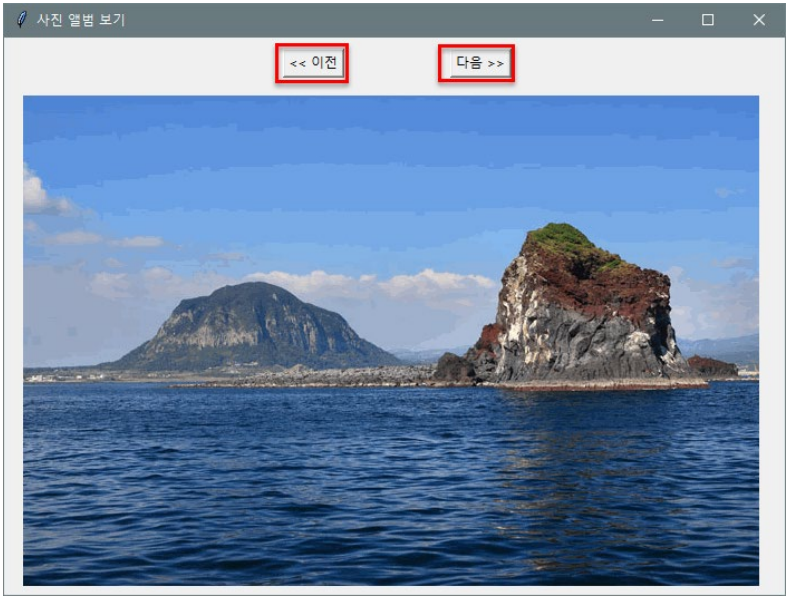
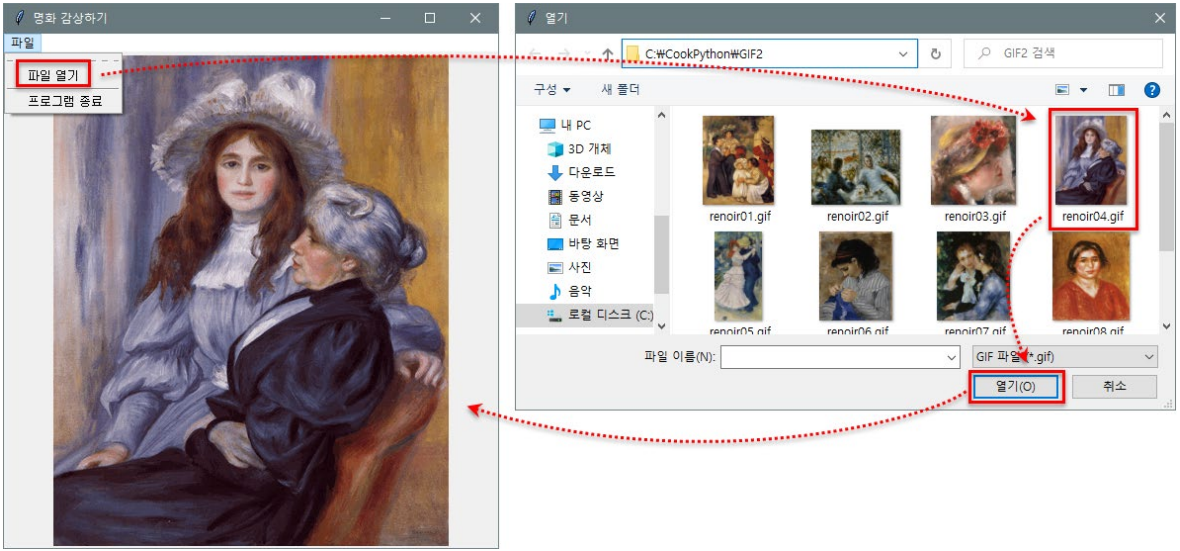
- [프로그램 1] 사진 앨범



