Tkinter

Seolyoung Jeong, Ph.D.

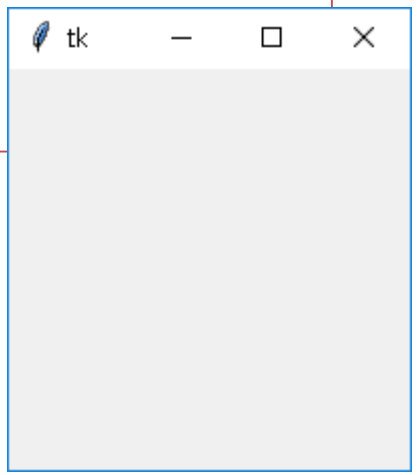
경북대학교 IT대학 컴퓨터학부

파이썬 GUI 라이브러리

Tkinter

- **GUI 관련 파이썬 표준 라이브러리**

```
1  # import tkinter module
2  from tkinter import *
3
4  # Tk instance
5  win = Tk()
6
7  win.mainloop()
```

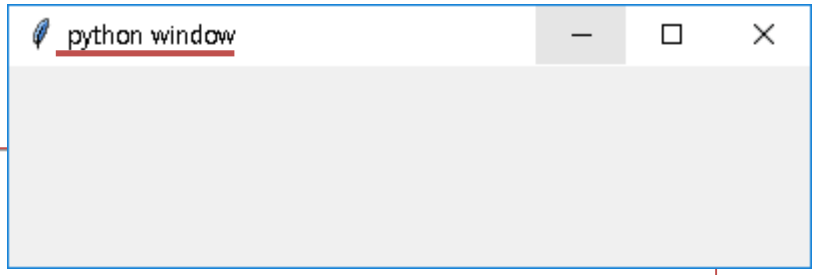


- 1행 : tkinter 모듈 import
- 5행 : Tk 클래스 인스턴스 생성 (기본 윈도우)
- 7행 : 생성된 인스턴스(win)의 mainloop() 함수 호출

- **mainloop() : 이벤트 메시지 루프**
  - 키보드나 마우스 혹은 화면 다시그리기와 같은 다양한 이벤트로부터 오는 메시지를 받고 전달하는 역할

기본 윈도우창

- **윈도우창 제목 및 윈도우 크기**
  - title("제목"), geometry("400x100")




```python
# import tkinter module
from tkinter import *

# Tk instance
win = Tk()
win.title("python window")      # window title
win.geometry("400x100")         # window size

win.mainloop()
```

- **윈도우 크기 고정 :** resizable(width=False, height=False)

```python
# import tkinter module
from tkinter import *

# Tk instance
win = Tk()
win.title("python window")      # window title
win.geometry("400x100")         # window size
win.resizable(width=False, height=False)

win.mainloop()
```

위젯

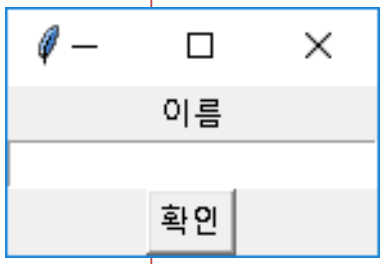
- **Widget** : 윈도우 창에 나올 수 있는 문자, 버튼, 체크박스, 라디오버튼 등의 컴포넌트

위젯	설명
Button	단순한 버튼
Label	텍스트 혹은 이미지 표시
CheckButton	체크박스
Entry	단순한 한 라인 텍스트 박스
ListBox	리스트 박스
RadioButton	옵션 버튼
Message	Label과 비슷하게 텍스트 표시. Label과 달리 자동 래핑 기능이 있다.
Scale	슬라이스 바
Scrollbar	스크롤 바
Text	멀티 라인 텍스트 박스로서 일부 Rich Text 기능 제공
Menu	메뉴 Pane
Menubutton	메뉴 버튼
Toplevel	새 윈도우를 생성할 때 사용. Tk()는 윈도우를 자동으로 생성하지만 추가로 새 윈도우 혹은 다이얼로그를 만들 경우 Toplevel를 사용한다
Frame	컨테이너 위젯. 다른 위젯들을 그룹화할 때 사용
Canvas	그래프와 점들로 그림을 그릴 수 있으며, 커스텀 위젯을 만드는데 사용될 수도 있다

