

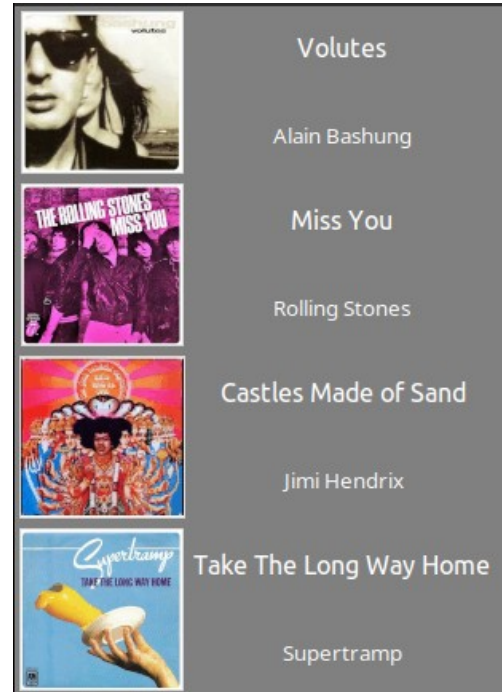
Projet Playlist

Un logiciel doit être développé pour une émission de radio. Il s'agit de stocker des chansons dans une playlist. Le projet consiste donc à concevoir une interface graphique (comme ci-contre) qui permette de visualiser la playlist qu'il conviendra bien sûr, d'alimenter par quelques chansons.

La première partie du projet porte sur une implémentation de la playlist par une liste chaînée simple.

La deuxième partie utilise la bibliothèque python *tkinter* pour représenter graphiquement la playlist.

Partie 1.



Les chansons d'une émission sont stockées dans une playlist **sous la forme d'une liste chaînée simple dont les chansons sont des éléments d'une classe *Chanson***.

Cette classe contiendra les attributs suivants :

- `id_song` : qui identifiera une chanson par un entier unique
- `titre`
- `artiste`
- `durée`
- `img` : qui stockera le chemin relatif pour accéder à l'image de la chanson
- `next` qui pointe vers la chanson suivante dans la playlist (et `None` si c'est la dernière chanson).

Dans un fichier *playlist.py*, créer la classe *Chanson* et une classe *Playlist* correspondante. Cette dernière contiendra une méthode *insérer(previous_song, new_song)* qui insère dans la playlist une nouvelle chanson *new_song* juste après la chanson *previous_song*. Elle contiendra une méthode permettant d'obtenir un affichage des chansons avec une numérotation respectant leur ordre dans la playlist. La classe *Playlist* pourra contenir toute autre méthode qu'on jugera nécessaire.

Tester la méthode d'affichage dans la console d'un éditeur python après avoir construit la playlist suivante :

1	Volutes	Alain Bashung
2	Miss you	Rolling Stones
3	Castles Made of Sand	Jimi Hendrix
4	Take The Long Way Home	Supertramp

Attention : l'ordre d'insertion d'une chanson dans une playlist est indépendant de l'identifiant `id_song` choisi pour cette chanson.

Projet Playlist

Partie 2 :

Concevoir une interface graphique de cette playlist de sorte que l'exemple précédent prenne la forme ci-dessus.

On respectera les consignes suivantes :

1. Le code devra être écrit dans un fichier nommé *interface.py*
2. On utilisera le fichier *playlist.py* comme un module.
3. On prévoira une fonction *affiche*, qui pour chaque chanson, crée une *frame* composée d'une image, le titre de la chanson et son interprète et insère cette *frame* dans l'interface.
4. La taille verticale de l'interface graphique sera calculée à l'aide d'une fonction prenant en paramètre la taille de la playlist.

Bonus : on pourra ajouter une fonction calculant la largeur « la plus esthétique » de l'interface graphique affichant la playlist.

Aide :

- pour fonctionner correctement avec python, on recommande de n'utiliser que des images au format gif.
- pour insérer des images , on pourra choisir un conteneur de type *Label* (cf lien 2) ou de type *Canvas* (cf lien 3).
Pour la disposition des widgets, on recommande d'utiliser la méthode *grid(row =...,column = ...)* qui prend en paramètre un numéro de ligne et un numéro de colonne.
Plus précisément, on pourra placer les *Frames* dans le conteneur racine en les disposant dans un tableau à une colonne unique
- Pour la disposition des éléments à l'intérieur d'une *Frame*, on pourra créer un tableau à deux lignes et deux colonnes, l'image utilisant les deux lignes (avec l'attribut *rowspan=2*)
- les images devront être stockées dans un conteneur python (liste ou dictionnaire) qui perdurera après les appels aux fonctions

Vous pouvez vous aider des trois liens suivants et de tous ceux que vous voulez:

- (1) <https://python.doctor/page-tkinter-interface-graphique-python-tutoriel>
- (2) <https://www.tutorialspoint.com/how-to-place-an-image-into-a-frame-in-tkinter>
- (3) <https://coderslegacy.com/create-frame-inside-canvas-in-tkinter/>