- [프로그램 2] 명화 감상

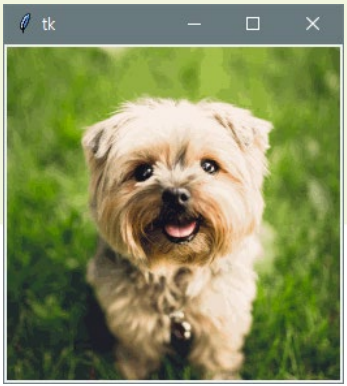

명화 감상하기
파일
파일 열기
프로그램 종료
열기
C:₩CookPython₩GIF2
GIF2 검색
구성
새 폴더
내 PC
3D 개체
다운로드
동영상
문서
바탕 화면
사진
음악
로컬 디스크 (C:)
renoir01.gif
renoir02.gif
renoir03.gif
renoir04.gif
renoir05.gif
renoir06.gif
renoir07.gif
renoir08.gif
파일 이름(N):
GIF 파일(*.gif)
열기(O)
취소

▪ 레이블에 글자 대신 이미지 넣기

- PhotoImage()는 GIF 파일만 지원, JPEG나 BMP 등은 지원하지 않음

Code10-04.py

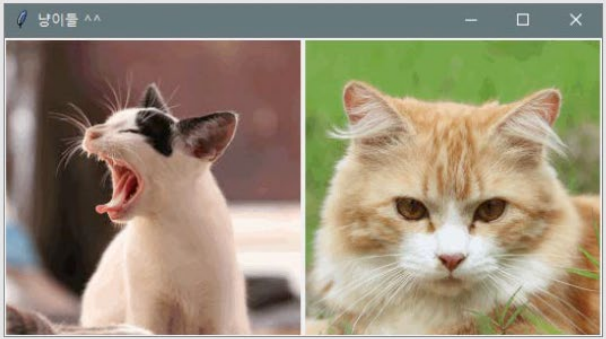
```
1  from tkinter import *
2  window = Tk()
3
4  photo = PhotoImage(file = "gif/dog.gif")
5  label1 = Label(window, image = photo)
6
7  label1.pack()
8
9  window.mainloop()
```



SELF STUDY 10-1

Code10-04.py를 수정해서 이미지를 2개 출력해 보자.

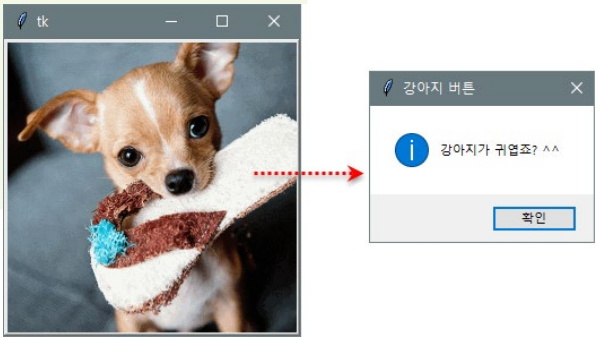
힌트 위젯을 가로로 나타내려면 pack(side=LEFT)를 사용한다.



- 예: 이미지 버튼을 누르면 간단한 메시지창이 나오는 코드

Code10-06.py

```
1  from tkinter import *
2  from tkinter import messagebox
3
4  ## 함수 선언 부분 ##
5  def myFunc() :
6      messagebox.showinfo("강아지 버튼", "강아지가 귀엽죠? ^^")
7
8  ## 메인 코드 부분 ##
9  window = Tk()
10
11 photo = PhotoImage(file = "gif/dog2.gif")
12 button1 = Button(window, image = photo, command = myFunc )
13
14 button1.pack()
15
16 window.mainloop()
```



