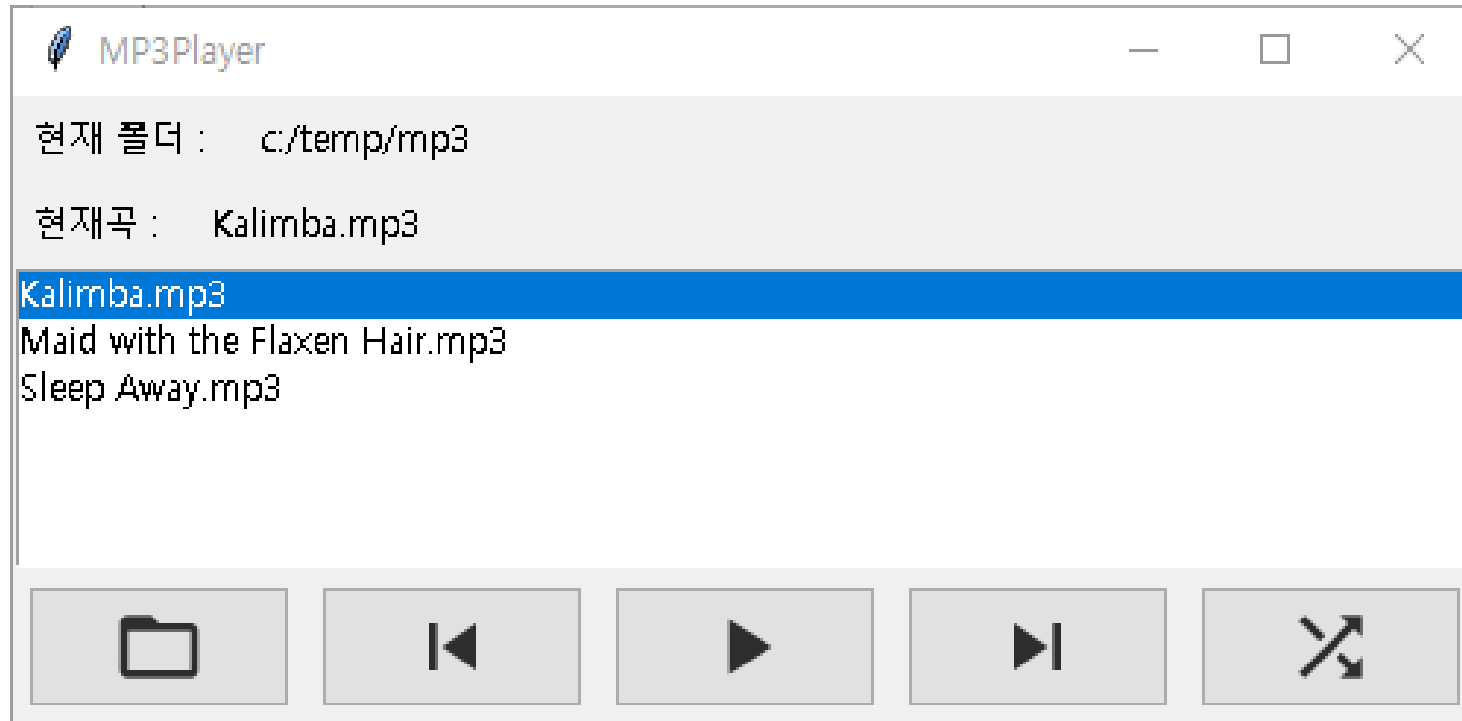


MP3Player (GUI)

MP3Player

❖ MP3Player GUI



MP3Player

❖ 기본 골격

```
from tkinter import filedialog
from tkinter import *
from tkinter.ttk import *
from random import shuffle
import glob
from pygame import mixer

_BASE_DIR = 'c:/temp/mp3'

class MP3Frame(Frame): # Frame - 윈도우 관리 클래스
    def __init__(self, master, base_dir=_BASE_DIR):
        Frame.__init__(self, master) # master는 부모 윈도우

        self.base_dir = base_dir
        self.master = master
        self.master.title("MP3Player")
        self.pack(fill=BOTH, expand=True) # 부모 윈도우 크기에 맞게 크기 조정
```

