

PROGRAMACION 2

Interfaz Gráfica



Esc. Sup. De Comercio N° 49 Cap. Gral. J.J. de Urquiza

Técnico Superior en Desarrollo de Software



Label

La clase anterior habíamos visto como configurar la ventana raiz y la de frame y habíamos comenzado a ver **Label**, en esta clase vamos a continuar y avanzar con otros widgets,

```
from tkinter import *
raiz=Tk()

miframe=Frame(raiz,width=300,height=200)
miframe.pack()

milabel=Label(miframe,text="Hola, Bienvenidos")
milabel.place(x=100, y=80)

raiz.mainloop()
```

Lo último que vimos fue el método place(x,y).En nuestro ejemplo habíamos puesto x=100 y=80 pero, dónde se posicionará? La coordenada x es la distancia que hay entre el borde izquierdo y el texto, en nuestro caso será de 100 pixeles y la otro coordenada es la distancia que hay desde el borde superior y el texto.

Dentro del método Label también podemos utilizar todas las opciones de configuración que vimos para frame y la ventana raiz , color de fondo, tipo de letra,etc.

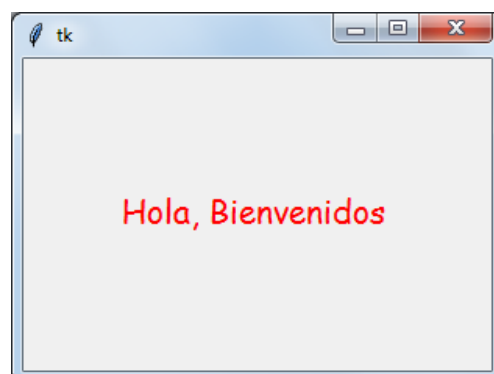
```
from tkinter import *
raiz=Tk()

miframe=Frame(raiz,width=300,height=200)
miframe.pack()

milabel=Label(miframe,text="Hola, Bienvenidos", fg="red", font=("Comic Sans MS",16))
milabel.place(x=60, y=80)

raiz.mainloop()
|
```

En el ejemplo foreground color y font

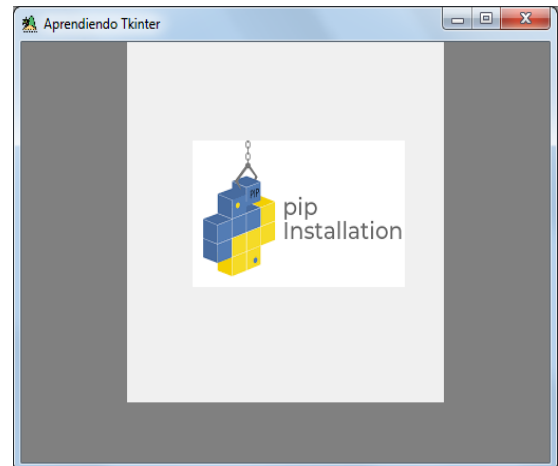


Si en lugar de texto queremos mostrar imágenes

Si en lugar de texto queremos mostrar una imagen usaremos la opción image como en el ejemplo

Usamos el método **PhotoImage** y asignamos a una variable el archivo donde se encuentra la imagen, si la misma no se encuentra en la misma carpeta que el archivo con el código escribimos la ruta. Luego en Label ponemos la opción imagen y le asignamos esta variable

```
File Edit Format Run Options Window Help
from tkinter import *
raiz=Tk()
raiz.title("Aprendiendo Tkinter")
raiz.resizable(True,True)
raiz.iconbitmap("palmera.ico")
raiz.geometry("500x350")
raiz.config(bg="grey")
miframe=Frame(raiz,width=300,height=300)
miframe.pack()
miimagen=PhotoImage(file="pip.png")
milabel=Label(miframe,image=miimagen)
milabel.place(x=60, y=80)
|
raiz.mainloop()
```

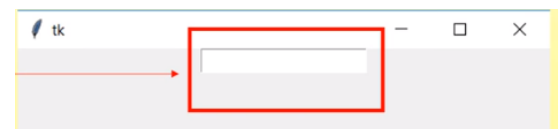


Entry

Utilizados para introducir datos. Los cuadros de textos se configuran de manera muy similar a las Labels.

```
entry.py - C:/Users/Master/AppData/Local/Programs/Py...
File Edit Format Run Options Window Help
from tkinter import *
raiz=Tk()
miFrame=Frame(raiz,width=900, height=600)
miFrame.pack()
cuadroTexto=Entry(miFrame)
cuadroTexto.place(x=100, y=100)
|
raiz.mainloop()
Ln: 11 Col: 0
```