기본 위젯 사용

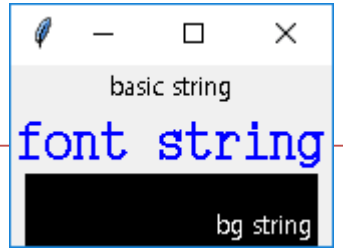
- 이름 문자 (Label Widget)
- 텍스트박스 입력창 (Entry Widget)
- 확인 버튼 (Button Widget)


```python
# import tkinter module
from tkinter import *

# Tk instance
win = Tk()
win.title("python window")      # window title
#win.geometry("400x100")         # window size
#win.resizable(width=False, height=False)

# label
la_name = Label(win, text="이름")
la_name.pack()

# textbox
in_tb = Entry(win)
in_tb.pack()

# button
btn_ok = Button(win, text="확인")
btn_ok.pack()

win.mainloop()
```



Label

- 문자 표시 : 폰트변경, 글자색, 배경색, 크기, 정렬 등

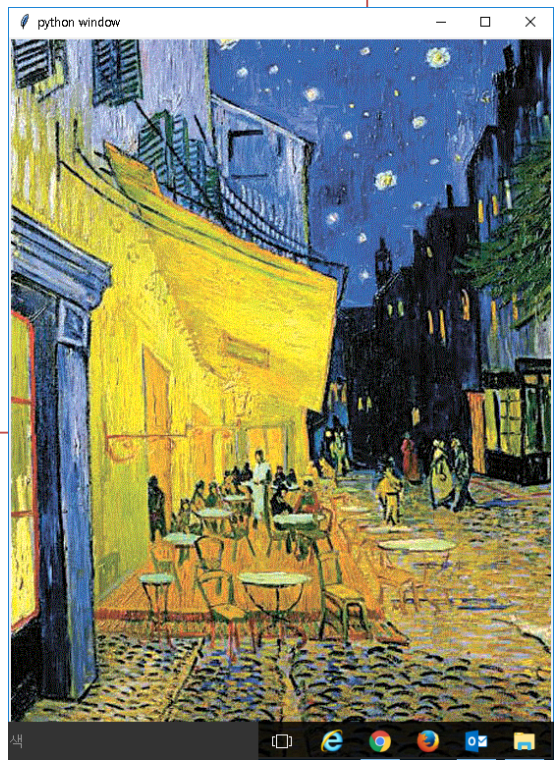



```python
# import tkinter module
from tkinter import *

# Tk instance
win = Tk()                      # root(base) window
win.title("python window")      # title

# label
la_str1 = Label(win, text="basic string")
la_str1.pack()

# font
la_str2 = Label(win, text="font string", font=("궁서체",20), fg="blue")
la_str2.pack()

# background
# anchor = CENTER(default), N, NE, NW, S, SE, SW, E, W
la_str3 = Label(win, text="bg string", bg="black", fg="white", width=20, height=2, anchor=SE)
la_str3.pack()

win.mainloop()
```

Label

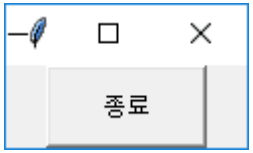
- 이미지 표시
- PhotoImage 클래스 : gif 혹은 png 이미지 파일 읽기 가능
  (다른 이미지 포맷을 읽기 위해서는 외부 모듈 사용해야 함)

```
1   # import tkinter module
2   from tkinter import *
3
4   # Tk instance
5   win = Tk()                          # root(base) window
6   win.title("python window")          # title
7
8   # label
9   image = PhotoImage(file="cafe.gif")
10  la_img = Label(win, image=image)
11  la_img.pack()
12
13  win.mainloop()
```



Button

- **“종료” 버튼 누르면 프로그램 종료**
  - command 파라미터 = quit

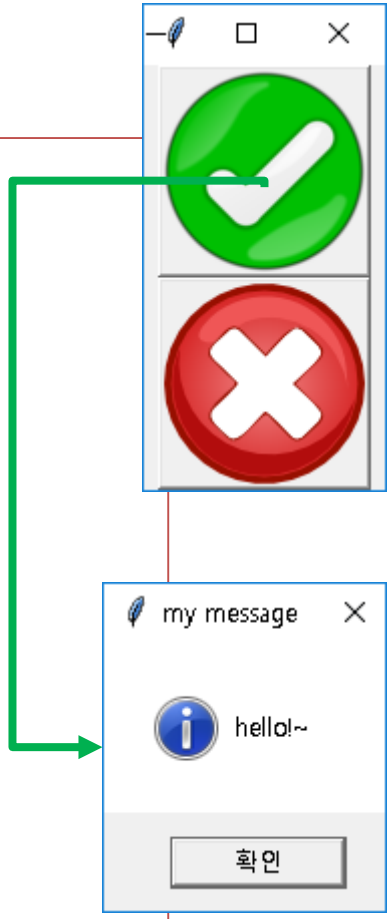



```
1   # import tkinter module
2   from tkinter import *
3
4   # Tk instance
5   win = Tk()                          # root(base) window
6   win.title("python window")          # title
7
8   # button
9   btn_ok = Button(win, text="종료", width=10, height=2, command=quit)
10  btn_ok.pack()
11
12  win.mainloop()
```

