

# Projet Juke Box

- Cahier des charges (avec vidéo)
- Gestion des données
- Étape 1 : affichage des images
- Étape 2 : Compléter les fonctions pour le son
- Étape 3 : Ajouter les fichiers audios avec l'extension wav
- Étape 4 : Compléter la fonction menu pour activer le son

## Cahier des charges

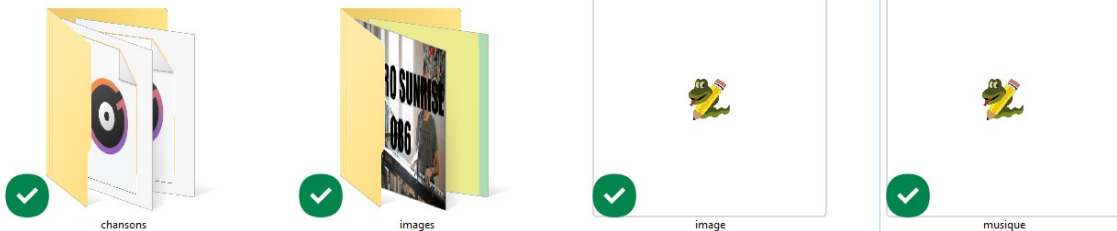
Objectifs du projet : - écrire un programme qui lit des fichiers audios.  
- se familiariser avec Tkinter pour préparer des projets plus ambitieux.  
- utiliser les notions apprises (POO, listes, fonctions, modularité ... )

Le travail peut se faire **seul ou à deux**.

1. Regardez la vidéo JukeBox.mp4 : elle montre le fonctionnement de l'application à programmer.
2. Recopiez l'ensemble des codes fournis. C'est une démarche importante qui nécessite de coder sans erreurs, en comprenant ce que vous écrivez. Suivez bien les 4 étapes du projet.
3. Lorsque vous avez terminé les 4 étapes et que l'application Juke Box fonctionne correctement :
  - a. Changez toutes les images (.png) ainsi que toutes les chansons (.wav) pour y mettre vos propres titres.
  - b. Ajoutez **au moins** une nouvelle fonctionnalité de votre choix (nouveau bouton, nouvelle fonction... )
4. Déposer votre travail sous forme d'un dossier zippé nommé **Nom\_prenom\_Nom\_prenom.zip** via l'application **casier** de l'ENT à Louis-Marie Madrias

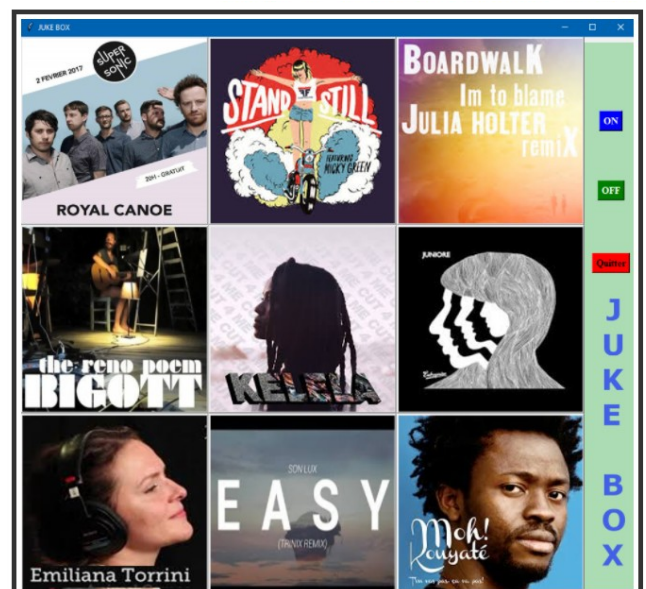
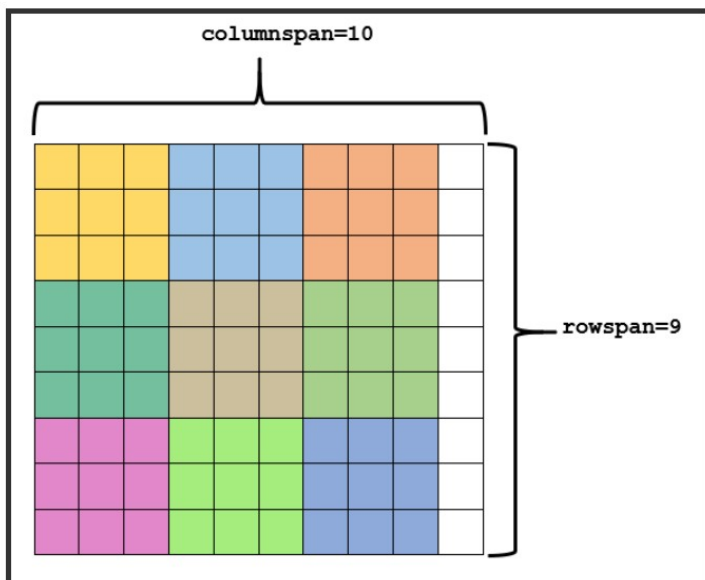
## Gestion des données

