

Tkinter

Tkinter es una librería del lenguaje de programación Python y funciona para la creación y el desarrollo de aplicaciones de escritorio. Esta librería facilita el posicionamiento y desarrollo de una interfaz gráfica de escritorio con Python. Tkinter es el paquete estándar de Python para interactuar con Tkt.

De acuerdo a la documentación de Python, [TK](#) se describe a sí mismo como el único *toolkit* o kit de herramientas para el desarrollo de una interfaz gráfica de usuario (GUI) que funciona en todos los sistemas operativos, es decir, funciona en Windows, Mac OS y Linux. Además, está diseñado y preparado para lenguajes dinámicos con un alto nivel, como pueden ser Tcl, Ruby, Perl o python aplicaciones de escritorio entre otros.

Por tanto, con su uso no es necesario que accedas a las rutinas de bajo nivel de cualquiera de los sistemas operativos para controlar gráficamente la pantalla de las interfaces de usuario que piensas crear. Saber qué es Tkinter te aporta mucha facilidad en el desarrollo de tus aplicaciones.

Elementos creados con Tkinter

Ventana de aplicación

Con los conceptos de la librería de tk inter es posible realizar la venta de tu aplicación. Una ventana de aplicación es un lienzo donde se dispondrán una serie de controles.

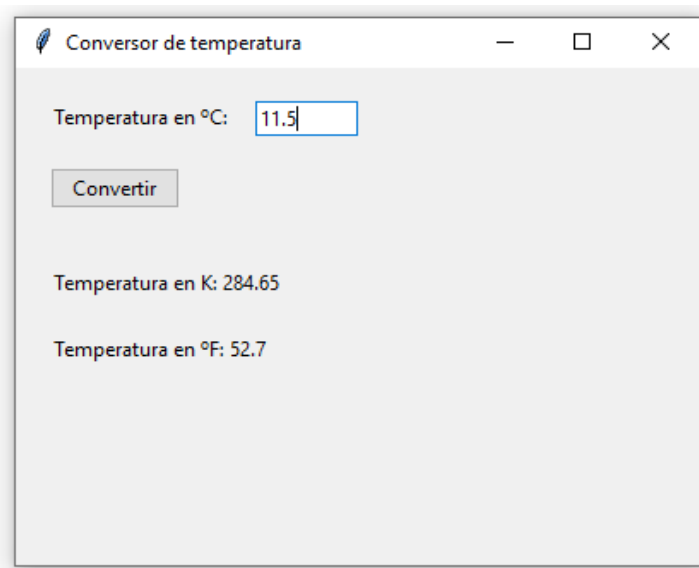
Widgets

En TK a los controles de las ventanas de aplicación se les llama widgets. Estos se encargan de realizar las acciones de nuestra aplicación. Pueden ser cuadros de texto (donde el usuario podrá escribir), etiquetas de texto (el programa comunicará texto que no se podrá modificar) o botones, entre otros. Los controles tienen una posición y estilo que pueden modificarse. Además, responderán acciones del usuario, es decir, los eventos gestionados.

Más ventanas

Se utilizan en programas donde se necesitan ventanas diferente a la ventana de aplicación. Funcionan para varios objetivos, como mensajes de aplicación o ventanas hijas para dar de alta a un usuario, entre otros.

Para ver como trabaja esta librería crearemos una pequeña aplicación que permite convertir un valor de temperatura expresado en grados Celsius a valores en Kelvin y grados Fahrenheit, como lo ilustra la siguiente imagen.



Para comenzar, el primer paso para cualquier aplicación de Tk es importar los módulos correspondientes:

```
import tkinter as tk
from tkinter import ttk
```

Aquí importamos el módulo principal tkinter abreviado como tk, una convención habitual entre los programadores de Python. En segundo lugar importamos el módulo ttk que se encuentra dentro de tkinter. Estaremos utilizando objetos que están dentro de ambos módulos. Para las diferencias entre tk y ttk, véase este artículo.

El segundo paso fundamental es crear la ventana principal. Todas las aplicaciones de Tk tendrán una ventana principal, y eventualmente algunas otras ventanas secundarias.

