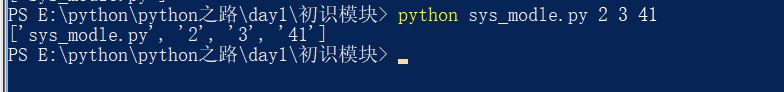
# encoding=utf8

import sys

#print(sys.path)

print(sys.argv)

print(sys.argv[2])

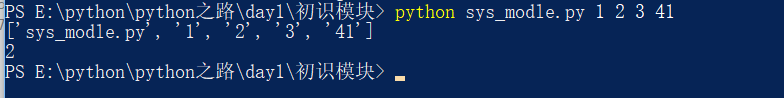


如图：我们将

['sys\_modle.py', '1', '2', '3', '41'] 这样的一组数据称为列表，就像shell的数组一样，逗号分隔每一个元素，从0开始进行标识，

提取列表中的数据：

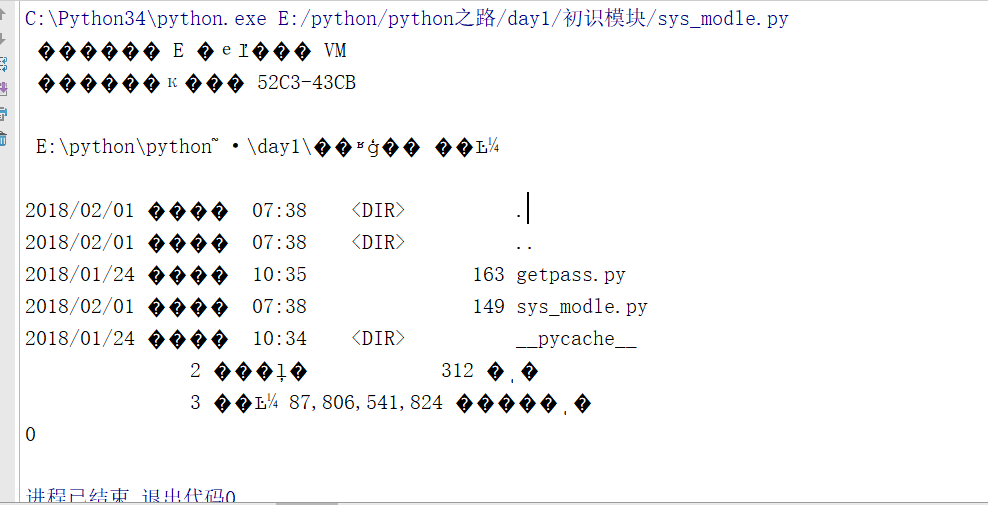
print(sys.argv[2])



