

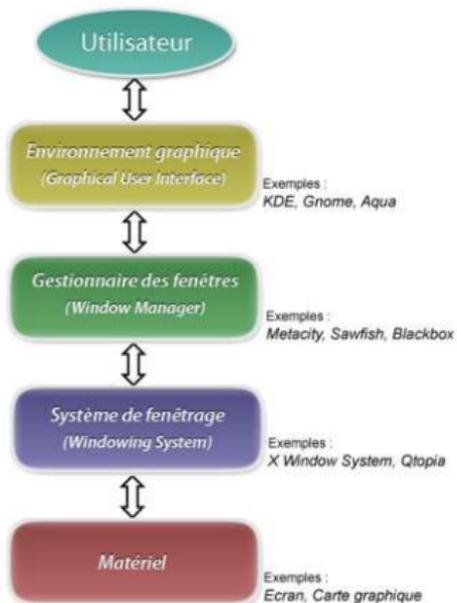
Interface Graphique en Python

April 22, 2008

Interface Graphique

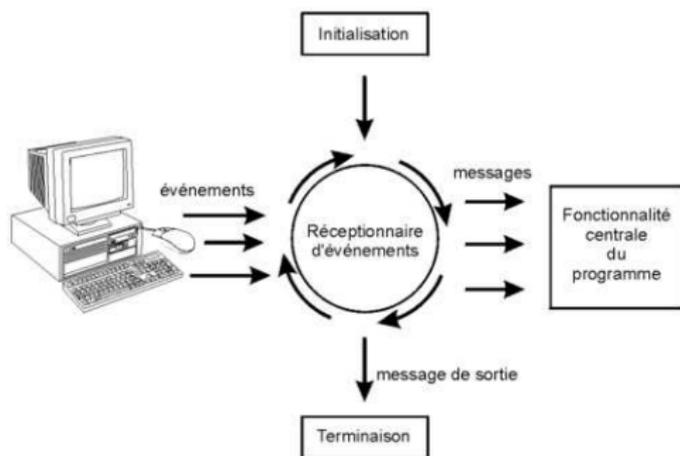
1. Connexion avec l'environnement graphique.
 - Gestion du contexte graphique, de la fenêtre, du clavier et de la souris.
 - Prise en compte des caractéristiques physiques.
 - Connexion à une bibliothèque spécialisée.
2. Gestion d'une boucle d'interaction avec l'utilisateur.
 - Définir un évènement.
 - Le relier à l'interface.
 - Ecrire la fonction correspondante: l'action.

Interface Graphique



Programmation Évènementielle

- Le programme va passer son temps à attendre des évènements - tels qu'un clic de l'utilisateur sur un bouton - et réagira en conséquence en exécutant certaines parties du code.
- Une fonction pré-définie est chargée de mettre en place la boucle d'écoute des évènements.



Python Tkinter

- Tkinter est une librairie dédiée à l'accès aux widgets Tk depuis le langage Python.
- Tk est l'une des bibliothèques graphiques les plus utilisées sous Unix/X-Window.
- site web :
<http://wikipython.flibuste.net/moin.py/Tkinter>
- Tkinter est disponible pour toutes les implémentations de Python.
- **Exemple :**

```
from Tkinter import *  
fenetre_principale = Tk()
```



Hello World

```
from Tkinter import *  
  
root = Tk()  
  
w = Label(root, text="Hello, world!")  
w.pack()  
  
root.mainloop()
```



Les Widgets

Widget : contraction de `windows gadget`. Les widgets sont tous les objets graphiques que l'on peut insérer dans une interface (fenêtre). Les principaux sont :

- **les boutons** : `Button` pour commander une action
- **Les labels** : `Label` pour insérer un texte
- **Les zones de saisie** : `Entry` pour entrer une donnée
- **Les canvas** : `Canvas` pour insérer des images

Chaque widget a des propriétés et des méthodes qui permettent de régler son apparence et les interactions de l'utilisateur.

