



Le 06/08/2024

Encadre par :OTMANE HAQAY

---

## Presentation Interphase Graphiques PYTHON (Tkinter)

Realise Par : Yasser Khat, Saad AIT  
ALLOUCH, Souhaib  
EDDAHMANI, MOHAMMED REDA MENYANI

# Introduction

- **Présentation de Tkinter** : Tkinter est la bibliothèque standard pour créer des interfaces graphiques en Python.
- **Objectifs de la présentation** : Comprendre les bases de Tkinter, apprendre à créer des interfaces simples, découvrir des exemples pratiques.

# Qu'est-ce que Tkinter ?

- Tkinter est un module de base intégré dans Python , normalement vous n'avez rien à faire pour pouvoir l'utiliser. L'un des avantages de Tkinter est sa portabilité sur les OS les plus utilisés par le grand public.


# Installer Tkinter

- Tkinter est installé par défaut, si ce n'est pas le cas, lancez la commande suivante:

```
sudo apt-get install python-tk
```

- En python 3:

```
sudo apt-get install python3-tk
```



# Python 2, python 3

Les modules ne sont pas les mêmes suivant votre version de python. Si le message suivant apparaît lors de l'exécution de votre script:

```
ImportError: No module named 'Tkinter'
```

C'est que le module appelé n'est pas le bon par rapport à votre version python.

Python 2		Python 3
Tkinter	→	tkinter
Tix	→	tkinter.tix
ttk	→	tkinter.ttk
tkMessageBox	→	tkinter.messagebox
tkColorChooser	→	tkinter.colorchooser
tkFileDialog	→	tkinter.filedialog
tkCommonDialog	→	tkinter.commondialog
tkSimpleDialog	→	tkinter.simpdialog
tkFont	→	tkinter.font
Tkdnd	→	tkinter.dnd
ScrolledText	→	tkinter.scrolledtext

# Hello world

- Voici le code de hello world :

```
# coding: utf-8

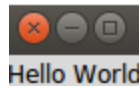
from tkinter import *

fenetre = Tk()

label = Label(fenetre, text="Hello World")
label.pack()

fenetre.mainloop()
```

- Une fenêtre comme celle-ci devrait apparaître:



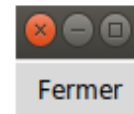
## Les widget Tkinter

Pour créer un logiciel graphique vous devez ajouter dans une fenêtre des éléments graphiques que l'on nomme widget. Ce widget peut être tout aussi bien une liste déroulante que du texte.

## Les boutons

Les boutons permettent de proposer une action à l'utilisateur. Dans l'exemple ci-dessous, on lui propose de fermer la fenêtre.

```
# bouton de sortie  
bouton=Button(fenetre, text="Fermer", command=fenetre.quit)  
bouton.pack()
```

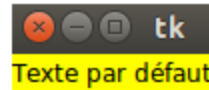


. . . .

## Les labels

Les labels sont des espaces prévus pour écrire du texte. Les labels servent souvent à décrire un widget comme un input

```
# label
label = Label(fenetre, text="Texte par défaut", bg="yellow")
label.pack()
```



## Entrée / input

```
# entrée
value = StringVar()
value.set("texte par défaut")
entree = Entry(fenetre, textvariable=string, width=30)
entree.pack()
```

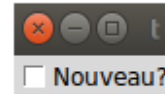




## Case à cocher

Les checkbox proposent à l'utilisateur de cocher une option.

```
# checkbox
bouton = Checkbutton(fenetre, text="Nouveau?")
bouton.pack()
```

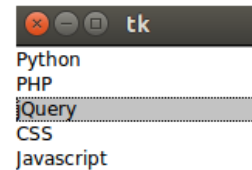


## Boutons radio

Les boutons radio sont des cases à cocher qui sont dans un groupe et dans ce groupe seul un élément peut être sélectionné.

```
# radiobutton
value = StringVar()
bouton1 = Radiobutton(fenetre, text="Oui", variable=value, value=1)
bouton2 = Radiobutton(fenetre, text="Non", variable=value, value=2)
bouton3 = Radiobutton(fenetre, text="Peu être", variable=value, value=3)
bouton1.pack()
bouton2.pack()
bouton3.pack()
```

- Les listes
- Les listes permettent de récupérer une valeur sélectionnée par l'utilisateur.



```
# liste
liste = Listbox(fenetre)
liste.insert(1, "Python")
liste.insert(2, "PHP")
liste.insert(3, "jQuery")
liste.insert(4, "CSS")
liste.insert(5, "Javascript")

liste.pack()
```

- Récupérer la valeur d'un input
- Pour récupérer la valeur d'un input il vous faudra utiliser la méthode `get()` :

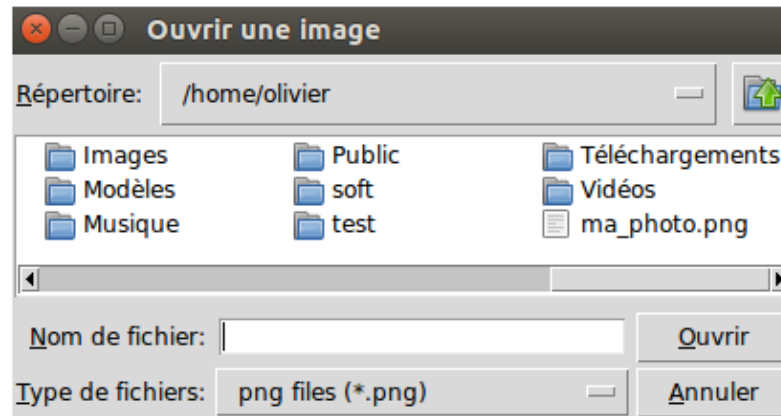
```
def recupere():  
    showinfo("Alerte", entree.get())  
  
value = StringVar()  
value.set("Valeur")  
entree = Entry(fenetre, textvariable=value, width=30)  
entree.pack()  
  
bouton = Button(fenetre, text="Valider", command=recupere)  
bouton.pack()
```

