Les bases d'un interface graphique avec Tkinter

Quelques références de base pour utiliser Tkinter

- Chapitre 8 du livre ”apprendre à programmer avec Python”, de Gérard Swinnen
  - version en wiki
- Tkinter reference: a GUI for Python (online or pdf) by John W. Shipman)
- An Introduction to Tkinter, de Fredrik Lundh (tutoriel Tk)
- An Introduction to Tkinter, sur effbot.org
- Tkinter tutorial, sur python-course.eu

> Entre Python 2 et Python 3, le nom de la librairie “Tkinter” est passé à tkinter ! (première lettre en bas de casse). L'utilisation sous Python 3 des exemples ci-dessous nécessite aussi de transformer les instructions print en print().

Une étiquette (Label) affichant "Bonjour !"

<sxh python; title : Tk-00.py> #!/usr/bin/env python # -*- coding: utf-8 -*-

from Tkinter import *

root=Tk() w=Label(root, text=“Bonjour !”) w.pack()

root.mainloop() </sxh>

Un bouton (Button) avec une action pour écrire

L'écriture va s'effectuer sur la console ! <sxh python; title : Tk-01.py> #!/usr/bin/env python # -*- coding: utf-8 -*-

from Tkinter import *

def action():

```
  print "Yes, we can !"
```

root=Tk() #w=Label(root, text=“Bonjour!”) #w.pack()

b=Button(root,text=“Click here !”,command=action) b.pack()

root.mainloop() </sxh>

Voyez à décommenter les deux lignes concernant l'étiquette “W” !

Champ d'entrée (Entry)

On peut mettre un champ d'entrée et y introduire du texte <sxh python; title : Tk-02.py> #!/usr/bin/env python # -*- coding: utf-8 -*-

from Tkinter import *

def action():

```
  print "Yes, we can !"
```

root=Tk() #w=Label(root, text=“Bonjour!”) #w.grid(row=?)

champ=Entry(root) champ.grid(row=1)

b=Button(root,text=“Click here !”,command=action) b.grid(row=2) root.mainloop() </sxh>

Si on à décommente les deux lignes concernant l'étiquette “W”, comment actualiser les “numéros” de row pour afficher l'étiquette, le champ d'entrée et le bouton ?!

Utiliser le texte rentré

En cliquant, on quitte et on écrit le texte rentré (on n'utilise pas la fonction “action”) <sxh python; title : Tk-03.py> #!/usr/bin/env python # -*- coding: utf-8 -*-

from Tkinter import *

def action():

```
  print "Yes, we can !"
```

root=Tk() #w=Label(root, text=“Bonjour!”)

champ=Entry(root) champ.grid(row=0)

b=Button(root,text=“Click here !”,command=root.quit) b.grid(row=1) root.mainloop()

lecture de la valeur du champ abcdef=champ.get() print abcdef # éliminer la fenêtre : root.destroy() </sxh>

Valeurs numériques et calcul

On fait un calcul avec la valeur rentrée, on quitte et on écrit <sxh python; title : Tk-04.py> #!/usr/bin/env python # -*- coding: utf-8 -*-

from Tkinter import *

def factorielle(argu):

```
  # calcul de la factorielle de argu
  a=1  # a contient une valeur qui va être incrémentée d'une unité à la fois
  b=1  # contient la factorielle de a-1
  while a<=argu:  # on arrêtera lorsque a sera > argu
      b=b * a
      a=a+1
  return b
```

def action():

```
  print "Yes, we can !"
```

root=Tk() #w=Label(root, text="Bonjour!")

champ=Entry(root) champ.grid(row=0)

b=Button(root,text="Click here !",command=root.quit) b.grid(row=1) root.mainloop()

lecture de la valeur du champ texte_n=champ.get() n=int(texte_n) print n, factorielle(n) # éliminer la fenêtre : root.destroy() </sxh>

Tout faire dans interface graphique

Ce programme utilise un Label pour afficher le résultat, on ne quitte plus et on peut recalculer sur d'autres valeurs entrées. Il y a un bouton pour terminer. <sxh python; title : Tk-05.py> #!/usr/bin/env python # -*- coding: utf-8 -*-

from Tkinter import *

def factorielle(argu):

```
  # calcul de la factorielle de argu
  a=1  # a contient une valeur qui va être incrémentée d'une unité à la fois
  b=1  # contient la factorielle de a-1
  while a<=argu:  # on arrêtera lorsque a sera > argu
      b=b * a
      a=a+1
  return b
```

def action():

```
  texte_n=champ.get()
  n=int(texte_n)
  affichefacto.configure(text =str(factorielle(n)))
```

root=Tk()

champ=Entry(root) champ.grid(row=0)

b=Button(root,text="Calcule la factorielle",command=action) b.grid(row=1)

affichefacto=Label(root) affichefacto.grid(row=2)

bfin=Button(root,text="Terminer",command=root.quit) bfin.grid(row=3)

root.mainloop()