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

En la primera línea creamos una instancia de Tk, que se ocupa de crear la ventana principal y de iniciar internamente un intérprete de Tcl y Tk, cosa que explica el nombre de la clase. En efecto, sería más claro si el nombre fuese tk.MainWindow o similar. Con propósitos didácticos hemos llamado ventana a la instancia en cuestión, pero la convención que habitúan seguir los programadores de Tk es nombrarla root (raíz, pues es el objeto del cual nacerá el resto de la interfaz).

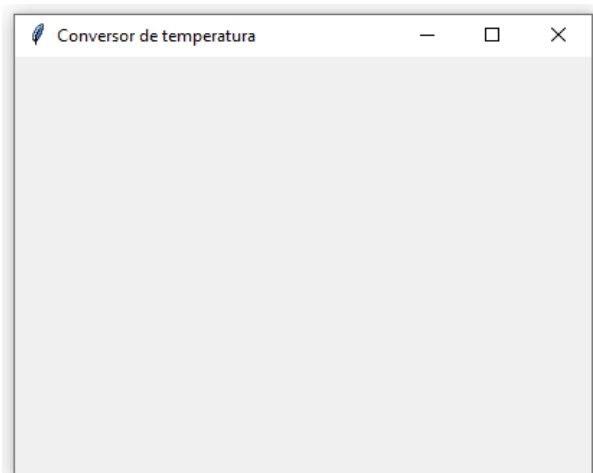
La segunda línea ejecuta el método **mainloop()**, que es el bucle principal del programa. Todas las aplicaciones de escritorio (en Tk o en cualquier otra herramienta afín) trabajan con un bucle principal que se ocupa de gestionar los eventos de la interfaz gráfica.

El bucle principal se está ejecutando constantemente (pues, justamente, es un bucle) y una de sus tareas principales es «dibujar» la ventana en la pantalla, por lo cual el método mainloop() solo finaliza cuando se cierra la última ventana de nuestra aplicación. Este es un dato importante a tener en cuenta siempre que atinemos a ubicar código debajo de la llamada al mainloop(). Por lo general, ventana.mainloop() será la última línea de nuestro código.

Ahora vamos a dar un título y un tamaño a la ventana, vía los métodos `title()` y `config()`.

1. `ventana = tk.Tk()`
2. `ventana.title("Conversor de temperatura")`
3. `ventana.config(width=400, height=300)`
4. `ventana.mainloop()`

En cualquier método, función o clase de Tk donde se requiera especificar un tamaño, será vía los argumentos `width` (ancho) y `height` (alto),



indicados en píxeles. Así, nuestra ventana tendrá al iniciarse un ancho de 400 píxeles y un alto de 300 píxeles.

Ahora bien, tenemos la ventana configurada y debemos empezar a llenarla de botones, cajas de texto, etiquetas, menús, casillas de verificación, etc. A cada uno de estos componentes que podemos incluir en nuestra interfaz y con los cuales el usuario puede interactuar se lo conoce como *control* o *widget*.

Empecemos por mostrar un mensaje en la ventana que indique al usuario que debe ingresar la temperatura en grados Celsius, para lo cual podemos usar un control o *widget* llamado etiqueta. La etiqueta está representada en Tk por la clase `ttk.Label`.

```
import tkinter as tk
from tkinter import ttk

ventana = tk.Tk()

ventana.title("Conversor de temperatura")
ventana.config(width=400, height=300)

etiqueta_temp_celsius = ttk.Label(text="Temperatura en °C:")
etiqueta_temp_celsius.place(x=20, y=20)

ventana.mainloop()
```

Para introducir un control en la interfaz, primero debemos crear una instancia de la clase correspondiente (`ttk.Label`, en este caso), asignársela a una variable (`etiqueta_temp_celsius`) y ubicarla en algún lugar de la ventana vía el método `place()`. Este método requiere que indiquemos de forma absoluta la posición del control en la ventana, esto es, especificando la posición en las coordenadas X e Y. Opcionalmente podemos pasar los argumentos `width` y `height` para asignarle al control un ancho y un alto fijos; de lo contrario, Tk provee un tamaño por defecto.

