

Atelier Création d'une application à Interface Graphique avec Tkinter

Exercice 1 :

Créez une application GUI simple qui affiche une fenêtre avec un bouton "Cliquez ici !" et un label vide. Lorsque le bouton est cliqué, le texte "Bonjour !" doit s'afficher dans le label.

Exercice 2 :

Concevez une application qui permet à l'utilisateur de saisir son nom dans un champ de texte. Après avoir appuyé sur un bouton, une nouvelle fenêtre doit apparaître avec un message de bienvenue personnalisé utilisant le nom saisi.

Exercice 3 :

Élaborez une application qui présente une liste déroulante (combobox) avec plusieurs options (par exemple, des couleurs). Lorsque l'utilisateur sélectionne une option et clique sur un bouton, la couleur de fond de la fenêtre doit changer en fonction de l'option sélectionnée.

Exercice 4 :

Créez une calculatrice simple avec une interface graphique. L'utilisateur devrait pouvoir saisir deux nombres dans des champs de texte, sélectionner une opération (addition, soustraction, multiplication, division) à l'aide de boutons radio, puis voir le résultat dans un label après avoir cliqué sur un bouton "Calculer".

Exercice 5 :

Concevez une application qui affiche une image dans une fenêtre lorsque l'utilisateur sélectionne un fichier image à partir de son système de fichiers à l'aide d'un bouton "Parcourir". Assurez-vous que l'application peut gérer différents formats d'images (par exemple, JPG, PNG).

Exercice 6 :

Élaborez une application de conversion de devises. L'utilisateur devrait pouvoir saisir un montant dans une devise, choisir une devise de destination dans une liste déroulante, puis voir le montant converti lorsque le bouton "Convertir" est cliqué. Utilisez une API de conversion de devises si possible.

Exercice 7 :

Créez une application de gestionnaire de tâches simple. L'utilisateur devrait pouvoir ajouter des tâches à une liste à l'aide d'un champ de texte et d'un bouton "Ajouter". Chaque tâche devrait être affichée dans une liste. L'utilisateur devrait pouvoir cocher une case à côté de chaque tâche pour indiquer qu'elle est terminée.

Exercice 8 :

Concevez une application de lecture de fichiers texte. L'utilisateur devrait pouvoir sélectionner un fichier texte à l'aide d'un bouton "Parcourir", puis afficher le contenu du fichier dans une zone de texte à l'intérieur de l'interface graphique.

Exercice 9 :

Élaborez une application de suivi de fitness qui permet à l'utilisateur de saisir son poids et sa taille à l'aide de champs de texte, puis de cliquer sur un bouton pour calculer l'indice de masse corporelle (IMC) et afficher le résultat dans une zone de texte avec une interprétation de l'IMC.

Solution

Exercice 1 :

```
import tkinter as tk

def afficher_message():
    label.config(text="Bonjour !")

# Création de la fenêtre principale
root = tk.Tk()
root.title("Application GUI")

# Création du bouton
bouton = tk.Button(root, text="Cliquez ici !", command=afficher_message)
bouton.pack()

# Création du label
label = tk.Label(root)
label.pack()

# Lancement de la boucle principale
root.mainloop()
```

Exercice 2 :

```
import tkinter as tk

from tkinter import messagebox

def afficher_message():

    nom = champ_nom.get()

    messagebox.showinfo("Bienvenue", f"Bienvenue, {nom} !")

root = tk.Tk()

root.title("Application GUI")

# Création du champ de texte pour le nom

champ_nom = tk.Entry(root)

champ_nom.pack()

bouton = tk.Button(root, text="Afficher message", command=afficher_message)

bouton.pack()

# Lancement de la boucle principale

root.mainloop()
```

Exercice 3 :

```
import tkinter as tk

from tkinter import ttk

def changer_couleur():

    couleur = combobox.get()

    root.config(bg=couleur)

# Création de la fenêtre principale

root = tk.Tk()

root.title("Application GUI")

# Création de la liste déroulante avec les couleurs

couleurs = ["red", "green", "blue", "yellow"]

combobox = ttk.Combobox(root, values=couleurs)

combobox.pack()

# Création du bouton pour changer la couleur

bouton = tk.Button(root, text="Changer couleur", command=changer_couleur)

bouton.pack()

# Lancement de la boucle principale

root.mainloop()
```

