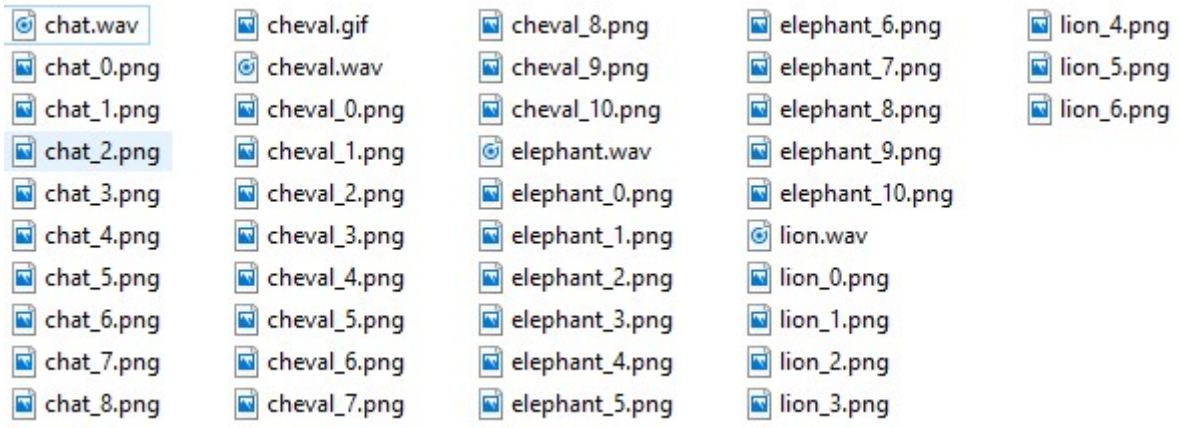
OBJECTIFS : L'objectif de ce TP est de réaliser sans trop d'aide, une animation en utilisant certains des outils vus dans les tps précédents.

L'évaluation de ce travail est basée sur le rendu du fichier *.py* qui sera constitué.

⇨ Pour débuter, télécharger le dossier *tp13.zip* contenant les fichiers images et sons qui seront utilisés. Le dézipper dans votre espace personnel sur ***U:*** . Ouvrir dans le dossier contenant ces images, un nouveau fichier *.py* et le sauvegarder sous le nom ***tp13.py*** .

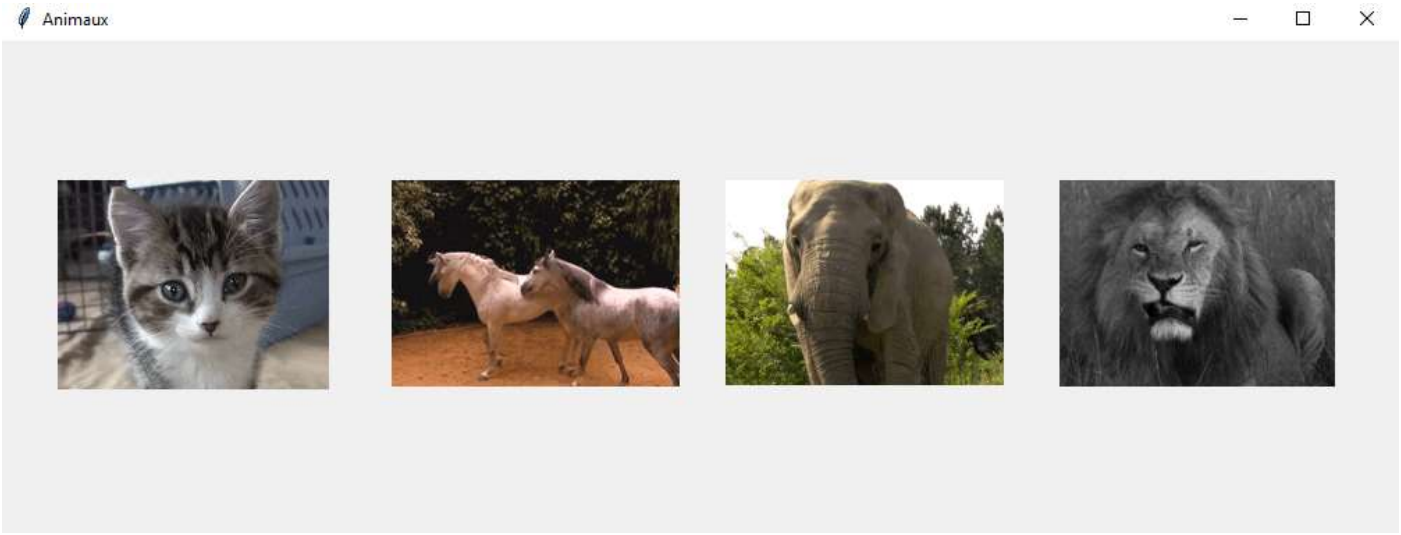
Ce dossier comprend pour 4 animaux différents, un fichier son de 2s qui donne le cri de cet animal et une dizaine de fichiers images de cet animal, qui ont été extraits d'une séquence vidéo où celui-ci émet ce cri. Ces images ont une taille de 200 x 150 px et sont au format png.