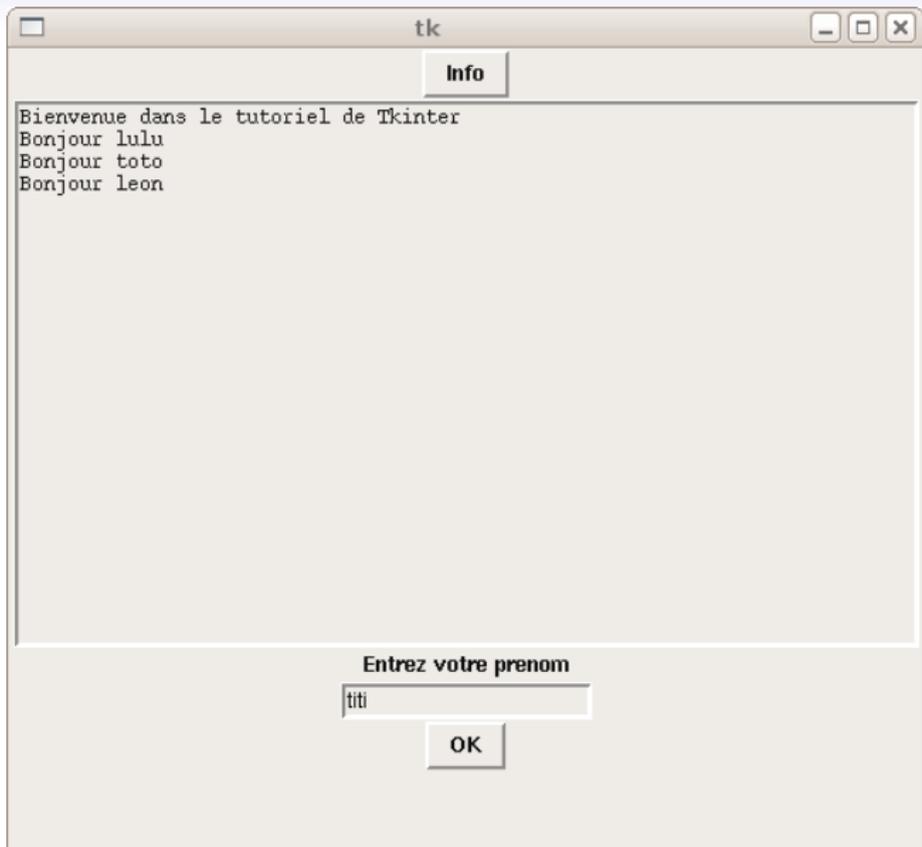
Créer un bouton

```
from Tkinter import *  
def aff_info():  
    lab=Label(root,text="Bienvenue dans le tutoriel de Tkinter")  
    lab.pack()  
  
root=Tk()  
but=Button(root,text="Info",command=aff_info)  
but.pack()  
root.mainloop()
```



Texte et zone de saisie

```
from Tkinter import *
def aff_info():
    txt.insert(END,"Bienvenue dans le tutoriel de Tkinter")
def get_prenom():
    prenom=ent.get()
    txt.insert(END,"Bonjour %s " %prenom)
root=Tk()
but=Button(root,text="Info",command=aff_info)
but.pack()
txt=Text(root)
txt.pack()
lab_ent=Label(root,text="Entrez votre prénom")
lab_ent.pack()
ent=Entry(root)
ent.pack()
b_valid=Button(root,text='OK',command=get_prenom)
b_valid.pack()
root.mainloop()
```



Créer un Canvas

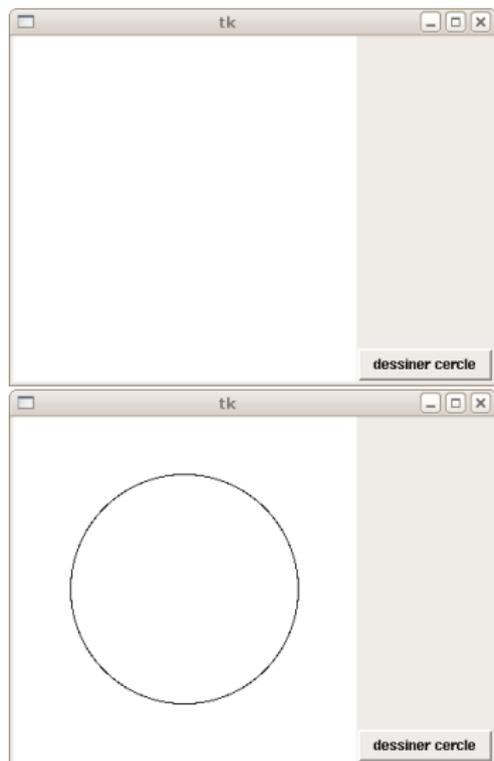
```
from Tkinter import *
from math import *
class App:
    def __init__(self, master):
        self.master=master
        self.fram_droite=Frame(master)
        self.fram_droite.pack(side='right')
        self.can=Canvas(self.fram_droite,bg='white',height=300,width=300)
        self.can.pack(side='left')
        self.b_cercle=Button(self.fram_droite,text='dessiner cercle',
                             command=self.dessine_cercle)
        self.b_cercle.pack(side='bottom')
    def dessine_cercle(self):
        self.can.delete(ALL)
        self.can.create_oval(50,50,250,250,tags="forme")

root=Tk()
App(root)
root.mainloop()
```

Créer un Canvas



Créer un Canvas



La suite

http://web.univ-pau.fr/bruel/Enseignements/TER/InterfacePython/tutoriel_kinter.html

http://sebsauvage.net/python/gui/index_fr.html