Button

- 이미지 버튼을 누르면 함수 호출(메시지 박스)

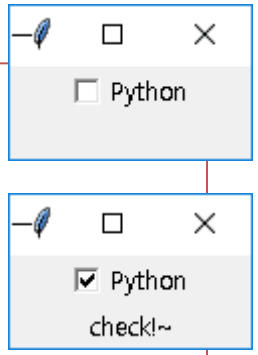
```
1   # import tkinter module
2   from tkinter import *
3   from tkinter import messagebox
4
5   # function
6   def btn_func():
7       messagebox.showinfo("my message", "hello!~")
8
9   # Tk instance
10  win = Tk()
11  win.title("python window")
12
13  # image buttons
14  img_ok = PhotoImage(file="btn_ok.png")
15  btn_ok = Button(win, image=img_ok, command=btn_func)
16  btn_ok.pack()
17
18  img_exit = PhotoImage(file="btn_exit.png")
19  btn_exit = Button(win, image=img_exit, command=quit)
20  btn_exit.pack()
21
22  win.mainloop()
```



Checkbox

- 선택 시 수행 함수 : 선택 결과 표시

```python
# import tkinter module
from tkinter import *
from tkinter import messagebox

# function
def check_func():
    if var_chk.get() == 1:
        la_result.configure(text="check!~")
    else:
        la_result.configure(text="")

# Tk instance
win = Tk()
win.title("python window")

# check variable of int type
var_chk = IntVar()

# check button
chk_ok = Checkbutton(win, text="Python", variable=var_chk, command=check_func)

# result text label
la_result = Label(win)

chk_ok.pack()
la_result.pack()

win.mainloop()
```



RadioButton (1)

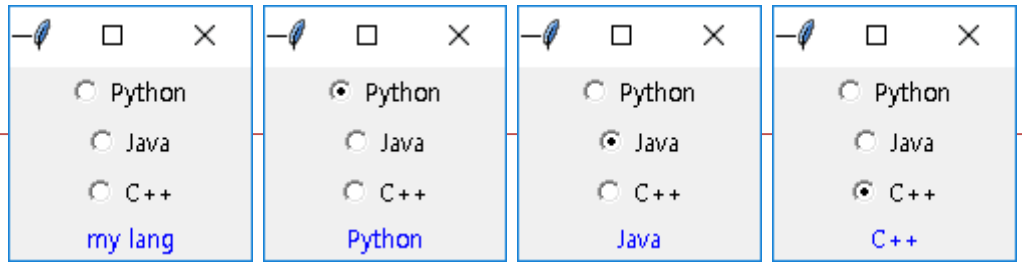
- 여러 개 옵션 중 한 개 선택
- 선택 시 수행 함수 : 선택에 따른 변수(var_lang)에 따라 문자 표시

```
1   # import tkinter module
2   from tkinter import *
3
4   # function
5   def radio_func():
6       if var_lang.get() == 1:
7           la_result.configure(text="Python")
8       elif var_lang.get() == 2:
9           la_result.configure(text="Java")
10      else:
11          la_result.configure(text="C++")
12
13  # Tk instance
14  win = Tk()
15  win.title("python window")
16
```

RadioButton (2)

- IntVar() : 정수형 타입의 변수 생성
- 각 radiobutton별 변수값 및 수행할 함수 파라미터 전달
- 결과 문자 표시할 Label

```
17  # radio variable of int type
18  var_lang = IntVar()
19
20  # radio button
21  rb1 = Radiobutton(win, text="Python", variable=var_lang, value=1, command=radio_func)
22  rb2 = Radiobutton(win, text="Java", variable=var_lang, value=2, command=radio_func)
23  rb3 = Radiobutton(win, text="C++", variable=var_lang, value=3, command=radio_func)
24
25  # result text label
26  la_result = Label(win, text="my lang", fg="blue")
27
28  rb1.pack()
29  rb2.pack()
30  rb3.pack()
31  la_result.pack()
32
33  win.mainloop()
```



