

# Interface Homme Machine

## Python-Tkinter

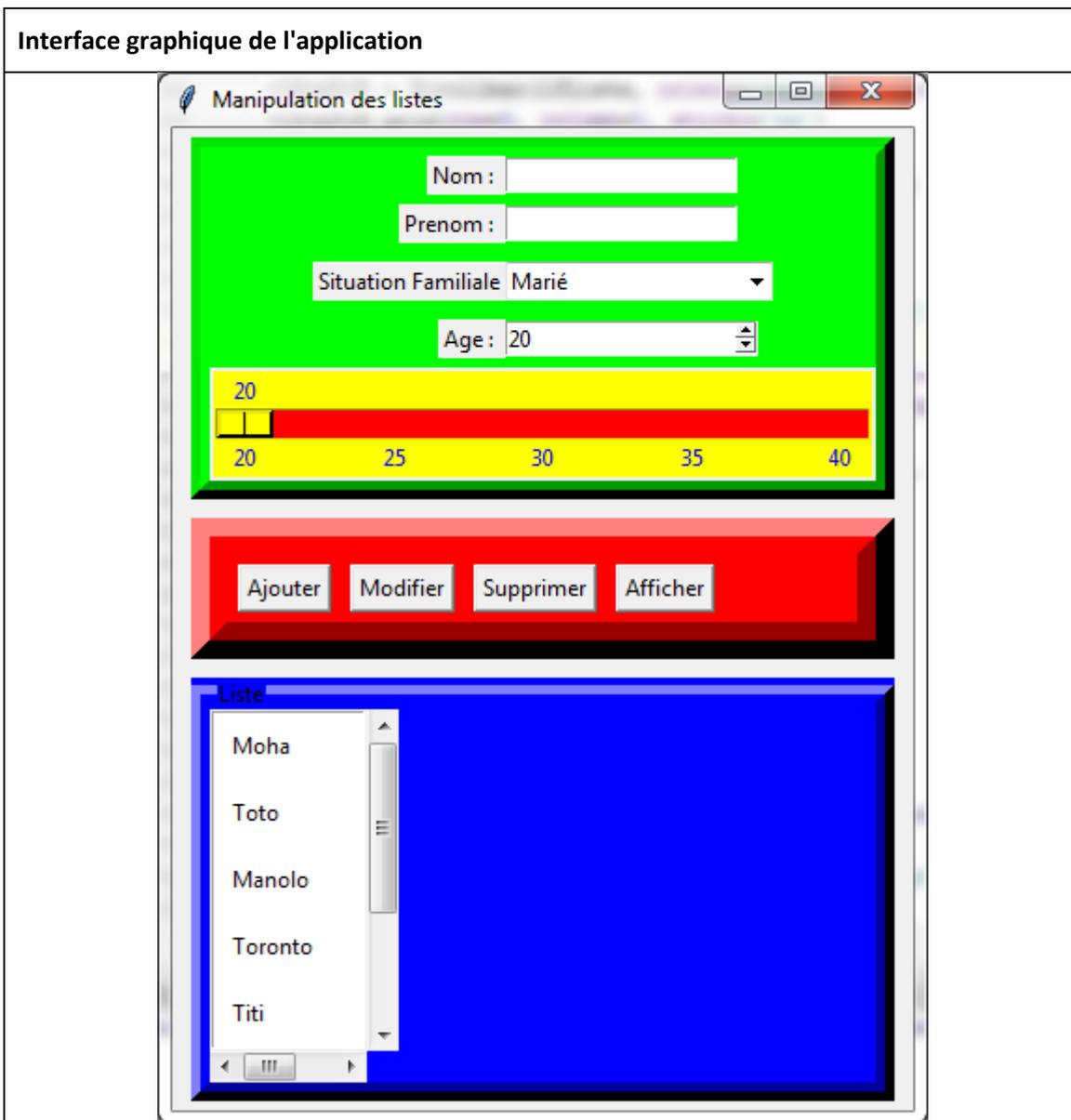
### Travaux Pratiques N° 3

#### Objetif :

- Réaliser une application avec une interface graphique qui permet de se familiariser avec les widgets ci-dessous:
  - ✓ Combobox (Zone de liste modifiable) du module ttk
  - ✓ Spinbox
  - ✓ Scale (Echelle)
  - ✓ Listbox (Zone de liste)
  - ✓ Scrolbar (Barre de défilement)
  - ✓ Boite d'affichage des messages (module messagebox)

L'interface graphique de l'application doit avoir le format ci-dessous.

