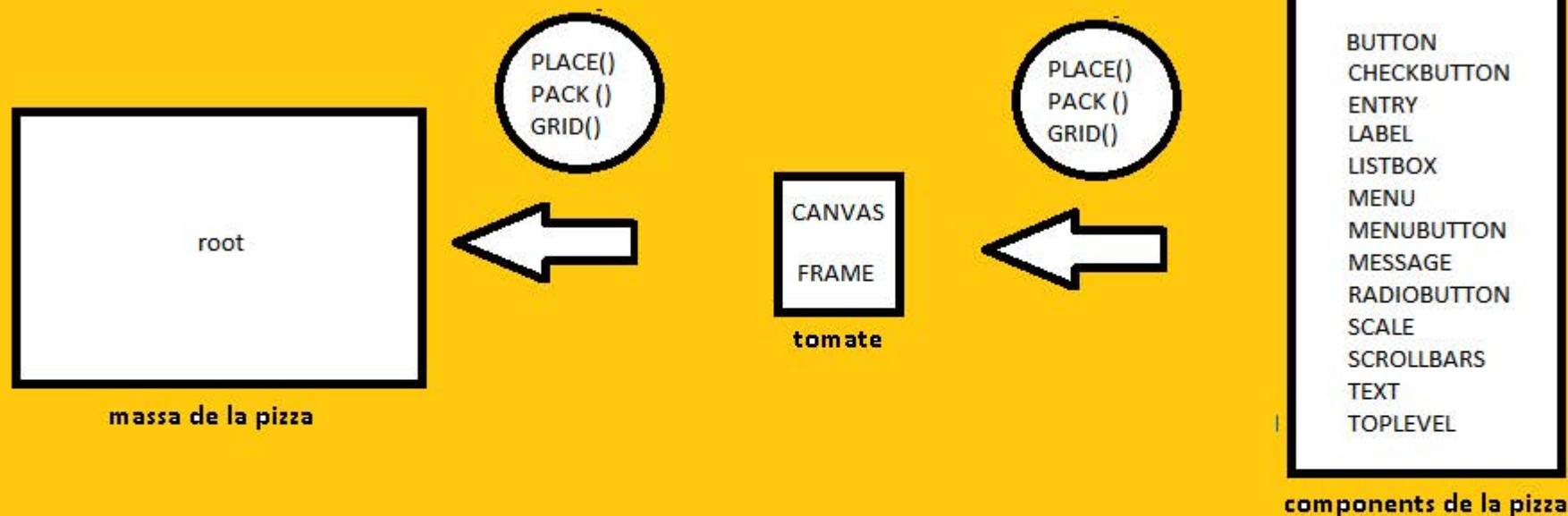


**INTERFÍCIES D'USUARI AMB TKINTER
PROGRAMACIÓ EN PYTHON
PRIMAVERA 2021**

JOAN MASDEMONT FONTÀS

INTERFÍCIES D'USUARI



ETAPES DE CONSTRUCCIO DE L'INTERFICIE D'USUARI

a) Inicialitzar root.

Massa de la pizza

b) Muntar un Canvas o Frame damunt de root.

Capa de Tomate de la pizza

c) Organitzar els altres widgets damunt de canvas o Frame
Ingredients

d) Lògica del programa.
Interacció entre Widgets

Aprofitament de la pizza

CODI INICIALITZADOR DE LA MASSA DE LA PIZZA.

DEFINICIÓ DE ROOT

Aquest és el codi inicialitzador de root estàndard. **A python 2 el tkinter s'importa en majúscules Tkinter**

```
import tkinter as tk
#
# FUNCIONS EVENTS
# RELACIONATS AMB ELS WIDGETS
#
root = tk.Tk()
amplada = root.winfo_screenwidth()
alcada = root.winfo_screenheight()
strink = '{}x{}'.format(amplada , alcada)
root.geometry(strink)
root.attributes('-fullscreen', True)
#
# ESPAI PER DEFINICIO DE WIDGETS
# CRIDA A FUNCIONS DE CADA WIDGET I EVENTS
# LÒGICA DEL PROGRAMA
# CRIDA A ALTRES FUNCIONS DE LA LÒGICA DEL PROGRAMA
#
root.mainloop()
```

AFEGIR UN CANVAS O FRAME DAMUNT DE ROOT.CAPA DE TOMATE

Damunt de root muntarem un canvas a on Frame el codi bàsic per fer-ho és el de sota.

```
c = tk.Canvas(root, bg ='orange')  
c.place(x = 0, y = 0, width = amplada, height = alcada)
```