위젯의 배치

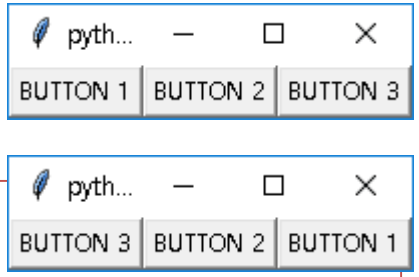
위젯의 화면 배치

- **위젯을 화면에 배치하는 세 가지 방법**
  - pack() : 위젯들을 부모에 패킹하여 불필요한 공간을 없앰
  - place() : 위젯 위치를 절대 좌표로 정함
    (윈도우 크기 변경에 따라 위젯들이 변경되지 않으므로 많이 사용되지 않음)
  - grid() : 위젯들을 테이블 레이아웃에 배치. 지정된 row, column에 위젯을 놓음

- **configure() : 이미 생성된 위젯을 변경하고자 할 때 사용하는 함수**

수평 정렬

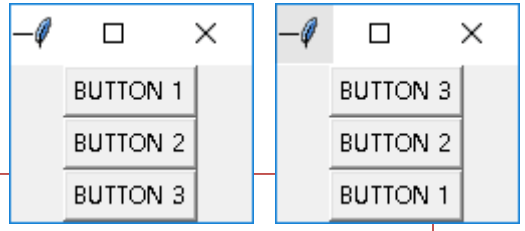
- 위젯.pack(side=LEFT or RIGHT)




```
1   # import tkinter module
2   from tkinter import *
3
4   # Tk instance
5   win = Tk()
6   win.title("python window")
7
8   # buttons
9   btn1 = Button(win, text="BUTTON 1")
10  btn2 = Button(win, text="BUTTON 2")
11  btn3 = Button(win, text="BUTTON 3")
12
13  btn1.pack(side=LEFT)
14  btn2.pack(side=LEFT)
15  btn3.pack(side=LEFT)
16
17  win.mainloop()
```

수직 정렬

- 위젯.pack(side=TOP or BOTTOM)

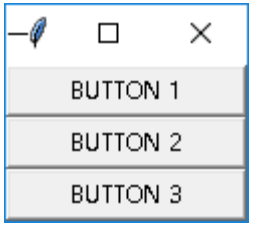


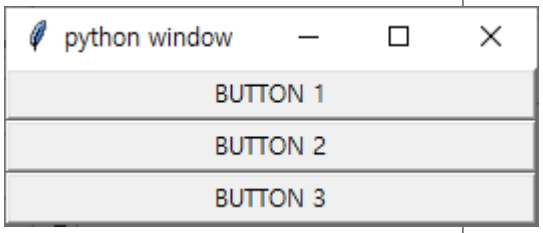

```
1   # import tkinter module
2   from tkinter import *
3
4   # Tk instance
5   win = Tk()
6   win.title("python window")
7
8   # buttons
9   btn1 = Button(win, text="BUTTON 1")
10  btn2 = Button(win, text="BUTTON 2")
11  btn3 = Button(win, text="BUTTON 3")
12
13  btn1.pack(side=TOP)
14  btn2.pack(side=TOP)
15  btn3.pack(side=TOP)
16
17  win.mainloop()
```

정렬 및 폭 맞추기

- fill=X(수평 맞추기) 또는 fill=Y(수직 맞추기)

```
1   # import tkinter module
2   from tkinter import *
3
4   # Tk instance
5   win = Tk()
6   win.title("python window")
7
8   # buttons
9   btn1 = Button(win, text="BUTTON 1")
10  btn2 = Button(win, text="BUTTON 2")
11  btn3 = Button(win, text="BUTTON 3")
12
13  btn1.pack(side=TOP, fill=X)
14  btn2.pack(side=TOP, fill=X)
15  btn3.pack(side=TOP, fill=X)
16
17  win.mainloop()
```





