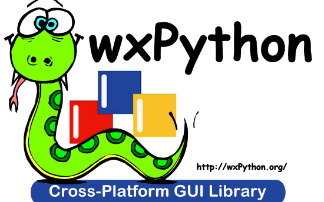
**Investigación Tkinter**

1. **¿Qué librerías incluye Python para Interfaces Graficas de Usuario?**

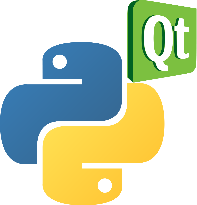
TkInter es la librería gráfica utilizada por Python y es una migración de la librería gráfica de Tk creada para el lenguaje TLC. Como gran ventaja cabe destacar que es fácil de implementar y se parece bastante a Gtk en su enfoque general.



wxPython es una librería que conecta los wxWidgets a Python. Es una alternativa a TkInter muy valorada y particularmente completa, próxima al sistema operativo. La librería wxWidgets es una librería gráfica de libre distribución escrita en C++ que permite escribir una interfaz gráfica que será idéntica sea cual sea el sistema operativo, pero tomando su apariencia. Esta librería es una referencia, y funciona en Python 3 a partir de la versión 3.3.

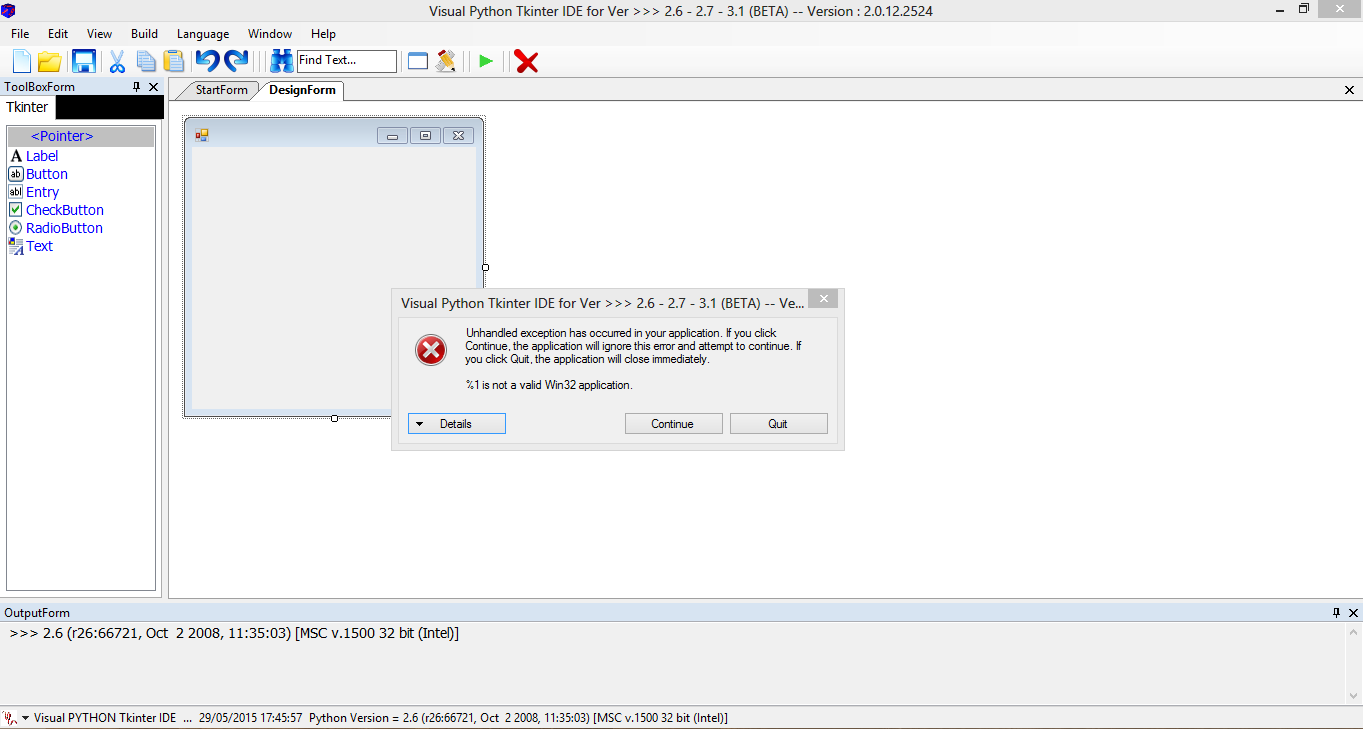


PyQt es una librería gráfica de libre distribución que conecta Qt a Python. Qt, que debe pronunciarse como la palabra inglesa «cute», se escribe en C++ y es muy portable. El entorno gráfico KDE está construido en Qt. Muchos módulos complementarios adicionales hacen de ella mucho más que una simple librería que permite realizar interfaces gráficas. Qt está vinculada a QtDesigner, que es, de manera similar a Glade, una aplicación que permite crear interfaces gráficas. Qt está completamente migrada a Python 3.x desde su versión 4.5, lo que hace de ella un candidato excelente.



