

1. Gestion du clavier

Voici un programme qui interagit avec l'appui sur une touche du clavier.

- Ajouter un test dans la fonction pour que le programme s'arrête quand l'utilisateur a appuyé successivement sur les touches "f", "i" et "n".

```
import tkinter as tk
def clavier(event) :
    """ Événement clavier """
    touche = event.keysym
    print("Touche appuyée :", touche)
fenetre = tk.Tk()
fenetre.bind("<Key>", clavier)
fenetre.mainloop()
```

2. Gestion de la souris

Voici un programme qui interagit avec le clic gauche de la souris.

Ajouter un cadre graphique et une fonction pour gérer le clic droit :

- Le clic droit choisit une couleur aléatoire dans une liste prédéfinie.
- Le clic gauche trace un petit cercle de la couleur choisie autour du point où se trouve le curseur de la souris.

```
import tkinter as tk
def clic_g(event) :
    """ Événement souris clic gauche """
    x = event.x
    y = event.y
    print("Clic gauche en", x, y)
fenetre = tk.Tk()
fenetre.bind("<Button-1>", clic_g)
fenetre.mainloop()
```

3. Gestion du temps

Voici un programme qui interagit avec le l'horloge interne, il affiche "ok" après un délai de 2 secondes (2000 millisecondes)

```
import tkinter as tk
def timer() :
    """ Événement déclenché par un timer """
    zone.itemconfig(compteur, text="ok")
    zone.pack()
fenetre = tk.Tk()
zone = tk.Canvas(fenetre, width=400, height=100, bg="white")
compteur = zone.create_text(200, 50, text="", font=('Helvetica', '80'))
zone.pack()
fenetre.after(2000, timer)
fenetre.mainloop()
```

- Remplacer le texte par un compteur qui commence à 0 et qui s'incrémente de 1 tous les dixièmes de secondes. Attention : il faudra relancer le timer à chaque fois qu'on l'utilise...
- Afficher le temps au format "00 :00.0" avec le détail des minutes, secondes, et dixièmes de secondes comme sur un chronomètre. Remarque : On pourra utiliser la notation :
`chrono = "{:0>2d}:{:0>2d}.:{:d}").format(minutes, secondes, dixiemes)`

4. Gestion de boutons

Voici un programme qui interagit avec deux boutons :

- Le premier bouton permet de changer de lettre.
- Le deuxième bouton permet de fermer la fenêtre.

```
import tkinter as tk
def bouton_1() :
    """ Bouton n°1 """
    global n
    n = (n + 1)%26
    zone.itemconfig(texte, text=chr(ord("a")+n))
    zone.pack()
n = 0
fenetre = tk.Tk()
zone = tk.Canvas(fenetre, width=300, height=100, bg="white")
zone.pack()
texte = zone.create_text(150, 50, text=chr(ord("a")+n), font=('arial', '50'))
b_1 = tk.Button(fenetre, text="Suivante", command=bouton_1)
b_1.pack(side=tk.LEFT, padx=5)
b_2 = tk.Button(fenetre, text="Sortir", command=fenetre.destroy)
b_2.pack(side=tk.RIGHT)
fenetre.mainloop()
```

Ajouter deux autres boutons permettant de déplacer de 10 pixels la lettre vers la droite ou vers la gauche

5. Gestion de cases à choisir et de cases à cocher

Voici un programme qui interagit avec deux types de cases :

- Les cases à choisir (Radiobutton) :
Elles permettent de choisir une option et une seule parmi une liste d'options.
- Les cases à cocher (Chekbutton) :
Elles permettent de cocher séparément différentes options, en indiquant pour chacune d'entre elles si on la veut ou pas...

```
import tkinter as tk
fenetre = tk.Tk()
zone = tk.Canvas(fenetre, width=200, height=50, bg="blue")
zone.grid(row = 0, columnspan = 3)
choix = tk.IntVar()
choix_1 = tk.Radiobutton(fenetre, text="1", variable=choix, value=10)
choix_1.grid(row=1, column=0, sticky=tk.W)
choix_2 = tk.Radiobutton(fenetre, text="2", variable=choix, value=20)
choix_2.grid(row=2, column=0, sticky=tk.W)
v_1 = tk.IntVar()
c_1 = tk.Checkbutton(fenetre, text="a", variable=v_1)
c_1.grid(row=1, column=2, sticky=tk.E)
v_2 = tk.IntVar()
c_2 = tk.Checkbutton(fenetre, text="b", variable=v_2)
c_2.grid(row=2, column=2, sticky=tk.E)
fenetre.mainloop()
print("choix :", choix.get())
print("v_1 =", v_1.get())
print("v_2 =", v_2.get())
```

Créer :

- Une zone graphique dans laquelle on pourra écrire.
- Trois cases à choisir "petit", "moyen" et "grand", permettant de choisir la taille de la police de caractère.
- Trois cases à cocher "gras", "italique" et "souligné", permettant de choisir un style à appliquer.
- Un bouton "Bonjour" sur lequel on devra cliquer pour écrire le mot "Bonjour" dans la taille choisie avec les effets de styles cochés.

6. Gestion de listes à choisir et de curseurs

Voici un programme qui interagit avec une liste et un curseur :

```
import tkinter as tk
fenetre = tk.Tk()
s_1 = tk.StringVar()
s_1.set("c")
spin_1 = tk.Spinbox(fenetre, value=['a','b','c','d'], textvariable=s_1, width=2)
spin_1.pack()
c_1 = tk.IntVar()
c_1.set(5)
curseur_1 = tk.Scale(fenetre, from_=0, to=10, orient="horizontal",
variable=c_1)
curseur_1.pack()
fenetre.mainloop()
print(s_1.get())
print(c_1.get())
```

Créer :

- Une zone graphique dans laquelle on pourra dessiner.
- Une liste de couleur à choisir.
- Une liste de valeurs à choisir la hauteur.
- (utiliser *increment=10* pour avoir des valeurs de 10 en 10)
- Un curseur pour choisir la largeur.
(utiliser *resolution=10* pour avoir des valeurs de 10 en 10)
- Un bouton "Dessiner" sur lequel on devra cliquer pour dessiner un rectangle ayant les caractéristiques choisies.

Autre exemple :
Gestion d'une
barre de menu

```
import tkinter as tk
from random import randrange
def couleur_rouge():
    rouge = zone.create_line(randrange(300), 0, randrange(300), 200, fill="red")
def couleur_bleu():
    bleu = zone.create_line(randrange(300), 0, randrange(300), 200, fill="blue")
fenetre = tk.Tk()
zone = tk.Canvas(fenetre, width=300, height=200, bg="white")
zone.pack()
menu_bar = tk.Menu(fenetre)
menu_couleur = tk.Menu(menu_bar, tearoff=0)
menu_couleur.add_command(label="rouge", command=couleur_rouge)
menu_couleur.add_command(label="bleu", command=couleur_bleu)
menu_bar.add_cascade(label="Couleurs", menu=menu_couleur)
fenetre.config(menu=menu_bar)
fenetre.mainloop()
```