**El codi amb root inicialitzat
i el Canvas o Frame.**

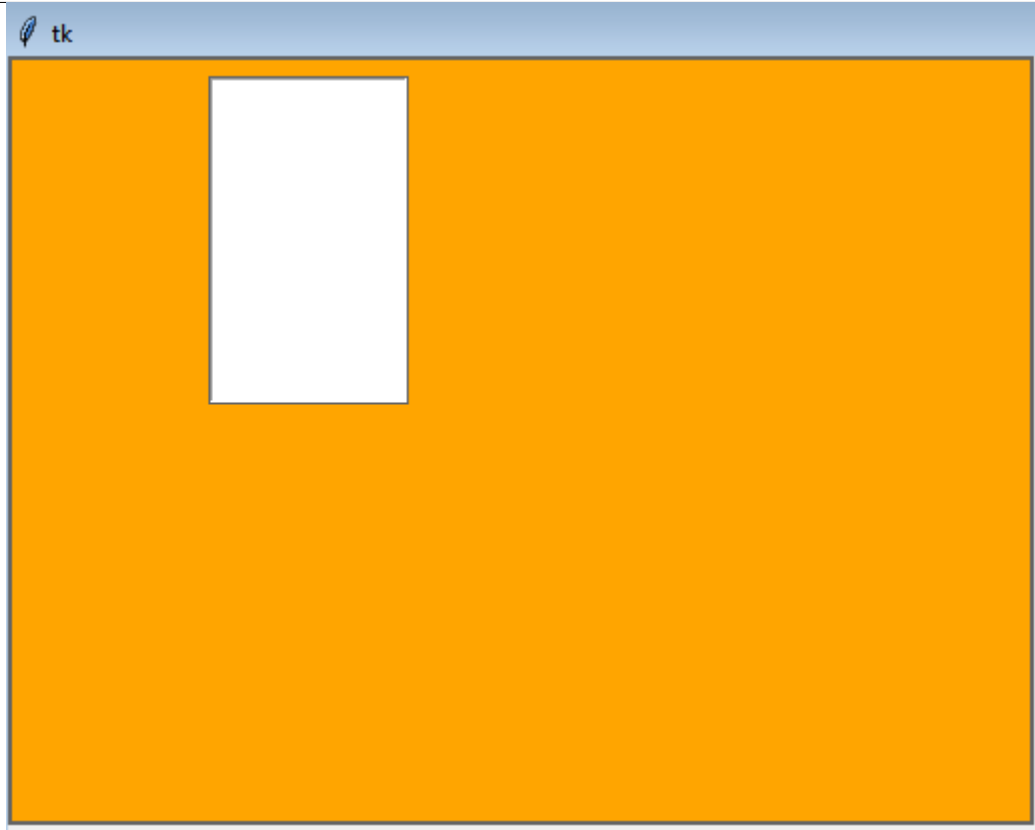
----->
----->
----->

```
import tkinter as tk  
root = tk.Tk()  
amplada = root.winfo_screenwidth()  
alcada = root.winfo_screenheight()  
strink = '{}x{}'.format(amplada,alcada)  
root.geometry(strink)  
root.attributes('-fullscreen',True)  
c = tk.Canvas(root,bg= 'yellow', width = amplada, height =alcada))  
c.place(x=0, y=0)
```

**ELS WIDGETS DE TKINTER PROGRAMACIÓ EN
PYTHON PRIMAVERA 2021**

JOAN MASDEMONT FONTÀS

1 WIDGET LISTBOX EN CANVAS O FRAME

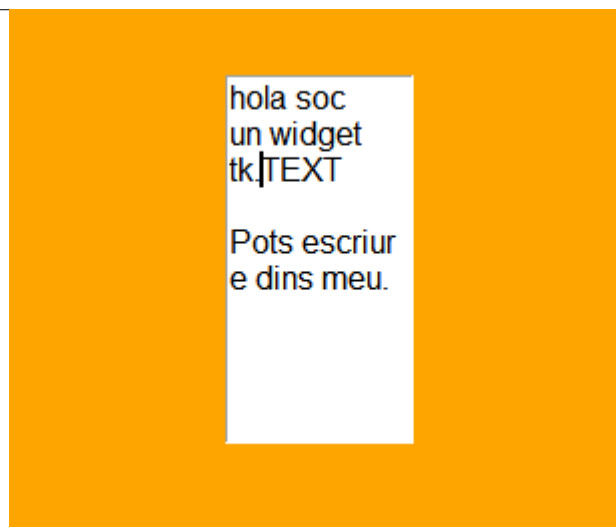


WIDGET LISTBOX

Codi Basic de creacio de listbox en el Canvas anterior que prenia el nom c.

```
Lb1 = tk.Listbox(c)  
Lb1.place( x = 100, y = 10, width = 100)
```

