

8. Ejecutando PS desde Python

Monday, November 9, 2020 11:12 AM

Para esto se necesita el módulo subprocess.

Ejemplo

```
import subprocess

comando = "Get-Process"
lineaPS = "powershell -Executionpolicy Bypass -Command "+ comando
runningProcesses = subprocess.check_output(lineaPS)
print(runningProcesses.decode())
```

Salida

Handles	NPM(K)	PM(K)	WS(K)	CPU(s)	Id	SI	ProcessName
1064	54	44956	57080	17.13	4172	2	ApplicationFrameHost
254	9	1816	6816	0.23	2476	2	brave
1791	64	179140	136124	2,497.70	5932	2	brave
465	23	21832	38412	141.19	8364	2	brave
302	22	67300	38792	67.17	8556	2	brave
303	24	188224	51948	31.09	9768	2	brave
279	18	8180	18328	24.33	10212	2	brave
274	16	21724	30112	4.33	10688	2	brave
242	16	25132	31304	1.56	10848	2	brave
224	15	25140	37084	2.27	10864	2	brave
300	18	37628	34952	3.56	11212	2	brave



ejemplo1...

Explicación

- El resultado del comando será almacenado en una variable, en este caso, **runningProcesses**. Esta es una variable de tipo bytes, por lo que requiere ser decodificada para convertirse en un str.
- La librería **subprocess** tiene el método **check_output** el cual toma el parámetro que recibe como la línea de comando que tu quieres ejecutar. Si ejecutamos el contenido de ese parámetro como un comando directamente en PowerShell (o en CMD) nos devolverá el mismo resultado que lo que obtuvimos de resultado en Python. La explicación de como se compone la línea enviada como parámetro es:
 - **powershell** - Es el proceso (intérprete) a ejecutar
 - **-Executionpolicy Bypass** - PowerShell no ejecutará scripts o CmdLets sin permiso explícito, este parámetro especifica que puede ejecutar el

comando sin bloquear nada y sin lanzar advertencias o desplegados en una terminal aparte.

- **-Command** - Nos permite especificar que lo que sigue es un comando de PowerShell. Si necesitamos ejecutar un script deberemos cambiarlo por **-File** y escribir un nombre válido de script de PowerShell.
- **Get-Process** - Es el nombre del CmdLet que se ejecutará, en este caso, sin parámetros.