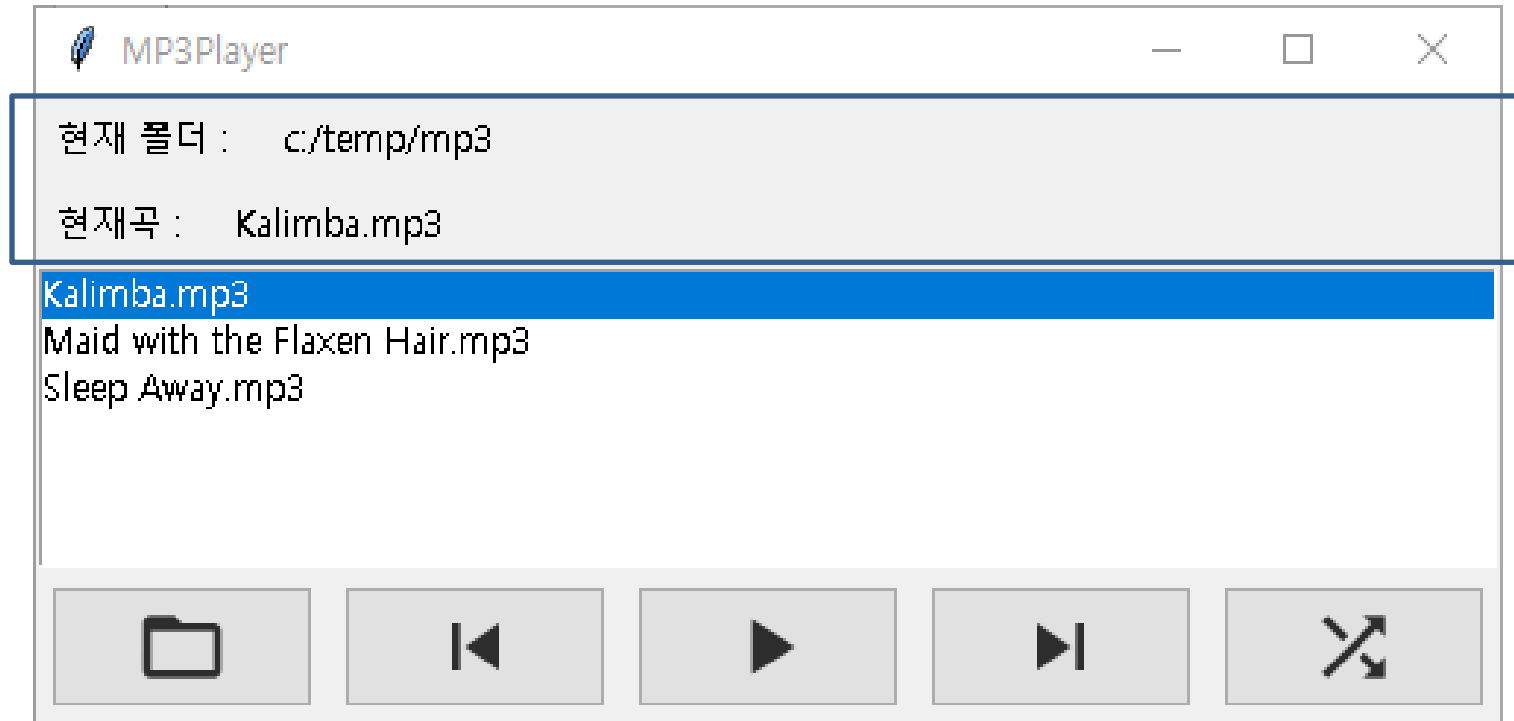
MP3Player

❖ 기본 골격

```
def main():  
    root = Tk() # 메인 윈도우  
    root.geometry("500x200+100+100") # 가로x세로+X위치+Y위치  
    MP3Frame(root)  
    root.mainloop()  
  
if __name__ == '__main__':  
    main()
```