## Récupérer une image et l'afficher

Pour cela, vous devez importer le module suivant:

```
from tkinter.filedialog import *
```

```
filepath = askopenfilename(title="Ouvrir une image",filetypes=[('png files','*.png'),('all files','*.*)'])  
photo = PhotoImage(file=filepath)  
canvas = Canvas(fenetre, width=photo.width(), height=photo.height(), bg="yellow")  
canvas.create_image(0, 0, anchor=NW, image=photo)  
canvas.pack()
```



La fonction `askopenfilename` retourne le chemin du fichier que vous avez choisi avec le nom de celui-ci.

Exemple: `/home/olivier/ma_photo.png`

## Récupérer un fichier texte et l'afficher

```
filename = askopenfilename(title="Ouvrir votre document",filetypes=[('txt files','.txt'),('all files','.*')])
fichier = open(filename, "r")
content = fichier.read()
fichier.close()

Label(fenetre, text=content).pack(padx=10, pady=10)
```

## Les évènements

Vous pouvez récupérer les actions utilisateurs à travers les [events](#) (évènement en français).

Pour chaque widget, vous pouvez *binder* (lier en français) un évènement, par exemple dire lorsque l'utilisateur appuie sur telle touche, faire cela.

Voici un exemple qui récupère les touches appuyées par l'utilisateur:

```
def clavier(event):
    touche = event.keysym
    print(touche)

canvas = Canvas(fenetre, width=500, height=500)
canvas.focus_set()
canvas.bind("<Key>", clavier)
canvas.pack()
```

On remarque que l' évènement est encadré par des chevrons.

D'autres évènements existent:

<Button-1>	: Click gauche
<Button-2>	: Click milieu
<Button-3>	: Click droit
<Double-Button-1>	: Double click droit
<Double-Button-2>	: Double click gauche
<KeyPress>	: Pression sur une touche
<KeyPress-a>	: Pression sur la touche A (minuscule)
<KeyPress-A>	: Pression sur la touche A (majuscule)
<Return>	: Pression sur la touche entrée
<Escape>	: Touche Echap
<Up>	: Pression sur la flèche directionnelle haut
<Down>	: Pression sur la flèche directionnelle bas
<ButtonRelease>	: Lorsque qu'on relache le click
<Motion>	: Mouvement de la souris
<B1-Motion>	: Mouvement de la souris avec click gauche
<Enter>	: Entrée du curseur dans un widget
<Leave>	: Sortie du curseur dans un widget
<Configure>	: Redimensionnement de la fenêtre
<Map> <Unmap>	: Ouverture et iconification de la fenêtre
<MouseWheel>	: Utilisation de la roulette

# Les alertes

Pour pouvoir utiliser les alertes de votre os, vous pouvez importer le module `tkMessageBox` (Python 2).

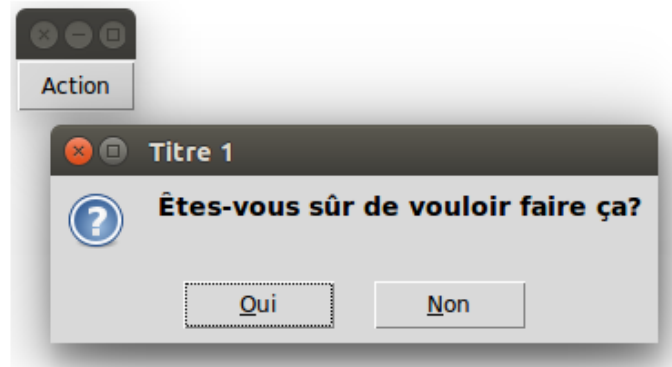
```
from tkMessageBox import *
```

Pour python 3:

```
from tkinter.messagebox import *
```

Exemple d'utilisation:

```
def callback():  
    if askyesno('Titre 1', 'Êtes-vous sûr de vouloir faire ça?):  
        showwarning('Titre 2', 'Tant pis...')  
    else:  
        showinfo('Titre 3', 'Vous avez peur!')  
        showerror("Titre 4", "Aha")  
  
Button(text='Action', command=callback).pack()
```



# Conclusion:

- **Résumé des Points Clés**

- **Tkinter** est la bibliothèque standard de Python pour créer des interfaces graphiques (GUI).
- Installation et configuration simples.
- Création facile d'interfaces utilisateur avec des widgets comme Label, Button, Entry.
- Gestion des dispositions (pack, grid, place) et des événements (clics de boutons, saisies de texte).
- Exemples de projets pratiques : calculatrice, gestionnaire de tâches, jeu Tic-Tac-Toe.

- **Importance de Tkinter dans le Développement Python**

- Intégré dans la bibliothèque standard de Python, idéal pour les débutants.
- Utilisé pour des applications de bureau simples à modérément complexes.