Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

Administración de sistemas con Python (8409 lectures)

Per **Antoni Aloy López**, ***aaloy*** (http://trespams.com)

Creado el 04/08/2001 12:16 modificado el 04/08/2001 12:16

Este pequeño artículo tiene por fin dar a conocer las posibilidades de Python como lenguaje para la administración de sistemas. Muestra el uso de la clase Cmd la cual nos permite implementar de manera rápida y fácil nuestro propio intérprete de comandos.

Utilizando Python para administrar nuestro sistema

Es habitual que con el paso del tiempo uno llege a tener una gran colección de rutinas, scripts y demás que le ayuden ha automatizar las tareas más habituales.

El problema estas rutinas tienden a desparramarse por los distintos directorios, tenemos que recordar porqué escribimos esa rutina y para qué servía.

Python nos proporciona un mecanismo muy sencillo y a la vez potente para mantener estas rutinas a través de un único interfaz y constuirnos un entorno más amigable: la **librería cmd**.

cmd contiene una única classe Cmd que implementa un procesador de comandos básico. Deberemos crear un descendiente de esa clase e implementar los métodos do_<nombre de funcion> y help_<nombre de funcion>. Podemos crear tantos procesadores de comandos como queramos e incluso anidar procesadores de comandos.

Un ejemplo muy básico de la utilización de esta clase.

Supongamos que somos ludópatas compulsivos y queremos agrupar todas las rutinitas que hemos ido creando a lo largo del tiempo para generar las apuestas.

Creemos un archivo llamado juegos.py con nuestro editor favorito

Tecleemos:

```
#!/usr/bin/env python

from cmd import Cmd
import sys
"""emplo de utilización de la classes Cmd
   Antoni Aloy
"""
class TestCmd(Cmd):
   def __init__(self):
      Cmd.__init__(self)
   def do_exit(self,arg):
      sys.exit()
   def help_exit(self):
      print "Cierra la sesión"
   def do_primitiva(self,arg):
      from random import Random
      aleat = Random()
      numeros = range(1,50)
      aleat.shuffle(numeros)
      laSuerte=numeros[0:6]
      laSuerte.sort()
```

```
      print laSuerte
   def help_help(self):
      print "Muestra la ayuda disponible"
   def help_primitiva(self):
      print "Nos hace la primitiva"
if __name__=='__main__':
   mySession = TestCmd()
   mySession.cmdloop()
```

Ahora no nos queda más que ejecutar el intérprete de Python con este programa como argumento o darle permisos de ejcución para poder iniciarlo directamente.

Podemos ejecutar tanto rutinas escritas en Python como cualquier otra utilidad, ya que Python nos proporciona funciones para ejecutar comandos del sistema o programas externos y capturar su salida, pero eso ya es otra historia...

E-mail del autor: aaloy _ARROBA_ bulma.net
Podrás encontrar este artículo e información adicional en: http://bulma.net/body.phtml?nIdNoticia=784