

INTRODUCCIÓN TKINTER

Para Python



Python Tkinter

ID: 3	1	AAA	BBB	ab@mail.com	17
First Name: EEE	3	EEE	FFF	ef@mail.com	91
Last Name: FFF	4	GGG	HHH	gh@mail.com	47
Email: ef@mail.com	7	MMM	NNN	mn@mail.com	25
Age: 91	8	PPP	QQQ	pq@mail.com	43
	9	RRR	SSS	rs@mail.com	94



Get Selected Row from Treeview

11 DE FEBRERO DE 2023

PERE MANEL VERDUGO ZAMORA

pereverdugo@gmail.com

Contenido

1.- Introducción a Tkinter	2
2.- Etiqueta, label	3
3.- Imágenes	6
4.- Message	11
5.- Button	13
6.- Radio Button	14
7.- CheckButton.....	16
8.- Scale	18
9.- Entry	19
10.- Canvas	20
11.- Scrollbar	22
12.- Messagebox I	23
13.- Messagebox II	25
14.- Diálogos comunes	30
15.- Pack	32
16.- Place	36
17.- Menu	38

1.- Introducción a Tkinter

```
# importamos tkinter
import tkinter as tk

# Tenemos que crear un widget que sea la raiz
# Se tiene que crear antes que los otros widgets y
# solo hay una sola raiz por aplicación.
# La raiz va a ser la ventana donde colocaremos
# los otros widgets

ventana = tk.Tk()

# Ahora vamos a crear una etiqueta.
# El primer parámetro es la ventana padre
# El segundo parámetro es el texto que deseamos
# mostrar en la etiqueta.

lblMensaje = tk.Label(ventana, text="Hola a todos")

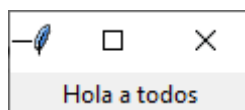
# Con pack la ventana se acopla al tamaño de la etiqueta.

lblMensaje.pack()

# Para desplegar la ventana es necesario invocar el
# ciclo principal, adentro de ese ciclo se lleva a
# cabo la administración de eventos.

ventana.mainloop()
```

Este será el resultado:



2.- Etiqueta, label

```
# importamos tkinter
import tkinter as tk

ventana = tk.Tk()

# Una forma sencilla de crear una etiqueta de múltiples
# líneas es creando una cadena y colocarla en la etiqueta.

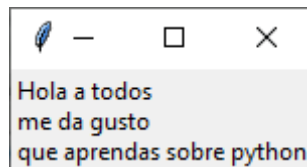
mensaje = """Hola a todos
me da gusto
que aprendas sobre python"""

# Podemos colocar pack en la misma línea
# con justify indicamos como justificar el texto
# LEFT, CENTER, RIGHT

lblMensaje = tk.Label(ventana, text=mensaje, justify=tk.LEFT).pack()

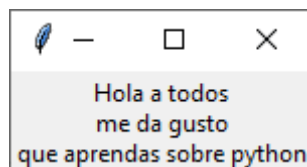
ventana.mainloop()
```

Este será el resultado:



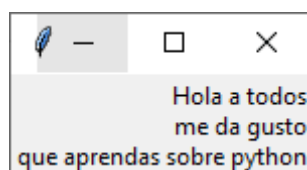
Cambiamos a CENTER:

```
lblMensaje = tk.Label(ventana, text=mensaje, justify=tk.CENTER).pack()
```



Cambiamos a RIGHT:

```
lblMensaje = tk.Label(ventana, text=mensaje, justify=tk.RIGHT).pack()
```



```

# importamos tkinter
import tkinter as tk

ventana = tk.Tk()

# Con fg indicamos el color de la fuente

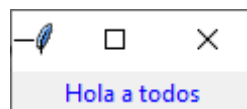
lblMensaje = tk.Label(ventana, text='Hola a todos', fg='blue').pack()

# font nos sirve para indicar el tipo de fuente y su estilo

ventana.mainloop()

```

Este será el resultado:



```

# importamos tkinter
import tkinter as tk

ventana = tk.Tk()

# font nos sirve para indicar el tipo fuente y su estilo.

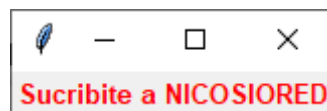
lblMensaje2 = tk.Label(ventana, text='Sucribite a NICOSIORED',
                       fg='red', font='Helvetica 10 bold').pack()

# font nos sirve para indicar el tipo de fuente y su estilo

ventana.mainloop()

```

Este será el resultado:



```

# importamos tkinter
import tkinter as tk

ventana = tk.Tk()

# bg nos permite indicar un color de fondo en el texto

lblMensaje3 = tk.Label(ventana, text='Ve los vídeos',
                       fg='white', bg='red', font='Times 18').pack()

```

font nos sirve para indicar el tipo de fuente y su estilo

```
ventana.mainloop()
```

Este será el resultado:



3.- Imágenes

En este capítulo vamos a aprender insertando imágenes aprovechando las etiquetas.

Vamos a buscar una imagen PNG que no sea muy grande y la guardamos en la carpeta de este proyecto.

```
# importamos tkinter
import tkinter as tk

ventana = tk.Tk()

# Cargamos la imagen
logo=tk.PhotoImage(file='pokemon.png')

# Con image indicamos cual es la imagen que se
# colocará en esa etiqueta.

lblMensaje = tk.Label(ventana, image=logo).pack()
ventana.mainloop()
```

Este será el resultado:



Vamos a agregar otra etiqueta:

```
# importamos tkinter
import tkinter as tk

ventana = tk.Tk()

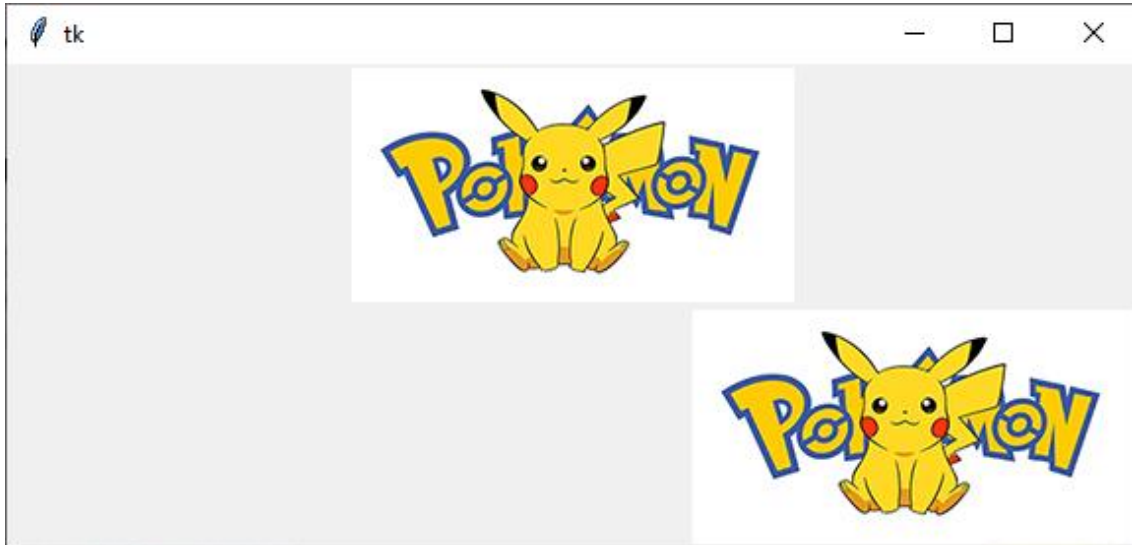
# Cargamos la imagen
logo=tk.PhotoImage(file='pokemon.png')

# Con image indicamos cual es la imagen que se
# colocará en esa etiqueta.

lblMensaje = tk.Label(ventana, image=logo).pack()
lblMensaje2 = tk.Label(ventana, image=logo).pack(side='right')

ventana.mainloop()
```

Este será el resultado habiendo modificado el ancho de la ventana:



```
# importamos tkinter
import tkinter as tk

ventana = tk.Tk()

# Cargamos la imagen
logo=tk.PhotoImage(file='pokemon.png')

# Con image indicamos cual es la imagen que se
# colocará en esa etiqueta.

lblMensaje = tk.Label(ventana, image=logo).pack()
lblMensaje2 = tk.Label(ventana, image=logo).pack(side='right')
lblMensaje3 = tk.Label(ventana, image=logo).pack(side='left')

ventana.mainloop()
```