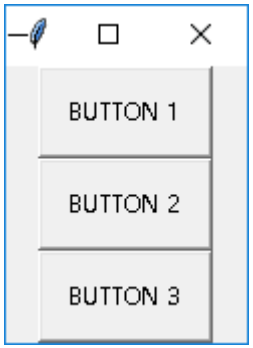
위젯 여백

- 외부 여백 : padx=픽셀값, pady=픽셀값

```
13  btn1.pack(side=TOP, fill=X, padx=10, pady=10)
14  btn2.pack(side=TOP, fill=X, padx=10, pady=10)
15  btn3.pack(side=TOP, fill=X, padx=10, pady=10)
```

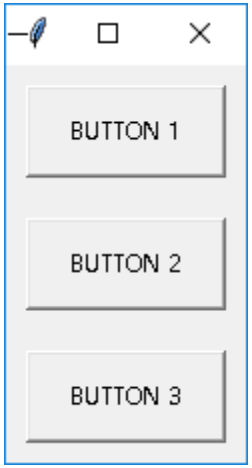
- 내부 여백 : ipadx=픽셀값, ipady=픽셀값

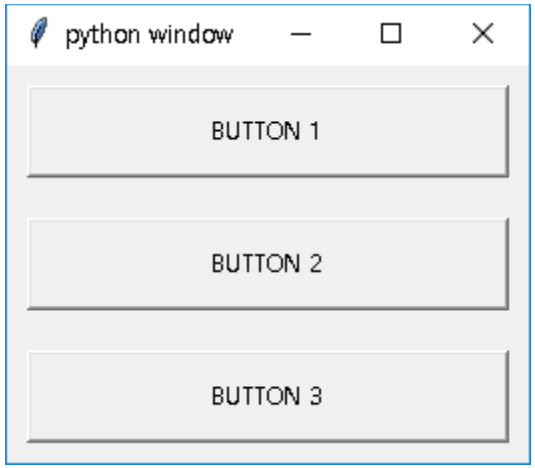
```
13  btn1.pack(side=TOP, fill=X, ipadx=10, ipady=10)
14  btn2.pack(side=TOP, fill=X, ipadx=10, ipady=10)
15  btn3.pack(side=TOP, fill=X, ipadx=10, ipady=10)
```



위젯 여백

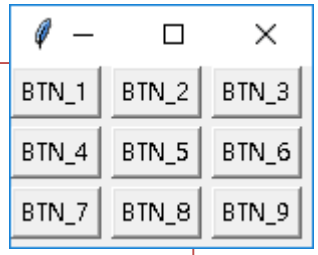
```
13  btn1.pack(side=TOP, fill=X, padx=10, pady=10, ipadx=10, ipady=10)
14  btn2.pack(side=TOP, fill=X, padx=10, pady=10, ipadx=10, ipady=10)
15  btn3.pack(side=TOP, fill=X, padx=10, pady=10, ipadx=10, ipady=10)
```





고정 위치 배치

- **place(x=X좌표, y=Y좌표, width=폭, height=높이)**

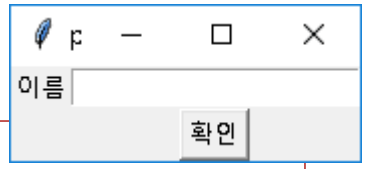



```python
# import tkinter module
from tkinter import *

# Tk instance
win = Tk()
win.title("python window")
win.geometry("150x90+10+10")

# buttons
btn_list = [None]*9

# create button
for i in range(9):
    btn_str = format("BTN_%d" % (i+1))
    btn_list[i] = Button(win, text=btn_str)

# set position
num = 0
pos_x, pos_y = 0, 0
for n in range(3):
    for m in range(3):
        btn_list[num].place(x=pos_x, y=pos_y)
        num += 1
        pos_x += 50
    pos_x = 0
    pos_y +=30

win.mainloop()
```

레이아웃 배치

- **grid(row=0, column=0)**

```python
# import tkinter module
from tkinter import *

# Tk instance
win = Tk()                          # root(base) window
win.title("python window")          # title

# label
la_name = Label(win, text="이름")
la_name.grid(row=0, column=0)

# textbox
in_tb = Entry(win)
in_tb.grid(row=0, column=1)

# button
btn_ok = Button(win, text="확인")
btn_ok.grid(row=1, column=1)

win.mainloop()
```

