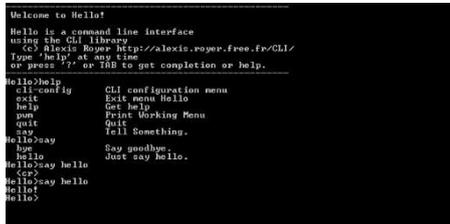
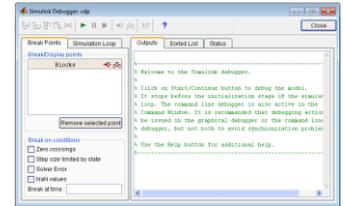


1. Interface graphique

Une interface graphique (GUI : Graphical User Interface) est un dispositif de dialogue homme-machine dans lequel les objets à manipuler sont dessinés sous forme de pictogramme à l'écran qui sont manipulables avec un dispositif de pointage, le plus souvent une souris.



Ce type d'interface a été mis sur le marché à la fin des années 1970 avec l'ordinateur Star de Xerox, pour remplacer les interfaces en ligne de commande (CUI : Command Line Interface).



2. Le module Tkinter

Le module Tkinter de Python permet de créer des interfaces graphiques. Dans la fenêtre principale, des éléments graphiques, que l'on nomme **widget** vont être ajoutés pour réaliser l'interface. La description qui suit va permettre de réaliser une interface pour une fonction de dé à multiples faces.

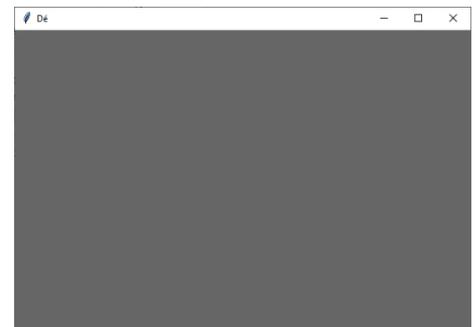
Le programme complet se situe à la fin de ce document.



2.1. Fenêtre principale

Le code suivant permet de créer une fenêtre.

```
# Importation du module Tkinter
from tkinter import *
from random import randint # pour le dé
# fonction du dé
def de(nbre_de_faces):
    val = randint(1, nbre_de_faces)
    return val
# Programme principal
# Création de la fenêtre principale
fenetre = Tk()
# Titre de la fenêtre
fenetre.title('Dé')
# Dimension et positionnement de la fenêtre
fenetre.geometry('600x400+100+100')
# Configuration du fond d'écran
fenetre.config(bg='gray40')
# Lancement du gestionnaire d'événements
fenetre.mainloop()
```



2.2. Label

Les labels (étiquettes) sont des espaces prévus pour écrire du texte. La méthode **pack** permet de placer le widget (**side** définit l'endroit du placement : **TOP**, **BOTTOM**, **LEFT**, **RIGHT** ; **padx** et **pady** définissent les marges extérieures du widget et **ipadx** et **ipady** définissent les marges intérieures du widget, le tout suivant les axes x et y).

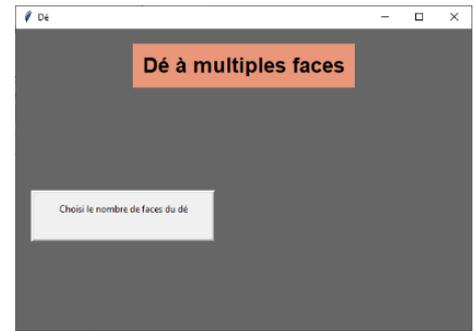
```
# Label
label_titre = Label(fenetre, text='Dé à multiples faces', font=('Arial', 20, 'bold'), bg='Dark Salmon')
# Affichage
label_titre.pack(side=TOP, padx=20, pady=20, ipadx=10, ipady=10)
```



2.3. Frame

Les frames (fenêtres) sont des zones rectangulaires qui peuvent contenir d'autres widgets.

```
# Frame
frame_gauche = Frame(fenetre, borderwidth=4, relief=RAISED)
# Label
label_choix = Label(frame_gauche, text='Choisi le nombre de faces du dé')
# Affichage
frame_gauche.pack(side=LEFT, padx=20, pady=20, ipadx=10, ipady=10)
label_choix.pack(side=TOP, padx=20, pady=10)
```



2.4. Spinbox

Spinbox est une proposition pour choisir un nombre.

```
# Spinbox
spinbox_nbre = Spinbox(frame_gauche, from_=1, to=100, width=5)
# Affichage
spinbox_nbre.pack(side=TOP, padx=20, pady=10)
```