#### Interface graphique de l'application





## Code complet source de l'application

```
from tkinter import *
from tkinter import ttk
from tkinter import messagebox
from functools import partial

def afficher(liste, indice):
    print(liste[indice])

root = Tk()
root.title("Manipulation des listes")
f = Frame(root, padx=10)
lfdonnees = Frame(f, bd=10, bg="#00FF00", relief="raised")

lbNom = Label(lfdonnees, text="Nom : ")
entNom = Entry(lfdonnees)

lbPrenom = Label(lfdonnees, text="Prenom : ")
entPrenom = Entry(lfdonnees)

lbSF = Label(lfdonnees, text="Situation Familiale")

#Utilisation du widget ttk.Combobox
lcbSF = ["Célibataire", "Marié", "Mariée", "Divorcé", "Divorcée", "Veuf"]
cbSFVar = StringVar(value="Marié")
cbSF = ttk.Combobox(lfdonnees, textvariable = cbSFVar, values=lcbSF)

#Définition d'un Spinbox
ageVar = IntVar(lfdonnees)
lbAge = Label(lfdonnees, text="Age : ")
spboxAge = Spinbox(lfdonnees, from_=20, to=40, increment=1, wrap=True,
textvariable=ageVar)

scaleAge = Scale(lfdonnees, orient='horizontal', from_=20, to=40,
resolution=1, tickinterval=5, length=350,
troughcolor="red", bg="yellow",
fg="blue", variable=ageVar)

lbNom.grid(row=0, column=0, sticky=E)
entNom.grid(row=0, column=1, sticky=W)
lbPrenom.grid(row=1, column=0, sticky=E, pady=5)
entPrenom.grid(row=1, column=1, sticky=W, pady=5)
lbSF.grid(row=2, column=0, sticky="e", pady=5)
cbSF.grid(row=2, column=1, stick="w", pady=5)
lbAge.grid(row=3, column=0, sticky="e", pady=5)
spboxAge.grid(row=3, column=1, sticky="w", pady=5)
scaleAge.grid(row=4, column=0, columnspan=2)
lfdonnees.grid(row=0, column=0, sticky="nsew", pady=5)

lcbSF.append("Veuve")
cbSF.configure(values=lcbSF)

lfListe = LabelFrame(f, text="Liste", bd=10, bg="blue", relief="raised")

#Définition d'une Lisbox et ses Scrolbar
vLNomScB = Scrollbar(lfListe, orient='vertical', relief="raised") #Barre
verticle
vLNomScB.grid(row=0, column=1, sticky='ns')

hLNomScB = Scrollbar(lfListe, orient='horizontal') #Barre horizontal
hLNomScB.grid(row=1, column=0, sticky='ew')
```

```

liste = ["Moha", "Toto", "Manolo", "Toronto", "Titi", "Tata",
"ABCDEFGHIJKLMN"]
lnom = StringVar(value=liste)
listNom = Listbox(lfListe, listvariable = lnom, activestyle="dotbox",
selectborderwidth=10, selectbackground="purple",
selectforeground="yellow",
selectmode="multiple", width=10, height=5,
xscrollcommand=hLNomScB.set,
yscrollcommand=vLNomScB.set)

hLNomScB['command'] = listNom.xview
vLNomScB['command'] = listNom.yview

liste.append("Habti")
lnom.set(liste)

listNom.grid(row=0, column=0, sticky="ew")
lfListe.grid(row=2, column=0, sticky="nsew", pady=5, padx=5)

lfactions = Frame(f, borderwidth=2, bd=20, bg="red", relief="raised")
btnAjouter = Button(lfactions, text="Ajouter")
btnModifier = Button(lfactions, text="Modifier")
btnSupprimer = Button(lfactions, text="Supprimer")
btnAfficher = Button(lfactions, text="Afficher")

btnAfficher["command"] = lambda: messagebox.showinfo("bouton afficher" ,
"bouton affiche clike")

btnAjouter.pack(side=LEFT, padx=5)
btnModifier.pack(side=LEFT, padx=5)
btnSupprimer.pack(side=LEFT, padx=5)
btnAfficher.pack(side=LEFT, padx=5)
lfactions.grid(row=1, column=0, stick="nswe", pady=5, ipady=5)

f.pack()

root.mainloop()

