

Tkinter

Navigation : [Accueil](#) -> [Python](#) -> Tkinter



Sommaire

- 1 Liens utiles
- 2 Introduction
 - ◆ 2.1 Ouvrir une fenêtre de démo
 - ◆ 2.2 Première fenêtre simple
- 3 Créer des objets graphiques
 - ◆ 3.1 Liste des widgets
 - ◆ 3.2 Les boutons (tk02.py)
 - ◆ 3.3 Le canvas (tk03.py)
 - ◆ 3.4 Le checkbox (tk04.py)
 - ◆ 3.5 Le widget Entry (tk05.py)
 - ◆ 3.6 Le radiobutton (tk06.py)
 - ◆ 3.7 La listbox (tk07.py)

Liens utiles[[modifier](#)]

<https://docs.python.org/fr/3/library/tkinter.html>

Introduction[[modifier](#)]

Le package tkinter (Interface TK) est un module qui encapsule la boîte à outils Tk.

- Tk, ne fait pas partie de python, il est maintenu par la société **Active State**
- Tk est une bibliothèque d'interfaces graphiques multiplate-forme. Conçue par John Ousterhout. (source wikipédia)

Ouvrir une fenêtre de démo[[modifier](#)]

La commande suivante , à lancer dans une invite de commande , permet de tester l'installation du package tkinter.

```
python -m tkinter
```

Si le module est absent, installez le avec la commande :

```
apt-get -y install python3-tk
```

Premiere fenêtre simple[modifier]

```
import tkinter as tk

class Application(tk.Frame):
    def __init__(self, master=None):
        super().__init__(master)
        self.master = master
        self.pack()
        self.create_widgets()

    def create_widgets(self):
        self.clickMe = tk.Button(self)
        self.clickMe["text"] = "Hello Wonderfull World\n(click me) "
        self.clickMe["command"] = self.dit_bonjour
        self.clickMe.pack(side="top")

        self.quit = tk.Button(self, text="QUIT", fg="red",
                               command=self.master.destroy)
        self.quit.pack(side="bottom")

    def dit_bonjour(self):
        print("Bonjour, everybody!")

root = tk.Tk()
app = Application(master=root)
app.mainloop()
```

- La classe `Application` hérite de la classe `tk.Frame`
- La fonction **`create_widgets()`** est appelée dans le constructeur pour créer les composants graphiques.
- `self.clickMe = tk.Button(self)` : création du bouton.
- On renseigne les propriétés des composants visuels :
 - ◆ `self.clickMe["text"] = "Hell...."`
 - ◆ `self.clickMe["command"] = self.dit_bonjour`
- La méthode **`dit_bonjour`** (fonction callback) est appelée lorsque l'utilisateur clique sur le bouton.

Créer des objets graphiques[modifier]

Liste des widgets[modifier]

No.	Description
1	Le widget Bouton permet d'afficher des boutons dans votre application.

- 2 Le widget **Canvas** permet de dessiner des formes, telles que des lignes, des ovales, des polygones et des rectangles, dans votre application.
- 3 Le widget **Checkbutton** permet d'afficher un certain nombre d'options sous forme de cases à cocher. L'utilisateur peut sélectionner plusieurs options à la fois.
- 4 Le widget **Entry** est utilisé pour afficher un champ de texte d'une seule ligne permettant d'accepter les valeurs d'un utilisateur.
- 5 Le widget **Frame** est utilisé en tant que widget conteneur pour organiser d'autres widgets.
- 6 Le widget **Label** est utilisé pour fournir une légende à une ligne pour les autres widgets. Il peut aussi contenir des images.
- 7 Le widget **Listbox** est utilisé pour fournir une liste d'options à un utilisateur.
- 8 Le widget **Menubutton** est utilisé pour afficher les menus dans votre application..
- 9 Le widget **Menu** est utilisé pour fournir diverses commandes à un utilisateur. Ces commandes sont contenues dans Menubutton.
- 10 Le widget **Message** est utilisé pour afficher des champs de texte multilignes permettant d'accepter les valeurs d'un utilisateur..
- 11 Le widget **Radiobutton** est utilisé pour afficher un certain nombre d'options sous forme de boutons radio. L'utilisateur ne peut sélectionner qu'une option à la fois.
- 12 Le widget **Scale** est utilisé pour fournir un widget curseur.
- 13 Le widget **Scrollbar** est utilisé pour ajouter une fonctionnalité de défilement à divers widgets, tels que les zones de liste..
- 14 Le widget **Text** est utilisé pour afficher du texte sur plusieurs lignes..
- 15 Le widget **Toplevel** est utilisé pour fournir un conteneur de fenêtre séparé..
- 16 Le widget **Spinbox** est une variante du widget standard Tkinter Entry, qui peut être utilisé pour sélectionner un nombre fixe de valeurs.
- 17 **PanedWindow** est un widget conteneur pouvant contenir un nombre quelconque de volets, disposés horizontalement ou verticalement..
- 18 Un **LabelFrame** est un simple widget conteneur. Son objectif principal est de servir d'espaceur ou de conteneur pour les dispositions complexes de fenêtres..
- 19 **tkMessageBox** est utilisé pour afficher des boîtes de message dans vos applications..

