目录

[其他 1](#_Toc532922379)

[简单http服务 1](#_Toc532922380)

[基础 2](#_Toc532922381)

[总结 2](#_Toc532922382)

[使用 2](#_Toc532922383)

[print 2](#_Toc532922384)

[指定解释器 2](#_Toc532922385)

[对比shell 3](#_Toc532922386)

[sys 3](#_Toc532922387)

[argv 3](#_Toc532922388)

[执行shell命令 3](#_Toc532922389)

[变量 5](#_Toc532922390)

[字符串 5](#_Toc532922391)

[列表 5](#_Toc532922392)

[python3 5](#_Toc532922393)

[print 5](#_Toc532922394)

# 其他

## 简单http服务

python -m SimpleHTTPServer 8000 （端口8000可以自己更改）(这是python的2.x版本的写法)

python -m http.server 8000（这是python的3.x版本的写法 ）

# 基础

## 总结

以单下划线开头 \_foo 的代表不能直接访问的类属性，需通过类提供的接口进行访问，不能用 from xxx import \* 而导入；

以双下划线开头的 \_\_foo 代表类的私有成员；

以双下划线开头和结尾的 \_\_foo\_\_ 代表 Python 里特殊方法专用的标识，如 \_\_init\_\_() 代表类的构造函数。

## 使用

1 chmod +x test.py # 脚本文件添加可执行权限

$ ./test.py

2 python test.py

### print

默认输出是换行的，如果要实现不换行需要在变量末尾加上逗号，这样输出会以空格隔开。

### 指定解释器

直接执行文件时（./test.py）才有用

#!/usr/bin/python 相当于写死了python路径;

#!/usr/bin/env python 会去环境设置寻找 python 目录,推荐这种写法

## 对比shell

## sys

### argv

用来获取命令行参数, sys.argv[0] 代表文件本身路径，所带参数从 sys.argv[1] 开始。

./test.py hello argv[0]为'./test.py'

python test.py argv[0]为'test.py'

#### shell中

$0表示文件名，$1开始表示参数。

## 执行shell命令

Python目前已经废弃了os.system，os.spawn\*，os.popen\*，popen2.\*，commands.\*来执行其他语言的命令，subprocesss是被推荐的方法；

subprocess允许你能创建很多子进程，创建的时候能指定子进程和子进程的输入、输出、错误输出管道，执行后能获取输出结果和执行状态。

# coding=utf-8

'''

Created on 2013年11月22日

@author: crazyant.net

'''

import shlex

import datetime

import subprocess

import time

def execute\_command(cmdstring, cwd=None, timeout=None, shell=False):

"""执行一个SHELL命令

封装了subprocess的Popen方法, 支持超时判断，支持读取stdout和stderr

参数:

cwd: 运行命令时更改路径，如果被设定，子进程会直接先更改当前路径到cwd

timeout: 超时时间，秒，支持小数，精度0.1秒

shell: 是否通过shell运行

Returns: return\_code

Raises: Exception: 执行超时

"""

if shell:

cmdstring\_list = cmdstring

else:

cmdstring\_list = shlex.split(cmdstring)

if timeout:

end\_time = datetime.datetime.now() + datetime.timedelta(seconds=timeout)

#没有指定标准输出和错误输出的管道，因此会打印到屏幕上；

sub = subprocess.Popen(cmdstring\_list, cwd=cwd, stdin=subprocess.PIPE,shell=shell,bufsize=4096)

#subprocess.poll()方法：检查子进程是否结束了，如果结束了，设定并返回码，放在subprocess.returncode变量中

while sub.poll() is None:

time.sleep(0.1)

if timeout:

if end\_time <= datetime.datetime.now():

raise Exception("Timeout：%s"%cmdstring)

return str(sub.returncode)

if \_\_name\_\_=="\_\_main\_\_":

print execute\_command("ls")

# 变量

一旦赋值数据类型就不会改变，重新赋值其他类型系统会分配一个新的变量。

不需要申明

## 字符串

截取字符串s[1:5] (取1-4)

print str[2:] # 输出从第三个字符开始的字符串

print str \* 2 # 输出字符串两次

print str + "TEST" # 输出连接的字符串

## 列表

支持[:], +, \*

# python3

## print

原来的print "xx"会报错，写成print("xx")

不需要引入编码就可以输出中文。