```

## Annexe1 : widget tkinter.Listbox : propriétés et méthodes

class <b>Listbox</b> (parent, option, ...)	
Ce constructeur retourne un objet instance de Listbox. L'argument option peut prendre les valeurs ci-dessous	
Option	Valeur
activestyle	Cette option sert à préciser l'apparence de la ligne active. Elle peut prendre l'une de ces valeurs: <ul style="list-style-type: none"> <li>➤ 'underline' - la ligne active est soulignée (valeur par défaut) ;</li> <li>➤ 'dotbox' - La ligne active est mise en valeur par une boîte pointillée ;</li> <li>➤ 'none' - aucune mise en valeur.</li> </ul>
bg – (ou background)	Couleur de fond de la liste de sélection. Voir : Attributs standards des widgets
bd – (ou borderwidth)	Largeur de la bordure de la liste de sélection. 2 pixels par défaut. Voir : Attributs standards des widgets
cursor	Cursor utilisé lorsque la souris survole le widget. Voir : Attributs standards des widgets
disabledforeground	Couleur du texte lorsque l'état du widget est 'disabled'.
exportselection	Par défaut, l'utilisateur peut sélectionner le texte à la souris qui est alors copié dans le presse-papier. Pour désactiver ce comportement, utiliser exportselection=0.
font	Police de caractère utilisée pour le texte de la liste de sélection. Voir : Attributs standards des widgets
fg – (ou foreground)	La couleur utilisée pour le texte. Voir : Attributs standards des widgets
height	Nombre de lignes et des pixels montrées dans la liste de sélection. 10 par défaut.
highlightbackground	Couleur de la ligne de focus lorsque le widget ne l'a pas.
highlightcolor	Couleur de la ligne de focus lorsque le widget l'obtient.
highlightthickness	Épaisseur de la ligne de focus.
listvariable	Une StringVar qui est associée à la liste de sélection dans son ensemble. (Voir <b>Variables de contrôle</b> ). L'appel de la méthode get() de cette variable de contrôle retourne une chaîne de la forme " ('ligne0', 'ligne1', ...)" où chaque lignei est le contenu de lignei ligne de la boîte de sélection. Pour modifier toutes les lignes de la boîte, appelez la méthode set(s) sur la variable de contrôle, où s est une chaîne qui contient les valeurs de chaque ligne séparées avec des espaces. Par exemple, si listVar est une StringVar associé à l'option listvariable d'une boîte de sélection, l'appel listVar.set('un deux trois') remplira la boîte avec trois lignes et l'appel listVar.get() retournera "(un', 'deux', 'trois)".
relief	Sert à régler le relief de la bordure. 'sunken' par défaut. Voir : Attributs standards des widgets
selectbackground	Couleur de fond utilisée pour les lignes sélectionnées.
selectborderwidth	Épaisseur de la bordure des lignes de texte sélectionnées. 0 par défaut. Elle est utilisée pour produire un effet de relief 'raised' plus ou moins fort autour du texte sélectionné.
selectforeground	Couleur du texte sélectionné.
selectforeground	Couleur du texte sélectionné.
selectmode	Détermine le nombre d'items qu'il est possible de sélectionner et la gestion du cliquer-glisser sur la sélection. <ul style="list-style-type: none"> <li>➤ 'browse' - Valeur par défaut, le cliquerglisser modifie la sélection.</li> <li>➤ 'single' - Une seule ligne peut être sélectionnée et il n'est pas possible de déplacer la sélection par cliquer-glisser.</li> <li>➤ 'multiple' – Vous pouvez sélectionner plusieurs lignes à la fois. Le fait de cliquer sur une ligne déjà sélectionnée la désélectionne et vice versa.</li> <li>➤ 'extended' - vous pouvez sélectionner des lignes adjacentes par cliquer-glisser.</li> </ul>
state	Valeurs possibles : 'normal' : valeur Par défaut, la liste de sélection est dans l'état normal. 'disabled' : désactive la liste'.
takefocus	Ce widget obtient le focus normalement. Mettre 0 pour sortir ce widget de la liste de «traversée du focus». Voir : Attributs standards des widgets

width	La largeur du widget mesurée en caractères et non en pixels. La largeur effective est basée sur la largeur moyenne des caractères de la fonte utilisée. La valeur par défaut est 20.
xscrollcommand	Si vous souhaitez que l'utilisateur puisse faire défiler la liste horizontalement, vous pouvez lier votre liste de sélection à une barre de défilement horizontale. Configurer cette option avec la méthode set() de la barre de défilement.
yscrollcommand	Similaire à l'option précédente mais pour un défilement vertical.

