

# 10\_2 윈도우 프로그래밍

## Section03 위젯의 배치와 크기 조절

### ■ [프로그램 1]의 완성

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

Code10-12.py

```
1 from tkinter import *
2 from time import *
3
4 ## 전역 변수 선언 부분 ##
5 fnameList = ["jeju1.gif", "jeju2.gif", "jeju3.gif", "jeju4.gif", "jeju5.gif",
               "jeju6.gif", "jeju7.gif", "jeju8.gif", "jeju9.gif"]
6 photoList = [None] * 9
7 num = 0
8
9 ## 함수 선언 부분 ##
10 def clickNext() :
11     global num
12     num += 1
13     if num > 8 :
```

5행 : fnameList 변수에 사진 9장의 파일명을 저장  
6행 : photoList에는 PhotoImage() 함수로 생성할 변수 9개 준비  
7행 : num은 현재 사진의 번호  
10~17행 : <다음> 버튼을 누르면 실행되는 함수  
11행 : global num은 num 전역 변수를 함수 안에서 사용 의미  
12행 : 사진 번호를 하나 증가  
13~14행 : 사진 번호가 최대 8, 8 넘으면 0번 사진 표시

## Section03 위젯의 배치와 크기 조절

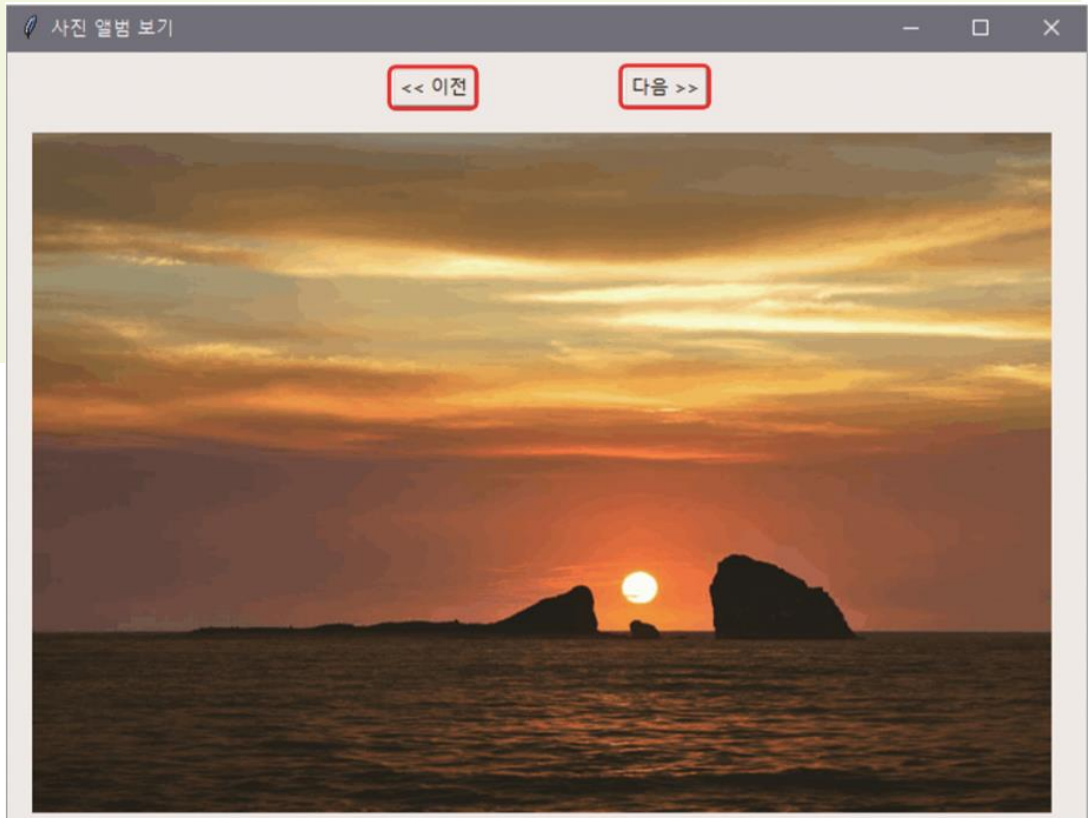
```
14         num = 0
15         photo = PhotoImage(file = "gif/" + fnameList[num])
16         pLabel.configure(image = photo)
17         pLabel.image = photo
18
19     def clickPrev() :
20         global num
21         num -= 1
22         if num < 0 :
23             num = 8
24         photo = PhotoImage(file = "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")
```

