Guia Tkinter Documentation

Publicación 0.1.1

Alvarez Alejandro

13 de April de 2016

Índice general

Algo muy buscado en Python son las guías sobre interfaces gráficas ya sea PyGTK, PyQT, WxPython o Tkinter entre las mas conocidas teniendo cada una sus ventajas y desventajas, así como cada una tiene facilidades y complicaciones en su uso y aplicación. En esta oportunidad les daremos un espacio a Tkinter e intentaremos ser lo mas lo mas claro posible sin pasarnos por alto los pequeños detalles en lo que compone la creación de una interfaz gráfica en Python con el toolkit Tkinter.

Contenido:

CAPÍTULO 1

Introduccion

Version 0.1.1. (Descarga)

Puede descargar la versión más reciente de esta guía gratuitamente en la web https://github.com/elilluminado/tutorial_tkinter

1.1 Agradecimientos

Ante todo gracias a todos los lectores y a aquellos que me enviaron correos con consultas y sugerencias y desde ya a los ayudaron a mejorar esta guía

CAPÍTULO 2

Creditos

Esta guia es escrita y mantenida por Alejandro Alvarez <elilluminado@codigopython.com.ar>

2.1 Colaboradores

Ninguno todavia. Por que no ser el primero?

CAPÍTULO 3

LICENCIA

CAPÍTULO 4

Historial de cambios

4.1 Version 0.1.2 (En desarrollo)

4.1.1 Añadidos

4.1.2 Correcciones

4.2 Version 0.1.1

- Se retoma el desarrollo de la guia utilizado Sphinx como motor

CAPÍTULO 5

Introduccion

Sean bienvenidos a esta noble guía que pretende a lo largo de sus capítulos mostrar las herramientas y conocimientos necesarios para el desarrollo de interfaces gráficas con las librerías Tk y el lenguaje de programación Python. El contenido de esta guía no pretende ser una referencia completa de lo que puede lograr pero si servir como una base al poco contenido que se logra encontrar en español sobre Tkinter.

CAPÍTULO 6

Acerca de Tk

Tkinter es un binding de la biblioteca gráfica Tcl/Tk para el lenguaje de programación Python, con estos queremos decir que Tk se encuentra disponible para varios lenguajes de programación entre los cuales se encuentra Python con el nombre de Tkinter. Este no es mas que una adaptación de esta librería para el lenguaje Python con lo cual usar Tk en otro lenguaje no nos supondrá un inconveniente.

Se considera un estándar para la interfaz gráfica de usuario (GUI) para Python y es el que viene por defecto con la instalación para Microsoft Windows y preinstalado en la mayoría de las distribuciones de GNU/Linux. Con Tkinter podremos conseguir resultados casi tan buenos como con otras librerías gráficas siempre teniendo en cuenta que quizás con otras herramientas podamos realizar trabajos mas complejos donde necesitemos una plataforma mas robusta, pero como herramienta didáctica e interfaces sencillas nos sobrara, dándonos una perspectiva de lo que se trata el desarrollo de una parte muy importante de una aplicación si deseamos distribuirla. Gracias a Tkinter veremos como interactuar con el usuario pidiéndole el ingreso de datos, capturando la pulsación de teclas, movimientos del mouse, entre algunas de las cosas que podremos lograr.

CAPÍTULO 7

Historia

CAPÍTULO 8

Virtudes y limitaciones

CAPÍTULO 9

Que es una interfaz grafica

CAPÍTULO 10

Buenas practicas

CAPÍTULO 11

Instalando Tkinter

CAPÍTULO 12

Linux

CAPÍTULO 13

Windows

CAPÍTULO 14

Mac

CAPÍTULO 15

Sobre esta guía

Para los que deseen colaborar con esta guía así como en su contenido, pueden hacerlo enviandome un correo electronico (elilumi nado@codigopython.com.ar), de preferencia en español, o a traves de GitHub desde el canal para envio de fallos o correcciones desde la siguiente url: https://github.com/elilumi nado/tutorial_tkinter/issues

Para los mas experimentados pueden colaborar realizando un fork de la guia y luego enviar los cambios entrando en https://github.com/elilumi nado/tutorial_tkinter

CAPÍTULO 16

Empezando por lo básico

Lo primero que debemos hacer al igual que con otros módulos, es que debemos importarlo para poder comenzar a utilizarlo, y al igual que con otros módulos no hay una sola forma de hacerlo.

La primer forma (y la mas popular):

Python 2.x

```
from Tkinter import *
```

Python 3.x

```
from tkinter import *
```

Y la segunda:

Python 2.x

```
import Tkinter
```

Python 3.x

```
import tkinter
```

Nota: En las versiones 3.x de Python el modulo Tk se debe llamar de esta forma *"from tkinter import *"* y no de esta otra forma *"from Tkinter import *"*, notar la t minuscula en el nombre, es un cambio menor que hay que tener en cuenta si están trabajando con Python 3.