Il faut mettre ces quatre données dans le même dossier :



## Étape 1: affichage des images

Organisation de la fenêtre :



Le module image :

```
1 #image.py
2 #module pour les images
3 #fabrique une liste d'images
4 from tkinter import *
5 def creer_item(tim):
6     """création d'un item"""
7     #tim+".png" = "inombre.png" (inombre = 1;2;...9;10;... )
8     pl = PhotoImage(file=tim+".png")
9     return pl
10
11 def liste_image(nombre):
12     """fonction qui permet de créer une liste qui comporte un nombre de photos"""
13     items = nombre*[""]
14     for i in range(nombre):
15         #le symbole / permet d'aller chercher les images dans le dossier images
16         tim = "images/"+str(i+1)
17         items[i]=creer_item(tim)
18     return items
19
20 def taille_photo(photo):
21     """Détermine la taille de la photo"""
22     largeur = photo.width()
23     hauteur = photo.height()
24     return largeur,hauteur
```

Le fichier principal  
musique.py :

```
1 #musique.py
2 from tkinter import *
3 from winsound import PlaySound
4 from tkinter import *
5 from image import *
6
7 def debutmusique(son):
8     pass
9
10 def stopmusique():
11     pass
12
13 def debut():
14     pass
15
16 class cdrom():
17     pass
18
19 def change(numero):
20     pass
21
22 def menu(fenetre, images):
23     global Continue, son
24     lb2 = Button(fenetre, image=images[1], command=lambda: change(0))
25     lb2.grid(row=0, column=0, columnspan=3, rowspan=3, sticky='wn')
26     lb2 = Button(fenetre, image=images[2], command=lambda: change(1))
27     lb2.grid(row=0, column=3, columnspan=3, rowspan=3, sticky='wn')
28     lb2 = Button(fenetre, image=images[3], command=lambda: change(2))
29     lb2.grid(row=0, column=6, columnspan=3, rowspan=3, sticky='wn')
30     lb2 = Button(fenetre, image=images[4], command=lambda: change(3))
31     lb2.grid(row=3, column=0, columnspan=3, rowspan=3, sticky='wn')
32     lb2 = Button(fenetre, image=images[9], command=lambda: change(4))
33     lb2.grid(row=3, column=3, columnspan=3, rowspan=3, sticky='wn')
34     lb2 = Button(fenetre, image=images[10], command=lambda: change(5))
35     lb2.grid(row=3, column=6, columnspan=3, rowspan=3, sticky='wn')
36     lb2 = Button(fenetre, image=images[11], command=lambda: change(6))
37     lb2.grid(row=6, column=0, columnspan=3, rowspan=3, sticky='wn')
38     lb2 = Button(fenetre, image=images[12], command=lambda: change(7))
39     lb2.grid(row=6, column=3, columnspan=3, rowspan=3, sticky='wn')
40     lb2 = Button(fenetre, image=images[13], command=lambda: change(8))
41     lb2.grid(row=6, column=6, columnspan=3, rowspan=3, sticky='wn')
42     Continue = False
43
44 ##### Programme principal #####
45 fenetre=Tk()
46 images=liste_image(15)
47 photo=PhotoImage(file="images/1.png")
48 largeur, hauteur=taille_photo(photo)
49 fenetre.geometry(str(largeur)+"x"+str(hauteur))
50 fenetre.title('JUKE BOX')
51 fond=Label(fenetre, image=photo)
52 fond.grid(row=0, column=0, rowspan=9, columnspan=10)
53 Continue=False
54 son=''
55 menu(fenetre,images)
56 fenetre.mainloop()
```