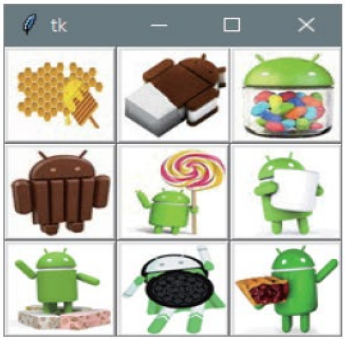
고정 위치에 배치

- 위젯을 고정 위치에 배치하려면 pack() 대신 place() 함수 사용
- 그림 9개를 2차원으로 배치하는 코드

Code10-11.py

```
1  from tkinter import *
2
3  ## 전역 변수 선언 부분 ##
4  btnList = [None] * 9
5  fnameList = ["froyo.gif", "gingerbread.gif", "honeycomb.gif", "icecream.gif",
       "jellybean.gif", "kitkat.gif", "lollipop.gif", "marshmallow.gif", "nougat.gif"]
6  photoList = [None] * 9
7  i, k = 0, 0
8  xPos, yPos = 0, 0
9  num = 0
10
11 ## 메인 코드 부분 ##
12 window = Tk()
13 window.geometry("210x210")
14
15 for i in range(0, 9) :
16     photoList[i] = PhotoImage(file = "gif/" + fnameList[i])
17     btnList[i] = Button(window, image = photoList[i])
```

```
18
19  for i in range(0, 3) :
20      for k in range(0, 3) :
21          btnList[num].place(x = xPos, y = yPos)
22          num += 1
23          xPos += 70
24      xPos = 0
25      yPos += 70
26
27  window.mainloop()
```



SELF STUDY 10-2

Code10–11.py를 실행할 때마다 그림을 임의로 뒤섞어서 나타내게 하자.

힌트 random 모듈을 임포트하고 shuffle(리스트) 함수를 사용하면 리스트를 임의로 뒤섞어 준다.

[프로그램 1]의 완성

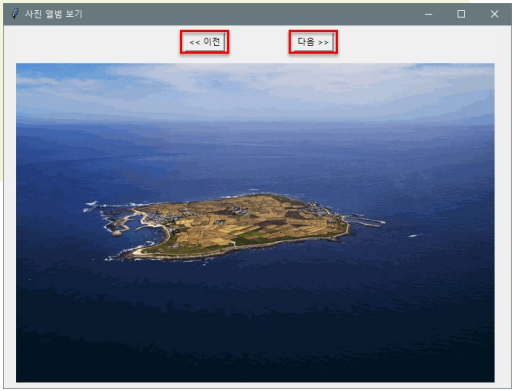
- <이전> 버튼이나 <다음> 버튼을 누르면 사진들을 표시하는 사진 앨범 프로그램

Code10-12.py

```
from tkinter import *
from time import *

## 전역 변수 선언 부분 ##
fnameList = ["jeju1.gif", "jeju2.gif", "jeju3.gif", "jeju4.gif", "jeju5.gif",
             "jeju6.gif", "jeju7.gif", "jeju8.gif", "jeju9.gif"]
photoList = [None] * 9
num = 0

## 함수 선언 부분 ##
def clickNext() :
    global num
    num += 1
    if num > 8 :
```

```
14          num = 0
15      photo = PhotoImage(file = "gif/" + fnameList[num])
16      pLabel.configure(image = photo)
17      pLabel.image = photo
18
19  def clickPrev() :
20      global num
21      num -= 1
22      if num < 0 :
23          num = 8
24      photo = PhotoImage(file = "gif/" + fnameList[num])
25      pLabel.configure(image = photo)
26      pLabel.image = photo
27
28  ## 메인 코드 부분 ##
29  window = Tk()
30  window.geometry("700x500")
31  window.title("사진 앨범 보기")
```

```
32
33  btnPrev = Button(window, text = "<< 이전", command = clickPrev)
34  btnNext = Button(window, text = "다음 >>", command = clickNext)
35
36  photo = PhotoImage(file = "gif/" + fnameList[0])
37  pLabel = Label(window, image = photo)
38
39  btnPrev.place(x = 250, y = 10)
40  btnNext.place(x = 400, y = 10)
41  pLabel.place(x = 15, y = 50)
42
43  window.mainloop()
```