15~17행 : 변경된 사진 번호에 해당하는 이미지 파일로 pLabel 변경  
19~26행 : <이전> 버튼을 누르면 처리되는 함수

## Section03 위젯의 배치와 크기 조절

```
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()
```

33~34행 : 버튼 누르면 이 함수와 연결  
36~37행 : 프로그램 실행 후 첫 번째  
          사진 표시  
39~41행 : 버튼 및 이미지 위치  
          place(x=좌표, y=좌표)로 지정



## Section03 위젯의 배치와 크기 조절

### SELF STUDY 10-3

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



## Section04 키보드와 마우스 이벤트 처리

### ■ 마우스 이벤트 기본 처리

표 10-1 마우스 이벤트

마우스 작동	마우스 버튼	이벤트 코드
클릭할 때	모든 버튼 공통	〈Button〉
	왼쪽 버튼	〈Button-1〉
	가운데 버튼	〈Button-2〉
	오른쪽 버튼	〈Button-3〉
떼었을 때	모든 버튼 공통	〈ButtonRelease〉
	왼쪽 버튼	〈ButtonRelease-1〉
	가운데 버튼	〈ButtonRelease-2〉
	오른쪽 버튼	〈ButtonRelease-3〉
더블클릭할 때	모든 버튼 공통	〈Double-Button〉
	왼쪽 버튼	〈Double-Button-1〉
	가운데 버튼	〈Double-Button-2〉
	오른쪽 버튼	〈Double-Button-3〉
드래그할 때	왼쪽 버튼	〈B1-Motion〉
	가운데 버튼	〈B2-Motion〉
	오른쪽 버튼	〈B3-Motion〉
마우스 커서가 위젯 위로 올라왔을 때		〈Enter〉
마우스 커서가 위젯에서 떠났을 때		〈Leave〉

## Section04 키보드와 마우스 이벤트 처리

- 마우스 이벤트가 처리 형식

```
def 이벤트처리함수(event) :  
    # 이 부분에 마우스 이벤트가 발생할 때 작동할 내용 작성  
  
위젯.bind("마우스이벤트", 이벤트처리함수)
```

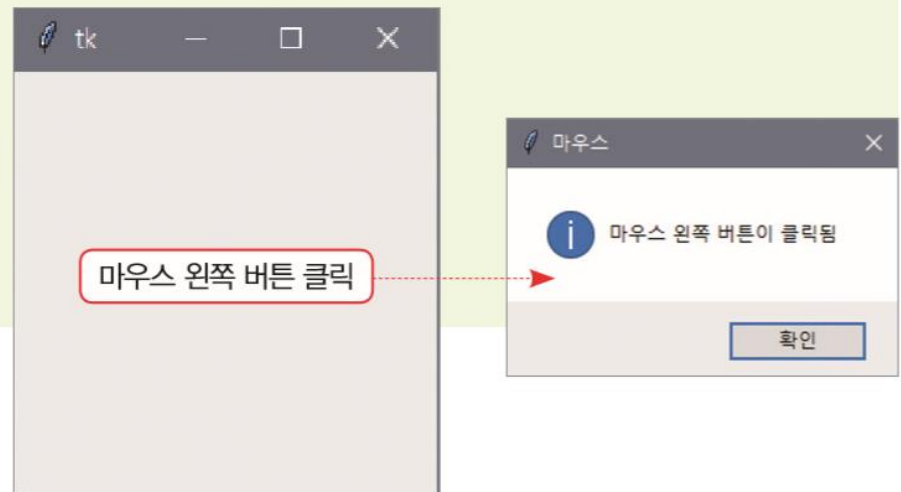
## Section04 키보드와 마우스 이벤트 처리

- 마우스 왼쪽 버튼을 클릭했을 때 처리하는 방법

Code10-13.py

```
1 from tkinter import *
2 from tkinter import messagebox
3
4 ## 함수 선언 부분 ##
5 def clickLeft(event) :
6     messagebox.showinfo("마우스", "마우스 왼쪽 버튼이 클릭됨")
7
8 ## 메인 코드 부분 ##
9 window = Tk()
10
11 window.bind("<Button-1>", clickLeft)
12
13 window.mainloop()
```

5~6행 : 마우스 이벤트가 발생할 때 작동할 함수 정의  
11행 : window.bind() 함수에는 마우스 왼쪽 버튼 클릭  
할 때 발생하는 이벤트인 <Button-1> 설정하고  
5행의 clickLeft 함수명을 지정