2. WIDGET TEXT EN CANVAS O FRAME

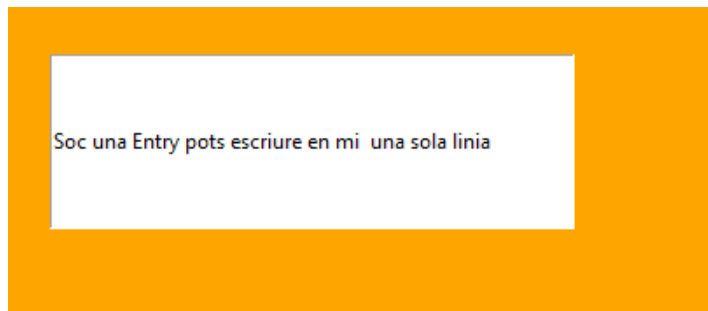


WIDGET TEXT

Codi Basic de creacio de Text en el Canvas anterior que prenia el nom c.

```
T= tk.Text(c, height = 30, width = 70, font = ("Helvetica", 12))  
T.place( x = 100, y = 10)
```

3. WIDGET ENTRY EN CANVAS O FRAME



WIDGET ENTRY

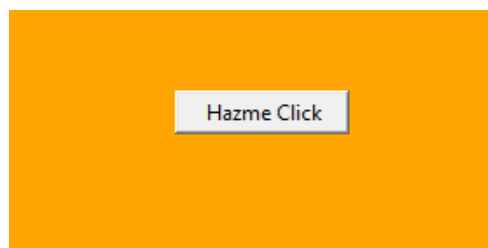
WIDGET ENTRY

Codi Basic de creacio de Entry en el Canvas anterior que prenia el nom c.

```
m=Entry(c)
```

```
m.place(x=630,y=100,width=100,height=25)
```

4. WIDGET BUTTON EN CANVAS O FRAME



WIDGET BUTTON

WIDGET BUTTON

Codi Basic de creacio de Button en el Canvas anterior que prenia el nom c.

```
def Buscar():
```

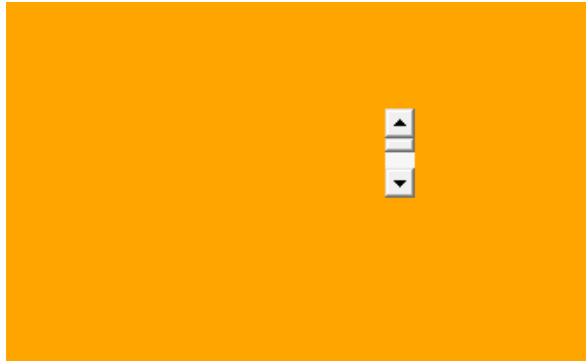
```
    pass
```

```
j = Button(c, text = "Hazme Click",command=Buscar)
```

```
j.place(x = 750, y = 50, width=100, height=25)
```

on Buscar es la funció a la que crida command al prémer el botó.

5. WIDGET SCROLLBAREN CANVAS O FRAME



WIDGET SCROLLBAR

WIDGET SCROLLBAR

Codi Basic de creacio de Scrollbar en el Canvas anterior que prenia el nom c.

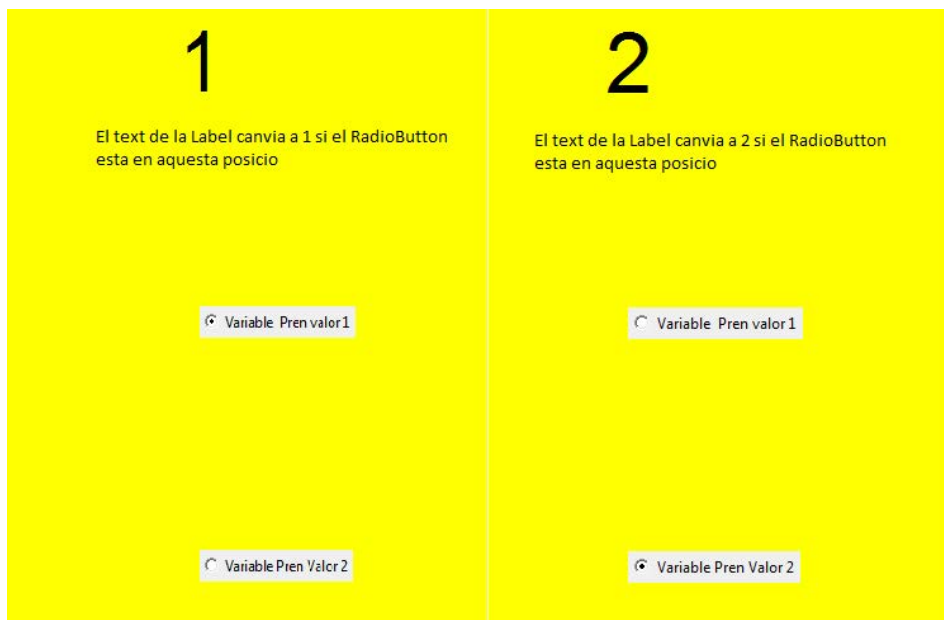
Pot utilitzar-se a altres widgets com Entry, Listbox i Text.

```
scrollbar = tk.Scrollbar(c)
```

```
scrollbar.place(x = 800, y = 120)
```