2.5. Button

Les boutons (button) permettent de proposer une action à l'utilisateur.

```
# Bouton
bouton_lancer = Button(frame_gauche, text="LANCER", command=lancer_de, width=10)
bouton_quit = Button(fenetre, text="QUIT", command=fenetre.destroy, width=10)
# Label
label_resultat = Label(frame_gauche, bg='Dark Salmon', width=5)
# Affichage
bouton_quit.pack(side = BOTTOM, padx=10, pady=20)
bouton_lancer.pack(side=TOP, padx=10, pady=10)
label_resultat.pack(side=TOP, padx=10, pady=10)
```

À l'appui sur le bouton LANCER, la fonction `lancer_de` (`command=lancer_de`) est appelée, donc il faut la créer.

```
def lancer_de():
    nbre_max = int(spinbox_nbre.get())
    nbre = de(nbre_max)
    label_resultat.config(text = nbre)
```

La méthode `get` permet de récupérer la valeur du champ de saisie.

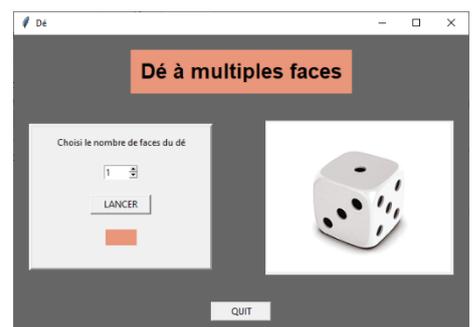
Le résultat de `de(nbre_max)` est écrit dans le label (ici `label_resultat`) pour l'affichage du résultat.



2.6. Image

Pour insérer une image, on peut créer un canevas et l'ajouter à l'intérieur. Un `canvas` (canevas) est un espace dans lequel on peut dessiner, écrire, insérer des éléments.

```
# Image
image_de = PhotoImage(file="de.png")
canvas = Canvas(fenetre, width=244, height=200, relief=RAISED)
canvas.create_image(122, 100, image=image_de)
# Affichage
canvas.pack(side=RIGHT, padx=20)
```



3. Positionnement des éléments

Pour un positionnement de peu d'éléments, la méthode `pack` peut être suffisante.
Pour un positionnement plus complexe, la méthode `grille` est plutôt conseillée.

4. Programme complet

```
from tkinter import *
from random import randint

def de(nombre_de_faces):
    val = randint(1, nombre_de_faces)
    return val

def lancer_de():
    nbre_max = int(spinbox_nbre.get())
    nbre = de(nbre_max)
    label_resultat.config(text = nbre)

#Programme principal
# Création de la fenêtre principale
fenetre = Tk()

# Titre de la fenêtre
fenetre.title('Dé')

# Dimension et positionnement de la fenetre
fenetre.geometry('600x400+100+100')

# Configuration du fond d'écran
fenetre.config(bg='gray40')

# Frame
frame_gauche = Frame(fenetre, borderwidth=4, relief=RAISED)

# Spinbox
spinbox_nbre = Spinbox(frame_gauche, from_=1, to=100, width=5)

# Image
image_de = PhotoImage(file="de.png")
canvas = Canvas(fenetre, width=244, height=200, relief=RAISED)
canvas.create_image(122, 100, image=image_de)

# Label
label_titre = Label(fenetre, text='Dé à multiples faces', font=('Arial', 20, 'bold'), bg='Dark Salmon')
label_choix = Label(frame_gauche, text='Choisi le nombre de faces du dé')
label_resultat = Label(frame_gauche, bg='Dark Salmon', width=5)

# Bouton
bouton_lancer = Button(frame_gauche, text="LANCER", command=lancer_de, width=10)
bouton_quit = Button(fenetre, text="QUIT", command=fenetre.destroy, width=10)

# Affichage
label_titre.pack(side = TOP, padx=20, pady=20, ipadx=10, ipady=10)
bouton_quit.pack(side = BOTTOM, padx=10, pady=20)
frame_gauche.pack(side = LEFT, padx=20, pady=20, ipadx=10, ipady=10)
label_choix.pack(side = TOP, padx=20, pady=10)
spinbox_nbre.pack(side=TOP, padx=20, pady=10)
bouton_lancer.pack(side=TOP, padx=10, pady=10)
label_resultat.pack(side=TOP, padx=10, pady=10)
canvas.pack(side=RIGHT, padx=20)

# Lancement du gestionnaire d'événements
fenetre.mainloop()
```

Source : <https://python.doctor/page-tkinter-interface-graphique-python-tutoriel>