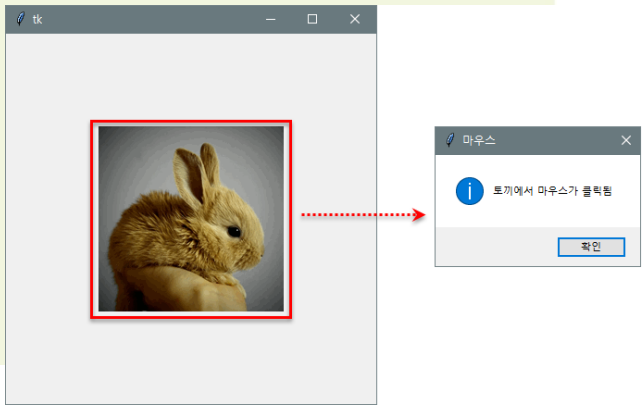
SELF STUDY 10-3

Code10-12.py를 수정해서 버튼 사이에 파일명을 표시해 보자.

- 지정된 위젯을 클릭했을 때 다른 함수 호출

Code10-14.py

```
1  from tkinter import *
2  from tkinter import messagebox
3
4  ## 함수 선언 부분 ##
5  def clickImage(event) :
6      messagebox.showinfo("마우스", "토끼에서 마우스가 클릭됨")
7
8  ## 메인 코드 부분 ##
9  window = Tk()
10 window.geometry("400x400")
11
12 photo = PhotoImage(file = "gif/rabbit.gif")
13 label1 = Label(window, image = photo)
14
15 label1.bind("<Button>", clickImage)
16
17 label1.pack(expand = 1, anchor = CENTER)
18 window.mainloop()
```



- 지정된 위젯을 클릭했을 때 다른 함수 호출

Code10-14.py

```
1  from tkinter import *
2  from tkinter import messagebox
3
4  ## 함수 선언 부분 ##
5  def clickImage(event) :
6      messagebox.showinfo("마우스", "토끼에서 마우스가 클릭됨")
7
8  ## 메인 코드 부분 ##
9  window = Tk()
10 window.geometry("400x400")
11
12 photo = PhotoImage(file = "gif/rabbit.gif")
13 label1 = Label(window, image = photo)
14
15 label1.bind("<Button>", clickImage)
16
17 label1.pack(expand = 1, anchor = CENTER)
18 window.mainloop()
```

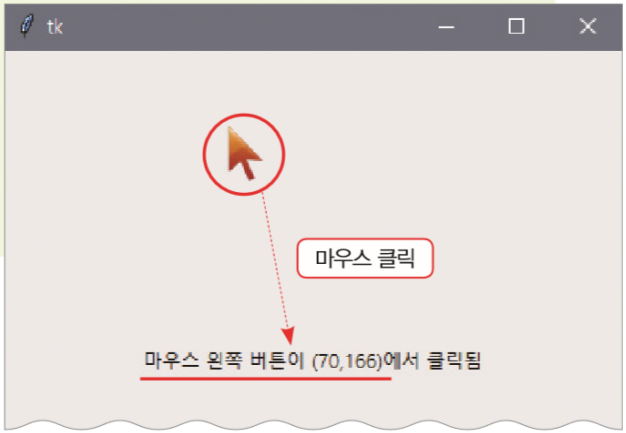
event 매개변수를 활용한 마우스 이벤트 처리

- 마우스를 클릭할 때마다 어떤 마우스가 클릭되었는지 보여 주고 클릭한 좌표 출력

Code10-15.py

```
from tkinter import *

## 함수 선언 부분 ##
def clickMouse( event ) :
    txt = ""
    if event.num == 1 :
        txt += "마우스 왼쪽 버튼이 ("
    elif event.num == 3 :
        txt += "마우스 오른쪽 버튼이 ("

    txt += str( event.y ) + "," + str( event.x ) + ")에서 클릭됨"
    label1.configure(text = txt)

## 메인 코드 부분 ##
window = Tk()
window.geometry("400x400")

