

Interface Homme Machine

Python-Tkinter

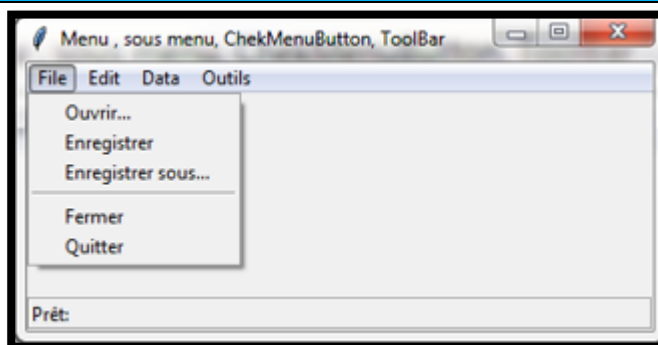
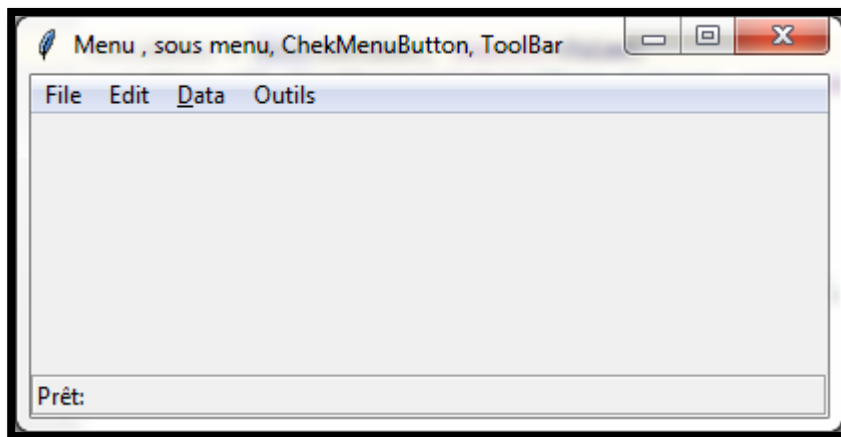
Travaux Pratiques N° 4

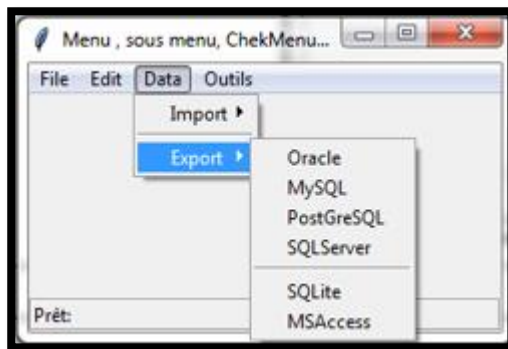
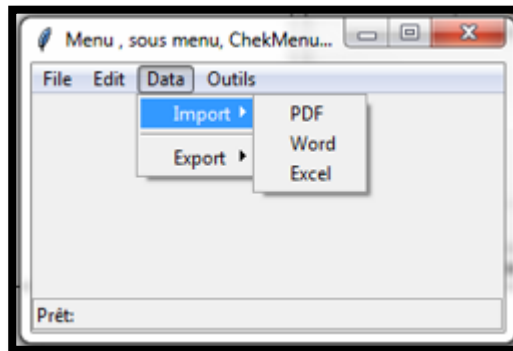
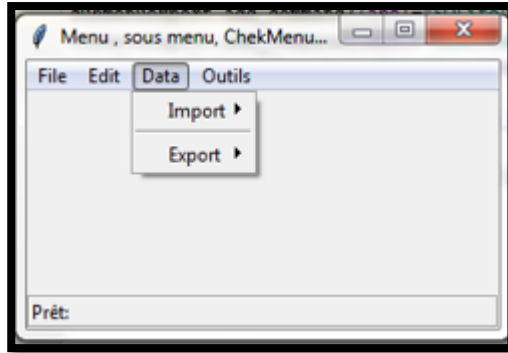
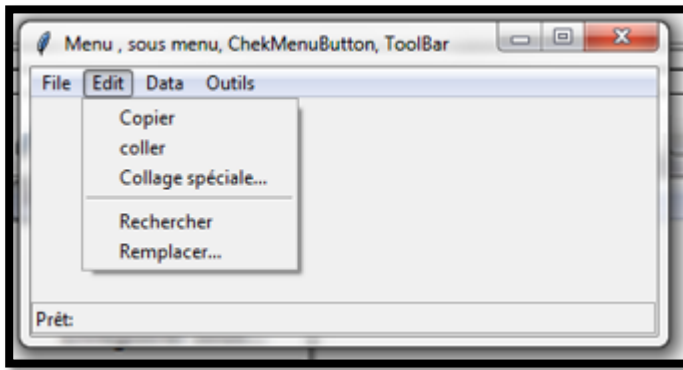
Objetif :

