

★ Python 3 para impacientes ★



"Simple es mejor que complejo" (Tim Peters)

Python	IPython	EasyGUI	Tkinter	JupyterLab	Numpy
--------	---------	---------	---------	------------	-------

miércoles, 12 de febrero de 2014

Ejecutar un comando externo

A continuación, se muestran varias posibilidades para ejecutar comandos externos desde un programa Python:

```
#!/usr/bin/env Python
# -*- coding: utf-8 -*-

import os, sys
import subprocess

# Ejecutar un comando con "os.system(comando)" y mostrar en
# pantalla la salida del comando y el resultado de la
# ejecución.
# Si su valor es 0 la ejecución finalizó con éxito.

valor1 = os.system("whoami")
print("Resultado:", valor1)

# Ejecutar un comando "erróneo" con "os.system(comando)"

valor2 = os.system("whoamix")
print("Resultado:", valor2)

# Ejecutar varios comandos con "os.system(comando)"

comando3 = "ls -l | grep 'txt' >textos.txt"
valor3 = os.system(comando3)
print("Resultado:", valor3)

# Ejecutar un comando con "os.popen(comando)" y capturar su
# salida

comando4 = "uname -srmo"
salida4 = os.popen(comando4).read()
print("Salida:\n", salida4)

# A continuación, se muestran varios ejemplos con el módulo
# "subprocess" que genera un nuevo proceso para ejecutar el
# comando permitiendo controlar su ejecución y obtener su
# salida y/o errores que pudieran darse.

# Ejecutar un comando externo con "subprocess.call", mostrar
# su salida y el resultado de su ejecución.

valor5 = subprocess.call(["ping", "-c 6", "www.elpais.es"])
print("Este texto se mostrará después de ejecutar el comando")
print("Resultado:", valor5)

# Ejecutar un comando externo en paralelo con
# "subprocess.Popen" con wait() se espera a que termine
# la ejecución del subprocesso
# Todas las líneas de código delante de wait() se
# ejecutarán en paralelo a la ejecución del comando.
# También, se mostrará si la ejecución terminó o no
# con éxito

proceso6 = subprocess.Popen(['ls', '-lha'])
for num in range(100):
    print ("Este texto se mostrará antes")

valor6 = proceso6.wait()
print ("Este texto se mostrará después")
print ("Resultado:", valor6)
```

Buscar

Python para impacientes

[Python](#)
[IPython](#)
[EasyGUI](#)
[Tkinter](#)
[JupyterLab](#)
[Numpy](#)

Anexos

[Guía urgente de MySQL](#)
[Guía rápida de SQLite3](#)

Entradas + populares

[Dar color a las salidas en la consola](#)

En Python para dar color a las salidas en la consola (o en la terminal de texto) existen varias posibilidades. Hay un método basado ...

[Instalación de Python, paso a paso](#)

Instalación de Python 3.6 A finales de 2016 se produjo el lanzamiento de Python 3.6. El propósito de esta entrada es mostrar, pas...

[Añadir, consultar, modificar y suprimir elementos en Numpy](#)

Acceder a los elementos de un array. [], [..], ... Acceder a un elemento de un array. Para acceder a un elemento se utiliz...

[Variables de control en Tkinter](#)

Variables de control Las variables de control son objetos especiales que se asocian a los widgets para almacenar sus valore...

[Cálculo con arrays Numpy](#)

Numpy ofrece todo lo necesario para obtener un buen rendimiento cuando se trata de hacer cálculos con arrays. Por como está concebido...

[Tkinter: interfaces gráficas en Python](#)

Introducción Con Python hay muchas posibilidades para programar una interfaz gráfica de usuario (GUI) pero Tkinter es fácil d...

[Operaciones con fechas y horas. Calendarios](#)

Los módulos datetime y calendar amplían las posibilidades del módulo time que provee funciones para manipular expresiones de ti...

[Convertir, copiar, ordenar, unir y dividir arrays Numpy](#)

Esta entrada trata sobre algunos métodos que se utilizan en Numpy para convertir listas en arrays y viceversa; para copiar arrays d...