```

```
18  label1 = Label(window, text = "이곳이 바뀜")
19
20  window.bind("<Button>", clickMouse)
21
22  label1.pack(expand = 1, anchor = CENTER)
23  window.mainloop()
```

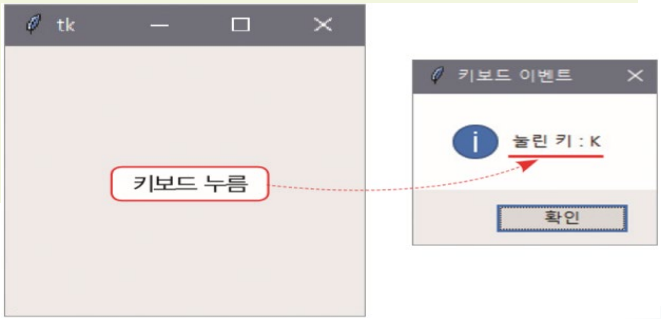


키보드 이벤트 기본 처리

- 키보드 이벤트는 위젯에서 키보드가 눌리면 발생

Code10-16.py

```
1  from tkinter import *
2  from tkinter import messagebox
3
4  ## 함수 선언 부분 ##
5  def keyEvent(event) :
6      messagebox.showinfo("키보드 이벤트", "눌린 키 : " + chr(event.keycode))
7
8  ## 메인 코드 부분 ##
9  window = Tk()
10
11 window.bind("<Key>", keyEvent)
12
13 window.mainloop()
```



- 키보드 이벤트

표 10-2 키보드 이벤트

키보드 작동	이벤트 코드
모든 키를 누를 때	<Key>
특수 키를 누를 때	<Return>, <BackSpace>, <Tab>, <Shift_L>, <Control_L>, <Alt_L>, <Pause>, <Caps_Lock>, <Escape>, <End>, <Home>, <Left>, <Right>, <Up>, <Down>, <Num_Lock>, <Delete>, <F1>~<F12> 등
일반 키를 누를 때	a~z, A~Z, 0~9, <space>, <less>
화살표 키와 조합	<Shift-Up>, <Shift-Down>, <Shift-Left>, <Shift-Right> 등

- Enter 를 처리하려면 Code10-16.py에서는 11행의 <Key> 대신 <Return>을 사용
- 대·소문자 등도 구분해서 처리 가능
- 소문자 r 은 11행의 <Key> 대신에 r을 사용해 처리
- 일반 키를 누를 때 주의할 점은 SpaceBar 는 <Space>로, < 는 <less>로 사용

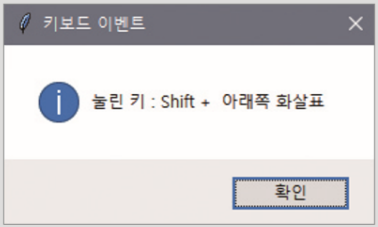
SELF STUDY 10-4

Code10-16.py를 수정해서 Shift+화살표 키를 누르면 화살표 키가 출력되도록 해 보자.

힌트1 이벤트 코드는 [표 10-2]를 참고한다.

힌트2 위젯.bind("이벤트", 함수) 코드를 한 위젯에 여러 개 작성해도 된다.

힌트3 화살표 키의 코드는 왼쪽 37, 위쪽 38, 오른쪽 39, 아래쪽 40이다.