Exercice 4 :

```
import tkinter as tk

def calculer():

    nombre1 = float(champ_nombre1.get())

    nombre2 = float(champ_nombre2.get())

    operation = operation_var.get()

    if operation == "+":

        resultat = nombre1 + nombre2

    elif operation == "-":

        resultat = nombre1 - nombre2

    elif operation == "*":

        resultat = nombre1 * nombre2

    elif operation == "/":

        if nombre2 != 0:

            resultat = nombre1 / nombre2

        else:

            resultat = "Erreur: Division par zéro"

    label_resultat.config(text=f"Résultat : {resultat}")

# Création de la fenêtre principale

root = tk.Tk()

root.title("Calculatrice")

# Création des champs de texte pour les nombres

champ_nombre1 = tk.Entry(root)

champ_nombre1.pack()

champ_nombre2 = tk.Entry(root)
```

```
champ_nombre2.pack()

# Création des boutons radio pour les opérations

operation_var = tk.StringVar()

operation_var.set("+") # Opération par défaut

bouton_radio1 = tk.Radiobutton(root, text="+", variable=operation_var,
value="+")

bouton_radio1.pack()

bouton_radio2 = tk.Radiobutton(root, text="-", variable=operation_var,
value="-")

bouton_radio2.pack()

bouton_radio3 = tk.Radiobutton(root, text="*", variable=operation_var,
value="*")

bouton_radio3.pack()

bouton_radio4 = tk.Radiobutton(root, text="/", variable=operation_var,
value="/")

bouton_radio4.pack()

# Création du bouton pour effectuer le calcul

bouton_calculer = tk.Button(root, text="Calculer", command=calculer)

bouton_calculer.pack()

# Affichage du résultat

label_resultat = tk.Label(root)

label_resultat.pack()

# Lancement de la boucle principale

root.mainloop()
```

Exercice 5 :

```
import tkinter as tk

from tkinter import filedialog

from PIL import Image, ImageTk

def parcourir_image():

    chemin_fichier = filedialog.askopenfilename(filetypes=[("Fichiers image",
"*.jpg;*.jpeg;*.png")])

    if chemin_fichier:

        afficher_image(chemin_fichier)

def afficher_image(chemin_image):

    image = Image.open(chemin_image)

    #image = image.resize((400, 400), Image.ANTIALIAS) # Redimensionner
l'image pour l'afficher correctement

    photo = ImageTk.PhotoImage(image)

    label_image.config(image=photo)

    label_image.image = photo # Garantir que la référence à l'image est
conservée

# Création de la fenêtre principale

root = tk.Tk()

root.title("Affichage d'image")

# Création du bouton "Parcourir"

bouton_parcourir = tk.Button(root, text="Parcourir", command=parcourir_image)

bouton_parcourir.pack()

label_image = tk.Label(root)

label_image.pack()

root.mainloop()
```


Exercice 6 :

```
import tkinter as tk

from tkinter import ttk

# Dictionnaire des taux de conversion (exemple : 1 USD = 0.85 EUR)

taux_conversion = {

    "USD": 0.85,

    "EUR": 1.0,

    "GBP": 1.15,

    "JPY": 0.0076}

def convertir_devises():

    montant = float(champ_montant.get())

    devise_source = "EUR" # Devise source (par défaut, vous pouvez la changer
si nécessaire)

    devise_destination = combobox_devises.get() # Devise sélectionnée dans la
liste déroulante

    taux = taux_conversion[devise_destination] /
taux_conversion[devise_source]

    montant_converti = montant * taux

    label_resultat.config(text=f"{montant} {devise_source} =
{montant_converti:.2f} {devise_destination}")

# Création de la fenêtre principale

root = tk.Tk()

root.title("Convertisseur de devises")

# Création du champ de texte pour le montant

champ_montant = tk.Entry(root)

champ_montant.pack()
```

```

# Création de la liste déroulante avec les devises

devises = ["USD", "EUR", "GBP", "JPY"] # Vous pouvez ajouter d'autres devises
si nécessaire

combobox_devises = ttk.Combobox(root, values=devises)

combobox_devises.pack()

# Création du bouton pour convertir les devises

bouton_convertir = tk.Button(root, text="Convertir",
command=convertir_devises)

bouton_convertir.pack()

# Affichage du résultat

label_resultat = tk.Label(root)

label_resultat.pack()

# Lancement de la boucle principale

root.mainloop()

```

Exercice 7 :

```

import tkinter as tk

from tkinter import ttk

def ajouter_tache():

    tache = champ_tache.get()

    if tache:

        liste_taches.insert("", "end", values=(tache, "Non"))