Este será el resultado:




```

# importamos tkinter
import tkinter as tk

ventana = tk.Tk()

# Cargamos la imagen
logo=tk.PhotoImage(file='pokemon.png')

# Ahora colocaremos un label con texto y otro con imagen.
mensaje = """Ahora colocamos texto
junto con una imagen
y experimentaremos como
van a lucir juntos."""

# Con image indicamos cual es la imagen que se
# colocará en esa etiqueta.

blbMensaje = tk.Label(ventana, image=logo).pack()
lblMensaje2 = tk.Label(ventana, image=logo).pack(side='right')
lblMensaje3 = tk.Label(ventana, text=mensaje).pack(side='left')

ventana.mainloop()

```

Este será el resultado:



Vamos a cambiar la imagen y el texto:

```

lblMensaje2 = tk.Label(ventana, image=logo).pack(side='left')
lblMensaje3 = tk.Label(ventana, text=mensaje).pack(side='right')

```

Este será el resultado:



Vamos con otro ejemplo:

```
# importamos tkinter
import tkinter as tk

ventana = tk.Tk()

# Cargamos la imagen
logo=tk.PhotoImage(file='pokemon.png')

# Ahora colocaremos un label con texto y otro con imagen.
mensaje = """Ahora colocamos texto
junto con una imagen
y experimentaremos como
van a lucir juntos."""

# Con image indicamos cual es la imagen que se
# colocará en esa etiqueta.

lblMensaje = tk.Label(ventana, image=logo).pack()
lblMensaje2 = tk.Label(ventana, image=logo).pack(side='left')
lblMensaje3 = tk.Label(ventana, text=mensaje).pack(side='right')

# Podemos hacer que el texto se coloque sobre la imagen
# al usar compound.
lblMensaje4 = tk.Label(ventana, text=mensaje, image=logo,
                        compound=tk.CENTER).pack()

ventana.mainloop()
```

Este será el resultado:



4.- Message

Ejemplo 1:

```
# importamos tkinter
import tkinter as tk

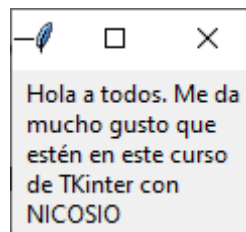
ventana = tk.Tk()

miTexto='Hola a todos. Me da mucho gusto que estén en este curso de
TKinter con NICOSIO'

# Ejemplo 1, sin más parámetros adapta a la ventana como esta.
msgMensaje=tk.Message(ventana, text=miTexto).pack()

ventana.mainloop()
```

Este será el resultado:



Ejemplo 2:

```
# importamos tkinter
import tkinter as tk

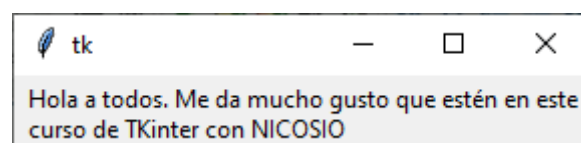
ventana = tk.Tk()

miTexto='Hola a todos. Me da mucho gusto que estén en este curso de
TKinter con NICOSIO'

# Ejemplo 2, podemos indicar el ancho que deseamos, vemos como se ajusta
el texto.
msgMensaje=tk.Message(ventana, text=miTexto, width=300).pack()

ventana.mainloop()
```

Este será el resultado:



Ejemplo 3:

```
# importamos tkinter
import tkinter as tk

ventana = tk.Tk()

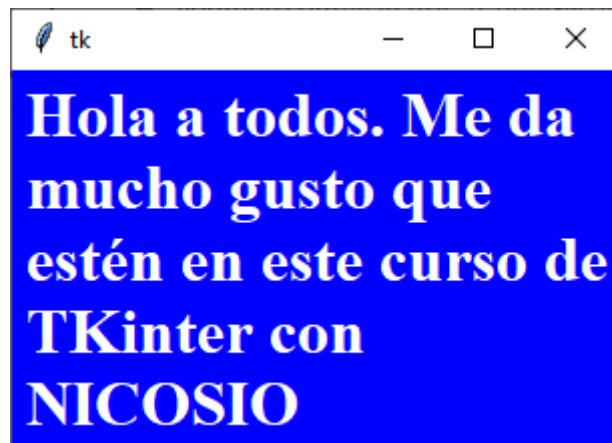
miTexto='Hola a todos. Me da mucho gusto que estén en este curso de
TKinter con NICOSIO'

# Ejemplo 3, podemos hacer una configuración paa el estilo

msgMensaje=tk.Message(ventana, text=miTexto, width=300)
msgMensaje.config(bg='blue', fg='white', font='times 24 bold')
msgMensaje.pack()

ventana.mainloop()
```

Este será el resultado:



Si vamos modificando el tamaño (width) el texto se ajustará.

5.- Button

```
import tkinter as tk

# Creamos una clase para nuestro ejemplo

class Test():
    def __init__(self):
        self.ventana = tk.Tk()

        # Tenemos que crear una variable de tipo StringVar para poder
        # cambiar el texto del un widget
        self.texto = tk.StringVar()
        # Le damos el valor inicial
        self.texto.set("Hola a todos")

        # Creamos una etiqueta
        # Con textVariable indicamos cual es la variable que tendrá el
        texto
        self.lblMiEtiqueta = tk.Label(self.ventana,
textvariable=self.texto, font='arial 24').pack()

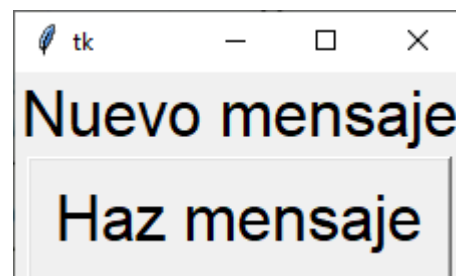
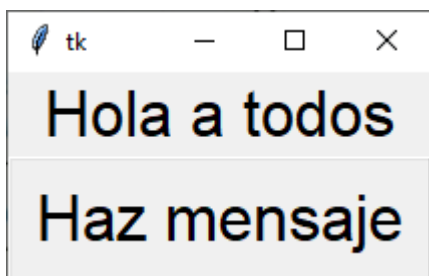
        # Creamos el botón
        self.btnMiBoton = tk.Button(self.ventana, text="Haz mensaje",
font='arial 24', command=self.chageText).pack()

        self.ventana.mainloop()

        # Este método es el handler
    def chageText(self):
        self.texto.set("Nuevo mensaje")

app=Test()
```

Este será el resultado antes y después de hacer clic en el botón.



6.- Radio Button

```
"""
Algunos widgets se conectan a las variables por medio de diferentes
opciones
variable
textvariable
onvalue
offvalue
Para esto necesitamos variables que sean subclases de variable, la cual
se define en
Tkinter
    StringVar
    IntVar
    DoubleVar
    BooleanVar
"""
import tkinter as tk

def Mostrar():
    if seleccion.get()==1:
        mensaje="Has seleccionado Python"
    if seleccion.get()==2:
        mensaje="Has seleccionado C#"
    if seleccion.get()==3:
        mensaje="Has seleccionado Java"

    lblMensaje.config(text=mensaje)

ventana = tk.Tk()
seleccion = tk.IntVar()

rbtnPython = tk.Radiobutton(ventana, text="Python", variable=seleccion,
value=1, command=Mostrar).pack(anchor=tk.W)

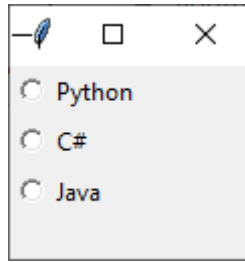
rbtnCSharp = tk.Radiobutton(ventana, text="C#", variable=seleccion,
value=2, command=Mostrar).pack(anchor=tk.W)

rbtnJava = tk.Radiobutton(ventana, text="Java", variable=seleccion,
value=3, command=Mostrar).pack(anchor=tk.W)

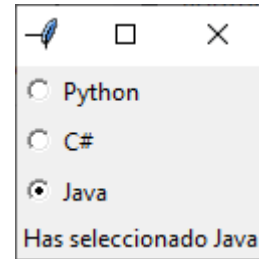
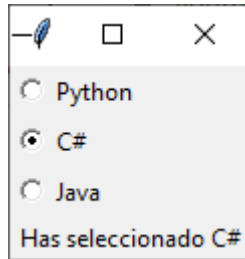
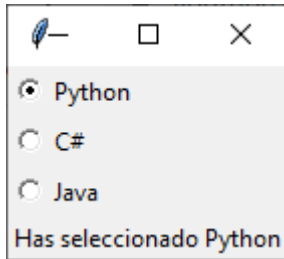
lblMensaje = tk.Label(ventana)
lblMensaje.pack()

ventana.mainloop()
```