6. WIDGET RADIOBUTTON A CANVAS O FRAME

Per entendre millor aquest widget mirem la següent imatge.



En un canvas hem afegit dos Radio buttons. El codi que mostraré a continuació fa el següent paper. Quan el radiobutton es troba en la posició Variable Pren valor 1, certa variable pren el valor 1 que és el text que acaba prenent la label del programa. Amb l'altre RadioButton seleccionat el text de la label pren el valor 2.

El codi que aconseguim això és el següent:

```
import tkinter as tk
import sys
def sel():
    Label1.config(text=v.get())

root = tk.Tk()
amplada = root.winfo_screenwidth()
alcada = root.winfo_screenheight()
strink = "{}x{}".format(amplada, alcada)
root.geometry(strink)
root.attributes('-fullscreen', True)
c = tk.Canvas(root, bg = "Yellow")
c.place(x = 0, y = 0, width = amplada, height = alcada)
Label1 = tk.Label(c, text = "", width = 0, font = ('Arial', 50), bg = c.cget('bg'))
Label1.place(x = amplada//2, y = alcada//4, anchor = "center")
v = tk.IntVar()
r1 = tk.Radiobutton(c, text = "Variable Pren valor 1", variable = v, value = 1, command = sel)
r1.place(x = amplada//2, y = alcada//2)
r2 = tk.Radiobutton(c, text = "Variable Pren valor 2", variable = v, value = 2, command = sel)
r2.place(x = amplada//2, y = 3*alcada//4)
root.mainloop()
```

Definim una variable sencera que el Tkinter interpreta i els RadioButtons modifiquen. El RadioButton r1 dóna a la variable el valor 1 i crida a la funció def sel() que modifica el valor del text de la label Label1 que pren el valor de v amb el comandament v.get(). El mateix per r2 que dóna valor 2 a la variable v.

7. WIDGET CHECKBUTTON A CANVAS O FRAME

Per entendre millor aquest widget mirem la següent imatge.



En un canvas hem afegit un CheckButton. El codi que mostraré a continuació fa el següent paper. Quan el Checkbutton es troba marcat una variable pren valor 1. Quant el Checkbutton no està marcat la variable pren el valor 2. En funció d'aquest valor el text de la label canviarà.

El codi que aconsegueix això és el següent:

```
import tkinter as tk
```

```
def sel():
```

```
    if v.get() == 1:
```

```
        text = "CheckButton Marcat"
```

```
    else:
```

```
        text = "CheckButton Desmarcat"
```

```
    Label1.config(text=text)
```

```
root = tk.Tk()
```

```
amplada = root.winfo_screenwidth()
```

```
alcada = root.winfo_screenheight()
```

```
strink = "{}x{}".format(amplada, alcada)
```

```
root.geometry(strink)
```

```
root.attributes('-fullscreen', True)
```

```
c = tk.Canvas(root, bg = "Yellow")
```

```
c.place(x = 0, y = 0, width = amplada, height = alcada)
```

```
Label1 = tk.Label(c, text = "", width = 0, font = ('Arial', 50), bg = c.cget('bg'))
```

```
Label1.place(x = amplada//2, y = alcada//4, anchor = "center")
```

```
v = tk.IntVar()
```

```
check1 = tk.Checkbutton(c, text = "Comprova Checkbutton", variable = v, onvalue = 1, offvalue = 2,  
command = sel)
```

