Introduction à Tkinter – Partie 2 /

Objectifs pédagogiques

- Savoir **organiser des widgets en grille** avec `.grid()`
- Utiliser d'autres **widgets utiles** : `Listbox`, `Canvas`, `Text`
- Créer des **fenêtres secondaires**
- Ajouter des **couleurs, icônes et images**
- Découvrir des **exemples d'applications complètes**

Table des matières

1. Disposition avec .grid() (placement en lignes/colonnes).....1
2. Un exemple d'interface simple expliqué en détails.....2
3. Présentation de quelques éléments graphiques (widgets) de plus.....2
   a. Listbox – liste d'éléments à sélectionner.....2
   b. Canvas – surface de dessin (formes, images).....2
   c. Text – zone de texte multiligne.....2
   d. Exercice :.....2
4. Exemples d'applications complètes à réaliser.....3
   a. Convertisseur d'unités.....3
   b. Formulaire de contact.....3
   c. Petit jeu (devine le nombre).....3
5. À retenir.....3

1. Disposition avec `.grid()` (placement en lignes/colonnes)

`.grid()` permet d'organiser les widgets **comme dans un tableau**, ligne par ligne, colonne par colonne. En anglais ligne se dit « row » et colonne se dit « column ».

```
import tkinter as tk

fenetre = tk.Tk()
fenetre.title("Disposition avec grid")

tk.Label(fenetre, text="Nom :").grid(row=0, column=0)
tk.Entry(fenetre).grid(row=0, column=1, padx=10, pady=10)

tk.Label(fenetre, text="Prénom :").grid(row=1, column=0)
tk.Entry(fenetre).grid(row=1, column=1)

# le bouton ci-dessous est centré sur deux colonnes à partir de la colonne 0
tk.Button(fenetre, text="Valider").grid(row=2, column=0, columnspan=2)

fenetre.mainloop()
```

✅ Les méthodes `.grid()` et `.pack()` ne doivent pas être utilisées dans la même fenêtre. En pratique nous utiliserons de préférence .grid() à pack().

Remarquez et comprenez bien l'usage des paramètres columnspan, padx et pady :

- columnspan : indique combien de **colonnes** un widget doit occuper (fusion horizontale).
- padx : espace **horizontal** à gauche et à droite, donné en pixels ;
- pady : espace **vertical** au dessus et au dessous, donné en pixels .

2. Un exemple d'interface simple expliqué en détails

Télécharger calculatrice_01.py et calculatrice_01.pdf

3. Présentation de quelques éléments graphiques (widgets) de plus...

a. `Listbox` – liste d'éléments à sélectionner

```
liste = tk.Listbox(fenetre, height = 9) # hauteur spécifiée en caractères
liste.insert(1, "Option A")
liste.insert(2, "Option B")
liste.insert(3, "Option C")
liste.pack()
```

b. `Canvas` – surface de dessin (formes, images)

```
canvas = tk.Canvas(fenetre, width=200, height=100, bg="white") #taille en pixels
canvas.create_rectangle(50, 20, 150, 80, fill="blue")
canvas.create_text(100, 50, text="Hello", fill="white")
canvas.pack()
```

c. `Text` – zone de texte multiligne

```
texte = tk.Text(fenetre, height=5, width=30) # dimensions en caractères
texte.pack()
texte.insert(tk.END, "Tape ton message ici…")
```

d. `Exercice :`

A titre d'exercice, créer une fenêtre qui ressemble à celle ci-dessous, en utilisant la methode grid() :



4. Exemples d’applications complètes à réaliser

a. Convertisseur d’unités

- Entrée : valeur
- Choix de l’unité d’entrée (mètre, cm, pouce, etc.)
- Bouton de conversion
- Affichage du résultat dans plusieurs unités différentes

b. Formulaire de contact

- Champs : nom, prénom, e-mail, message
- Bouton "Envoyer"
- Message de confirmation

c. Petit jeu (devine le nombre)

- Entrée d’un nombre
- Bouton "Tester"
- Indication : "Trop grand", "Trop petit", "Bravo"

5. À retenir

Élément	Utilité
`.grid()`	Placement ligne/colonne
`Listbox`	Liste d’options
`Canvas`	Dessin, graphiques, images
`Text`	Zone de texte multiligne
`Toplevel`	Fenêtre secondaire
`PhotoImage`	Affichage d’images `.png`