

[首页](#) [笔记首页](#) [ANDROID](#) [ES6 教程](#) [排序算法](#) [程序员人生](#) [程序员笑话](#) [编程技术](#) [登录](#)

Python3 subprocess

分类 **编程技术**

subprocess 模块允许我们启动一个新进程，并连接到它们的输入/输出/错误管道，从而获取返回值。

使用 subprocess 模块

subprocess 模块首先推荐使用使用的是它的 run 方法，更高级的用法可以直接使用 Popen 接口。

run 方法语法格式如下：

```
subprocess.run(args, *, stdin=None, input=None, stdout=None, stderr=None, capture_output=False, shell=False, cwd=None, timeout=None, check=False, encoding=None, errors=None, text=None, env=None, universal_newlines=None)
```

args：表示要执行的命令。必须是一个字符串，字符串参数列表。

stdin、stdout 和 stderr：子进程的标准输入、输出和错误。其值可以是 subprocess.PIPE、subprocess.DEVNULL、一个已经存在的文件描述符、已经打开的文件对象或者 None。subprocess.PIPE 表示为子进程创建新的管道。subprocess.DEVNULL 表示使用 os.devnull。默认使用的是 None，表示什么都不做。另外，stderr 可以合并到 stdout 里一起输出。

timeout：设置命令超时时间。如果命令执行时间超时，子进程将被杀死，并弹出 TimeoutExpired 异常。

check：如果该参数设置为 True，并且进程退出状态码不是 0，则弹出 CalledProcessError 异常。

encoding：如果指定了该参数，则 stdin、stdout 和 stderr 可以接收字符串数据，并以该编码方式编码。否则只接收 bytes 类型的数据。

shell：如果该参数为 True，将通过操作系统的 shell 执行指定的命令。

run 方法调用方式返回 CompletedProcess 实例，和直接 Popen 差不多，实现是一样的，实际也是调用 Popen，与 Popen 构造函数大致相同，例如：

教程列表

ADO 教程 Ajax 教程 Android 教
Angular2 AngularJS AppML 教
ASP 教程 ASP.NET Bootstrap
Bootstrap4 C 教程 C# 教程
C++ 教程 CSS 参考 CSS 教程
CSS3 教程 Django 教 Docker 教
DTD 教程 ECharts 教 Eclipse 教
Firebug 教 Font Foundation
Git 教程 Go 语言教 Google 地
Highcharts HTML HTML 参考
HTML 字符 HTML 教程 HTTP 教程
ionic 教程 iOS 教程 Java 教程
JavaScript Javascript jQuery
jQuery jQuery UI jQuery 教
JSON 教程 JSP 教程 Kotlin 教程
Linux 教程 Lua 教程 Markdown
Maven 教 Memcached MongoDB
MySQL 教 Node.js 教 NumPy 教
Perl 教程 PHP 教程 PostgreSQL
Python 3 Python 基 R 教程
RDF 教程 React 教程 Redis 教程
RSS 教程 Ruby 教程 Rust 教程
Sass 教程 Scala 教程 Servlet 教
SOAP 教程 SQL 教程 SQLite 教
SVG 教程 SVN 教程 Swift 教程
TCP/IP 教 TypeScript VBScript
Vue.js 教程 W3C 教程 Web
WSDL 教 XLink 教程 XML DC



实例

```
#执行ls -l /dev/null 命令
>>> subprocess.run(["ls", "-l", "/dev/null"])
crw-rw-rw- 1 root wheel  3,  2  5  4 13:34 /dev/null
CompletedProcess(args=['ls', '-l', '/dev/null'], returncode=0)
```

returncode: 执行完子进程状态，通常返回状态为0则表明它已经运行完毕，若值为负值 "-N",表明子进程被终。

简单实例：

实例

```
import subprocess
def runcmd(command):
    ret = subprocess.run(command,shell=True,stdout=subprocess.PIPE,stderr=subprocess.PIPE,encoding="utf-8",timeout=1)
    if ret.returncode == 0:
        print("success:",ret)
    else:
        print("error:",ret)

runcmd(["dir", "/b"])#序列参数
runcmd("exit 1")#字符串参数
```

