TKINTER - CANVAS et EVENEMENTS SOURIS ou CLAVIER

Widget Canvas

Déclaration : nom_canvas=Canvas(parent, width=800, height=600, bg='white'), puis pack ou grid
« nom_canvas » est le nom donné au Canvas, dans la suite ce sera « dessin »

Tracé de figures

Ligne : dessin.create_line(x1,y1,x2,y2, *fill='blue',width=2*)
Rectangle dessin.create_rectangle(x1,y1,x2,y2, *fill='yellow',outline='red',width=4*)
Ellipse dessin.create_oval(....)
Polygon dessin.create_polygon(x1,y1,x2,y2,.....x1,y1)

Trouver un objet

2 méthodes utiles : **find_closest** (le plus près du point précisé)
find_enclosed (situé dans un rectangle précisé)
Les deux méthodes renvoient une **liste**, éventuellement vide (None)
objet=dessin.find_closest(x1,y1)
objet=dessin.find_enclosed(x1,y1,x2,y2)
à suivre toujours de « if objet : », au cas où il n'y en ait pas,
puis de objet_select=objet[0], au cas où il y en ait plusieurs.....

Modifier objet

Exemple AB= dessin.create_line(x1,y1,x2,y2)
Modifier dessin.itemconfig(AB,fill='blue')
Effacer dessin.delete(AB)
Déplacer dessin.move(AB,x,y) (translation)
dessin.coords(AB,x3,y3,x4,y4) (change les 4 coordonnées)
Remarque : « coords » peut aussi être utilisé pour récupérer les coordonnées de l'objet : dessin.coords(AB) renvoie la liste [x1,y1,x2,y2]

Événements souris

On applique d'abord la méthode « bind » au widget qui doit réagir à certaines actions de la souris.
Syntaxe : **dessin.bind(action, fonction)** où « dessin» est le nom du widget,« action » décrit l'action sur la souris et « fonction », la fonction appelée lors de l'action. Cette fonction doit dépendre du paramètre « event » qui permet entre autres de récupérer les coordonnées du pointeur de souris, avec event.x et event.y.

Exemple :

```
dessin.bind("<Button-1>", affiche)
def affiche(event) :
   print (event.x,event.y)
```

Actions possibles : "<Button-1>", "<Button1-Motion>","<Button1-ButtonRelease>"
<Double-Button-1>, "<Enter>", "<Leave>", idem avec Button 2 et 3.

dessin.focus_set() pour donner le focus au widget

Événements clavier dessin.bind('<Key>', fonction)

On récupère le code de la touche enfoncée avec event.keycode
Certaines touches peuvent être « bindées » directement : **Cancel**, **BackSpace**, **Tab**, **Return**, **Shift_L, Control_L**, **Alt_L** , **Pause**, **Caps_Lock**, **Escape**, **Prior** , **Next** , **End**, **Home**, **Left**, **Up**, **Right**, **Down**, **Print**, **Insert**, **Delete**, **F1**, **F2**, **F3**, **F4**, **F5**, **F6**, **F7**, **F8**, **F9**, **F10**, **F11**, **F12**, ….