메뉴의 생성

메뉴의 구성 개념과 형식

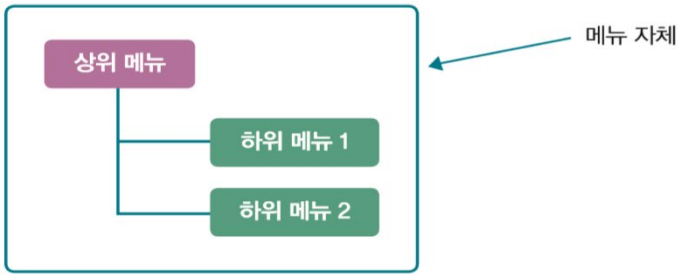


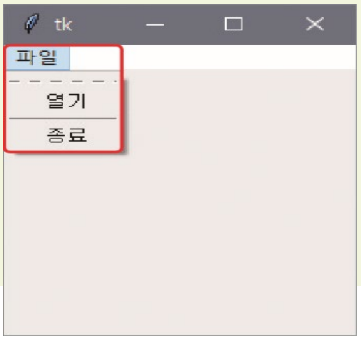
그림 10-1 메뉴의 구성 개념도

```
메뉴자체 = Menu(부모윈도)
부모윈도.config(menu = 메뉴자체)

상위메뉴 = Menu(메뉴자체)
메뉴자체.add_cascade(label = "상위메뉴텍스트", menu = 상위메뉴)
상위메뉴.add_command(label = "하위메뉴1", command = 함수1)
상위메뉴.add_command(label = "하위메뉴2", command = 함수2)
```

- [파일] 메뉴 아래에 [열기]와 [종료] 하위 메뉴가 있는 코드

Code10-17.py

```
1  from tkinter import *
2
3  window = Tk()
4
5  mainMenu = Menu(window)
6  window.config(menu = mainMenu)
7
8  fileMenu = Menu(mainMenu)
9  mainMenu.add_cascade(label = "파일", menu = fileMenu)
10 fileMenu.add_command(label = "열기")
11 fileMenu.add_separator()
12 fileMenu.add_command(label = "종료")
13
14 window.mainloop()
```

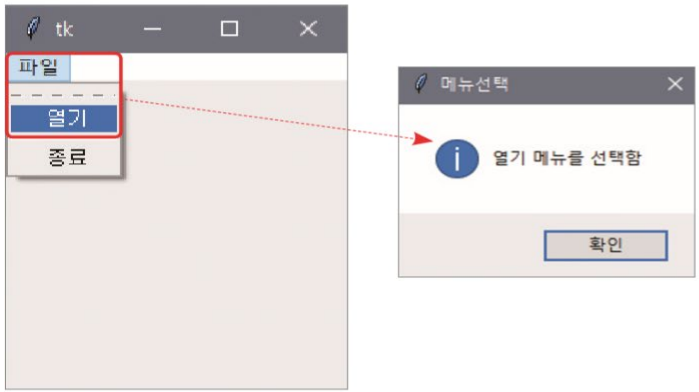


- 메뉴를 선택 하면 작동할 수 있도록 코드 추가

Code10-18.py

```
1  from tkinter import *
2  from tkinter import messagebox
3
4  ## 함수 선언 부분 ##
5  def func_open() :
6      messagebox.showinfo("메뉴선택", "열기 메뉴를 선택함")
7
8  def func_exit() :
9      window.quit()
10     window.destroy()
11
12 ## 메인 코드 부분 ##
13 window = Tk()
14
15 mainMenu = Menu(window)
16 window.config(menu = mainMenu)
17
```

```
18  fileMenu = Menu(mainMenu)
19  mainMenu.add_cascade(label = "파일", menu = fileMenu)
20  fileMenu.add_command(label = "열기", command = func_open)
21  fileMenu.add_separator()
22  fileMenu.add_command(label = "종료", command = func_exit)
23
24  window.mainloop()
```



■ 대화상자의 생성과 사용

- tkinter.simpledialog 모듈을 임포트한 후 askinteger() 및 askstring() 등을 사용

Code10-19.py

```
1  from tkinter import *
2  from tkinter.simpledialog import *
3
4  ## 함수 선언 부분 ##
5  window = Tk()
6  window.geometry("400x100")
7
8  label1 = Label(window, text = "입력된 값")
9  label1.pack()
10
11 value = askinteger("확대배수", "주사위 숫자(1~6)을 입력하세요", minvalue = 1, maxvalue = 6)
12
13 label1.configure(text = str(value))
14 window.mainloop()
```