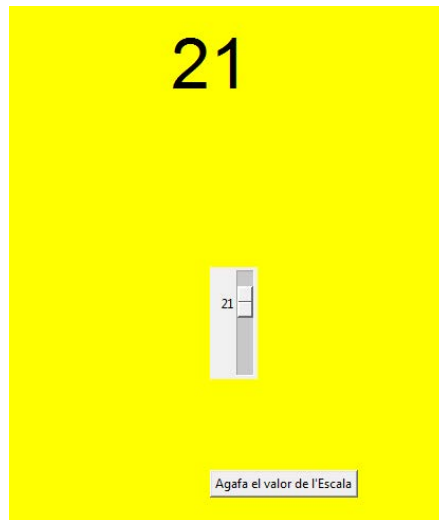
```
check1.place(x = amplada//2, y = alcada//2)
```

```
root.mainloop()
```

Definim una variable sencera que el Tkinter interpreta i el Checkbutton modifica, variable **v**. **onvalue** és el valor que pren la variable quan el Checkbutton està marcat i **offvalue** és el valor que pren la variable quan el Checkbutton està desmarcat. El Checkbutton crida a la **funció sel()** que aprofitem per modificar el **valor del text de la Label**. El text oscil·la entre *Checkbutton Marcat* o *Checkbutton desmarcat*.

8. WIDGET SCALE A CANVAS O FRAME

Per entendre millor aquest widget mirem la següent imatge.



En un canvas hem afegit un widget Scale que modifica el valor d'una variable quan canviem l'escala, p.e. del 0 al 100. Un Botó activarà una funció que llegirà el valor de la variable definit a l'indicador del widget scale. El text de la Label mostrarà el valor de l'escala.

El codi que aconseguim això és el següent:

```
import tkinter as tk
```

```
def sel():
```

```
    text = v.get()
```

```
    Label1.config(text=text)
```

```
root = tk.Tk()
```

```
amplada = root.winfo_screenwidth()
```

```
alcada = root.winfo_screenheight()
```

```
strink = "{}x{}".format(amplada, alcada)
```

```
root.geometry(strink)
```

```
root.attributes('-fullscreen', True)
```

```
c = tk.Canvas(root, bg = "Yellow")
```

```
c.place(x = 0, y = 0, width = amplada, height = alcada)
```

```
Label1 = tk.Label(c, text = "", width = 0, font = ('Arial', 50), bg = c.cget('bg'))
```

```
Label1.place(x = amplada//2, y = alcada//4, anchor = "center")
```

```
v = tk.IntVar()
```

```
scale1 = tk.Scale(c, variable = v)
```

```
scale1.place(x = amplada//2, y = alcada//2)
```

```
Button1 = tk.Button(c, text = "Agafa el valor de l'escala", command = sel)
```

```
Button1.place(x = amplada//2, y = 3 * alcada//4)
```

```
root.mainloop()
```

Definim una variable sencera que el Tkinter interpreta i el Widget Scale modifica. El Button crida a la funció sel() que aprofitem per llegir el valor de la variable v i modificar el valor del text de la Label, que mostrarà el valor de l'escala.

9. WIDGET MENU A CANVAS O FRAME

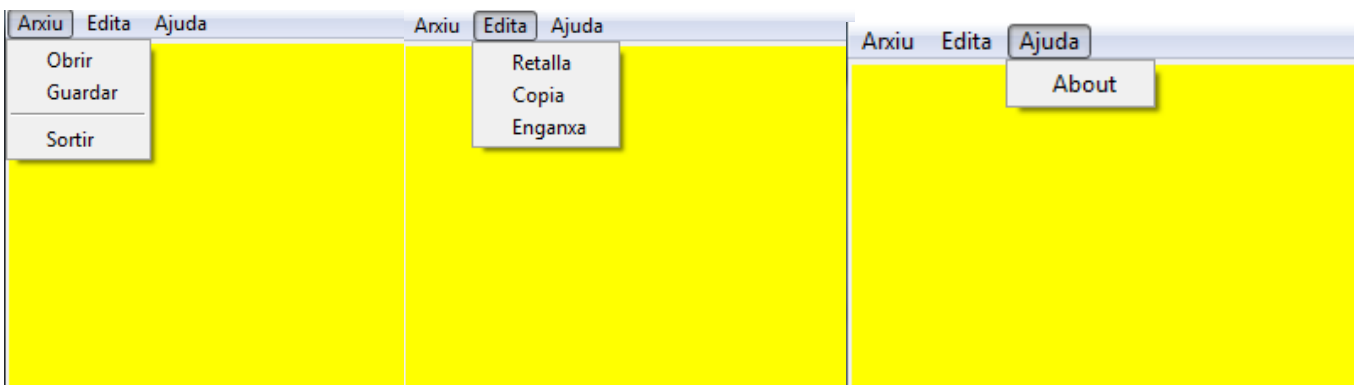
A continuació es mostra un codi que utilitza aquest widget Menu i la imatge dels resultats obtinguts