## Les méthodes

Les méthodes des listes de sélection utilisent fréquemment des index. Un index peut être spécifié en utilisant les valeurs suivantes :

- 'n' : n est un entier qui correspond à la ligne de la liste de sélection qui possède cet index, en comptant à partir de 0.
- 'end' : correspond à la dernière ligne de la liste de sélection.
- 'active' : correspond à la ligne sélectionnée. Si la liste de sélection permet la multisélection, il correspond à la dernière ligne sélectionnée.
- '@x,y' : correspond à la ligne qui est la plus proche du point de coordonnées (x,y) relativement au coin supérieur gauche du widget.

Le tableau ci-dessous présente quelques méthodes des listes de sélection :

Méthodes	Description
<b>activate</b> (index)	Sélectionne la ligne ayant l'index indiqué.
<b>bbox</b> (index)	Abréviation de " <b>bounding box</b> ". Retourne la boîte englobante de la ligne ayant l'index indiqué sous la forme d'un tuple à 4 éléments (x, y, largeur, hauteur), où le pixel le plus en haut et à gauche de cette ligne est situé en (x,y) et la largeur et hauteur sont données en pixels. La largeur correspond à la partie de la ligne qui contient le texte. Si la ligne de numéro index n'est pas visible, cette méthode retourne None. Si elle est partiellement visible, la boîte englobante peut excéder la zone visible.
<b>curselection</b> ()	Retourne un tuple qui contient les numéros ou index de la ou des lignes sélectionnées, en comptant à partir de 0. Si aucune ligne n'est sélectionnée, le tuple est vide.
<b>delete</b> (debut, fin=None)	Supprime les lignes dont les indices sont dans l'intervalle [debut, fin] (extrémités incluses). Si le deuxième argument est omis, seule la ligne d'index debut est supprimée.
<b>get</b> (debut, fin=None)	Retourne un tuple qui contient les textes des lignes dont les indices appartiennent à l'intervalle [deb, fin]. Si le deuxième argument est omis, seul le texte de la ligne d'indice debut est retourné.
<b>index</b> (i)	Si c'est possible, positionne la partie visible de la liste de sélection de telle sorte que la ligne numéro i soit tout en haut de la liste.
<b>insert</b> (index, *elements)	Insert une ou plusieurs lignes (autant que d'éléments fournis après le premier argument) dans la liste avant la ligne de numéro index. Utiliser 'end' comme premier argument si vous souhaitez ajouter de nouvelles lignes à la fin de la liste.
<b>itemcget</b> (index, option)	Retourne l'une des valeurs d'option de la ligne de numéro index de la liste. Pour les options possibles, voir la méthode itemconfig() ci-dessous. Si l'option donnée n'a pas été configurée pour la ligne indiquée, la valeur de retour est une chaîne vide.
<b>itemconfig</b> (index, option=value, ...)	Modifie une ou des options de configuration de la ligne de numéro index. Les options incluent les paramètres : background : La couleur de fond de la ligne. foreground : La couleur du texte de la ligne. selectbackground : La couleur de fond utilisée lorsque la ligne est sélectionnée. selectforeground : La couleur du texte utilisée lorsque la ligne est sélectionnée.
<b>nearest</b> (y)	Retourne l'index de la ligne visible la plus proche du niveau y (vertical) exprimé en pixels relativement au bord supérieur du widget.
<b>scan_dragto</b> (x, y)	Voir la méthode scan_mark() ci-dessous.
<b>scan_mark</b> (x, y)	Utilisez cette méthode pour implémenter le défilement rapide de la liste de sélection à la souris. Pour réaliser cette fonctionnalité, lier un événement «appui sur l'un des boutons de la souris» à un gestionnaire qui se chargera d'appeler la méthode