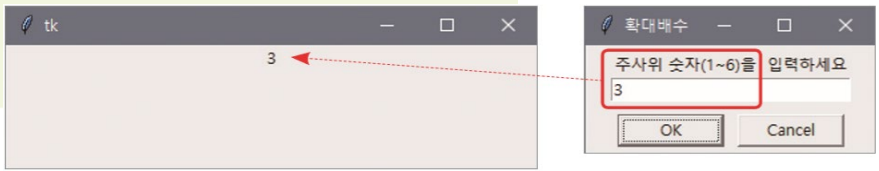
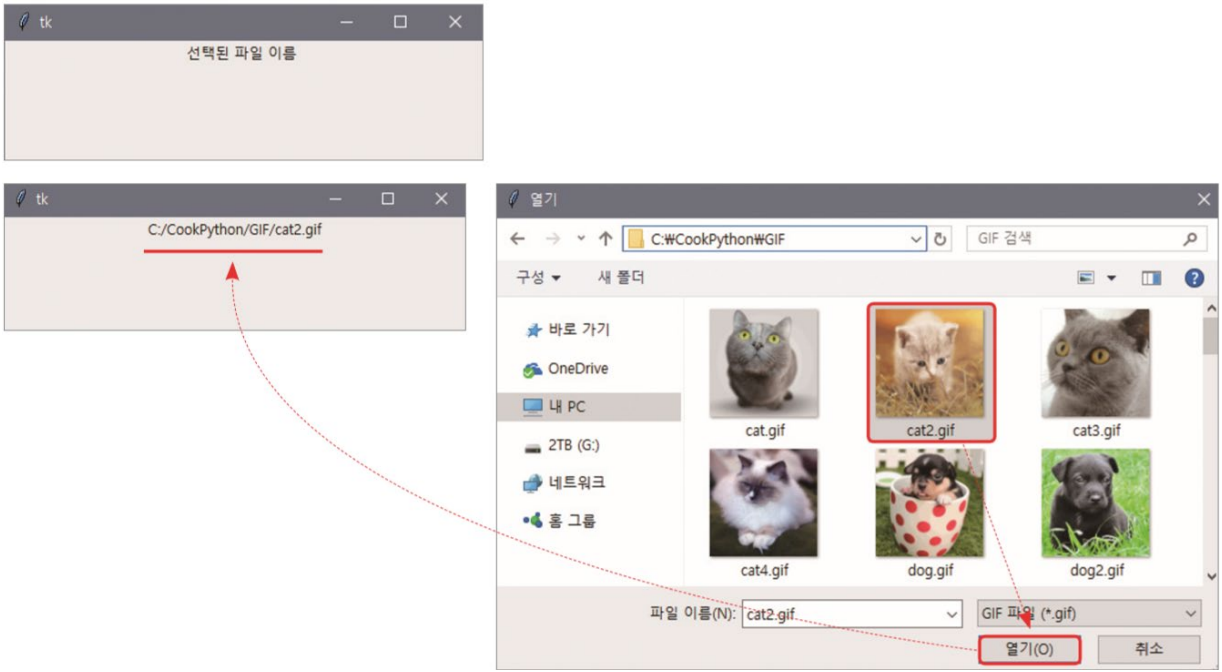