[Tkinter: Tipos de ventanas](#)

Ventanas de aplicación y de diálogos En la entrada anterior tratamos los distintos gestores de geometría que se utilizan para di...

[Threading: programación con hilos \(I\)](#)

En programación, la técnica que permite que una aplicación ejecute

```

# Ejecutar un comando externo en paralelo con
# "subprocess.Popen"

proceso7 = subprocess.Popen(['ls', '-lha'])
for num in range(300):
    print ("Este texto se mostrará en paralelo")

# Ejecutar un comando en paralelo con
# "subprocess.Popen" y capturar su salida y
# los errores que pudieran darse

proceso8 = subprocess.Popen(['ls', '-a'],
                             stdout=subprocess.PIPE,
                             stderr=subprocess.PIPE)
errores8 = proceso8.stderr.read()
salida8 = proceso8.stdout.read()
proceso8.stderr.close()
proceso8.stdout.close()
print("Salida de errores:\n")
errores8 = errores8.decode(sys.getdefaultencoding())
print(errores8)
salida8 = salida8.decode(sys.getdefaultencoding())
print("Salida del comando:\n")
print(salida8)

# Ejecutar un comando externo con "subprocess.Popen" y
# obtener su salida y los errores que puedan darse (Otra
# manera de hacerlo)
# En el ejemplo se provoca un error y se muestra el texto
# de su salida

proceso9 = subprocess.Popen(["ping",
                             "-zz 4",
                             "www.elpais.es"],
                             stdout=subprocess.PIPE)
salida9, errores9 = proceso9.communicate()
print(salida9.decode(sys.getdefaultencoding()))
if errores9 != None:
    print(errores9.decode(sys.getdefaultencoding()))

# Para resolver variables de entorno, patrones globales y
# otras características de la shell hay que incluir el
# parámetro "shell = True"

print("Correcto : subprocess.call('echo $HOME', shell=True)")
print("Incorrecto: subprocess.call('echo $HOME')")
# Es equivalente a shell = False

subprocess.call('echo $HOME', shell=True)

# Ejecutar un comando externo con subprocess.Popen() y
# obtener su salida en tiempo real

comando11 = "ping -c3 www.elpais.es"
resultado11 = subprocess.Popen(comando11,
                               shell=True,
                               stdout=subprocess.PIPE)
for salida11 in resultado11.stdout:
    print(salida11.decode(sys.getdefaultencoding()).rstrip())

# Ejecutar un comando externo con subprocess.Popen()
# y obtener la salida en tiempo real (Otra forma de
# hacerlo)

comando12 = "ping -c2 www.elpais.es"
resultado12 = subprocess.Popen(comando12,
                               shell=True,
                               stdout=subprocess.PIPE)
while resultado12.poll() is None:
    salida12 = resultado12.stdout.readline()
    print(salida12.decode(sys.getdefaultencoding()).rstrip())

```

simultáneamente varias operaciones en el mismo espacio de proceso se...

Archivo

febrero 2014 (17) ▾

python.org



pypi.org



Sitios

- [ActivePython](#)
- [Anaconda](#)
- [Bpython](#)
- [Django](#)
- [Flask](#)
- [Ipython](#)
- [IronPython](#)
- [Matplotlib](#)
- [MicroPython](#)
- [Numpy](#)
- [Pandas](#)
- [Pillow](#)
- [PortablePython](#)
- [PyBrain](#)
- [PyCharm](#)
- [PyDev](#)
- [PyGame](#)
- [Pypi](#)
- [PyPy](#)
- [Pyramid](#)
- [Python.org](#)
- [PyTorch](#)
- [SciPy.org](#)
- [Spyder](#)
- [Tensorflow](#)
- [TurboGears](#)

[Ir al índice del tutorial de Python](#)

Publicado por Pherkad en [10:09](#)



Etiquetas: [Python3](#)

[Entrada más reciente](#)

[Inicio](#)

[Entrada antigua](#)

2014-2020 | Alejandro Suárez Lamadrid y Antonio Suárez Jiménez, Andalucía - España
. Tema Sencillo. Con la tecnología de [Blogger](#).