	scan_mark() à la position courante de la souris. Ensuite, lier l'événement «déplacement de la souris» (Motion) à un gestionnaire qui appellera scan_dragto() avec la position courante de la souris. La liste de sélection défilera alors à un rythme proportionnel à la distance qui sépare la position enregistrée par scan_mark et la position courante.
<b>see(index)</b>	Ajuste la position de la liste de sélection de telle sorte que la ligne de numéro index soit visible.
<b>selection_anchor(index)</b>	Positionne l'«ancrage de sélection» sur la ligne de numéro index. Une fois que cette ancre a été positionnée, vous pouvez y faire référence en utilisant l'index spécial 'anchor'. Par exemple, si votre liste est lbox, ces instructions sélectionneront les lignes 3, 4 et 5: lbox.selection_anchor(3) lbox.selection_set(tk.ANCHOR,5)
<b>selection_clear(debut, fin=None)</b>	Désélectionne toutes les lignes dont les index appartiennent à l'intervalle [debut, fin]. Si le second argument est omis, seule la ligne de numéro debut est désélectionnée.
<b>selection_includes(index)</b>	Retourne 1 si la ligne d'index donné est sélectionnée et retourne 0 autrement.
<b>selection_set(debut, fin=None)</b>	Sélectionne toute les lignes dont les index appartiennent à l'intervalle [debut, fin]. Si le deuxième argument est omis, seule la ligne d'index debut est sélectionnée.
<b>size()</b>	Retourne le nombre de lignes de la liste de sélection.
<b>xview()</b>	Pour faire défiler la liste horizontalement, configurez l'option command du widget barre de défilement horizontale avec cette méthode. <code>horizontalDefilBare['command'] = listbox.xview</code> Voir <a href="#">Défilement d'une liste de sélection</a> .
<b>xview_moveto(fraction)</b>	Fait défiler la liste de sélection horizontalement de telle sorte que le côté gauche de la fraction de sa ligne la plus longue soit placé contre le bord gauche de la zone visible. L'argument fraction appartient à l'intervalle [0,1].
<b>xview_scroll(nombre, quoi)</b>	Fait défiler la liste de sélection horizontalement. Pour l'argument quoi, utiliser soit: <ul style="list-style-type: none"> <li>➤ 'units' pour un défilement d'unité «un caractère»,</li> <li>➤ 'pages' pour un défilement où l'unité est la «largeur effective de la liste de sélection».</li> </ul> L'argument nombre indique le nombre d'unités du défilement: <ul style="list-style-type: none"> <li>➤ les valeurs négatives font défiler vers la droite,</li> <li>➤ les valeurs positives vers la gauche.</li> </ul>
<b>yview()</b>	Similaire à la méthode xview(), mais pour un défilement vertical.
<b>yview_moveto(fraction)</b>	Similaire à la méthode xview_moveto() pour un défilement vertical.
<b>yview_scroll(nombre, quoi)</b>	Similaire à la méthode xview_scroll() mais pour un défilement vertical. 'units' se réfère à l'unité «ligne» et 'pages' à l'unité «hauteur visible de la liste».

## Défilement d'une liste de sélection

Le programme ci-dessous présente comment créer une Listbox avec une barre de défilement horizontal et vertical.

```

from tkinter import *

root = Tk()
root.title("Listbox ave Scrolbar horizontal et vertical ")
#Définition d'une Scrollbar pour le défilement vertical
verticalScrolbar = Scrollbar(root, orient='vertical')

#Définition d'une Scrollbar pour le défilement horizontal
horizontalScrolbar = Scrollbar(root, orient='horizontal')

#Définition d'une list qui contient les valeurs à afficher dans la liste
listValue = ["Moha", "Toto", "Manolo", "Tronto", "ABCDEFGHJKLMN", "Habti", "Bardi",
"Tarhi", "Ouiddi"]
listVar = StringVar(value=listValue)

#Définition d'une Listbox ayant horizontalScrolbar comme valeur de l'option

```

```
xscrollcommand.  
#et verticalScrolbar comme valeur de l'option yscrollcommand.  
listbox = Listbox(root, listvariable = listVar,  
xscrollcommand=horizontalScrolbar.set, yscrollcommand=verticalScrolbar.set, width=10,  
height=7)  
  
horizontalScrolbar['command'] = listbox.xview  
verticalScrolbar['command'] = listbox.yview  
  
verticalScrolbar.grid(row=0, column=1, sticky='ns')  
listbox.grid(row=0, column=0, sticky='nsew')  
horizontalScrolbar.grid(row=1, column=0, sticky='ew')  
  
root.mainloop()
```