éliminer la fenêtre après avoir quitté : root.destroy() </sxh>

Pour d'autres exemples, voir par exemple :

- http://www.python-course.eu/tkinter_entry_widgets.php

Canvas : des rectangles et des mouvements

<sxh python; title : Tk_canvas_rectangles_move.py> #! /usr/bin/env python # -*- coding: utf-8 -*- # Exemple utilisation du Canvas Tk pour gérer une boite avec couvercle mobile

from Tkinter import *

def move():

```
  "déplacement du couvercle"
  global hauteur,v
  hauteur = hauteur + v
  if hauteur > 250 or hauteur < 130:
      v = -v
  can.coords(couvercle,100,hauteur-20, 300, hauteur)
  flag=1
  root.after(1,move)        # boucler après 50 millisecondes
```

root = Tk() can = Canvas(root, width=500, height=400) can.pack()

can.create_rectangle(95,100, 100, 355,fill='blue') can.create_rectangle(300,100, 305, 355,fill='green') can.create_rectangle(100,350, 300, 355,fill='red') hauteur = 150 couvercle = can.create_rectangle(100,hauteur-20, 300, hauteur,fill='black')

animation simple: v = 0.1 # incrément/vitesse verticale move()

can.mainloop() </sxh>

Une étiquette dynamique

<sxh python; title : compteur-01.py> #! /usr/bin/env python # -*- coding: utf-8 -*- # Exemple d'une étiquette dynamique par récursion

import Tkinter as tk

def compteur_label(lab):

```
  def compte():
      global compteur
      compteur += 1
      lab.config(text=str(compteur))
      lab.after(1000, compte)
  compte()
```

root = tk.Tk() root.title(“Comptage en secondes”) label = tk.Label(root, fg=“dark green”) label.pack() compteur = -1 compteur_label(label) button = tk.Button(root, text='Arrêtez !', width=25, command=root.destroy) button.pack() root.mainloop() </sxh>

Créer des points avec la souris

<sxh python; title : points_souris-02.py> #!/usr/bin/env python # -*- coding: utf-8 -*- # créer des points à l'aide de la souris # refs : # http://effbot.org/tkinterbook/tkinter-events-and-bindings.htm

from Tkinter import *

def point(event):

```
can.create_oval(event.x-4, event.y-4, event.x+4, event.y+4, outline="black",
fill="red")
points.append([event.x,event.y])
return
```

root = Tk() root.title(“Créer des points !”) points = [] can = Canvas(root, bg=“grey”, width=640, height= 480) can.configure(cursor=“crosshair”) can.grid(row=0) can.bind(“<Button-1>”, point) b=Button(root,text=“Quitter”,command=root.quit) b.grid(row=1) root.mainloop() print points </sxh>

Utiliser des boutons radio (radiobuttons)

<sxh python; title : radiobuttons.py> #! /usr/bin/env python # -*- coding: utf-8 -*- # Exemple d'utilisation des boutons radio

import Tkinter as tk

def affiche_choix():

```
  i=v.get()
  print i, positions[i-1][0]
```

root = tk.Tk() v = tk.IntVar() v.set(1) # choix par défaut

positions = [(“ortho”,1),(“meta”,2),(“para”,3)]

lab=tk.Label(root, text=“Choix de la position”, fg=“dark blue”) lab.pack()

for txt, val in positions:

```
  b=tk.Radiobutton(root, text=txt, padx = 10, variable=v,
command=affiche_choix, value=val)
  b.pack()
```

tk.mainloop() </sxh>

Utiliser des cases à cocher (checkbuttons)

<sxh python; title : checkbuttons-03.py> #! /usr/bin/env python # -*- coding: utf-8 -*- # Exemple d'utilisation des cases à cocher

import Tkinter as tk

def affiche_choix():

```
  print zip(elements, [etats[i].get() for i in range(len(elements))])
```

root = tk.Tk() lab=tk.Label(root, text=“Cochez les éléments présents”, bg=“red”, fg=“dark blue”) lab.grid(row = 0)

elements=['C','H','O','N','P','S',u'éléments métalliques',u'halogénures',u'autres'] etats=[] nelem=len(elements)

for i in range(nelem):

```
  etat=tk.IntVar()
  caco = tk.Checkbutton(root, text=elements[i], variable=etat,width =
20,padx=50,anchor = tk.W)
  caco.grid(row = i+1)
  etats.append(etat)
```

button = tk.Button(root, text='Affichez !', width=25, command=affiche_choix) button.grid(row = nelem+1)

tk.mainloop() </sxh>

From:
https://dvillers.umons.ac.be/wiki/ - **Didier Villers, UMONS - wiki**

Permanent link:
https://dvillers.umons.ac.be/wiki/teaching:progappchim:tkinter_gui_simple?rev=1426783363

Last update: **2015/03/19 17:42**