MP3Player

❖ 상태 라벨 만들기



MP3Player

❖ 상태 라벨 만들기

```
class MP3Frame(Frame): # Frame - 윈도우 관리 클래스
    def __init__(self, master, base_dir=_BASE_DIR):
        :

        self.make_info_frame()

    def make_info_frame(self):
        frameInfo = Frame(self)
        frameInfo.pack(fill=X)

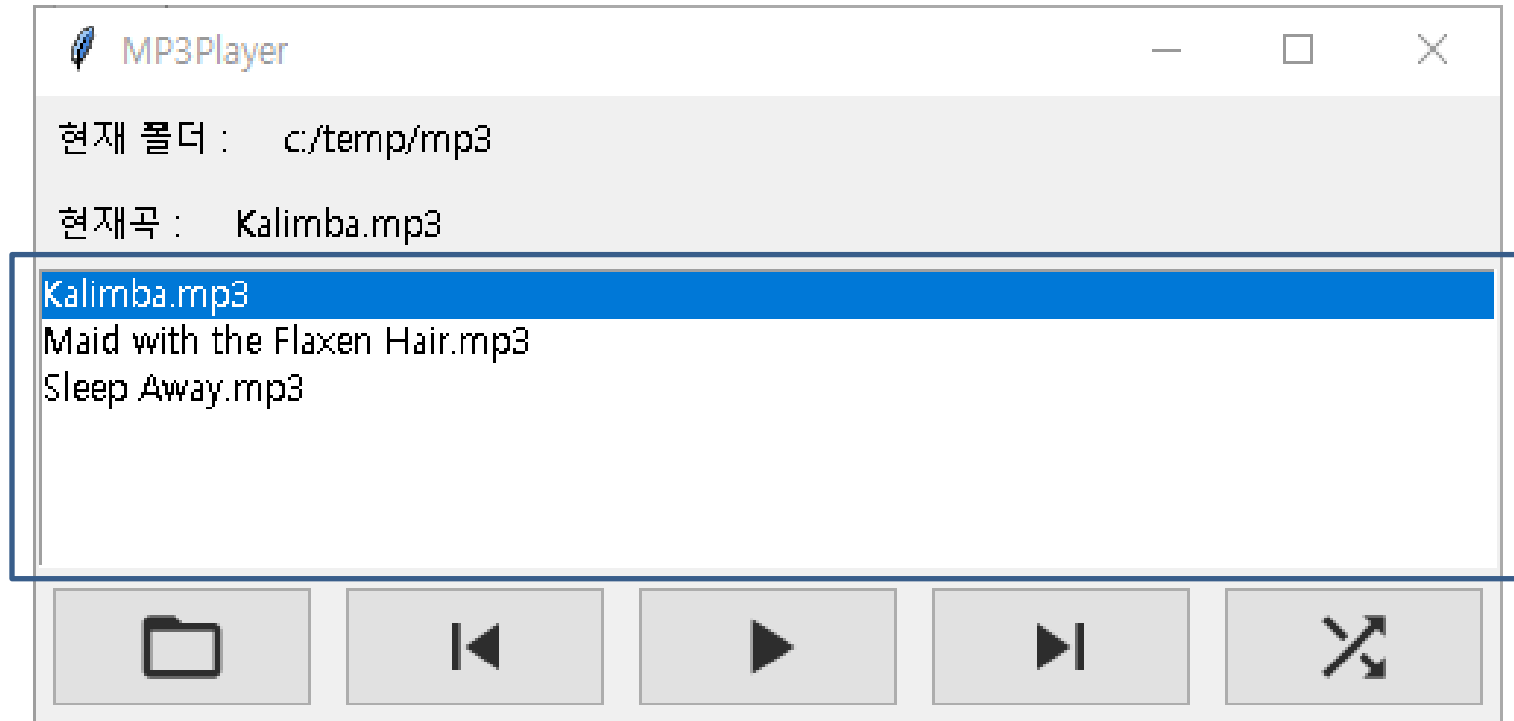
        Label(frameInfo, text="현재 폴더 : ").pack(side=LEFT, padx=5, pady=5)
        self.lblBaseDir = Label(frameInfo, text=self.base_dir)
        self.lblBaseDir.pack(side=LEFT, padx=5, pady=5, fill=X)

        frameInfo = Frame(self)
        frameInfo.pack(fill=X)

        Label(frameInfo, text="현재곡 : ").pack(side=LEFT, padx=5, pady=5)
        self.lblSongTitle = Label(frameInfo, text="----")
        self.lblSongTitle.pack(side=LEFT, padx=5, pady=5, fill=X)
```

