Python 3 para impacientes

"Simple es mejor que complejo" (Tim Peters)

sábado, 7 de junio de 2014

Egstore: archivos de configuración con EasyGUI

Como hemos visto hasta ahora el módulo **EasyGUI** proporciona lo esencial para construir una interfaz gráfica para nuestros programas y scripts, pero hay algo más.

Además, mediante la clase **Egstore** podemos usar ficheros de configuración para guardar y leer información en el disco de manera permanente, que puede ser usada por los programas en cualquier momento y estar disponible en futuras sesiones.

Es típico que un programa guarde datos sobre su instalación: la ubicación donde está instalado, información sobre su apariencia (colores, fuentes), idioma, formatos empleados, datos del usuario y/o empresa, etc. Toda esta información puede ser leída al principio y utilizada o cambiarse mientras el programa está ejecutándose.

En el siguiente ejemplo crearemos la subclase '***AppConfig***' a partir de la clase ***eg.Egstore*** para leer y grabar datos de configuración.

En la claúsula ***__init__*** se definen los atributos que contendrá la información que se leerá/guardará en el disco: ***self***.usuario, ***self***.idioma y ***self***.rutabasedatos (***self.filename*** es obligatorio).

También, importaremos el módulo ***os*** para utilizar alguna de sus funciones.

```
#!/usr/bin/python3
# -*- coding: utf-8 -*-

import os
import easygui as eg

class AppConfig(eg.EgStore):
 def __init__(self, filename):
  self.usuario = ""
  self.idioma = ""
  self.rutabasedatos = ""
  self.filename = filename
```

A continuación, se leerá el archivo de configuración "**config-gui.txt**" y en el caso de no existir será creado en la ruta del programa:

```
car = os.getcwd()  # Obtiene la ruta de la carpeta actual
ArchivoConfig = os.path.join(car,"","config-gui.txt")
MiConfig = AppConfig(ArchivoConfig)
MiConfig.restore()

if MiConfig.usuario == "":
    usu = os.getenv('USER')  # Captura el Usuario
    idi = "ES"  # Asigna el idioma de la aplicación
    rbd = car  # Asigna ruta de la base de datos

    # Guardar datos en el archivo de configuración

    MiConfig.usuario = usu
    MiConfig.idioma = idi
    MiConfig.rutabasedatos = rbd
    MiConfig.store()
else:
    usu = MiConfig.usuario
    idi = MiConfig.idioma
    rbd = MiConfig.rutabasedatos
```

Buscar

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 concebid...

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

En cualquier parte del programa podemos leer y guardar la información que necesitemos. A continuación, se muestra una ventana para teclear la información y después guardarla.

```
MiConfig.restore()

usu =  MiConfig.usuario
idi =  MiConfig.idioma
rbd =  MiConfig.rutabasedatos

campos = ['Usuario', 'Idioma', 'Ruta base datos']
datos = [usu, idi, rbd]
datos = eg.multenterbox(msg='Modifique los datos',
                        title='EgStore: Guardar datos',
                        fields=campos, values=datos)

MiConfig.usuario = datos[0]
MiConfig.idioma = datos[1]
MiConfig.rutabasedatos = datos[2]
MiConfig.store()
```

Finalmente, se leerán del archivo de configuración los últimos datos almacenados:

```
MiConfig.restore() # Lee archivo de configuración
usu = MiConfig.usuario
idi = MiConfig.idioma
rbd = MiConfig.rutabasedatos
eg.msgbox(usu + " " + idi + " " +rbd,
          "EgStore: Leer",
          ok_button="Seguir")
```

Ir al índice del tutorial de EasyGUI

Publicado por Pherkad en 4:50

Etiquetas: EasyGUI, Python3

Entrada más reciente Inicio Entrada antigua

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

Archivo

junio 2014 (2)

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