Debajo de la creación de la etiqueta, agreguemos la caja de texto para introducir la temperatura y el botón para realizar la conversión. Para ello usaremos dos clases nuevas: `ttk.Entry` y `ttk.Button`.

```
import tkinter as tk
from tkinter import ttk

ventana = tk.Tk()
ventana.title("Conversor de temperatura")
ventana.config(width=400, height=300)
etiqueta_temp_celsius = ttk.Label(text="Temperatura en °C:")
etiqueta_temp_celsius.place(x=20, y=20)
caja_temp_celsius = ttk.Entry()
caja_temp_celsius.place(x=140, y=20, width=60)
boton_convertir = ttk.Button(text="Convertir")
```

```
boton_convertir.place(x=20, y=60)
ventana.mainloop()
```

Como se observa, la lógica es siempre la misma: primero la creación del control, luego su posicionamiento. La clase `Entry()` no lleva argumento `text` puesto que se espera que el usuario ingrese allí un dato.

Respecto del diseño de la interfaz nos resta únicamente agregar las dos etiquetas en las cuales se mostrarán los resultados de la conversión, o sea, los valores en Kelvin y grados Fahrenheit.

```
import tkinter as tk
from tkinter import ttk
ventana = tk.Tk()
ventana.title("Conversor de temperatura")
ventana.config(width=400, height=300)
etiqueta_temp_celsius = ttk.Label(text="Temperatura en °C:")
etiqueta_temp_celsius.place(x=20, y=20)
caja_temp_celsius = ttk.Entry()
caja_temp_celsius.place(x=140, y=20, width=60)
boton_convertir = ttk.Button(text="Convertir")
boton_convertir.place(x=20, y=60)
etiqueta_temp_kelvin = ttk.Label(text="Temperatura en K: n/a")
etiqueta_temp_kelvin.place(x=20, y=120)
etiqueta_temp_fahrenheit = ttk.Label(text="Temperatura en °F:
n/a")
etiqueta_temp_fahrenheit.place(x=20, y=160)
ventana.mainloop()
```

Hasta aquí el diseño de la ventana. Ahora debemos hacer que efectivamente al presionar el botón nuestro programa convierta la temperatura ingresada en la caja de texto. Para ello debemos crear una función y asociarla al `boton_convertir` vía el argumento `command`. Luego, desde el interior de la función obtenemos el contenido de la caja de texto vía el método `get()` y arrojamos el resultado de la conversión en las `etiqueta_temp_kelvin` y `etiqueta_temp_fahrenheit` vía el método `config()`

```
import tkinter as tk
from tkinter import ttk
def convertir_temp():
    temp_celsius = float(caja_temp_celsius.get())
    temp_kelvin = temp_celsius + 273.15
    temp_fahrenheit = temp_celsius*1.8 + 32
    etiqueta_temp_kelvin.config(text=f"Temperatura en K:
{temp_kelvin}")
    etiqueta_temp_fahrenheit.config(
```

```
        text=f"Temperatura en °F: {temp_fahrenheit}")
ventana = tk.Tk()
ventana.title("Conversor de temperatura")
ventana.config(width=400, height=300)
etiqueta_temp_celsius = ttk.Label(text="Temperatura en °C:")
etiqueta_temp_celsius.place(x=20, y=20)
caja_temp_celsius = ttk.Entry()
caja_temp_celsius.place(x=140, y=20, width=60)
boton_convertir = ttk.Button(text="Convertir",
command=convertir_temp)
boton_convertir.place(x=20, y=60)
etiqueta_temp_kelvin = ttk.Label(text="Temperatura en K: n/a")
etiqueta_temp_kelvin.place(x=20, y=120)
etiqueta_temp_fahrenheit = ttk.Label(text="Temperatura en °F: n/a")
etiqueta_temp_fahrenheit.place(x=20, y=160)
ventana.mainloop()
```

La función `convertir_temp()` será invocada por Tk cada vez que el usuario presione el `boton_convertir`. El contenido de la caja de texto hemos tenido que convertirlo a un número de coma flotante vía la función incorporada `float()`, ya que por defecto el resultado de `get()` es siempre una cadena.