1. **¿Qué es Tkinter?**

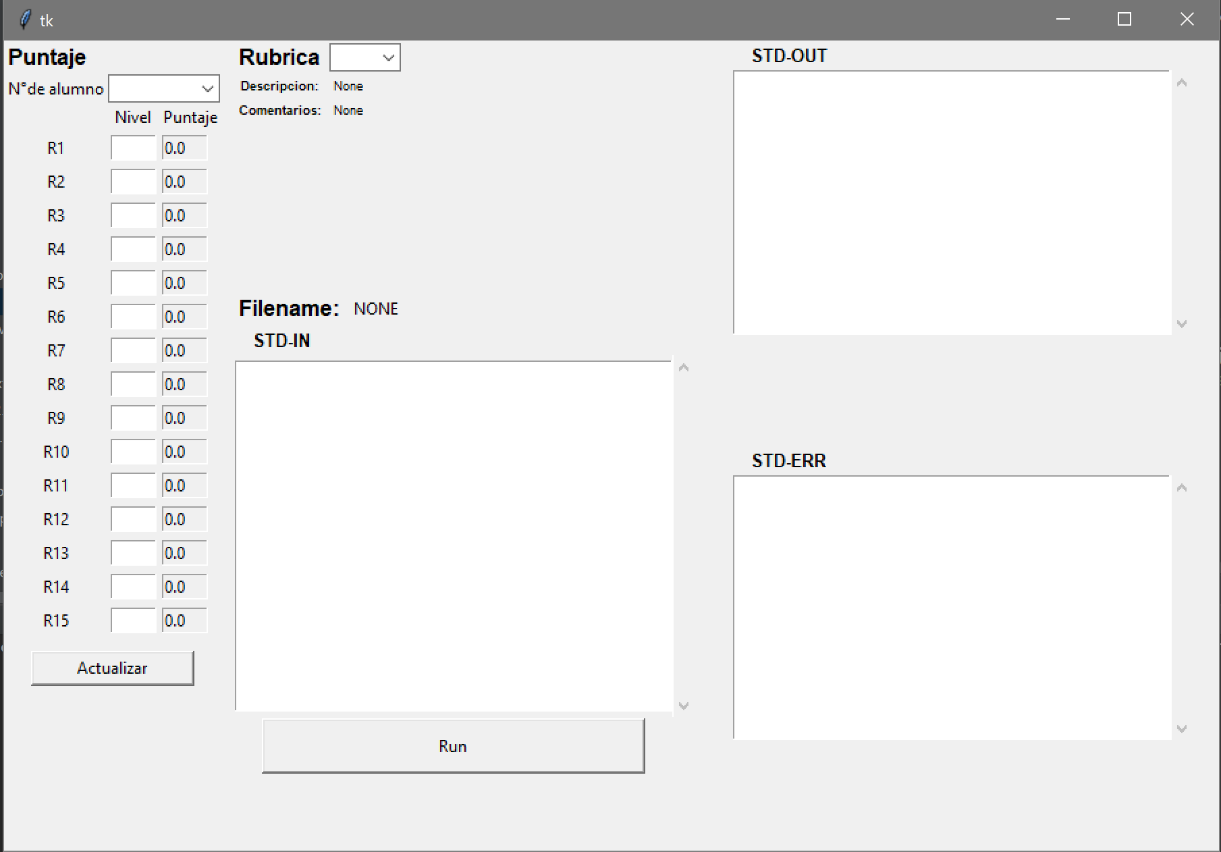
El paquete tkinter es la interfaz por defecto de Python para el toolkit de la GUI Tk. Tanto Tk como tkinter están disponibles en la mayoría de las plataformas Unix, así como en sistemas Windows (Tk en sí no es parte de Python, es mantenido por ActiveState).



1. **¿Qué es un widget?**

En general, el widget es un elemento de la interfaz gráfica de usuario (GUI) que muestra/ilustra información o proporciona una forma para que el usuario interactúe con el sistema operativo. En Tkinter, los widgets son objetos; instancias de clases que representan botones, marcos, etc.

Cada widget por separado es un objeto de Python. Al crear un widget, debe pasar su padre como parámetro a la función de creación del widget. La única excepción es la ventana "raíz", que es la ventana de nivel superior que contendrá todo lo demás y no tiene un padre.