Les boutons (tk02.py)[modifier]

```
import tkinter as tk
from tkinter import messagebox
```

```
mainFrame= tk.Tk()
```

```
""" Fonction callback """
```

```
def helloCallBack():
    messagebox.showinfo( "Titre Message", "Texte du message")

button = tk.Button(mainFrame, text ="Btn clic", command = helloCallBack)

button.pack()
mainFrame.mainloop()
```

Le canvas (tk03.py)[[modifier](#)]

Permet de dessiner :

- des lignes,
- des polygones,
- des rectangles
- des ovales
- ...

```
import tkinter as tk

mainFrame = tk.Tk()

canvas = tk.Canvas(mainFrame, bg="blue", height=250, width=300)

coord = 10, 50, 240, 210
angle=150
arc = canvas.create_arc(coord, start=0, extent=angle, fill="lightblue")

canvas.pack()
mainFrame.mainloop()
```

Le checkbox (tk04.py)[[modifier](#)]

```
import tkinter as tk
from tkinter import messagebox
from tkinter import Checkbutton

mainFrame = tk.Tk()
""" Callback """
def checkboxCallBack():
    str="Securisation:{0}\nAccélération:{1},\nTerminée:{2}".format(checkVar1.get(), checkVar2.get(), checkVar3.get())
    messagebox.showinfo("Résultat", str)
""" IntVar() est un wrapper pour des valeurs entières """
checkVar1 = tk.IntVar()
checkVar2 = tk.IntVar()
checkVar3 = tk.IntVar()
chkbtn1 = Checkbutton(mainFrame, text = "Sécurisation", variable = checkVar1,
                      onvalue = 1, offvalue = 0, height=2, \
                      width = 20, command = checkboxCallBack)
chkbtn2 = Checkbutton(mainFrame, text = "Accélération", variable = checkVar2,
                      onvalue = 1, offvalue = 0, height=2, \
                      width = 20, command = checkboxCallBack)
chkbtn3 = Checkbutton(mainFrame, text = "Terminée", variable = checkVar3, \
                      onvalue = 1, offvalue = 0, height=2, \
```

```

width = 20, command = checkboxCallBack)
chkbtn1.pack()
chkbtn2.pack()
chkbtn3.pack()
mainFrame.mainloop()

```

Le widget Entry (tk05.py)[[modifier](#)]

```

import tkinter as tk
from tkinter import messagebox
from tkinter import Label
from tkinter import Entry

mainFrame = tk.Tk()
label1 = Label(mainFrame, text="Nom")
label1.grid(row=0, column=0)

entry1 = Entry(mainFrame, bd =5)
entry1.grid(row=0, column=1)

label2 = Label(mainFrame, text="Prenom")
label2.grid(row=1, column=0)

entry2 = Entry(mainFrame, bd =5)
entry2.grid(row=1, column=1)

def btnCallBack():
    str = "Les valeurs sont\n - nom:{0},\n - prenom:{1}".format(entry1.get(),
        messagebox.showinfo( "Les valeurs ", str)

button = tk.Button(mainFrame, text ="Voir", command = btnCallBack)
button.grid(row=2, column=0, columnspan=2)

mainFrame.mainloop()

```

Le radiobutton (tk06.py)[[modifier](#)]

```

import tkinter as tk
from tkinter import messagebox
from tkinter import Radiobutton
from tkinter import Label

def sel():
    selection = "Vous avez sélectionné l'option " + str(var.get())
    label.config(text = selection)

mainFrame = tk.Tk()
var = tk.IntVar()
R1 = Radiobutton(mainFrame, text="Option 1", variable=var, value=1,
    command=sel)
R1.pack( anchor = tk.W )

```

```

R2 = Radiobutton(mainFrame, text="Option 2", variable=var, value=2,
                  command=sel)
R2.pack( anchor = tk.W )

R3 = Radiobutton(mainFrame, text="Option 3", variable=var, value=3,
                  command=sel)
R3.pack( anchor = tk.W)

label = Label(mainFrame)
label.pack()
mainFrame.mainloop()

```

La listbox (tk07.py)[[modifier](#)]

Exemple illustrant une listbox avec scrollbar

```

import sys
import tkinter as tk
from tkinter import messagebox
from tkinter import Listbox
from tkinter import Scrollbar
from tkinter import Label

mainFrame= tk.Tk()

compteur = 0

# def listboxCallback():
#     sys.stdout.write("Ajout d'un item")

scrollbar = Scrollbar(mainFrame)
scrollbar.pack( side = tk.RIGHT, fill = tk.Y )

listbox = Listbox(mainFrame, yscrollcommand = scrollbar.set)
scrollbar.config( command = listbox.yview )

""" Fonction callback """
def addCallBack():
    global compteur
    compteur += 1
    listbox.insert(tk.END, "item n{0}".format(compteur))
    #messagebox.showinfo( "Titre Message", "Texte du message")
listbox.pack()

button = tk.Button(mainFrame, text ="ajoute un item", command = addCallBack)

button.pack()
mainFrame.mainloop()

```