Este será el resultado:



Según la opción que seleccionemos:



7.- CheckButton

```
import tkinter as tk

def mostrar():
    mensaje= ' '
    if(intchk1.get()==1):
        mensaje=mensaje+ ' Manzana'
    if(intchk2.get()==1):
        mensaje=mensaje+ ' Pera'
    else:
        mensaje=mensaje+ ' no olvide comprar pera'

    lblMensaje.config(text=mensaje)

ventana=tk.Tk()

# Estas variables las usaremos para conocer el estado de los checkbox
intchk1 = tk.IntVar()
intchk2 = tk.IntVar()

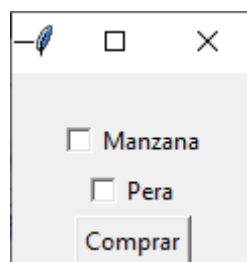
# Etiqueta para mostrar mensaje
lblMensaje = tk.Label(ventana)
lblMensaje.pack()

chk1 = tk.Checkbutton(ventana, text='Manzana', variable=intchk1).pack()
chk2 = tk.Checkbutton(ventana, text='Pera', variable=intchk2).pack()

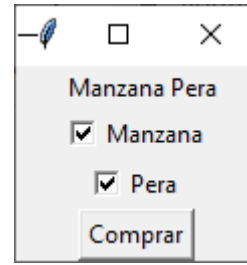
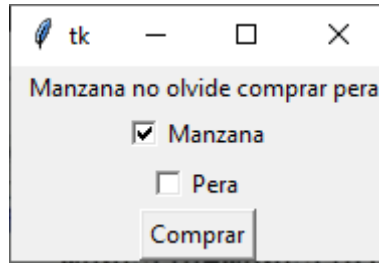
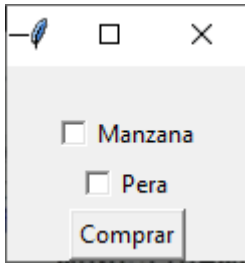
# Botón para llevar a cabo la acción de mostrar
btnMostrar = tk.Button(ventana, text='Comprar', command=mostrar).pack()

ventana.mainloop()
```

Este será el resultado:



Posibles respuestas:



8.- Scale

```
import tkinter as tk

# La función necesita un parámetro pues scale le manda su valor.
def Sumar(n):
    r=vertical.get()+horizontal.get()
    lblMensaje.config(text=str(r)+' --- '+str(n))

ventana = tk.Tk()

# Creamos los slides.
# La variable con la que lo controlaremos
vertical=tk.IntVar()

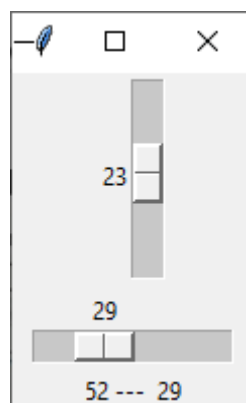
# Creamos un slider vertical con rango de 0 a 50 y empezando por 25
sliderv=tk.Scale(ventana,from_=0, to=50, variable=vertical,
command=Sumar).pack()
vertical.set(25)

# Ahora hacemos el slider horizontal
horizontal=tk.IntVar()
sliderh=tk.Scale(ventana,from_=0, to=100, orient=tk.HORIZONTAL,
variable=horizontal, command=Sumar).pack()

# Etiqueta para mostrar mensaje
lblMensaje = tk.Label(ventana)
lblMensaje.pack()

ventana.mainloop()
```

Este será el resultado:



9.- Entry

```
import tkinter as tk

def Calcular():
    dias=int(edad.get())*365
    mensaje=nombre.get()+' has vivido '+str(dias)+' dias'
    lblMensaje.config(text=mensaje)

ventana = tk.Tk()

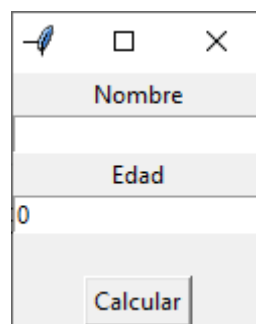
# Creamos una etiqueta para indicar que ahí va el nombre
tk.Label(ventana, text='Nombre').pack()
# Colocamos un Entry para poder tener la entrada usuario
nombre=tk.StringVar()
nomb=tk.Entry(ventana, textvariable=nombre).pack()

tk.Label(ventana, text='Edad').pack()
edad=tk.IntVar()
ed=tk.Entry(ventana, textvariable=edad).pack()

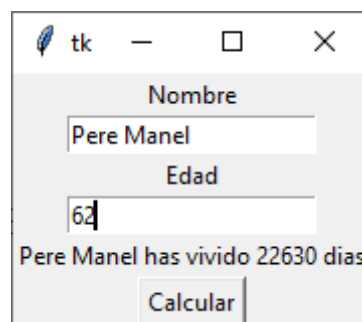
# Etiqueta para mostrar el mensaje
lblMensaje=tk.Label(ventana)
lblMensaje.pack()

btnCalcula=tk.Button(ventana, text='Calcular', command=Calcular).pack()
ventana.mainloop()
```

Este será el resultado:



Vamos a introducir un nombre y una edad, seguido de hacer clic en Calcular.



10.- Canvas

```
# Canvas permite llevar a cabo operaciones básicas de graficación
# Importamos Tkinter y también Canvas
import tkinter as tk
from tkinter import Canvas

# El objeto que representa a la ventana
ventana=tk.Tk()

# Creamos el objeto Canvas y pasamos como parámetros su tamaño.
canvas=Canvas(ventana, width=640, height=480)
canvas.pack()

# Creamos una línea que va de (10,20) a (100, 150) de color rojo
canvas.create_line(10,20,100,150, fill='#FF0000', width=3)

canvas.create_rectangle(150,120,250,180, outline='#00FF00', width=5)
canvas.create_rectangle(250, 220, 350, 280, outline='#00AAAA',
fill='#0000FF', width=5)

canvas.create_arc(10,300,400,200, outline='#00AAAA', fill='#0000FF',
width=5)

# tk.PIESLICE, tk.CHORD, tk.ARC
canvas.create_arc(10, 300, 400, 200, star=45, extent=120, style=tk.CHORD)

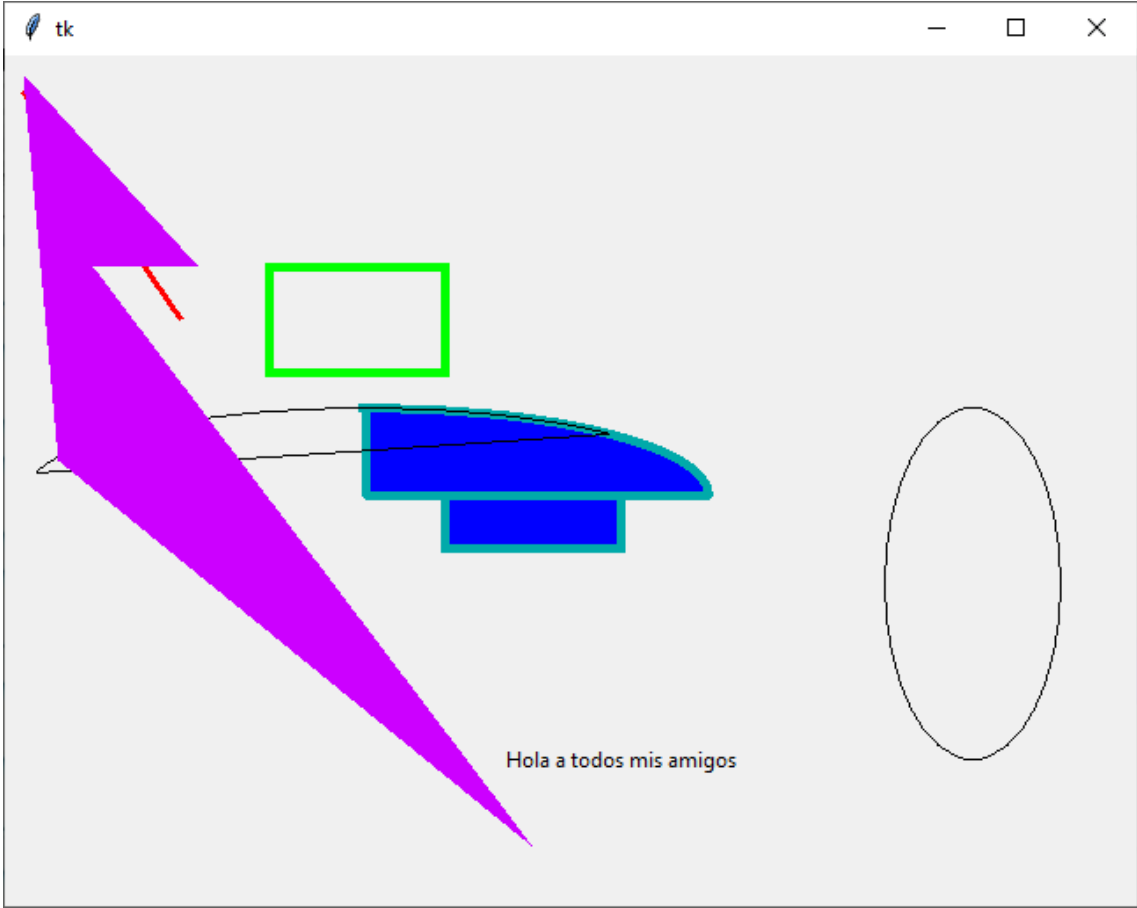
canvas.create_oval(500,200, 600, 400)

# En el polígono, vamos pasando los pares de coordenadas de cada punto
# que lo conforman
canvas.create_polygon(10, 10, 110, 120, 50, 120, 300, 450, 30, 230,
fill='#CC00FF')

canvas.create_text(350, 400,text='Hola a todos mis amigos')

ventana.mainloop()
```