Si executes aquest codi s'acaba entenent què fa cada línia. Ho comentarem a classe.

```
import tkinter as tk

def hello():
    pass

root = tk.Tk()
amplada = root.winfo_screenwidth()
alcada = root.winfo_screenheight()
strink = '{}x{}'.format(amplada,alcada)
root.geometry(strink)
c= tk.Canvas(root,bg='yellow')
c.place(x=0,y=0,width=amplada,height=alcada)
barramenu=tk.Menu(root)
menuarxiu= tk.Menu(barramenu, tearoff=0)
menuarxiu.add_command(label="Obrir", command=hello)
menuarxiu.add_command(label="Guardar", command=hello)
menuarxiu.add_separator()
menuarxiu.add_command(label="Sortir", command=root.quit)
barramenu.add_cascade(label="Arxiu", menu=menuarxiu)
menuedicio = tk.Menu(barramenu, tearoff=0)
menuedicio.add_command(label="Retalla", command=hello)
menuedicio.add_command(label="Copia", command=hello)
menuedicio.add_command(label="Enganxa", command=hello)
barramenu.add_cascade(label="Edita", menu=menuedicio)
menuajuda = tk.Menu(barramenu, tearoff=0)
menuajuda.add_command(label="About", command=hello)
barramenu.add_cascade(label="Ajuda", menu=menuajuda)
root.config(menu=barramenu)
root.mainloop()
```



**ANNEXOS EXPLICATIUS
PROGRAMACIÓ EN PYTHON
PRIMAVERA 2021**

JOAN MASDEMONT FONTÀS

TKINTER

A) INICIALIZACIO BASICA LINIES QUE NO PODEN FALTAR

import Tkinter as tk (a Python 2 es Tkinter a Python 3 es tkinter)

root = tk.Tk() (definicio de l'objecte principal sobre la que muntarem el canvas i altres widgets.

root.mainloop() (Definició del bucle principal perquè es mantingui l'estructura gràfica tot el temps)

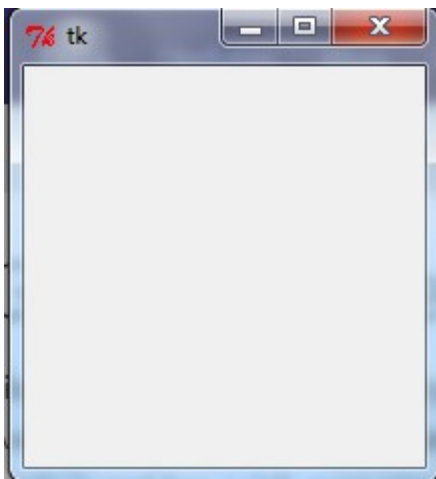


Figura 1. Gràfic obtingut amb les tres línies de dalt

B) AGAFANT LA RESOLUCIO DE LA PANTALLA DE L'ORDINADOR PER ESCALAR OBJECTES

amplada = root.winfo_screenwidth()
alcada = root.winfo_screenheight()

Podem fer `print(amplada, alcada)` per veure que hem capturat les dimensions de la resolució de pantalla correctament

```
C:\Python>C:/Python27/python.exe c:/Python/grafic.py  
(1366, 768)
```

C) ESPECIFICANT EL TAMANY DE L'OBJECTE PRINCIPAL

```
strink = '{}x{}'.format(amplada,alcada)
root.geometry(strink)
```

En aquest exemple utilitzem un string format a partir de amplada i alcada que eren el resultat de capturar la resolució de la pantalla d'un sistema operatiu per definir que l'objecte principal tindrà el tamany igual a tota la pantalla o sigui que ocupara tot l'escriptori.

D) ASSEGURANT QUE S'OBRE EL NOSTRE OBJECTE TKINTER A PANTALLA COMPLETA

Podem assegurar que quan executem el programa s'obrirà sempre l'objecte principal del nostre tkinter a pantalla completa, dient

```
root.attributes('-fullscreen', True)
```

El nostre codi de moment queda així

```
import Tkinter as tk
root = tk.Tk()
amplada = root.winfo_screenwidth()
alcada = root.winfo_screenheight()
strink = '{}x{}'.format(amplada,alcada)
root.geometry(strink)
root.attributes('-fullscreen', True)
root.mainloop()
```

ARA JA ESTEM A PUNT D'OMPLIR EL NOSTRE OBJECTE PRINCIPAL DE WIDGETS

Tkinter Classes

Widget classes

Tkinter supports 15 core widgets:

Button

A simple button, used to execute a command or other operation.