MP3Player

❖ 리스트박스 만들기



MP3Player

❖ 리스트박스 만들기

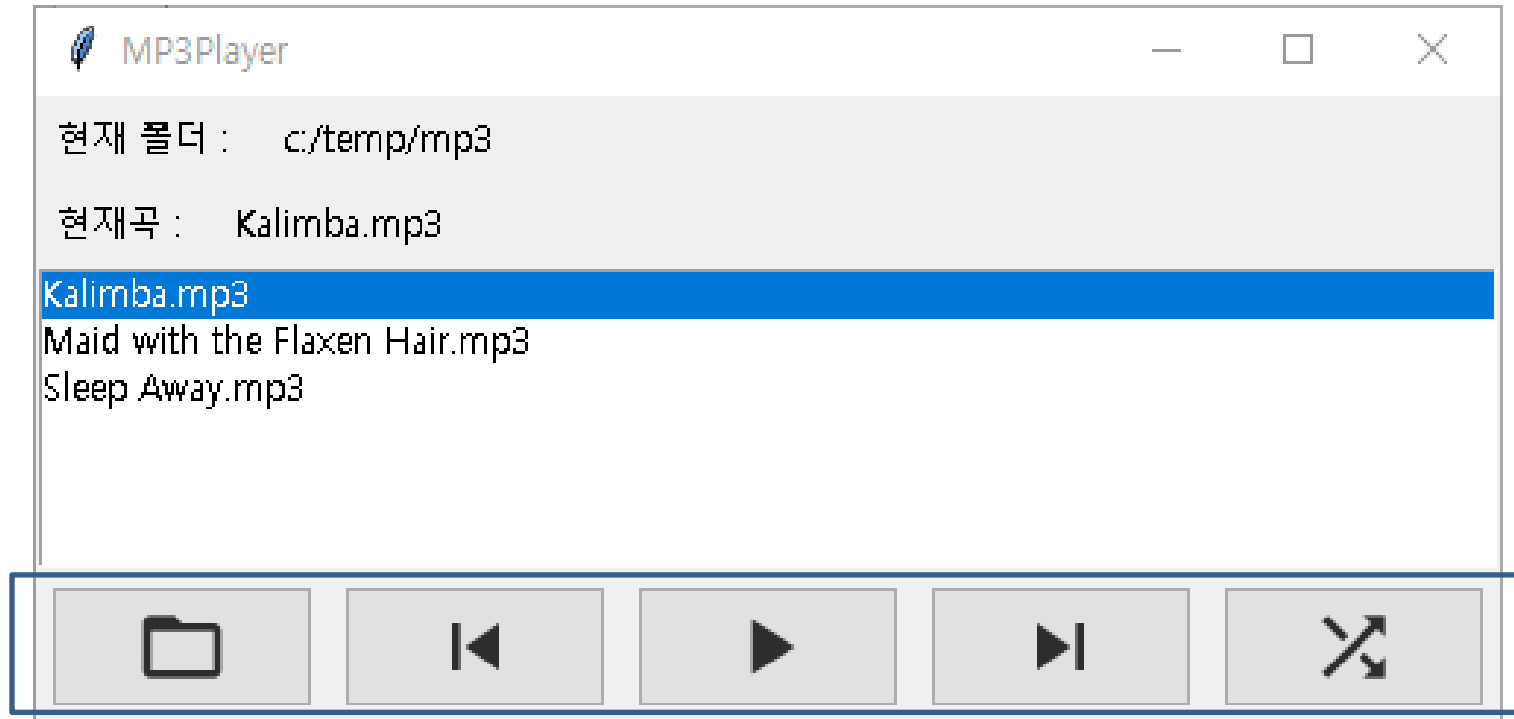
```
class MP3Frame(Frame): # Frame - 윈도우 관리 클래스
    def __init__(self, master, base_dir=_BASE_DIR):
        :

        self.make_info_frame()

        self.listbox = Listbox(self, height=5)
        self.listbox.pack(fill=BOTH, expand=True) # 윈도우 크기 확장시 확장
```


MP3Player

❖ 제어 버튼 만들기



MP3Player

❖ 제어 버튼 만들기

```
class MP3Frame(Frame): # Frame - 윈도우 관리 클래스
    def __init__(self, master, base_dir=_BASE_DIR):
        :

        self.make_button_frame()

    def make_button_frame(self):
        self.frameButtons = Frame(self)
        self.frameButtons.pack(fill=X)

        self.play_img = PhotoImage(file='play.png')
        self.pause_img = PhotoImage(file='pause.png')
        self.prev_img = PhotoImage(file='previous.png')
        self.next_img = PhotoImage(file='next.png')
        self.shuffle_img = PhotoImage(file='shuffle.png')
        self.foler_img = PhotoImage(file='folder.png')
```