Este será el resultado:



11.- Scrollbar

```
import tkinter as tk
```

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

```
# Creamos el scrool y el texto.
```

```
scroll = tk.Scrollbar(ventana)
```

```
texto = tk.Text(ventana, height=10,width=30 )
```

```
# Ahora colocamos el scrool y el texto
```

```
scroll.pack(side=tk.RIGHT, fill=tk.Y)
```

```
texto.pack(side=tk.LEFT, fill=tk.Y)
```

```
# Configuramos el widget
```

```
# Indicamos que modificará el texto en un scroll y invocando el modo
```

```
# yview, xview
```

```
scroll.config(command=texto.yview)
```

```
# Asociamos con el scroll par poder colocarlo en la posición invocando
```

```
# método set
```

```
texto.config(yscrollcommand=scroll.set)
```

```
mensaje="""¿Por qué esta mangifica tecnología científica,
```

```
que ahora trabajo y nos hace la vida más fácil
```

```
nos aporta tan poca felidad? La respuesta es esta,
```

```
simplemente: porque aún no hemos aprendido a usarlas con tino.
```

```
Nunca consideres el estudio como una obligación, sino como una
```

```
oportunidad para penetrar en el bello y maravilloso mundo del saber.
```

```
Hay una fueza motriz más poderosa que el vapor,
```

```
la electricidad y la energía atómica: la voluntad.
```

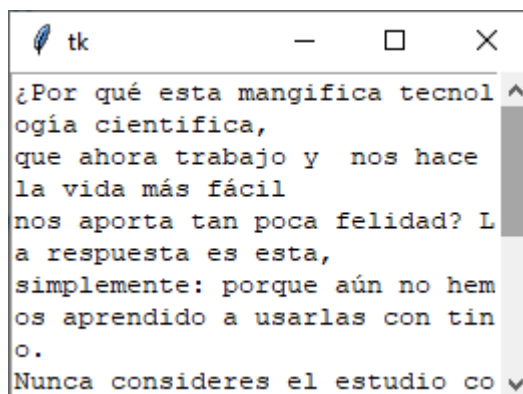
```
-Albert Eisntein"""
```

```
# Insertamos el mensaje al final del texto.
```

```
texto.insert(tk.END, mensaje)
```

```
ventana.mainloop()
```

Este será el resultado:



12.- MessageBox I

```
import tkinter as tk
from tkinter import messagebox

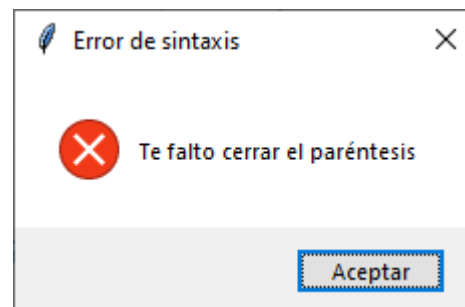
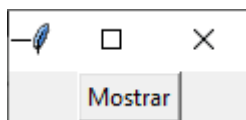
def mostrarBoton():
    # Para mostrar error
    messagebox.showerror('Error de sintaxis', 'Te faltó cerrar el
paréntesis')

ventana=tk.Tk()

boton=tk.Button(text='Mostrar', command=mostrarBoton).pack()

ventana.mainloop()
```

Este será el resultado:



```
import tkinter as tk
from tkinter import messagebox

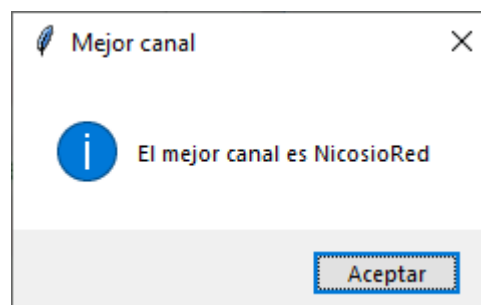
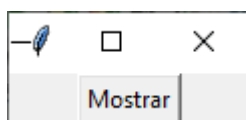
def mostrarBoton():
    # Para mostrar información
    messagebox.showinfo('Mejor canal', 'El mejor canal es NicosioRed')

ventana=tk.Tk()

boton=tk.Button(text='Mostrar', command=mostrarBoton).pack()

ventana.mainloop()
```

Este será el resultado:




```
import tkinter as tk
from tkinter import messagebox

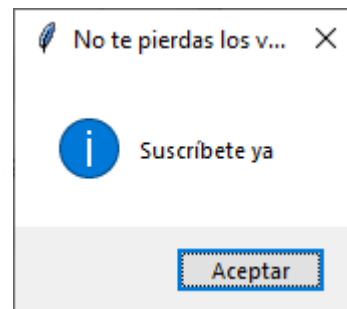
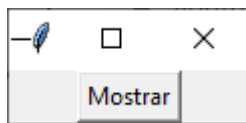
def mostrarBoton():
    # Para mostrar una advertencia
    messagebox.showinfo('No te pierdas los vídeos', 'Suscríbete ya')

ventana=tk.Tk()

boton=tk.Button(text='Mostrar', command=mostrarBoton).pack()

ventana.mainloop()
```

Este será el resultado:



13.- MessageBox II

```
import tkinter as tk
from tkinter import messagebox

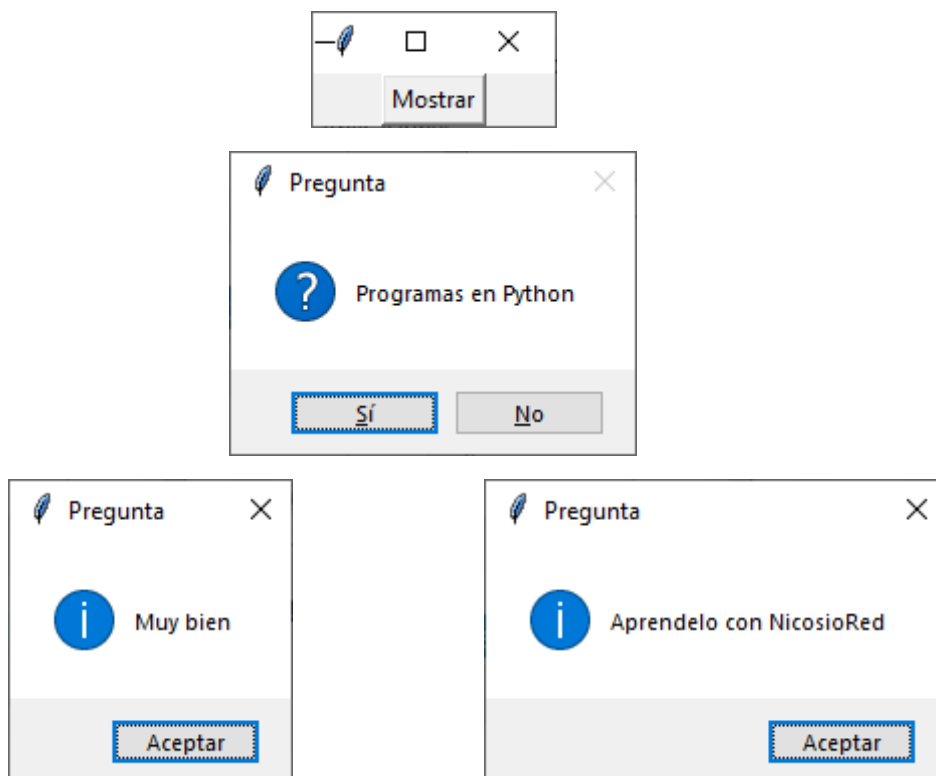
def mostrarBoton():
    # Para preguntar algo
    r=messagebox.askquestion('Pregunta', 'Programas en Python')
    if r=='yes':
        messagebox.showinfo('Pregunta', 'Muy bien')
    else:
        messagebox.showinfo('Pregunta', 'Aprendelo con NicosioRed')

ventana=tk.Tk()

boton=tk.Button(text='Mostrar', command=mostrarBoton).pack()

ventana.mainloop()
```

Este será el resultado:



```
import tkinter as tk
from tkinter import messagebox

def mostrarBoton():
    # Para preguntar si esta hace algo o no
    r=messagebox.askokcancel('Ejecutar', 'Ejecutamos programa')
```

```

if r==True:
    messagebox.showinfo('Programa', 'Se ejecuta el programa')
else:
    messagebox.showinfo('Programa', 'No se ejecuta nada')

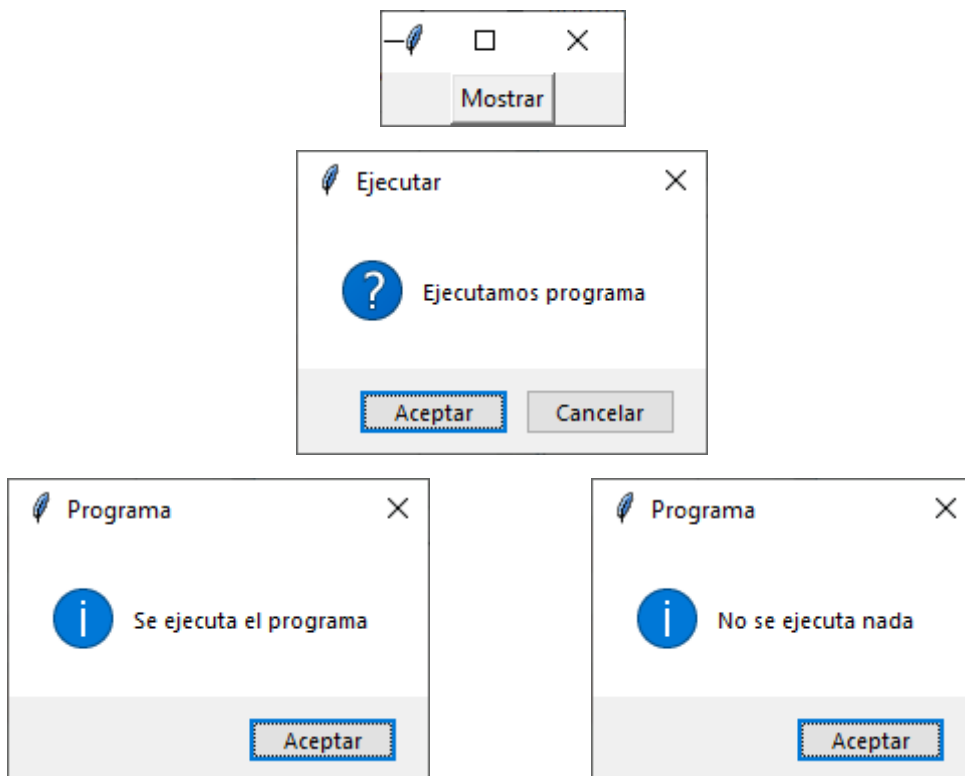
```

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

```
boton=tk.Button(text='Mostrar', command=mostrarBoton).pack()
```

```
ventana.mainloop()
```

Este será el resultado:



```

import tkinter as tk
from tkinter import messagebox

```

```

def mostrarBoton():
    # Para preguntar con retry o cancel
    r=messagebox.askretrycancel('Programa', 'Ejecutamos de nuevo el programa')
    if r==True:
        messagebox.showinfo('Programa', 'Se ejecuta de nuevo')
    else:
        messagebox.showinfo('Programa', 'Ya no se ejecuta de nuevo')

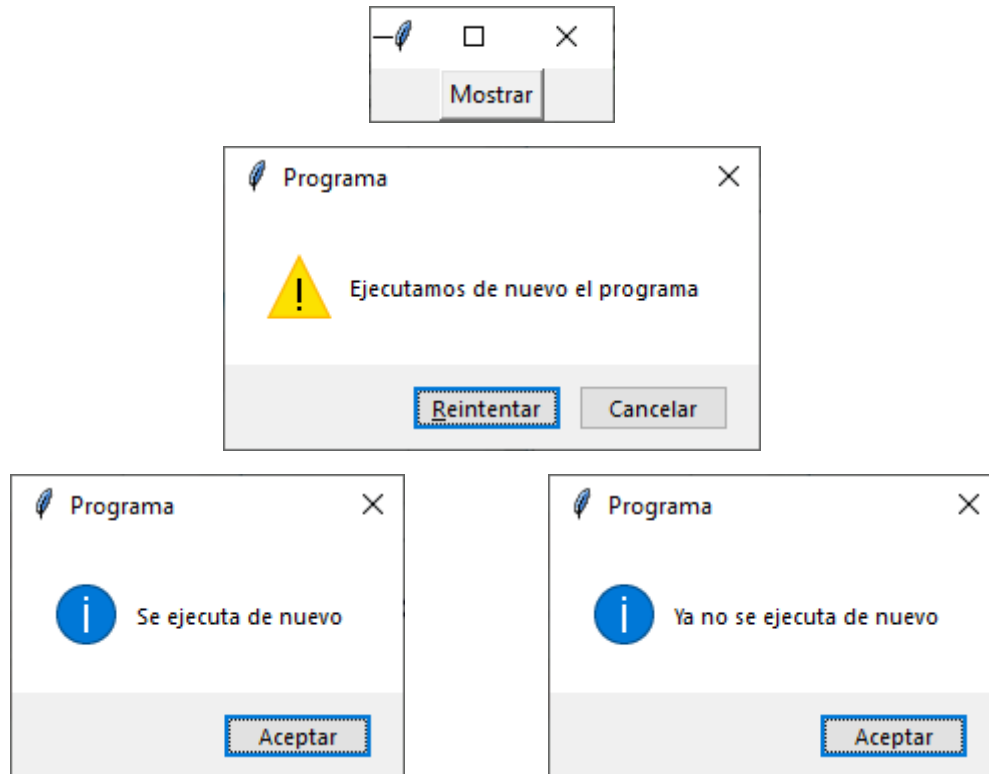
```

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

```
boton=tk.Button(text='Mostrar', command=mostrarBoton).pack()
```

```
ventana.mainloop()
```

Este será el resultado:



```
import tkinter as tk
from tkinter import messagebox

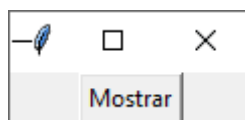
def mostrarBoton():
    # Para preguntar con yes y no
    r=messagebox.askyesno('Programa', 'Ejecutamos de nuevo el programa')
    if r==True:
        messagebox.showinfo('Programa', 'Se ejecuta de nuevo')
    else:
        messagebox.showinfo('Programa', 'Ya no se ejecuta de nuevo')
```