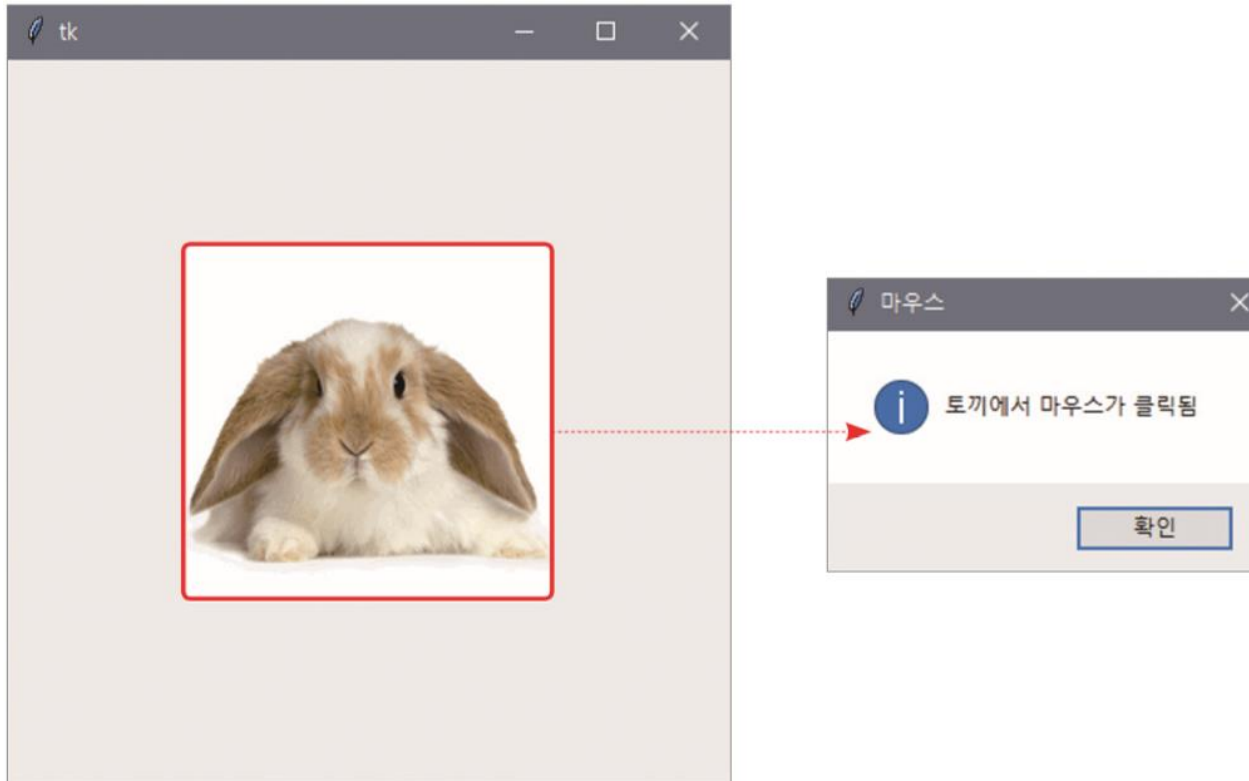
## Section04 키보드와 마우스 이벤트 처리

- 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()
```

## Section04 키보드와 마우스 이벤트 처리



Code10-14.py는 이미지 클릭할 때만 이벤트 처리(15행에서 `window.bind()`가 아닌 `label1.bind()`를 사용했기 때문)  
또 `<Button>` 이벤트를 사용했기 때문에 어떤 마우스 버튼 눌러도 메시지창 표시

## Section04 키보드와 마우스 이벤트 처리

- 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()
```