그림 파일인 GIF 파일을 선택하는 코드

- Code10-20.py는 Code10-19.py의 2행과 11행만 변경

Code10-20.py

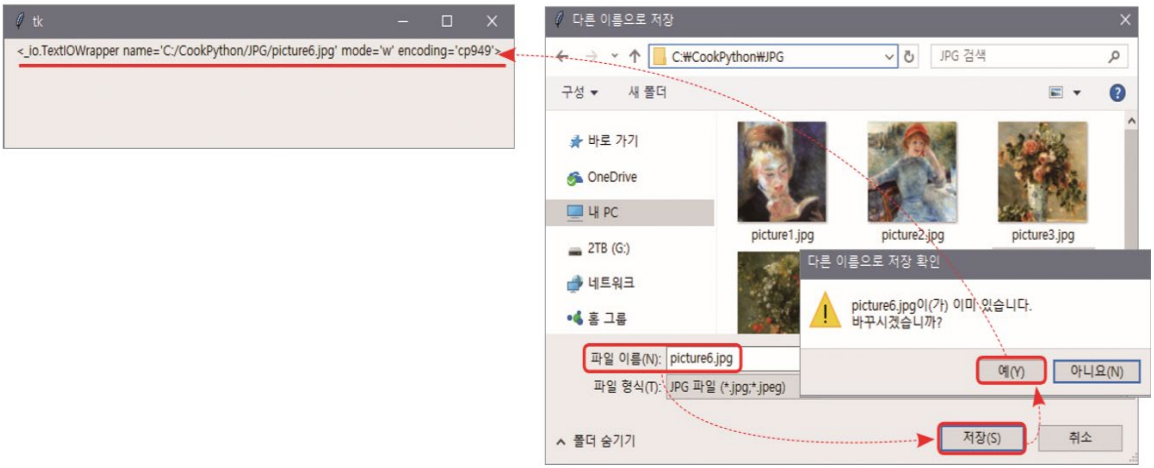
```
1  from tkinter import *
2  from tkinter.filedialog import *
3
4  ## 함수 선언 부분 ##
5  window = Tk()
6  window.geometry("400x100")
7
8  label1 = Label(window, text = "선택된 파일 이름")
9  label1.pack()
10
11 filename = askopenfilename(parent = window, filetypes = (("GIF 파일", "*.gif"),
               ("모든 파일", "*.*")))
12
13 label1.configure(text = str(filename))
14
15 window.mainloop()
```


tk
선택된 파일 이름
tk
C:/CookPython/GIF/cat2.gif
열기
C:₩CookPython₩GIF
GIF 검색
구성
새 폴더
바로 가기
OneDrive
내 PC
2TB (G:)
네트워크
홈 그룹
cat.gif
cat2.gif
cat3.gif
cat4.gif
dog.gif
dog2.gif
파일 이름(N):
cat2.gif
GIF 파일 (*.gif)
열기(O)
취소

- Code10-20.py의 11~13행 변경하고 실행

Code10-21.py

```
11  saveFp = asksaveasfile(parent = window, mode = "w", defaultextension = ".jpg",
            filetypes = (("JPG 파일", "*.jpg;*.jpeg"), ("모든 파일", "*.*")))
12  label1.configure(text = saveFp)
13  saveFp.close()
```



[프로그램 2]의 완성

- 메뉴 처리와 파일 처리가 핵심

Code10-22.py

```
from tkinter import *
from tkinter.filedialog import *

## 함수 선언 부분 ##
def func_open() :
    filename = askopenfilename(parent = window, filetypes = (("GIF 파일", "*.gif"),
                ("모든 파일", "*.*")))
    photo = PhotoImage(file = filename)
    pLabel.configure(image = photo)
    pLabel.image = photo

def func_exit() :
    window.quit()
    window.destroy()

## 메인 코드 부분 ##
window = Tk()
window.geometry("400x400")
```

```
18  window.title("명화 감상하기")
19
20  photo = PhotoImage()
21  pLabel = Label(window, image = photo)
22  pLabel.pack(expand = 1, anchor = CENTER)
23
24  mainMenu = Menu(window)
25  window.config(menu = mainMenu)
26  fileMenu = Menu(mainMenu)
27  mainMenu.add_cascade(label = "파일", menu = fileMenu)
28  fileMenu.add_command(label = "파일 열기", command = func_open)
29  fileMenu.add_separator()
30  fileMenu.add_command(label = "프로그램 종료", command = func_exit)
31
32  window.mainloop()
```