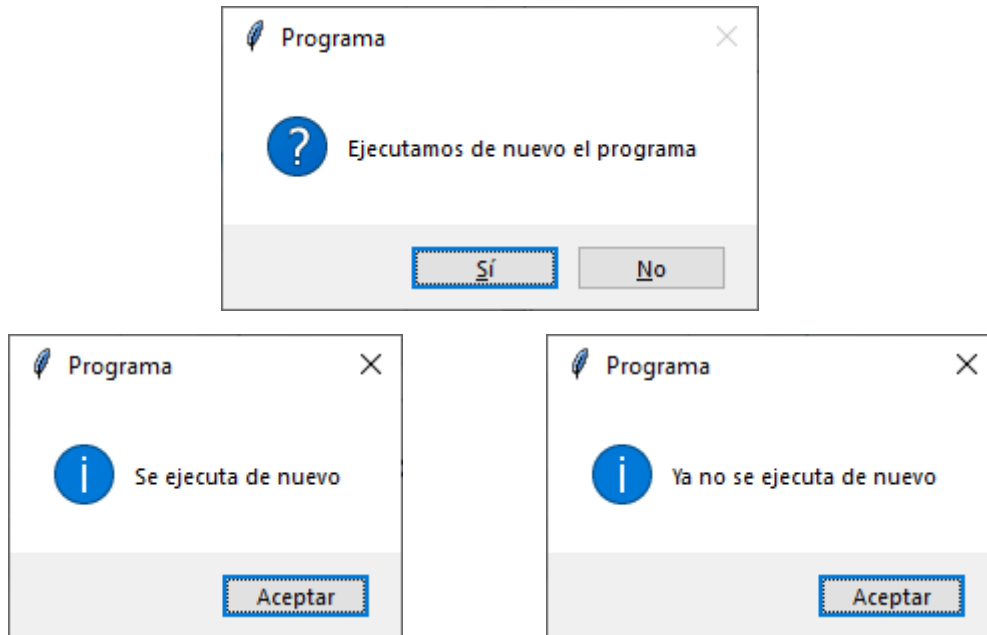
```
ventana=tk.Tk()
```

```
boton=tk.Button(text='Mostrar', command=mostrarBoton).pack()
```

```
ventana.mainloop()
```

este será el resultado:





```
import tkinter as tk
from tkinter import messagebox

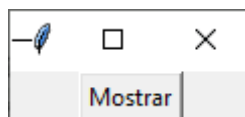
def mostrarBoton():
    # Para preguntar con yes, no y cancel
    r=messagebox.askyesnocancel('Pregunta', 'Ejecutamos de nuevo el programa?')
    if r==True:
        messagebox.showinfo('Programa', 'Se ejecuta de nuevo')
    if r==False:
        messagebox.showinfo('Programa', 'Ya no se ejecuta de nuevo')
    if r==None:
        messagebox.showinfo('Programa', 'Cancelaste la ejecución')

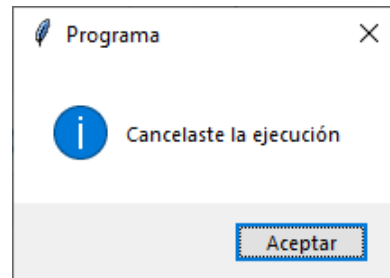
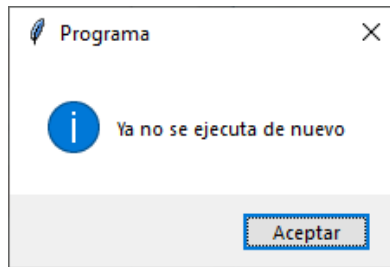
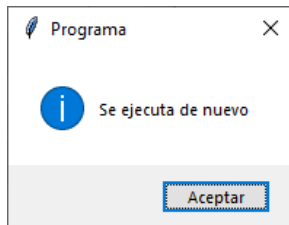
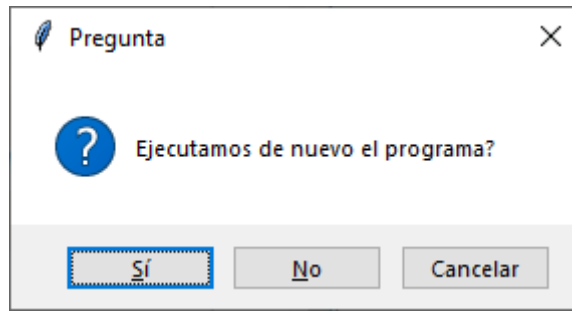
ventana=tk.Tk()

boton=tk.Button(text='Mostrar', command=mostrarBoton).pack()

ventana.mainloop()
```

Este será el resultado:





14.- Diálogos comunes

```
import tkinter as tk
from tkinter import filedialog as fd
from tkinter.colorchooser import askcolor
```

```
def MostrarDialogo():
    ruta= fd.askopenfilename()
    print(ruta)
```

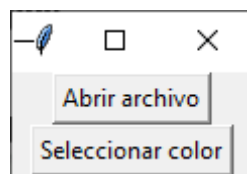
```
def SeleccionaColor():
    colorS=askcolor(color='#FFFFFF', title='Selecciona un color ')
    print(colorS)
```

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

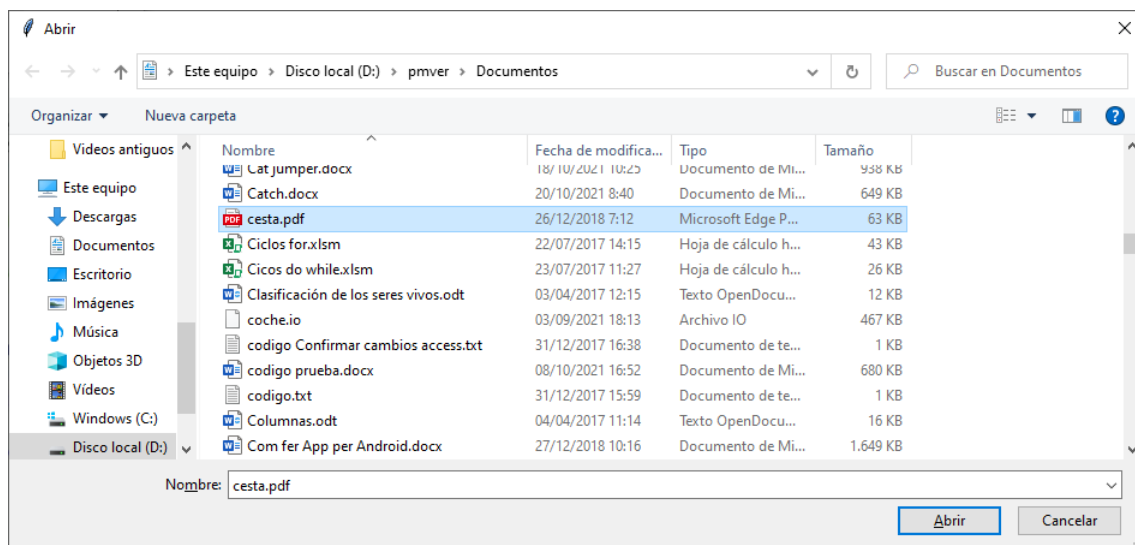
```
btnAbrir=tk.Button(text='Abrir archivo', command=MostrarDialogo).pack()
btnColor=tk.Button(text='Seleccionar color',
command=SeleccionaColor).pack()
```

```
ventana.mainloop()
```

este será el resultado:



Vamos a seleccionar 'Abrir archivo':

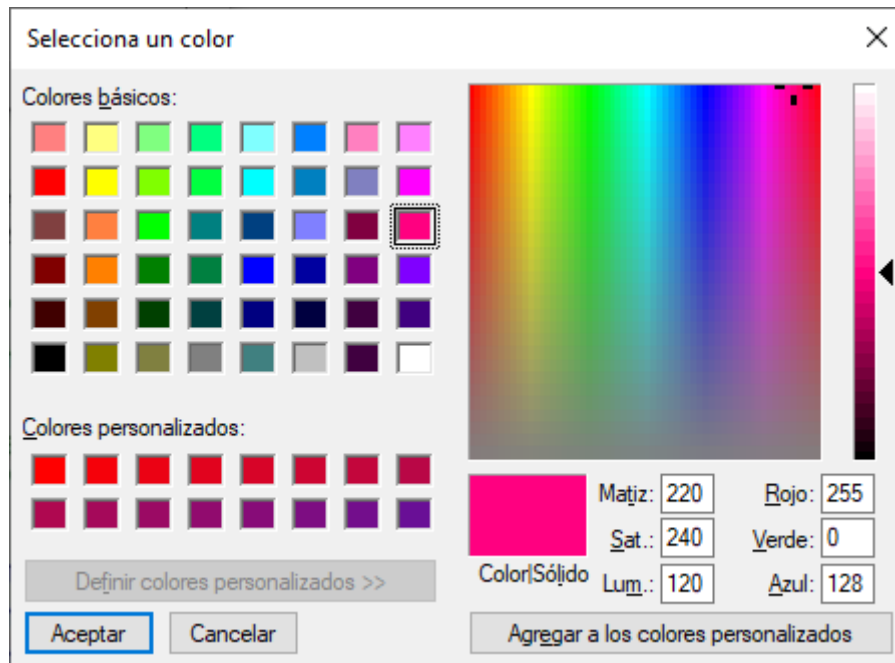


Seleccionamos el archivo seguido del botón Abrir.