Canvas

Structured graphics. This widget can be used to draw graphs and plots, create graphics editors, and to implement custom widgets.

Checkbutton

Represents a variable that can have two distinct values. Clicking the button toggles between the values.

Entry

A text entry field.

Frame

A container widget. The frame can have a border and a background, and is used to group other widgets when creating an application or dialog layout.

Label

Displays a text or an image.

Listbox

Displays a list of alternatives. The listbox can be configured to get radiobutton or checklist behavior.

Menu

A menu pane. Used to implement pulldown and popup menus.

Menubutton

A menubutton. Used to implement pulldown menus.

Message

Display a text. Similar to the label widget, but can automatically wrap text to a given width or aspect ratio.

Radiobutton

Represents one value of a variable that can have one of many values. Clicking the button sets the variable to that value, and clears all other radiobuttons associated with the same variable.

Scale

Allows you to set a numerical value by dragging a “slider”.

Scrollbar

Standard scrollbars for use with canvas, entry, listbox, and text widgets.

Text

Formatted text display. Allows you to display and edit text with various styles and attributes. Also supports embedded images and windows.

Toplevel

A container widget displayed as a separate, top-level window.

CODI. WIDGET SCALE QUE ACTUALITZA NOMES GRADUAR L'SCALE (SENSE CAP ALTRE WIDGET INTERMEDIARI)

```
import tkinter as tk
import sys

def sel(self):
    text = v.get()
    print(text)
    Label1.config(text=text)

root = tk.Tk()
amplada = root.winfo_screenwidth()
alcada = root.winfo_screenheight()
strink = '{}x{}'.format(amplada,alcada)
root.geometry(strink)
root.attributes('-fullscreen', True)
c= tk.Canvas(root,bg='yellow')
c.place(x=0,y=0,width=amplada,height=alcada)
Label1 = tk.Label(c, text='', width = 0, font=('Arial',50),bg = c.cget('bg'))
Label1.place(x = amplada//2, y = alcada//4, anchor ="center")
v = tk.IntVar()
scale1 = tk.Scale(c, variable = v, orient='horizontal', from_=0, to=128, command=sel)
scale1.place( x = amplada // 2, y= alcada // 2)
root.mainloop()
```

CODI INTEGRACIO SCROLLBAR VERTICAL A WIDGET TEXT. METODE IN_

```
import tkinter as tk
root = tk.Tk()
amplada = root.winfo_screenwidth()
alcada = root.winfo_screenheight()
strink = '{}x{}'.format(amplada,alcada)
root.geometry(strink)
root.attributes('-fullscreen', True)
c= tk.Canvas(root,bg='yellow')
c.place(x=0,y=0,width=amplada,height=alcada)
Label1 = tk.Label(c, text='EL XIFRAT DEL CESAR', width = 0, font=('Arial',50),bg=c.cget('bg'))
Label1.place(x = amplada//2, y = 2*alcada//32, anchor ="center")
txt1 = tk.Text(c, height = 10, width = 10, font=('Arial',30))
txt1.place(x =25*amplada//32, y = alcada//2, anchor ="center")
scrollbar = tk.Scrollbar(c,command=txt1.yview)
scrollbar.place(in_=txt1, relx=1.0, relheight=1.0, bordermode="outside")

txt1.config(yscrollcommand=scrollbar.set)
txt2 = tk.Text(c, height = 10, width = 10, font=('Arial',30))
txt2.place(x =7*amplada//32, y = alcada//2, anchor ="center")
scrollbar2 = tk.Scrollbar(c,command=txt2.yview)
scrollbar2.place(in_=txt2, relx=1.0, relheight=1.0, bordermode="outside")
txt2.config(yscrollcommand=scrollbar2.set)
root.mainloop()
```