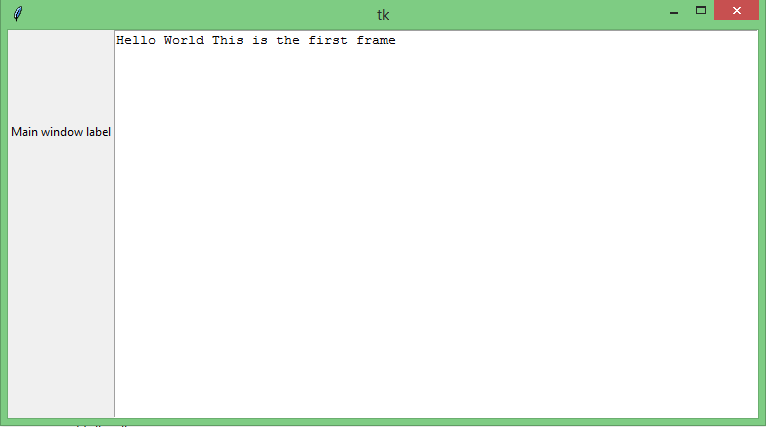
1. **¿Cómo funcionan los siguientes controles? (Agregar imagen):**
   1. **TK (frame).**

El widget Frame es muy importante para el proceso de agrupar y organizar otros widgets de una manera amigable. Funciona como un contenedor, que se encarga de organizar la posición de otros widgets.

Utiliza áreas rectangulares en la pantalla para organizar el diseño y proporcionar relleno de estos widgets. Un marco también se puede utilizar como clase básica para implementar widgets complejos.

Sintaxis

w = Frame ( master, option, ... )

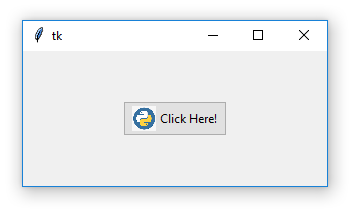


* 1. **Button.**

El widget de button se utiliza para agregar botones en una aplicación de Python. Estos botones pueden mostrar texto o imágenes que transmiten el propósito de los botones. Puede adjuntar una función o un método a un botón que se llama automáticamente cuando hace clic en el botón.

Sintaxis

w = Button ( master, option=value, ... )



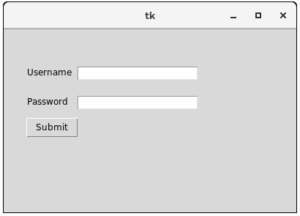
* 1. **Label.**

Este widget implementa un cuadro de visualización donde puede colocar texto o imágenes. El texto que muestra este widget se puede actualizar en cualquier momento que desee.

También es posible subrayar parte del texto (como para identificar un atajo de teclado) y extender el texto en varias líneas.

Sintaxis

w = Label ( master, option, ... )

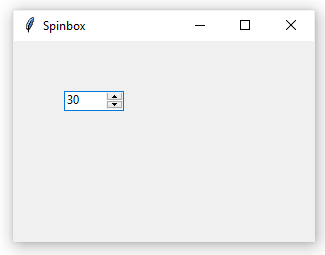


* 1. **Spinbox.**

El widget Spinbox es una variante del widget Tkinter Entry estándar, que se puede utilizar para seleccionar entre un número fijo de valores.

Sintaxis

w = Spinbox( master, option, ... )



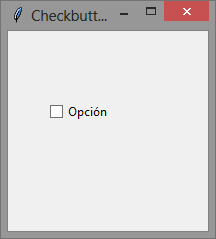
* 1. **Checkbox.**

El widget Checkbutton se utiliza para mostrar una serie de opciones a un usuario como botones de alternancia. A continuación, el usuario puede seleccionar una o más opciones haciendo clic en el botón correspondiente a cada opción.

También puede mostrar imágenes en lugar de texto.

Sintaxis

w = Checkbutton ( master, option, ... )



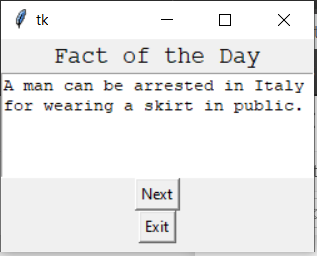
* 1. **Text.**

Los widgets de texto brindan capacidades avanzadas que le permiten editar un texto de varias líneas y formatear la forma en que debe mostrarse, como cambiar su color y fuente.

También puede usar estructuras elegantes como pestañas y marcas para ubicar secciones específicas del texto y aplicar cambios a esas áreas. Además, puede incrustar ventanas e imágenes en el texto porque este widget fue diseñado para manejar texto sin formato y con formato.

Sintaxis

w = Text ( master, option, ... )



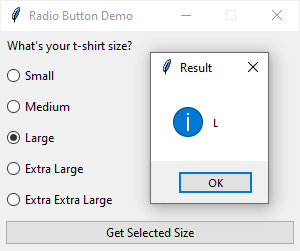
* 1. **RadioButton.**

Este widget implementa un botón de opción múltiple, que es una forma de ofrecer muchas selecciones posibles al usuario y le permite elegir solo una de ellas.

Para implementar esta funcionalidad, cada grupo de radiobotones debe estar asociado a la misma variable y cada uno de los botones debe simbolizar un único valor. Puede usar la tecla Tab para cambiar de un botón de radio a otro.

Sintaxis

w = Radiobutton ( master, option, ... )



* 1. **Image.**

El widget de imagen se utiliza para crear y manipular imágenes. La sintaxis para crear una imagen es la siguiente:

image create type name options