Nos muestra por consola la ubicación del archivo.

D:/pmver/Documents/cesta.pdf

Ahora vamos a seleccionar color:



Selecciona el color que tu desees seguido del botón Aceptar.

`((255, 0, 128), '#ff0080')`

Retorna el color en modo RGB y hexadecimal.

15.- Pack

```
# Hemos estado usando pack, pero sin entrar en detalle.  
# No solo nos permite colocar los elementos en la ventana,  
# también darle orden.  
# Pack es un Layout manager junto con grid y place.  
# Lo usamos para indicar la posición de un widget con relación  
# a otros.  
# Es sencillo, pero limitado.
```

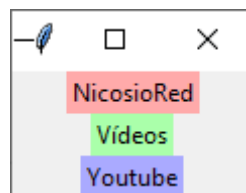
```
import tkinter as tk  
ventana=tk.Tk()
```

```
# Al colocar pack solamente se loca el widget en el tamaño que  
# necesita centrado y uno abajo de otro.
```

```
l1 = tk.Label(ventana, text='NicosioRed', bg='#FFAAAA')  
l1.pack()  
l2 = tk.Label(ventana, text='Vídeos', bg='#AAFFAA')  
l2.pack()  
l3 = tk.Label(ventana, text='Youtube', bg='#AAAAFF')  
l3.pack()
```

```
ventana.mainloop()
```

Este será el resultado:

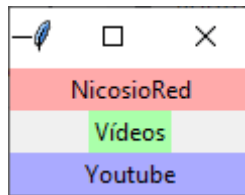


```
import tkinter as tk  
ventana=tk.Tk()
```

```
# Con fill hacemos que el widget sea tan ancho (tk.X) como el  
# widget que lo contiene, en este caso la ventana  
l1 = tk.Label(ventana, text='NicosioRed', bg='#FFAAAA')  
l1.pack(fill=tk.X)  
l2 = tk.Label(ventana, text='Vídeos', bg='#AAFFAA')  
l2.pack()  
l3 = tk.Label(ventana, text='Youtube', bg='#AAAAFF')  
l3.pack(fill=tk.X)
```

```
ventana.mainloop()
```

Este será el resultado:

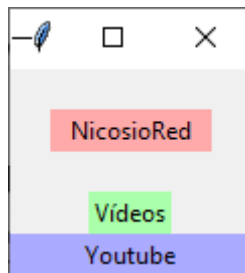


```
import tkinter as tk
ventana=tk.Tk()

# Con pad creamos un "marco" alrededor del widget
l1 = tk.Label(ventana, text='NicosioRed', bg='#FFAAAA')
l1.pack(fill=tk.X, padx=20, pady=20)
l2 = tk.Label(ventana, text='VÍdeos', bg='#AAFFAA')
l2.pack()
l3 = tk.Label(ventana, text='Youtube', bg='#AAAAFF')
l3.pack(fill=tk.X)

ventana.mainloop()
```

Este será el resultado:

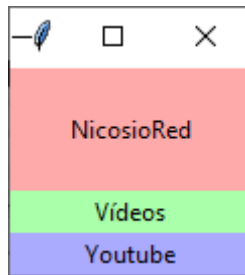


```
import tkinter as tk
ventana=tk.Tk()

# Con ipad creamos un "marco" interno en el widget
l1 = tk.Label(ventana, text='NicosioRed', bg='#FFAAAA')
l1.pack(fill=tk.X, ipadx=20, ipady=20)
l2 = tk.Label(ventana, text='VÍdeos', bg='#AAFFAA')
l2.pack(fill=tk.X)
l3 = tk.Label(ventana, text='Youtube', bg='#AAAAFF')
l3.pack(fill=tk.X)

ventana.mainloop()
```

Este será el resultado:

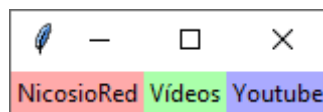


```
import tkinter as tk
ventana=tk.Tk()

# Para poner los widgets al lado en lugar de arriba o abajo
# usamos side, e indicamos la posición relativa tk.LEFT, tk.RIGHT
l1 = tk.Label(ventana, text='NicosioRed', bg='#FFAAAA')
l1.pack(fill=tk.X, side=tk.LEFT)
l2 = tk.Label(ventana, text='VÍdeos', bg='#AAFFAA')
l2.pack(fill=tk.X, side=tk.LEFT)
l3 = tk.Label(ventana, text='Youtube', bg='#AAAAFF')
l3.pack(fill=tk.X, side=tk.LEFT)

ventana.mainloop()
```

Este será el resultado:

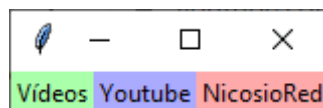


```
import tkinter as tk
ventana=tk.Tk()

l1 = tk.Label(ventana, text='NicosioRed', bg='#FFAAAA')
l1.pack(fill=tk.X, side=tk.RIGHT)
l2 = tk.Label(ventana, text='VÍdeos', bg='#AAFFAA')
l2.pack(fill=tk.X, side=tk.LEFT)
l3 = tk.Label(ventana, text='Youtube', bg='#AAAAFF')
l3.pack(fill=tk.X)

ventana.mainloop()
```

Este será el resultado:

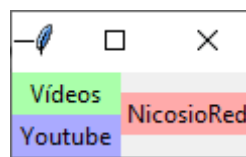


```
import tkinter as tk
ventana=tk.Tk()

l1 = tk.Label(ventana, text='NicosioRed', bg='#FFAAAA')
l1.pack(fill=tk.X, side=tk.RIGHT)
l2 = tk.Label(ventana, text='VÍdeos', bg='#AAFFAA')
l2.pack(fill=tk.X)
l3 = tk.Label(ventana, text='Youtube', bg='AAAAFF')
l3.pack(fill=tk.X)

ventana.mainloop()
```

Este será el resultado:



16.- Place

```
import tkinter as tk
ventana=tk.Tk()
# Definimos la geometría de widgets que es nuestra ventana.
ventana.geometry('640x480')

# Colocamos una etiqueta
l1=tk.Label(ventana, text='NicosioRed', bg='#FFAAAA')

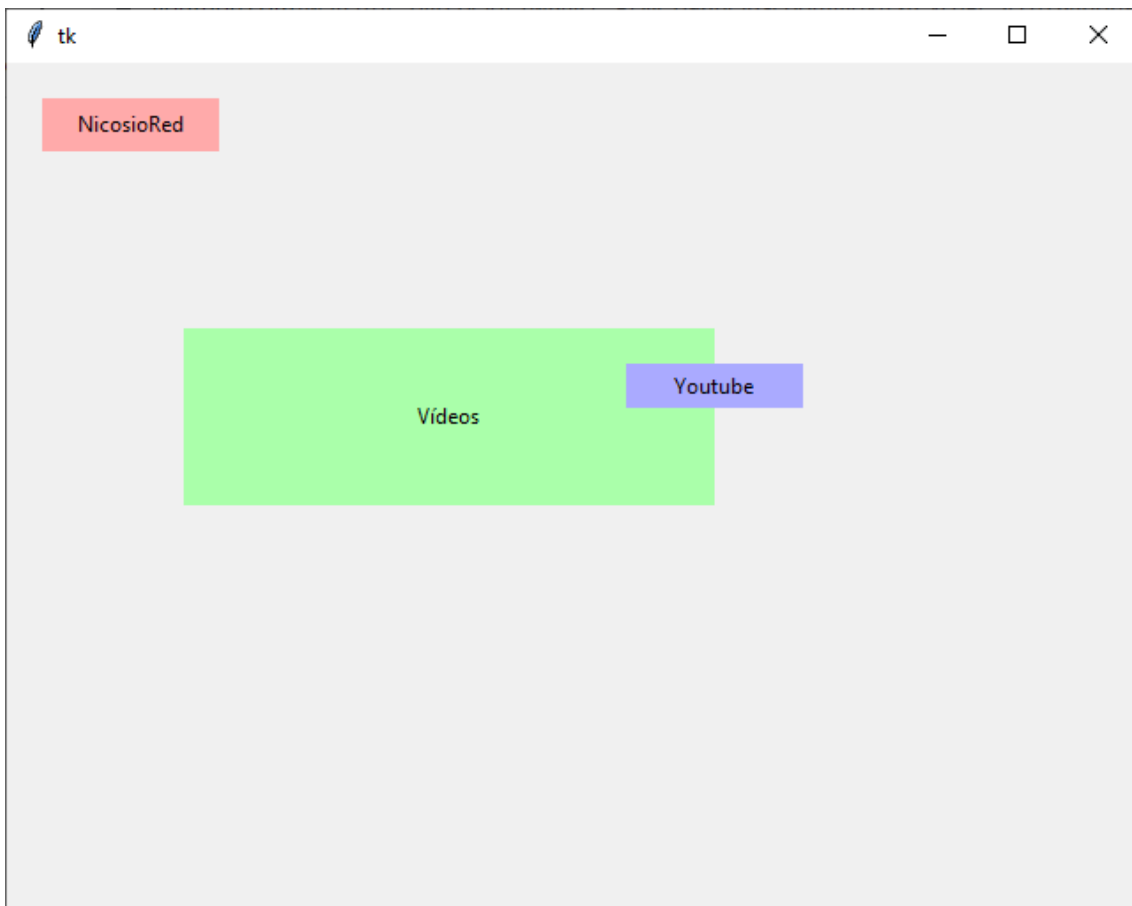
# Indicamos las cooredenadas donde deseamos que aparezca el widget
# Luego el ancho y el alto del area que ocupará en la ventana
l1.place(x=20, y=20, width=100, height=30)

l2=tk.Label(ventana, text='VÍdeos', bg='#AAFFAA')
l2.place(x=100, y=150, width=300, height=100)

l3=tk.Label(ventana, text='Youtube', bg='#AAAAFF')
l3.place(x=350, y=170, width=100, height=25)

ventana.mainloop()
```

Este será el resultado:



Es importante el orden, ahora vamos a cambiar el orden.

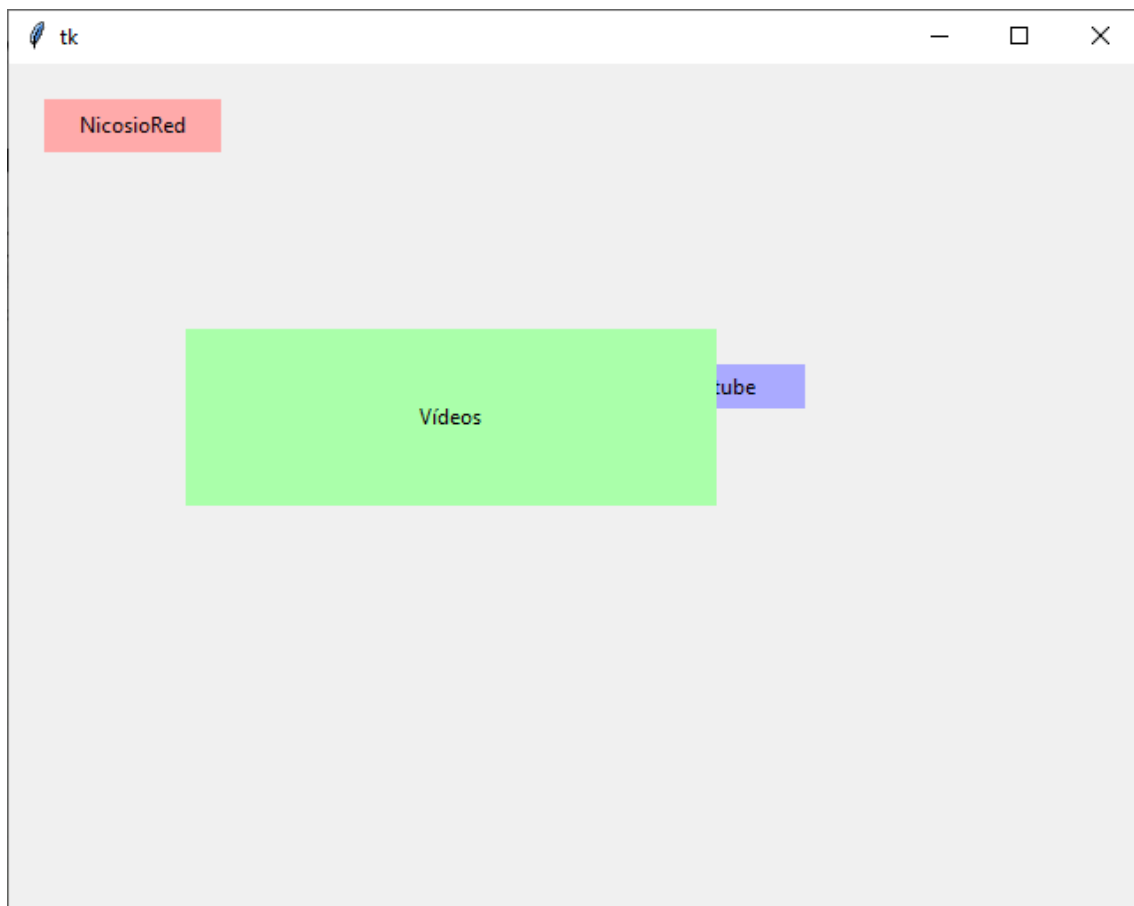
```
# Colocamos una etiqueta
l1=tk.Label(ventana, text='NicosioRed', bg='#FFAAAA')

# Indicamos las cooredenadas donde deseamos que aparezca el widget
# Luego el ancho y el alto del area que ocupará en la ventana
l1.place(x=20, y=20, width=100, height=30)

l3=tk.Label(ventana, text='Youtube', bg='#AAAAFF')
l3.place(x=350, y=170, width=100, height=25)

l2=tk.Label(ventana, text='VÍdeos', bg='#AAFFAA')
l2.place(x=100, y=150, width=300, height=100)
```

Este será el resultado:



17.- Menu

```
import tkinter as tk
from tkinter import Menu

ventana=tk.Tk()

def Menu1():
    print('Se ejecuta el menu1')

def Menu2():
    print('Se ejecuta el menu2')

def Salir():
    ventana.destroy()

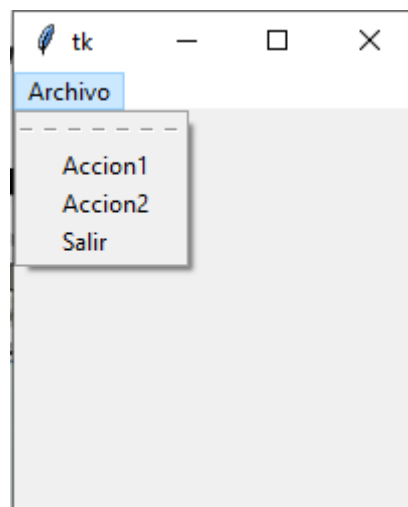
# Creamos el menu de la ventana
miMenu=Menu(ventana)
ventana.config(menu=miMenu)

# Le ponemos una primera sección y lo asignamos al menú de la ventana
menuPrincipal=Menu(miMenu)
miMenu.add_cascade(label='Archivo', menu=menuPrincipal)

# Creamos los elementos para ese menu
menuPrincipal.add_command(label='Accion1', command=Menu1)
menuPrincipal.add_command(label='Accion2', command=Menu2)
menuPrincipal.add_command(label='Salir', command=Salir)

ventana.mainloop()
```

Este será el resultado:



Si seleccionamos Acción1 en la consola aparecerá el siguiente mensaje:

Se ejecuta el menu1

Si seleccionamos Accion2 en la consola aparecerá el siguiente mensaje:

Se ejecuta el menu2

Si seleccionamos Salir se cerrará al ventana.