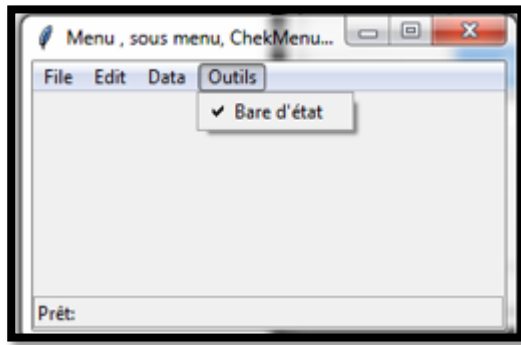
- Réaliser une application avec une interface graphique qui permet de se familiariser avec les:
 - ✓ Menus simples
 - ✓ Menus contextuels
 - ✓ Sous menus
 - ✓ Barre d'état

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 messagebox
from functools import partial

def afficher(s):
    messagebox.showinfo("Choix d'un Menu", "Menu : " + s + " : a été cliqué" )

def onExit():
    quit()

root = Tk()
root.geometry("300x150+300+300")
root.title("Menu , sous menu, ChekMenuButton, PopupMenu")

#Création de la barre de menu principale
menubar = Menu(root)
root.config(menu=menubar)

#Création d'un premier menu simple
fileMenu = Menu(menubar, tearoff=False)
fileMenu.add_command(label="Ouvrir...", command=partial(afficher, "Ouvrir"))
fileMenu.add_command(label="Enregistrer", command=partial(afficher, "Enregistrer"))
fileMenu.add_command(label="Enregistrer sous...", command=partial(afficher,
"Enregistrer sous"))
fileMenu.add_separator() #Ligne séparateur
fileMenu.add_command(label="Fermer", command=onExit)
fileMenu.add_command(label="Quitter", command=onExit)

#Ajout du menu à la barre de menu : Il faut utiliser add_cascade
menubar.add_cascade(label="File", menu=fileMenu)

#Création d'un deuxième meun simple
editMenu = Menu(menubar, tearoff=False)
editMenu.add_command(label="Copier", command=partial(afficher, "Copier"))
editMenu.add_command(label="Coller", command=partial(afficher, "Coller"))
editMenu.add_command(label="Collage spéciale...", command=partial(afficher, "Collage
spéciale"))
editMenu.add_separator()
editMenu.add_command(label="Rechercher", command=quit)
editMenu.add_command(label="Remplacer...", command=quit)

#Ajout du menu à la barre de menu : Il faut utiliser add cascade
menubar.add_cascade(label="Edit", menu=editMenu)

#Création d'un popupmenu
def showMenu(e):
    popupmenu.post(e.x_root, e.y_root)

def doMinimize():
    root.iconify()

popupmenu = Menu(root, tearoff=False)
popupmenu.add_command(label="Minimize", command=doMinimize)
popupmenu.add_command(label="Exit", command=onExit)

root.bind("<Button-3>", showMenu)
```

```

#Création d'un menu avec sous menu
dataMenu = Menu(menuubar, tearoff=False)

submenuimport = Menu(dataMenu, tearoff=False)
submenuimport.add_command(label="PDF")
submenuimport.add_command(label="Word")
submenuimport.add_command(label="Excel")
dataMenu.add_cascade(label='Import', menu=submenuimport, underline=0)

dataMenu.add_separator()

submenuexport = Menu(dataMenu, tearoff=False)
submenuexport.add_command(label="Oracle")
submenuexport.add_command(label="MySQL")
submenuexport.add_command(label="PostGreSQL")
submenuexport.add_command(label="SQLServer")
submenuexport.add_separator()
submenuexport.add_command(label="SQLite")
submenuexport.add_command(label="MSAccess")
dataMenu.add_cascade(label='Export', menu=submenuexport, underline=0)

menuubar.add_cascade(label="Data", underline=0, menu=dataMenu)

```

```

#Création d'un CheckMenuButton pour afficher une barre d'état
def onClick():
    if (showStat.get() == True):
        sb.pack(side=BOTTOM, fill=X)
    else:
        sb.pack_forget()

showStat = IntVar()
showStat.set(1)

toolsMenu = Menu(menuubar, tearoff=False)
toolsMenu.add_checkbutton(label="Bare d'état", command=onClick, variable=showStat,
onvalue=1, offvalue=0)
menuubar.add_cascade(label="Outils", menu=toolsMenu)

svar = StringVar()
svar.set("Prêt:")

sb = Label(root, textvariable = svar, relief="groove", anchor="w")
sb.pack(side=BOTTOM, fill=X)

root.mainloop()

```