Este código ubicaría el **Entry** dentro de la ventana raiz y dentro del Frame, le podemos



Si queremos ponerle una etiqueta para indicar que datos se debe ingresar por ejemplo *Nombre:* podríamos poner una Label y ubicarla a la izquierda del Entry. El problema es saber qué coordenadas x, y debo darle al place() ya que estas toman como referencia el borde superior y el izquierdo.

```
from tkinter import *

raiz= Tk()

miFrame=Frame(raiz, width=1200, height=600)
miFrame.pack()

cuadroTexto=Entry(miFrame)
cuadroTexto.place(x=100, y=100)

nombreLabel=Label(miFrame, text="Nombre:")
nombreLabel.place(x=100, y=100)

raiz.mainloop()
```



Si ejecutamos vamos a ver que ubica el cuadro de texto a la derecha del Label.

¿Por qué?

Si usamos la misma coordenada de x la primera "empuja" a la siguiente hacia la derecha.

Pero esto no es aconsejable realizarlo así, ya que si por ejemplo ponemos:

```
nombreLabel.place(x=120, y=100)
```

Nos aparecerá algo así.

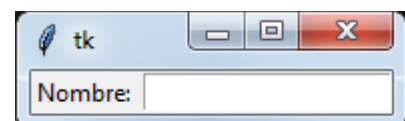
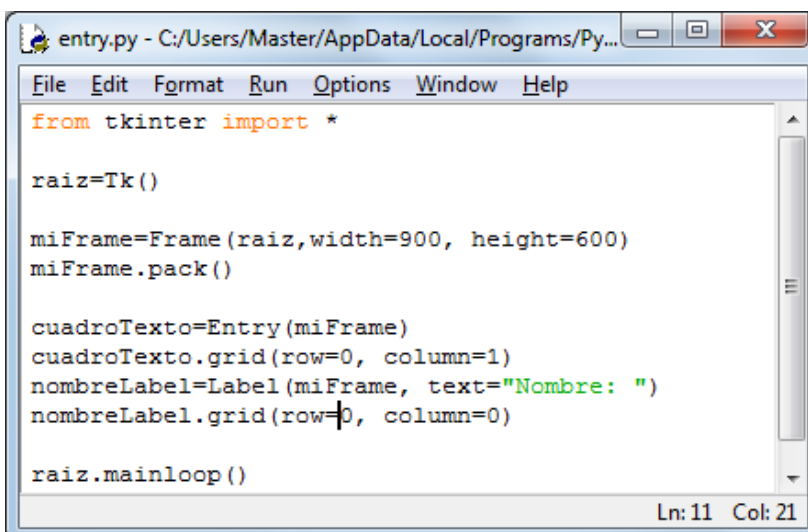


La forma para ubicar esto correctamente es trabajar con un método llamado **grid()**

Grid()

Este método construye una grilla o tabla con tantas filas y columnas como nosotros queramos. Y de esta forma poder colocar todos los elementos en las ubicaciones elegidas.

En el código sustituimos el place por el grid. Los parámetros de este método son **column** y **row**, que empiezan a contabilizar desde 0 (cero)



Como vemos ocurre lo mismo que con pack() no nos respeta las dimensiones que le dimos a la ventana.

En cuanto a la alineación por defecto aparecerá el texto centrado. Para poder alinear a utilizamos la propiedad **sticky**. Utiliza los puntos cardinales para colocar adecuadamente los elementos.

```
nombreLabel=Label(miFrame, text="Nombre: ")
nombreLabel.grid(row=0, column=0, sticky="e")
```

Si queremos que el texto "Nombre:" esté alineado a la derecha poner "e" de east.

Y si queremos darle **padding** (distancias entre elementos) utilizamos la opción **padx** es para configurar padding en el eje de las x y **pady** para configurarlo en el eje de las y

```
nombreLabel=Label(miFrame, text="Nombre: ")
nombreLabel.grid(row=0, column=0, sticky="e", padx=10, pady=10)
```

```
cuadroNombre=Entry(miFrame)
cuadroNombre.grid(row=0, column=1, padx=10, pady=10)
cuadroNombre.config(fg="red", justify)
cuadroApellido=Entry(miFrame)
```

También podemos utilizar estas opciones para el Entry, cambiar el color o que al escribir esté justificado.

```
cuadroPass.config(show="*")
```

Otra opción que nos puede ser útil por ejemplo si queremos ingresar una contraseña es show para que en lugar de mostrarse lo que estoy escribiendo se vea un

carácter que puede ser cualquiera, en este caso usamos el *

Practicamos

Realizar una ventana que ingrese los siguientes datos:

- Apellido y Nombre
- Domicilio
- Teléfono
- DNI