Le programme principal  
est terminé, il ne faut  
plus le modifier

## Étape 2: Compléter les fonctions pour le son

```
7 def debutmusique(son):
8     PlaySound(son,1)
9
10 def stopmusique():
11     global Continue
12     Continue=False
13     PlaySound(None,1)
14
15 def debut():
16     global Continue, son
17     if Continue==True and son !='':
18         debutmusique(son)
19
20 class cdrom():
21     def __init__(self, son=''):
22         self.son=son
23     def start(self):
24         global son
25         debut()
```

## Étape 3 : Ajouter les fichiers audios avec l'extension wav

```
27 def change(numero):
28     global Continue, compt, son
29     Continue=True
30     mu=['chansons/01 - Stemming.wav',
31         "chansons/02 - Stand Still (Featuring Micky Green).wav",
32         "chansons/03 - I m To Blame.wav",
33         "chansons/04 - The Reno Poem.wav",
34         "chansons/05 - Guns & Synth.wav",
35         "chansons/06 - Christine.wav",
36         "chansons/09 - Animal Games.wav",
37         "chansons/10 - Easy.wav",
38         "chansons/11 - T en vas pas, ca va pas.wav",
39         "chansons/17 - 086.wav"]
40     son=mu[numero]
41     musique=cdrom()
42     musique.start()
```

## Étape 4 : Compléter la fonction menu pour activer le son

Il faut rajouter les lignes 65 à 71.

```
45 def menu(fenetre, images):
46     global Continue, son
47     lb2 = Button(fenetre, image=images[1], command=lambda: change(0))
48     lb2.grid(row=0, column=0, columnspan=3, rowspan=3, sticky='wn')
49     lb2 = Button(fenetre, image=images[2], command=lambda: change(1))
50     lb2.grid(row=0, column=3, columnspan=3, rowspan=3, sticky='wn')
51     lb2 = Button(fenetre, image=images[3], command=lambda: change(2))
52     lb2.grid(row=0, column=6, columnspan=3, rowspan=3, sticky='wn')
53     lb2 = Button(fenetre, image=images[4], command=lambda: change(3))
54     lb2.grid(row=3, column=0, columnspan=3, rowspan=3, sticky='wn')
55     lb2 = Button(fenetre, image=images[9], command=lambda: change(4))
56     lb2.grid(row=3, column=3, columnspan=3, rowspan=3, sticky='wn')
57     lb2 = Button(fenetre, image=images[10], command=lambda: change(5))
58     lb2.grid(row=3, column=6, columnspan=3, rowspan=3, sticky='wn')
59     lb2 = Button(fenetre, image=images[11], command=lambda: change(6))
60     lb2.grid(row=6, column=0, columnspan=3, rowspan=3, sticky='wn')
61     lb2 = Button(fenetre, image=images[12], command=lambda: change(7))
62     lb2.grid(row=6, column=3, columnspan=3, rowspan=3, sticky='wn')
63     lb2 = Button(fenetre, image=images[13], command=lambda: change(8))
64     lb2.grid(row=6, column=6, columnspan=3, rowspan=3, sticky='wn')
65     ft="Times 12 bold"
66     b2=Button(fenetre, text='Quitter', bg="red", font=ft, command=fenetre.destroy)
67     b2.grid(row=3, column=9)
68     b2=Button(fenetre, text='ON', bg="blue", fg="white", font=ft, command=debut)
69     b2.grid(row=1, column=9)
70     b2=Button(fenetre, text='OFF', bg="green", fg="white", font=ft, command=stopmusique)
71     b2.grid(row=2, column=9)
72     Continue = False
```