❖ 제어 버튼 만들기

```
def make_button_frame(self):
    :
    self.btnSelect = Button(self.frameButtons, image=self.foler_img,
                             command=lambda : self.on_change_dir())
    self.btnSelect.pack(side=LEFT, padx=5, pady=5, fill=X, expand=True)

    self.btnPrev = Button(self.frameButtons,,image=self.prev_img,
                           command=lambda : self.on_prev())
    self.btnPrev.pack(side=LEFT, padx=5, pady=5, fill=X, expand=True)

    self.btnPlay = Button(self.frameButtons,image=self.play_img,
                           command=lambda : self.on_play())
    self.btnPlay.pack(side=LEFT, padx=5, pady=5, fill=X, expand=True)

    self.btnNext = Button(self.frameButtons, image=self.next_img,
                           command=lambda : self.on_next())
    self.btnNext.pack(side=LEFT, padx=5, pady=5, fill=X, expand=True)

    self.btnShuffle = Button(self.frameButtons, image=self.shuffle_img,
                              command=lambda : self.on_shuffle())
    self.btnShuffle.pack(side=LEFT, padx=5, pady=5, fill=X, expand=True)
```

❖ 제어 버튼 만들기 : 이벤트 핸들러

```
def on_change_dir(self):  
    pass  
  
def on_prev(self):  
    pass  
  
def on_play(self):  
    pass  
  
def on_next(self):  
    pass  
  
def on_shuffle(self):  
    pass
```

MP3Player

❖ 초기화

```
class MP3Frame(Frame): # Frame - 윈도우 관리 클래스
    def __init__(self, master, base_dir=_BASE_DIR):
        :

        self.make_button_frame()

        # 디렉토리에서 목록 읽기
        self.load_list()

        self.current = 0          # 현재 재생 파일 인덱스
        self.position = 0         # 재생 위치
        self.status = False       # 재생 상태

        mixer.init()              # mixer 초기화
```

❖ 초기화 : 디렉토리의 곡 목록 읽기

```
def load_list(self):
    self.lblBaseDir.config(text=self.base_dir)
    file_list = glob.glob1(self.base_dir, '*.mp3')

    # 리스트 박스 구성하기
    self.listbox.delete(0, END)
    for file in file_list:
        self.listbox.insert(END, file)

    self.current = 0
    self.listbox.select_set(self.current)
    title = self.listbox.get(self.current)
```

MP3Player

❖ 음악 재생하기

```
def play(self):
    self.position = 0
    mp3 = self.base_dir + '/' + self.listbox.get(self.current)
    mixer.music.load(mp3)
    mixer.music.play()

    self.status = True

    # 재생 정보 갱신
    title = self.listbox.get(self.current)
    self.lblSongTitle.config(text=title)
```

❖ 재생 버튼 이벤트 핸들러

```
def on_play(self):  
  
    if self.status : # 재생 중이면 중지로  
        self.btnPlay.config(image=self.pause_img)  
        self.status = False  
        self.position = mixer.music.get_pos()  
        mixer.music.pause()  
  
    else : # 중지면 재생으로  
        self.btnPlay.config(image=self.play_img)  
        self.status = True  
        if self.position :  
            mixer.music.unpause()  
        else:  
            self.play()
```


❖ 다음 버튼 이벤트 핸들러

```
def on_next(self):  
  
    self.listbox.select_clear(self.current) # 선택 해제  
    self.current = (self.current+1)%self.listbox.size()  
    self.listbox.select_set(self.current)   # 새로운 선택 설정  
  
    self.play()
```

MP3Player

❖ 다음 이전 이벤트 핸들러

```
def on_prev(self):
    self.listbox.select_clear(self.current)      # 선택 해제
    if self.current == 0:
        self.current = self.listbox.size()-1
    else:
        self.current -= 1

    self.listbox.select_set(self.current)        # 새로운 선택 설정

    self.play()
```

❖ 다음 셔플 이벤트 핸들러

```
def on_shuffle(self):  
    # 목록 가져오기  
    mp3_list = list(self.listbox.get(0, END))  
    shuffle(mp3_list)  
  
    # 이전 목록 지우기  
    self.listbox.delete(0, END)  
  
    # 새로운 목록 넣기  
    for file in mp3_list:  
        self.listbox.insert(END, file)
```

MP3Player

❖ 재생 디렉토리 변경 이벤트 핸들러

```
def on_change_dir(self):  
  
    # 디렉토리 선택 대화상자 띄우기  
    dir = filedialog.askdirectory()  
    if dir :  
        self.base_dir = dir  
        self.load_list()
```