

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](#)

Un 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 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>
```

Champ d'entrée

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.pack()
```

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

```
b=Button(root,text="Click here !",command=action) b.grid(row=1) root.mainloop() </sxh>
```

Utiliser le texte rentré

En cliquant, on quitte et on écrit le texte rentré <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 : root.destroy() </sxh>
```

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 boîte 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>
```

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=1385041936

Last update: **2013/11/21 14:52**