Lorsque ces images sont affichées à des intervalles de temps rapprochés, on recrée le mouvement. On obtient ainsi une image animée plus connue sous le nom de gif.

L'objectif de ce tp est de réaliser un code qui affiche dans une fenêtre la 1ère image de chaque animal. Lorsque l'utilisateur click sur un animal, celui-ci se met en mouvement et on entend le son de son cri.

Les images seront centrées dans une fenêtre graphique de 1000 x 350 px :



Pour cadrer légèrement ce travail, on demande de respecter les indications suivantes :

Les bibliothèques à importer seront les suivantes :

```
# Modules -----------------------------------------------------
from tkinter import Tk , Canvas
from PIL import Image, ImageTk
import winsound
```

Le programme principal de votre code sera le suivant :

```
# Main ----------------------------------------------------------------
fenetre = creer_fenetre()
zone_graphique = creer_widgets()
animaux = creation_cadres()
zone_graphique.bind("<ButtonPress-1>",click)

fenetre.mainloop()
```

La fonction *creation_cadres()* permettra d'afficher les 4 cadres contenant chacun l'image 0 de l'animal. Elle retournera un dictionnaire nommé *animaux* qui contiendra les informations suivantes :

animaux[''chat''] = [[obj0, obj1, obj2, obj3, obj4, obj5, obj6, obj7, obj8] , imCV]

= [liste des 9 objets images Tk du chat , **item de l'image affichée dans le canevas**]

animaux[''cheval''] = ...

animaux['elephant''] = ...

animaux['lion''] = ...

Pour vous aider, on donne le début du code de cette fonction :

```
def creation_cadres() :
    animaux = {}

    imTk = []
    for i in range(9) :
        nom_fichier = "chat_"+str(i)+".png"
        obj = ImageTk.PhotoImage(Image.open(nom_fichier) , master = fenetre)
        imTk.append(obj)
    x = 40
    y = 100
    imCv = zone_graphique.create_image(x,y , anchor = "nw" , image = imTk[0])
    animaux["chat"] = [imTk , imCv]

    imTk = []
    for i in range(11) :
        nom_fichier = "cheval_"+str(i)+".png"
        obj = ImageTk.PhotoImage(Image.open(nom_fichier) , master = fenetre)
        imTk.append(obj)
    x = 280
    y = 100
    imCv = zone_graphique.create_image(x,y , anchor = "nw" , image = imTk[0])
    animaux["cheval"] = [imTk , imCv]
```

.... suite à définir ....

On donne aussi le code de la fonction *click()* qui est appelée après un click sur une image :

```
def click(event) :
    x = event.x
    y = event.y
    for key in animaux :
        xmin , xmax , ymin , ymax = limites(key)
        if x > xmin and x < xmax and y > ymin and y < ymax :
            winsound.PlaySound(None, winsound.SND_PURGE)
            fichier_son = key + ".wav"
            winsound.PlaySound(fichier_son, winsound.SND_ASYNC)
            gif(key, 0)
```

Dans cette fonction on retrouve la procédure qui permet de jouer les fichiers sons. On retrouve aussi les fonctions *limites()* et *gif()* qui sont à compléter :

- *limites()* retourne les coordonnées des points '***nw***'et '***se***' de l'image repérée par la clé *key*. Dans cette fonction, il faudra récupérer les coordonnées du point '**nw'** de l'image en utilisant la méthode *coords()* (voir documentation) et prendre en compte le fait que les images ont toute une taille de 200 x 150 px.

- *gif()* permet de réaliser l'animation. Dans cette fonction, il faudra modifier l'objet image Tk affectée à l'item image en utilisant la méthode *itemconfigure()* (voir documentation). Il faudra relancer l'appel de cette fonction autant de fois qu'il y a d'images pour cette clé, en utilisant la méthode *after()* (voir documentation).

⇨ Terminer le code et tester son bon fonctionnement.

<u>En bonus :</u>

Compléter le code en ajoutant des animations supplémentaires :

- En double-cliquant 2 fois sur un point de la fenêtre, la dernière image qui a été animée se déplace sur ce point. Il s'agit ainsi de créer un nouvel évènement lié au double-click qui modifie les coordonnées du point repérant l'image concernée.

- En appuyant sur une touche du clavier, la dernière image qui a été animée tombe vers le bas jusqu'à disparaître lorsqu'elle sort de la fenêtre.

- ….

⇨ Transférer le fichier tp13.py **par l'intermédiaire de l'onglet transfert** du site en utilisant le code : **tp13**