FENT US DEL WIDGET LISTBOX. LISTBOX IN ACTION

```
import tkinter as tk
```

```
def CurSelet(evt):
```

```
    # Agafem el text triat i ho convertim en el text de la Label.
```

```
    Lb1.config(text =listbox.get(tk.ANCHOR))
```

```
root = tk.Tk()
```

```
amplada = root.winfo_screenwidth()
```

```
alcada = root.winfo_screenheight()
```

```
strink = '{}x{}'.format(amplada,alcada)
```

```
root.geometry(strink)
```

```
root.attributes('-fullscreen', True)
```

```
c= tk.Canvas(root,bg='yellow')
```

```
c.place(x=0,y=0,width=amplada,height=alcada)
```

```
listbox = tk.Listbox(c)
```

```
listbox.place(x = amplada//2, y= alcada//2, anchor = 'center' )
```

```
listbox.insert(tk.END, "LLISTAT")
```

```
for item in ["one", "two", "three", "four"]:
```

```
    listbox.insert(tk.END, item)
```

```
Lb1 = tk.Label(c, text =", width =0,height = 30, font = ('Helvetica', 12),
```

```
bg = c.cget('bg') )
```

```
Lb1.place(x = amplada//4, y= alcada//2, anchor = 'center' )
```

```
# Es defineix què passa quan se selecciona un element de la listbox, per
```

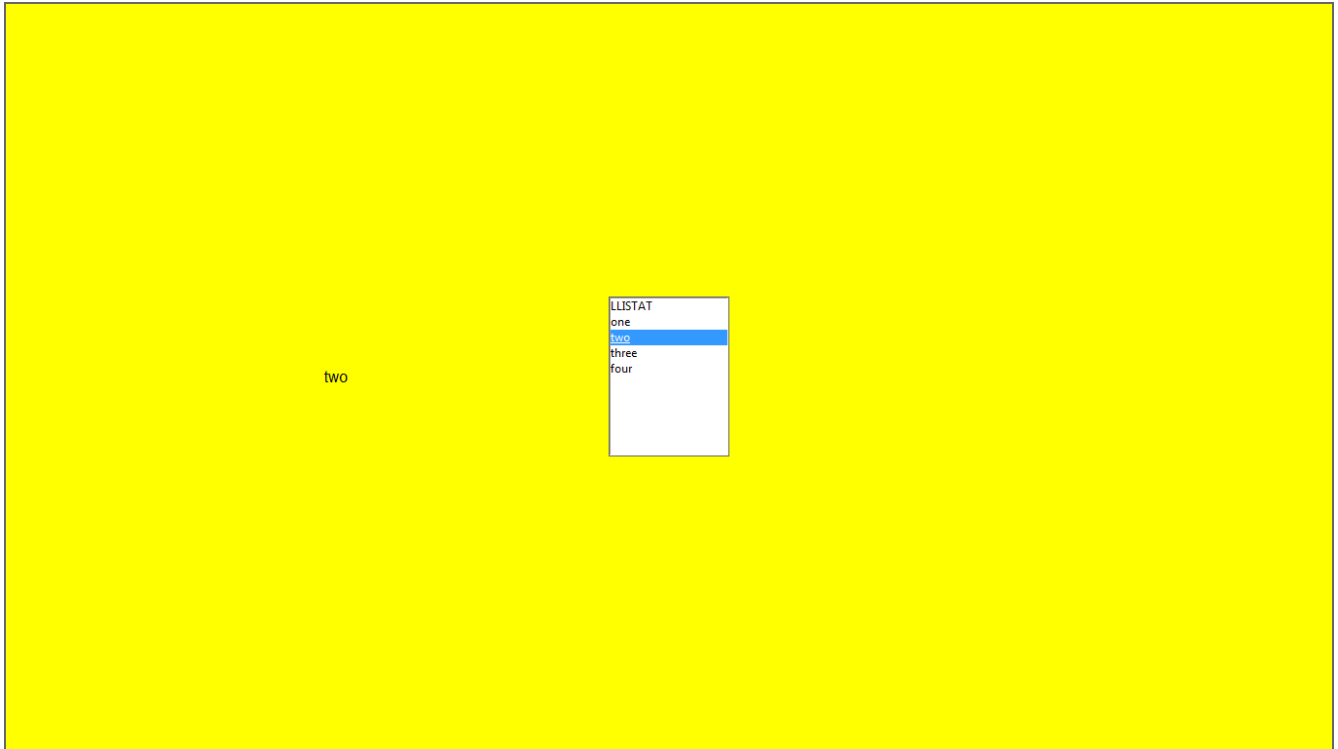
```
# exemple amb el ratoli i es crida a la funcio CurSelet
```

```
listbox.bind('<<ListboxSelect>>',CurSelet)
```

```
root.mainloop()
```

```
#METODE PER ESBORRAR-HO TOT
```

```
#listbox.delete(0,END)
```



**AL FER CLIC A UN ELEMENT ES MOSTRA EL TEXT A
UNA LABEL**