输出结果如下：

```
success: CompletedProcess(args=['dir', '/b'], returncode=0, stdout='test.py\n', stderr='')
error: CompletedProcess(args='exit 1', returncode=1, stdout='', stderr='')
```

Popen() 方法

Popen 是 subprocess的核心，子进程的创建和管理都靠它处理。

构造函数：

```
class subprocess.Popen(args, bufsize=-1, executable=None, stdin=None, stdout=None, stderr=None, preexec_fn=None, close_fds=True, shell=False, cwd=None, env=None, universal_newlines=False, startupinfo=None, creationflags=0, restore_signals=True, start_new_session=False, pass_fds=(), *, encoding=None, errors=None)
```

常用参数：

args: shell命令，可以是字符串或者序列类型（如：list，元组）

[XML](#) [XML 教程](#) [XPath 教程](#)

[XQuery 教](#) [XSLFO 教](#) [XSLT 教程](#)

[正则表达式](#) [测验](#) [浏览器](#)

[网站品质](#) [网站建设指](#) [网站服务器](#)

[设计模式](#)



bufsize: 缓冲区大小。当创建标准流的管道对象时使用, 默认-1。

0: 不使用缓冲区

1: 表示行缓冲, 仅当universal_newlines=True时可用, 也就是文本模式

正数: 表示缓冲区大小

负数: 表示使用系统默认的缓冲区大小。

stdin, stdout, stderr: 分别表示程序的标准输入、输出、错误句柄

preexec_fn: 只在 Unix 平台下有效, 用于指定一个可执行对象 (callable object) , 它将在子进程运行之前被调用

shell: 如果该参数为 True, 将通过操作系统的 shell 执行指定的命令。

cwd: 用于设置子进程的当前目录。

env: 用于指定子进程的环境变量。如果 env = None, 子进程的环境变量将从父进程中继承。

创建一个子进程, 然后执行一个简单的命令:

实例

```
>>> import subprocess
>>> p = subprocess.Popen('ls -l', shell=True)
>>> total 164
-rw-r--r--  1 root root   133 Jul  4 16:25 admin-openrc.sh
-rw-r--r--  1 root root   268 Jul 10 15:55 admin-openrc-v3.s
h
...
>>> p.returncode
>>> p.wait()
0
>>> p.returncode
```

这里也可以使用 `p = subprocess.Popen(['ls', '-cl'])` 来创建子进程。

Popen 对象方法

poll(): 检查进程是否终止, 如果终止返回 returncode, 否则返回 None。

wait(timeout): 等待子进程终止。

communicate(input,timeout): 和子进程交互, 发送和读取数据。

send_signal(signal): 发送信号到子进程。

terminate(): 停止子进程, 也就是发送SIGTERM信号到子进程。

kill(): 杀死子进程。发送 SIGKILL 信号到子进程。

实例



```
import time
import subprocess

def cmd(command):
    subp = subprocess.Popen(command, shell=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE, encoding="utf-8")
    subp.wait(2)
    if subp.poll() == 0:
        print(subp.communicate()[1])
    else:
        print("失败")

cmd("java -version")
cmd("exit 1")
```

输出结果如下:

```
java version "1.8.0_31"
Java(TM) SE Runtime Environment (build 1.8.0_31-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.31-b07, mixed mode)
```

失败

参考文章:

https://blog.csdn.net/weixin_42547344/article/details/80894760

<https://blog.csdn.net/maosijunzi/article/details/80138458>

← localStorage 必知必会

JavaScript window.onload →

 点我分享笔记

在线实例

- HTML 实例
- CSS 实例
- JavaScript 实例
- Ajax 实例
- jQuery 实例
- XML 实例
- Java 实例

字符集&工具

- HTML 字符集设置
- HTML ASCII 字符集
- HTML ISO-8859-1
- HTML 实体符号

最新更新

- R 数据重塑
- Redis GEO
- Java 中操作 R
- R 绘图 - ...
- R 绘图 - ...
- R 绘图 - ...
- R 绘图 - ...

站点信息

- 意见反馈
- 免责声明
- 关于我们
- 文章归档



- [HTML 拾色器](#)
- [JSON 格式化工具](#)

关注微信



Copyright © 2013-2020 菜鸟教程
runoob.com All Rights Reserved.
备案号: 闽ICP备15012807号-1

