



Scripts administración

El módulo subprocess

Índice

- Fundamentos
- Ejecución de procesos
- Salida estándar
- El objeto Popen

Fundamentos

- A través de este módulo es posible lanzar procesos externos y comunicarse con ellos
- Muy utilizado en scripts de administración en los que se necesita ejecutar comandos del sistema operativo.
- Permite controlar un proceso durante su ejecución.

```
import subprocess
```


Ejecución de procesos

- El módulo subprocess proporciona los métodos run(versión 3.5 en adelante) y call(versiones anteriores a 3.5), para ejecutar procesos del sistema o programas externos.
- El comando y los argumentos se pasan como una lista de valores:

```
p = subprocess.run(["python", "--version"])
```

- Devuelve un objeto CompletedProcess, cuyo atributo returncode contiene el código de respuesta.
- Si es un comando del sistema operativo, se debe incluir el parámetro shell=True:

```
subprocess.run(["cls"], shell=True) #limpia la pantalla
```


Salida estándar

- Para recoger la salida de un proceso en la salida estándar de python, debemos incluir el argumento `stdout=subprocess.PIPE`
- Para mostrar la lista de subdirectorios del directorio actual:

```
p=subprocess.run(["dir"], stdout=subprocess.PIPE, shell=True)
print(p.stdout)
```

- En caso de error, se puede capturar también:

```
p=subprocess.run(["dir"], stdout=subprocess.PIPE, stderr=subprocess.PIPE, shell=True)
print(p.stdout)
print(p.stderr)
```


Objeto Popen

- Al igual que el método call/run, Popen es un objeto del módulo subprocess que se emplea para ejecutar procesos:

```
from subprocess import Popen
proc=Popen(["dir"], stdout=subprocess.PIPE, stderr=subprocess.PIPE, shell=True)
```

- A diferencia de los métodos anteriores, Popen ofrece un mayor control sobre el proceso, ya que proporciona los siguientes métodos y atributos:

- pid. Atributo que contiene el pid del proceso
- terminate(). Método que realiza la terminación de un proceso
- kill(). Permite destruir el proceso
- communicate(). Se comunica con el proceso para enviarle datos (stdin) o recibir la salida del mismo (stdout)