

Initiation à la programmation en Python

Damien Vergnaud

École Normale Supérieure

8 avril 2015

Tkinter : premier exemple

```
from Tkinter import *
```

```
>>> fen1 = Tk()
>>> tex1 = Label(fen1, text='Bonjour tout le monde !', fg='red')
>>> tex1.pack()
>>> bou1 = Button(fen1, text='Quitter', command = fen1.destroy)
>>> bou1.pack()
>>> fen1.mainloop()
```

Tracé de lignes

```
# les variables suivantes seront utilisées de manière globale :
x1, y1, x2, y2 = 10, 190, 190, 10           # coordonnées de la ligne
coul = 'dark green'                          # couleur de la ligne

# Création du widget principal ("maître") :
fen1 = Tk()

# création des widgets "esclaves" :
can1 = Canvas(fen1,bg='dark grey',height=200,width=200)
can1.pack(side=LEFT)
bou1 = Button(fen1,text='Quitter',command=fen1.quit)
bou1.pack(side=BOTTOM)
bou2 = Button(fen1,text='Tracer une ligne',command=drawline)
bou2.pack()
bou3 = Button(fen1,text='Autre couleur',command=change_color)
bou3.pack()

fen1.mainloop()                               # démarrage du récepteur d'événements

fen1.destroy()                                # destruction (fermeture) de la fenêtre
```

Tracé de lignes

```
# --- définition des fonctions gestionnaires d'événements : ---
def drawline():
    "Tracé d'une ligne dans le canevas can1"
    global x1, y1, x2, y2, coul
    can1.create_line(x1,y1,x2,y2,width=2,fill=coul)

    # modification des coordonnées pour la ligne suivante :
    y2, y1 = y2+10, y1-10

def changecolor():
    "Changement aléatoire de la couleur du tracé"
    global coul
    pal=['purple','cyan','maroon','green','red','blue','orange','yellow']
    c = randrange(8)          # => génère un nombre aléatoire de 0 à 7
    coul = pal[c]
```

Gestion de la souris

```
# Détection et positionnement d'un clic de souris dans une fenêtre :

from Tkinter import *

def pointeur(event):
    chaine.configure(text = "Clic détecté en X =" + str(event.x) + \
        ", Y =" + str(event.y))

fen = Tk()
cadre = Frame(fen, width =200, height =150, bg="light yellow")
cadre.bind("<Button-1>", pointeur)
cadre.pack()
chaine = Label(fen)
chaine.pack()

fen.mainloop()
```

Les classes de widget Tkinter

- Button** Un bouton classique, à utiliser pour provoquer l'exécution d'une commande quelconque.
- Canvas** Un espace pour disposer divers éléments graphiques.
- Checkbutton** Une case à cocher qui peut prendre deux états distincts (la case est cochée ou non).
- Entry** Un champ d'entrée, dans lequel l'utilisateur peut insérer un texte à partir du clavier.
- Frame** Une surface rectangulaire dans la fenêtre, où l'on peut disposer d'autres widgets.
- Label** Un texte (ou libellé) quelconque (éventuellement une image).
- Listbox** Une liste de choix proposés à l'utilisateur

Les classes de widget Tkinter

Menu	Un menu déroulant ou bien un menu pop up
Menubutton	Un bouton-menu, à utiliser pour les menus déroulants.
Message	Permet d'afficher un texte. (variante du widget Label, qui permet d'adapter automatiquement le texte)
Radiobutton	Représente (par un point noir dans un petit cercle) une des valeurs d'une variable qui peut en posséder plusieurs.
Scale	Permet de faire varier de manière visuelle la valeur d'une variable
Scrollbar	Barre de défilement
Text	Affichage de texte formaté. Permet aussi à l'utilisateur d'éditer le texte affiché. Des images peuvent également être insérées.
Toplevel	Une fenêtre affichée séparément (par-dessus).

Méthode grid

```
from Tkinter import *

fen1 = Tk()

# création de widgets 'Label' et 'Entry' :
txt1 = Label(fen1, text = 'Premier champ :')
txt2 = Label(fen1, text = 'Second :')
txt3 = Label(fen1, text = 'Troisième :')
entr1 = Entry(fen1)
entr2 = Entry(fen1)
entr3 = Entry(fen1)

# création d'un widget 'Canvas' contenant une image bitmap :
can1 = Canvas(fen1, width =160, height =160, bg = 'white')
photo = PhotoImage(file = 'photo.gif')
item = can1.create_image(80, 80, image =photo)
```


Méthode grid

```
# Mise en page à l'aide de la méthode 'grid' :
txt1.grid(row =1, sticky =E)
txt2.grid(row =2, sticky =E)
txt3.grid(row =3, sticky =E)
entr1.grid(row =1, column =2)
entr2.grid(row =2, column =2)
entr3.grid(row =3, column =2)
can1.grid(row =1, column =3, rowspan =3, padx =10, pady =5)

# démarrage :
fen1.mainloop()
```