```

```

def marquer_terminee():

    selection = liste_taches.selection()

    if selection:

        for item in selection:

            liste_taches.item(item,
values=(liste_taches.item(item)["values"][0], "Oui"))

# Création de la fenêtre principale

root = tk.Tk()

root.title("Gestionnaire de tâches")

# Création du champ de texte pour ajouter une tâche

champ_tache = tk.Entry(root, width=30)

champ_tache.grid(row=0, column=0, padx=5, pady=5)

# Création du bouton pour ajouter une tâche

bouton_ajouter = tk.Button(root, text="Ajouter", command=ajouter_tache)

bouton_ajouter.grid(row=0, column=1, padx=5, pady=5)

# Création de la liste des tâches

liste_taches = ttk.Treeview(root, columns=("Tâche", "Terminée"),
show="headings", selectmode="extended")

liste_taches.heading("Tâche", text="Tâche")

liste_taches.heading("Terminée", text="Terminée")

liste_taches.grid(row=1, column=0, columnspan=2, padx=5, pady=5)

# Création du bouton pour marquer une tâche comme terminée

bouton_terminee = tk.Button(root, text="Marquer comme terminée",
command=marquer_terminee)

bouton_terminee.grid(row=2, column=0, columnspan=2, padx=5, pady=5)

```

```
# Lancement de la boucle principale  
  
root.mainloop()
```

Exercice 8:

```
import tkinter as tk  
  
from tkinter import filedialog  
  
def parcourir_fichier():  
    chemin_fichier = filedialog.askopenfilename(filetypes=[("Fichiers texte",  
                                                             "*.txt")])  
  
    if chemin_fichier:  
        with open(chemin_fichier, "r") as fichier:  
            contenu = fichier.read()  
  
            zone_texte.delete(1.0, tk.END) # Effacer le contenu précédent  
  
            zone_texte.insert(tk.END, contenu)  
  
# Création de la fenêtre principale  
  
root = tk.Tk()  
  
root.title("Lecteur de fichiers texte")  
  
# Création du bouton "Parcourir"
```

```
bouton_parcourir = tk.Button(root, text="Parcourir",  
command=parcourir_fichier)  
  
bouton_parcourir.pack(pady=10)  
  
# Création de la zone de texte pour afficher le contenu du fichier  
  
zone_texte = tk.Text(root, wrap="word", width=40, height=10)  
  
zone_texte.pack(padx=10, pady=5)  
  
# Lancement de la boucle principale  
  
root.mainloop()
```

Exercice 9 :

```
import tkinter as tk  
  
def calculer_imc():
```

```

poids = float(champ_poids.get())

taille = float(champ_taille.get())

# Calcul de l'IMC

imc = poids / (taille * taille)

# Interprétation de l'IMC

interpretation = ""

if imc < 18.5:

    interpretation = "Sous-poids"

elif 18.5 <= imc < 25:

    interpretation = "Poids normal"

elif 25 <= imc < 30:

    interpretation = "Surpoids"

else:

    interpretation = "Obésité"

# Affichage du résultat

resultat = f"Votre IMC est : {imc:.2f}\nInterprétation : {interpretation}"

zone_resultat.config(state=tk.NORMAL)

zone_resultat.delete(1.0, tk.END)

zone_resultat.insert(tk.END, resultat)

zone_resultat.config(state=tk.DISABLED)

# Création de la fenêtre principale

root = tk.Tk()

root.title("Calculateur d'IMC")

# Création du champ de texte pour le poids

label_poids = tk.Label(root, text="Poids (kg):")

label_poids.grid(row=0, column=0, padx=5, pady=5)

```

```
champ_poids = tk.Entry(root)

champ_poids.grid(row=0, column=1, padx=5, pady=5)

# Création du champ de texte pour la taille

label_taille = tk.Label(root, text="Taille (m):")

label_taille.grid(row=1, column=0, padx=5, pady=5)

champ_taille = tk.Entry(root)

champ_taille.grid(row=1, column=1, padx=5, pady=5)

# Création du bouton pour calculer l'IMC

bouton_calculer = tk.Button(root, text="Calculer IMC", command=calculer_imc)

bouton_calculer.grid(row=2, column=0, columnspan=2, padx=5, pady=5)

# Création de la zone de texte pour afficher le résultat

zone_resultat = tk.Text(root, width=40, height=5, state=tk.DISABLED)

zone_resultat.grid(row=3, column=0, columnspan=2, padx=5, pady=5)

# Lancement de la boucle principale

root.mainloop()
```