## Section04 키보드와 마우스 이벤트 처리

### ■ event 매개변수를 활용한 마우스 이벤트 처리

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

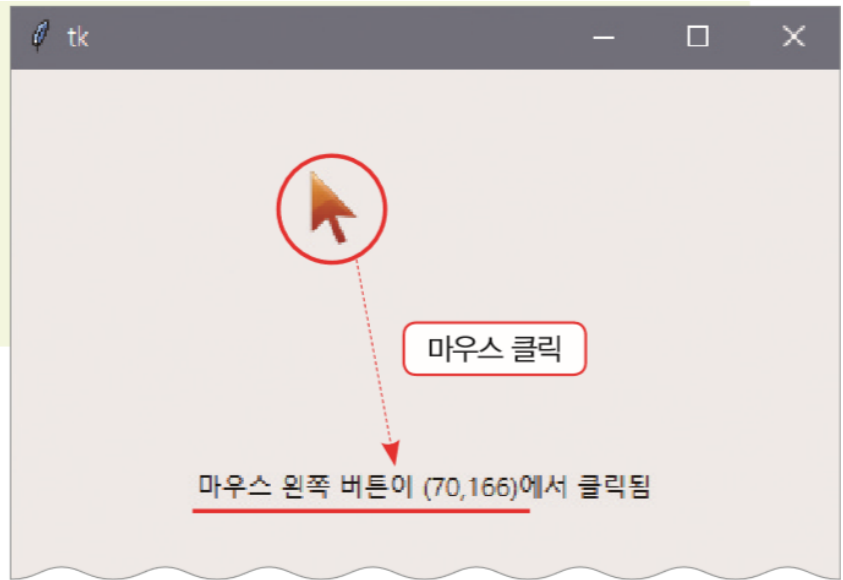
Code10-15.py

```
1  from tkinter import *
2
3  ## 함수 선언 부분 ##
4  def clickMouse(event) :
5      txt = ""
6      if event.num == 1 :
7          txt += "마우스 왼쪽 버튼이 ("
8      elif event.num == 3 :
9          txt += "마우스 오른쪽 버튼이 ("
10
11     txt += str(event.y) + "," + str(event.x) + ")에서 클릭됨"
12     label1.configure(text = txt)
13
14  ## 메인 코드 부분 ##
15  window = Tk()
16  window.geometry("400x400")
17
```

4~12행 : 마우스 클릭할 때 실행될 이벤트 함수 선언  
20행 : 마우스 클릭하면 함수 호출  
6~9행 : event.num값은 마우스 왼쪽 버튼 클릭했을 때 1값 가지고, 마우스 오른쪽 버튼 클릭했을 때 2값 가짐  
11행 : event.x와 event.y는 클릭한 위치의 좌표를 가짐  
12행 : 18행에서 화면에 표시한 레이블의 글자 변경

## Section04 키보드와 마우스 이벤트 처리

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



# Section04 키보드와 마우스 이벤트 처리

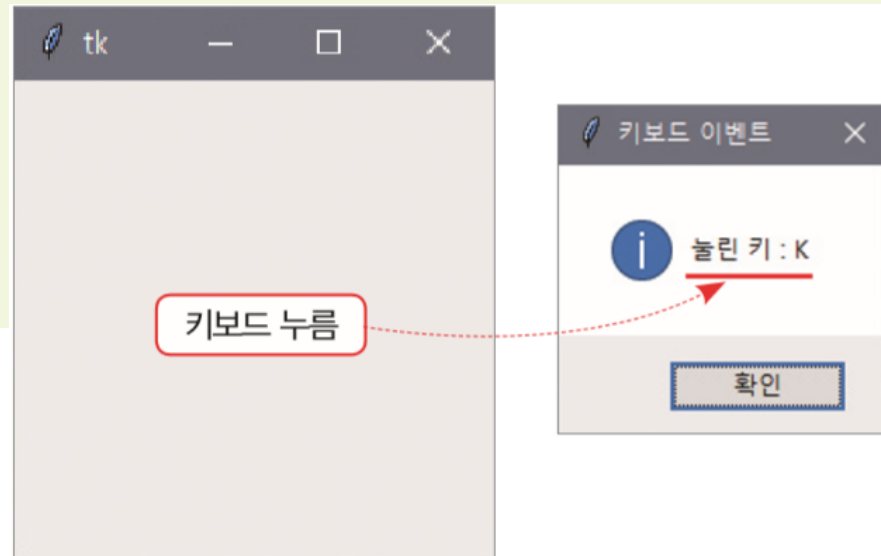
## ■ 키보드 이벤트 기본 처리

- 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()
```

5~6행 : 키보드가 눌릴 때 작동하는 함수 선언  
event.keycode에는 눌려진 키의 숫자 값 들어 있으므로  
chr() 함수를 사용하면 문자로 변환.  
11행 : <Key> 이벤트를 윈도우창에 사용



## Section04 키보드와 마우스 이벤트 처리

### ■ 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

## Section04 키보드와 마우스 이벤트 처리

### 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.