La diferencia entre usar la primera o la segunda forma es la misma con la que nos podemos encontrar a la hora de importar un modulo en Python, para verlo les muestro esta diferencia utilizando al modulo *'time'* y usamos el siguiente ejemplo:

```
1 import time
2 time.sleep(10)
```

```
1 from time import sleep
2 sleep(10)
```

Usando la forma *"import Tkinter"* cada vez que utilicemos una función de este modulo tendremos que anteponer la palabra *'Tkinter'*, en cambio usando la segunda forma *"from Tkinter import *"* simplemente deberemos usar el nombre de la función sin el nombre del modulo. Hay otras formas pero dependerá de tu forma de trabajar o la que te resulte mas cómoda, en todo caso puedes darle un vistazo a esta traducción de la guía de estilo escrita por Guido van Rossum y Barry Warsaw http://mundogeek.net/traducciones/guia-estilo-python.htm

Por ahora a modo didactico usaremos una variacion de la segunda forma para que los ejemplos sean mas claros, pero esto queda a tu eleccion. Ahora si retomemos nuestro camino.

Una observación que tendremos que tener antes de continuar es no debemos pasa por alto la posibilidad de que el usuario no tenga instalado las librerías de Tkinter y en consecuencia nuestra aplicación no podrá funcionar, lo mejor en este caso como en muchos otros es anticiparnos a los posibles errores que puedan ir surgiendo y manejar las excepciones de la siguiente forma:

Python 2.x

```
try:
    import Tkinter
except ImportError:
    raise ImportError("Se requiere el modulo Tkinter")
```

Python 3.x

```
try:
    import tkinter
except ImportError:
    raise ImportError("Se requiere el modulo tkinter")
```

Tambien podriamos ir un poco mas lejos y suponer que no sabemos que version esta usando el usuario y llevar el codigo de arriba para que sea compatible con ambas versiones de Python

```
import sys

PYTHON_VERSION = sys.version_info.major

if PYTHON_VERSION < 3:
    try:
        import Tkinter as tk
    except ImportError:
        raise ImportError("Se requiere el modulo Tkinter")
else:
    try:
        import tkinter as tk
    except ImportError:
        raise ImportError("Se requiere el modulo tkinter")
```

Otra forma igual de efectiva

```
try:
    import Tkinter as tk
except ImportError:
    import tkinter as tk
```

La forma mas elegante y eficiente para mi gusto (Aclaro que solo es mi humilde opinion), es a traves de una libreria externa llamada *six* que la pueden bajar e instalar desde aqui https://pypi.python.org/pypi/six. Con esta podemos obtener la primer forma mostrada pero mas compacta. Dependera de ustedes si van a agregar librerias externas a su aplicacion.

```
try:
    from six.moves import tkinter as tk
except ImportError:
    raise ImportError("Se requiere el modulo Tkinter")
```

De esta forma en caso de que el potencial usuario de nuestra preciada aplicación, no pueda ejecutarla pueda saber cual es motivo (en este caso es que no tenga instalado Tkinter) por el cual no pudo iniciarla.

Una vez importado el modulo Tkinter correctamente podemos utilizarlo para crear nuestra primera ventana de la siguiente forma:

```
from six.moves import tkinter as tk

root = tk.Tk()
root.mainloop()
```

Estas lineas son fundamentales, ya que de ellas dependerá gran parte del contenido así como pueden ser botones y menús, aunque mas adelante conforme vayamos viendo temas mas avanzados podremos hacer uso de otras herramientas y técnicas.

En la primera linea (No tengamos en cuenta la importación vista anteriormente) se crea un identificador que sera el que utilizaremos para referirnos a la ventana, en este caso lo llamamos 'root' y es una de las funciones mas importantes de Tkinter. Siempre que iniciamos un identificador que en este caso lo llamamos 'root' debemos cerrarlo para capturar como veremos mas adelante los eventos.

Nota: Para que el codigo escrito sea mantenible se utilizara de ahora en adelante el modulo *six*, esto permite ignorar la version de Python utilizada.

Nota: La variable 'root' usada para nombrar al identificador, puede ser reemplazada por cualquier otro nombre siempre y cuando se respeten las palabras reservadas de Python, aunque es muy utilizado usar el nombre 'root' para la ventana principal y puede ser una buena practica para que resulte mas familiar a los demás programadores que se quieran unir al desarrollo de nuestra aplicación.

Con esto ya tendremos una ventana vacía que nos servirá para comenzar a trabajar, a partir de ahora iremos ampliando el contenido mostrando los distintos widgets con los que contamos en Tkinter y luego para finalizar crearemos un ejemplo sencillo para unir lo se vio a lo largo de este material.

CAPÍTULO 17

Indices and tables

- genindex
- search