

Python Programming

Módulo 06

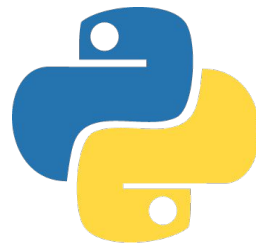
Widgets básicos

Widgets básicos

Estos 3 widgets muchos los vieron en el curso introductorio, no está demás un repaso de estos.

- Botón.
- Caja.
- Etiqueta.

Luego veremos otros widgets más avanzados para crear mejores aplicaciones.



Botón

Tk provee dos *widgets* para botones:

`ttk.Button` y `tk.Button`. El primero es un *widget* con apariencia básica, el segundo tiene mejoras en el aspecto.

Por ejemplo, el siguiente código crea un botón con el texto “Hola mundo” y lo ubica en la posición 50, 10.

```
boton = ttk.Button(text="Hola mundo")
boton.place(x=50, y=10)
```

El parámetro `text` indica el texto que mostrará el botón.

Podemos usar la función `help()` en la consola interactiva para conocer los argumentos de cada control, o mirar la [documentación oficial](#).

Cuando se omiten los valores `width` y `height` en una llamada a `place()`, son inferidos de acuerdo con la longitud del texto.

Para asociar la presión del botón con una función de Python, podemos crear una función como la siguiente:

```
def boton_presionado():
    print("Presionado")
```

Y en la aplicación usamos el parámetro `command`.

```
boton = ttk.Button(text="Hola mundo", command=boton_presionado)
boton.place(x=50, y=10)
```

Los argumentos pasados al crear un widget pueden ser luego adulterados vía la función `config()`. Por ejemplo, para cambiar el texto del botón en algún otro lugar del programa:

```
boton.config(text="Nuevo texto")
```

Hemos visto cómo crear una ventana, un botón y cómo asociarlo a una función de Python cuando es presionado. A continuación, haremos un pequeño recorrido por el resto de los widgets principales que componen una interfaz de usuario: etiquetas, cajas de texto, listas, y más.

Cajas de texto (Entry y Text)

Tk provee dos widgets para ingresar texto: `ttk.Entry` y `tk.Text`. El primero es generalmente útil para textos pequeños en una misma línea, soportando únicamente texto plano. El segundo, además de permitir texto de múltiples líneas, también puede albergar imágenes y texto con formato.

```
entry = ttk.Entry()
```

Para obtener el texto que el usuario ha ingresado en el control utilizamos la función `get()`.

A continuación vemos una función asociada a un botón la cual toma lo que hay en la caja:

```
def imprimir_texto():  
    print(entry.get())
```

En la aplicación:

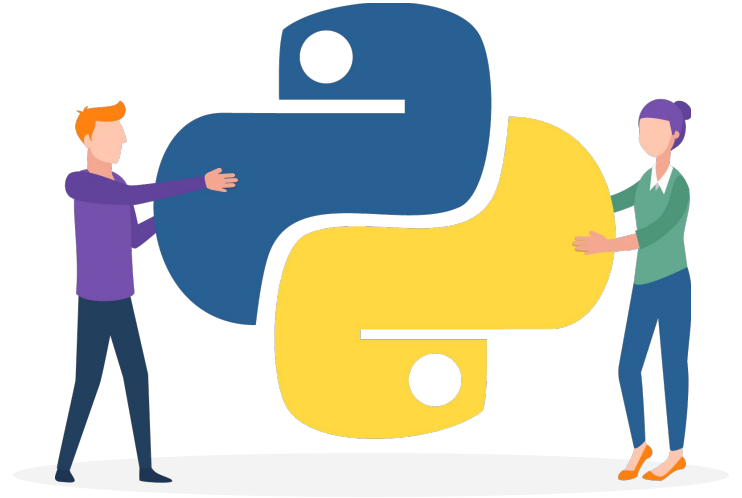
```
entry = ttk.Entry()  
entry.place(x=10, y=10)  
boton = ttk.Button(text="Imprimir texto",  
    command=imprimir_texto)  
boton.place(x=10, y=50)
```

Vía las funciones `insert()` y `delete()` podemos añadir y remover texto del control en una determinada posición:

```
# Inserta el texto al comienzo del control.  
entry.insert(0, "Hola mundo")
```

```
# Borra los primeros 10 caracteres.  
entry.delete(0, 10)
```

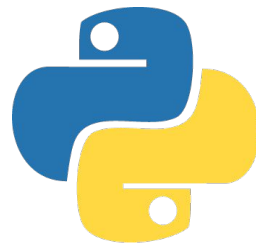
```
# Borra todo el texto.  
entry.delete(0, tk.END)
```



Etiqueta (Label)

Una etiqueta es un control cuya única funcionalidad es contener un texto o una imagen. Al igual que los botones, la propiedad que controla el texto es `text`.

```
label = ttk.Label(text="Hola mundo")
```



¡Sigamos trabajando!