Tutoriel Tkinter

Tutoriel Tkinter

Pour aller plus loin : TkDocs Tutorial

Code de base

Créer une fenêtre vide :

```
import tkinter as tk
from tkinter import ttk

root = tk.Tk()

root.mainloop()
```

Ajouter des boutons dans la fenêtre :

```
import tkinter as tk
from tkinter import ttk

root = tk.Tk()

bt1 = ttk.Button(root, text="Bonjour")
bt1.grid(column=0, row=0)

bt2 = ttk.Button(root, text="Au revoir")
bt2.grid(column=1, row=0)

root.mainloop()
```

Positionnement

On peut changer le nombre de colonnes ou de lignes occupées par un widget en ajoutant les paramètres `columnspan` ou `rowspan` en plus de `column` et `row`.

Pour coller un widget au bord de la grille, on peut ajouter `sticky` :

```
bt2.grid(column=1, row=0, sticky=(tk.W, tk.N))
```

`sticky` prend une liste de 0 à 4 éléments parmi :

- `tk.E` : *East* = droite

- `tk.N` : *North* = haut
- `tk.S` : *South* = bas
- `tk.W` : *West* = gauche

Redimensionner la fenêtre

Il faut indiquer quelles lignes et colonnes doivent changer de taille quand la fenêtre change de taille.

Dans cet exemple, ce sont la ligne 1 et la colonne 0.

```
root.rowconfigure(1, weight=1)
root.columnconfigure(0, weight=1)
```

On peut appeler plusieurs fois ces fonctions pour choisir plusieurs lignes et colonnes, et changer les `weight` pour modifier les proportions.

Widgets

Tous les widgets peuvent être positionnés avec `grid`, de la même manière que ci-dessus.

Label

Juste du texte.

```
label = ttk.Label(root, text="Je suis un texte. Wow.")
```

On peut aussi modifier le texte plus tard :

```
label_text = tk.StringVar()
label["textvariable"] = label_text
label_text.set("Le nouveau texte. Tout ça pour ça !")
```

Bouton

Un bouton qu'on peut cliquer.

```
def onclick():
    print("On a cliqué")

bt = ttk.Button(root, text="Cliquez-moi", command=onclick)
```

Image

Un bouton ou un label peut afficher une image :

```
img = tk.PhotoImage(file='image.png')
label["image"] = img
```

Entrée de texte

Boîte dans laquelle on peut écrire une ligne de texte.

```
text = tk.StringVar()
entry = ttk.Entry(root, textvariable=text)
```

Plus tard, on peut récupérer le texte :

```
print("L'utilisateur a écrit :", text.get())
```

Entrée de texte multiligne

Zone dans laquelle on peut écrire du texte en plusieurs lignes.

```
text = tk.Text(root)
```

Il y a plusieurs méthodes pour interagir avec le texte :

```
print(area.get("1.0", "end")) # obtenir tout le texte
print(area.get("2.0", "8.0")) # le texte de la ligne 2 à la ligne 8
area.replace("1.0", "end", "le nouveau texte") # remplacer du texte
area.insert("1.0", "le nouveau texte") # insérer du texte
```

Barre de défilement

Certains widgets (`Text`) peuvent défiler avec la molette, mais pour afficher la barre de défilement il faut un widget supplémentaire.

```
scroll = ttk.Scrollbar(root, orient=tk.VERTICAL, command=mon_widget_qui_défile.yview)
scroll.grid(column=1, row=0, sticky=(tk.N, tk.S, tk.E, tk.W))
mon_widget_qui_défile.configure(yscrollcommand=scroll.set)
```

Cadre

Un cadre qui peut avoir une bordure et contenir d'autres widgets.

```
frame = ttk.Frame(root, borderwidth=5, relief="ridge", width=200, height=100)
```

Les options ne sont pas obligatoires.

Le cadre contient ses propres lignes et colonnes. Pour placer des widgets dedans, il suffit de remplacer `root` par `frame` en les créant.

Barre de menus

La barre de menus en haut de la fenêtre.

```
root.option_add('*tearOff', False)
menubar = tk.Menu(root)
root['menu'] = menubar

menu_file = tk.Menu(menubar)
menu_edit = tk.Menu(menubar)

menubar.add_cascade(menu=menu_file, label='Fichier')
menubar.add_cascade(menu=menu_edit, label='Édition')

menu_file.add_command(label='Ouvrir', command=on_open)
```

Canevas

Le canevas est une zone de dessin.

```
canvas = tk.Canvas(root, background='white')
```

On peut y dessiner des formes :

```
canvas.create_line(x1, y1, x2, y2)
canvas.create_line(x1, y1, x2, y2, fill="red", width=3, dash=6)
canvas.create_rectangle(x1, y1, x2, y2, fill="red", outline="blue")
canvas.create_oval(x1, y1, x2, y2, fill="red", outline="blue")
```

Événements

Quand il se passe quelque chose sur un widget, un événement est généré. On peut écouter les événements de certains types, c'est-à-dire lancer une fonction quand l'action se produit.

Par exemple, on affiche des informations quand une touche du clavier est relâchée dans la zone de texte :

```
text = tk.Text(root)

def onrelease(event):
    print("On a frappé le clavier !")
    print(event)

area.bind("<KeyRelease>", onrelease)
```

On peut aussi utiliser les infos contenues dans l'événement, par exemple `event.keycode`.

Liste des événements

- Souris
  - `ButtonPress`: clic enfoncé d'un bouton de la souris
  - `Button-1`: comme `ButtonPress` mais seulement pour le clic gauche
  - `Button-2`: comme `ButtonPress` mais seulement pour le clic molette

    - `Button-3`: comme `ButtonPress` mais seulement pour le clic droit
    - `ButtonRelease`: clic relâché d'un bouton de la souris
    - `Double-1`: double-clic gauche
    - `Double-3`: double-clic droit
    - `Enter`: le pointeur de la souris entre
    - `Leave`: le pointeur de la souris sort
    - `Motion`: le pointeur de la souris bouge dedans
    - `B1-Motion`: comme `Motion` mais avec le clic gauche enfoncé
    - `B2-Motion`: comme `Motion` mais avec le clic molette enfoncé
    - `B3-Motion`: comme `Motion` mais avec le clic droit enfoncé
- Clavier
    - `KeyPress`: appui d'une touche du clavier
    - `KeyRelease`: relâche d'une touche du clavier
- Widget
    - `Configure`: le widget est créé
    - `Expose`: le widget est affiché ou réaffiché
    - `FocusIn`: le widget reçoit le focus
    - `FocusOut`: le widget pert le focus

Dialogues

On peut ouvrir des fenêtres pour dire ou demander des choses.

Fichiers

Demander où enregistrer un fichier, quel fichier ouvrir, choisir un dossier.

```
from tkinter import filedialog

filetypes = [
    ("Texte", "*.txt"),
    ("Image PNG", "*.png"),
    ("Autre", "*.*")
]
chemin = tk.filedialog.asksaveasfilename(filetypes=filetypes)
print(chemin)
```

On peut remplacer `asksaveasfilename` par `askopenfilename` ou `askdirectory`.

Couleur

Demander de choisir une couleur.

```
from tkinter import colorchooser

color = colorchooser.askcolor(initialcolor='black')
print(color)
```

On peut aussi indiquer la couleur initiale avec sa notation hexadécimale `"#000000"`.