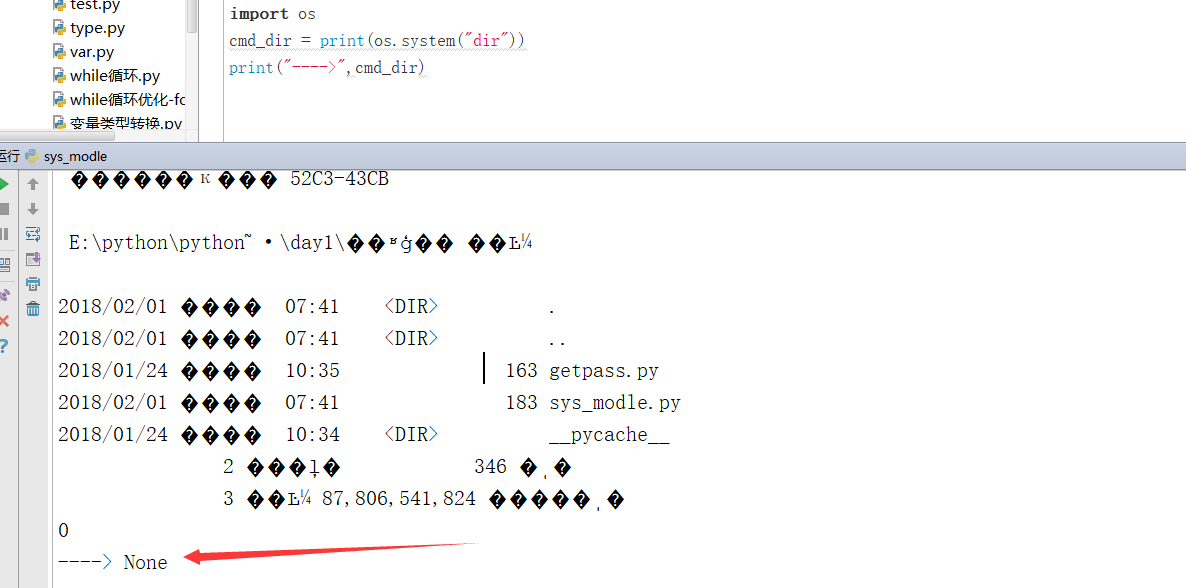
# 标准库-os模块

os模块就是调用相关的系统命令

## os.system—调用系统命令，只是打印结果，并不保存结果，保存的是状态码 import os print(os.system("dir")) #打印当前目录

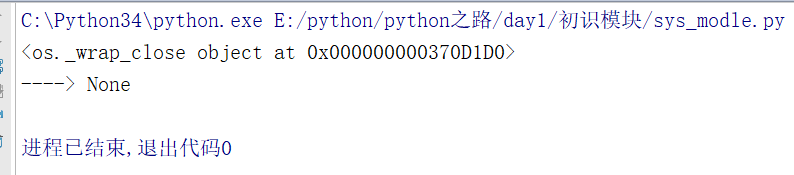


执行命令并不保存结果，其结果是放回的是“状态码”，打印的内容并不能被引用。



## os.popen –

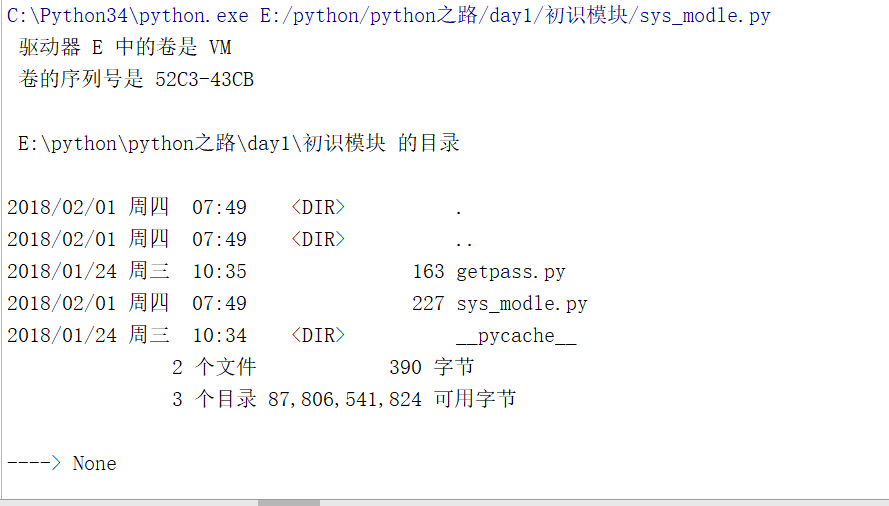
**import** os  
  
*#cmd\_dir = print(os.system("dir"))*cmd\_dir **=** print(os.popen("dir"))  
print("---->",cmd\_dir)



#此时返回的只是这个对象在内存中的地址，

需要读取这个地址

**import** os  
  
*#cmd\_dir = print(os.system("dir"))*cmd\_dir **=** print(os.popen("dir").read()) #.read 即可读取，这个内存地址中的数据  
print("---->",cmd\_dir)



## os.mkdir--- 创建目录

os.mkdir("new\_dir") #当前目录下创建一个新目录

# 写一个模块

将写好.py 文件放到sys.path 中，然后在程序中直接导入即可。

*# Author：Li Ang  
# encoding=utf8***import** sys  
print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")  
print(sys.path)  
**import** password

C:\Python34\python.exe E:/python/python之路/day1/login.py

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

['E:\\python\\python之路\\day1', 'E:\\python\\python之路', 'C:\\WINDOWS\\SYSTEM32\\python34.zip', 'C:\\Python34\\DLLs', 'C:\\Python34\\lib', 'C:\\Python34', 'C:\\Python34\\lib\\site-packages']

Username:li

Password:123

Wellcome user li login...

进程已结束,退出代码0

#此时是可以运行的

现在把password.py 移动走，然后运行login.py

C:\Python34\python.exe E:/python/python之路/day1/login.py

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

['E:\\python\\python之路\\day1', 'E:\\python\\python之路', 'C:\\WINDOWS\\SYSTEM32\\python34.zip', 'C:\\Python34\\DLLs', 'C:\\Python34\\lib', 'C:\\Python34', 'C:\\Python34\\lib\\site-packages']

Traceback (most recent call last):

File "E:/python/python之路/day1/login.py", line 7, in <module>

import password

ImportError: No module named 'password' #找不到password 模块

进程已结束,退出代码1

因为此时 password.py 已经不在sys.path 中了，

1. 修改sys.path 将password.py 的路径加入进去
2. 移动passwd.py 到sys.path 路径下