Memento Tkinter avec Python

from tkinter import *

Fenêtre

```
fen1=Tk()                    #ouvre une instance de fenêtre
fen1.title('Titre')          # titre
lbl1=Label(fen1,text='Texte',bg='red')   # label rouge
lbl1.pack()                  #ajuste au contenu et affiche
btn1=Button(fen1,text='Quitter', command=fen1.destroy)
btn1.pack
fen1.mainloop    #démarre la réception d'évènements
```

options couleurs: fg=foreground, bg=background

Widgets containers : Frame et Canvas

Widget Frame= cadre avec couleur et bordure pour contenir des widgets

```
Frame(fen1,bd=2,bg="#d0d0b0",relief=FLAT)
```

reliefs: RAISED (sortant), SUNKEN (rentrant), FLAT (plat), RIDGE (crête), GROOVE (sillon), SOLID (bordure)

Widget Canvas : surface pour dessiner

```
can1=Canvas(fen1 ou Tk() , bg='dark gray',height=200, width=200)
can1.pack(side=LEFT)
```

Méthodes de positionnements

1.pack(): positionnés dans le flux

```
btn.pack(side=LEFT,padx=5,pady=5)
```

side : LEFT, RIGHT, TOP, BOTTOM
padx : espace bord gauche dte
pady: espace bord haut bas

2.Méthode grid: positionnés avec une grille

```
txt1.grid(row=0)          #1er champ en haut à gche
txt2.grid(row=1)          #2è ligne
txt3.grid(row=0,column=1) #1ere ligne 2ème col
```

options: sticky N, S, E, O alignement haut, bas...
colspan=2 deux colonnes / rowspan=2 2 lignes
padx =5 espacement gche dte
pady= 5 espacement haut bas

3.place() en coordonnées

Tracés dans un canevas

```
can1.create_line(x1,y1,x2,y2,fill='couleur')
can1.create_rectangle(x1,y1,x2,y2)
can1.create_arc(x1,y1,x2,y2,start=0,extend=90)
```

start : angle de départ / extend angle

ovale dans rectangle fictif du coin sup gche au inf. droit

```
can1.create_oval(x1,y1,x2,y2,outline='couleur')
```

cercle de centre x,y de rayon r :

```
can1.create_oval(x-r,y-r,x+r,y+r,outline='blue')
```

polygone fermé: dernier pt rejoint le 1er

```
create_polygon(x1,y1,x2,y2,...)
```

texte:

```
can1.create_text(x,y,text="mon texte", font="Arial 12")
```

modifier par montexte.configure(text="nouveau")

Paramètres :

fill ='couleur' / outline='couleur' / width=1

Effacer: can1.delete(ALL)

```
fen.update()   #force la mise à jour
```

Widgets de base

Button - Label –Entry (champ d'entrée texte) – Text
Listbox – Menu - MenuButton (déroulant)
Message(texte mis en forme sur largeur hauteur)
Checkbutton – Scrollbar (ascenseur)

Widget: Scale (curseur)

```
Scale( Tk(), length=200, orient=HORIZONTAL,
label="Echelle", throughcolor='darkgray',
sliderlength=20, showvalue=0, _from=-25, to=125,
tickinterval=25, command=nomcommande)
```

Widget: RadioButton

```
Radiobutton(self,text="texte",variable=var, value=valeur,
command=nomcde)
```

où variable sera la même pour tous les boutons liés
on récupère la valeur choisie par: var.get()

Widget Entry (champ d'entrée texte)

```
entree= Entry(fen1)
```

ou entree=Entry(fen1,text variable= var)

Accès au widget Entry

```
entree.get() #récupere la chaine complète
entree.insert(pos,"texte à insérer")  #insertion ss-chaine
```

où pos = entier entre 0 et len -1,
0 pour début de chaîne, END pour fin

```
entree.configure(font="Arial 15 Normal") #modif param.
```

Detection d'entrées clavier

```
#associe la fonction fct par référence sans () au retour
entree.bind("<Return>",fct)
def fct(event) :
        #interprete un calcul entré au clavier
        result= eval(entree.get() )
        # affichage sans un Label
        lbl1.configure(text="Affichage")
event.char   #contient la lettre pressée
```

Détection de la souris

event transmet ts les événements clavier-souris
event.x , event.y coordonnées du click

```
fen.bind("<Button-1>",fct) #click
```

Eveènements souris

```
<Button-1>          #btn souris gauche droit
<Double-1>          #double click btn gauche
<Button1-Motion>  #déplacement bouton gche enfoncé
<ButtonRelease-1>  #relaché du  bouton gauche
<Button-2>          #click sur molette centrale
<Button-3>          #bouton droit souris
```

Evènements clavier

```
<KeyPress-A >                  #Clavier touche A Majuscule
<KeyPress-a >                  #Clavier touche a minuscule
<Control-Shift-KeyPress-A>  #Ctrl Shift A
<Any-KeyPress>=<Key>    #clavier n'importe quelle touche
<KP-Down> <KP-Enter><KP-1>   #clavier numérique
<Delete> <Escape> <F1> <Insert><Tab> <Return> <Pause>
<Down> <Up><Left><Right>
<Enter><Leave>   #entrée et sortie de souris sur widget
<Configure>   #Redimensionnement fenêtre
```