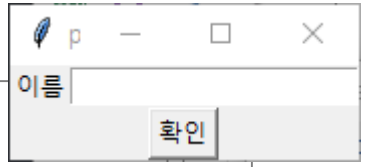


row=0, column=0	row=0, column=1
row=1, column=0	row=1, column=1

레이아웃 배치

- **grid(rowspan=2, columnspan=2)**

```python
# import tkinter module
from tkinter import *

# Tk instance
win = Tk()                          # root(base) window
win.title("python window")          # title

# label
la_name = Label(win, text="이름")
la_name.grid(row=0, column=0)

# textbox
in_tb = Entry(win)
in_tb.grid(row=0, column=1)

# button
btn_ok = Button(win, text="확인")
btn_ok.grid(row=1, columnspan=2)

win.mainloop()
```



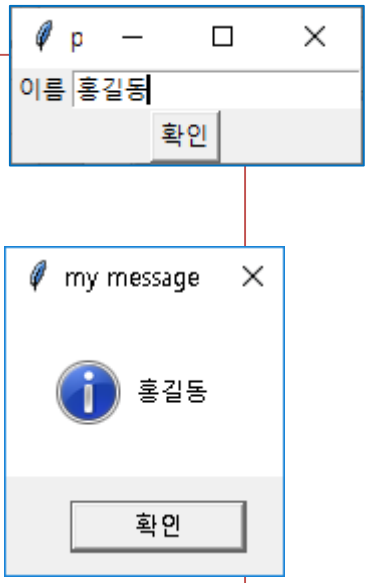
row=0, column=0	row=0, column=1
row=1, columnspan=2	

이벤트 처리

버튼 기본 이벤트

- 버튼 클릭 시 입력창 데이터 보여줌

```python
# import tkinter module
from tkinter import *
from tkinter import messagebox

# function
def btn_func():
    str_output = in_tb.get()
    messagebox.showinfo("my message", str_output)

# Tk instance
win = Tk()
win.title("python window")

# label
la_name = Label(win, text="이름")
la_name.grid(row=0, column=0)

# textbox
in_tb = Entry(win)
in_tb.grid(row=0, column=1)

# button
btn_ok = Button(win, text="확인", command=btn_func)
btn_ok.grid(row=1, columnspan=2)

win.mainloop()
```



마우스 이벤트

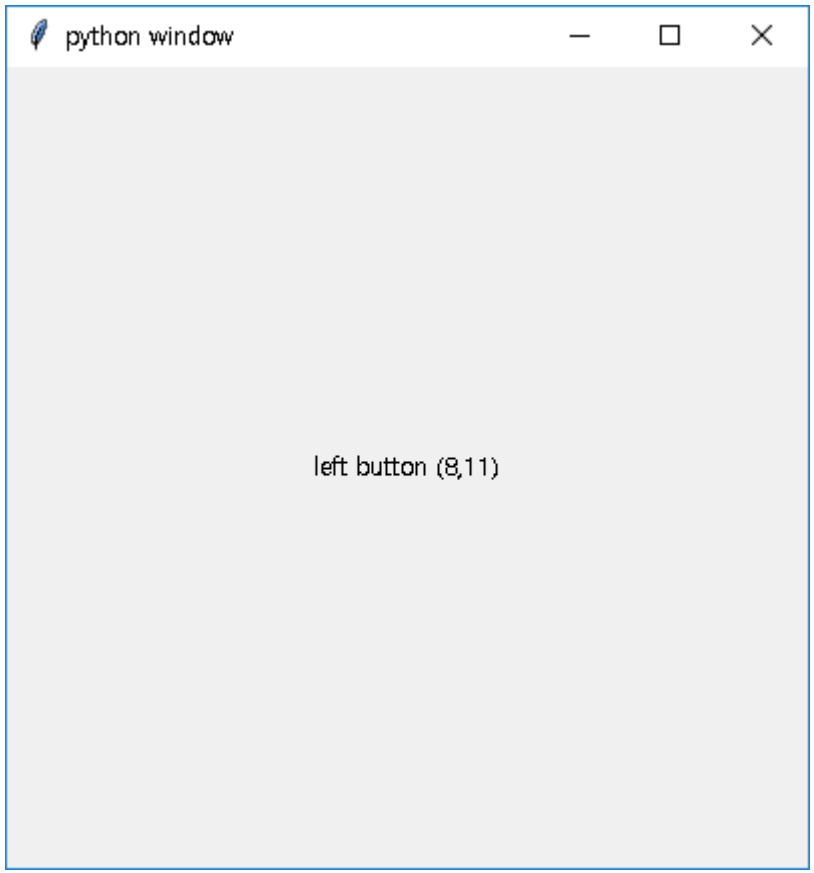
- **mainloop() 함수에서 이벤트 발생을 기다림**

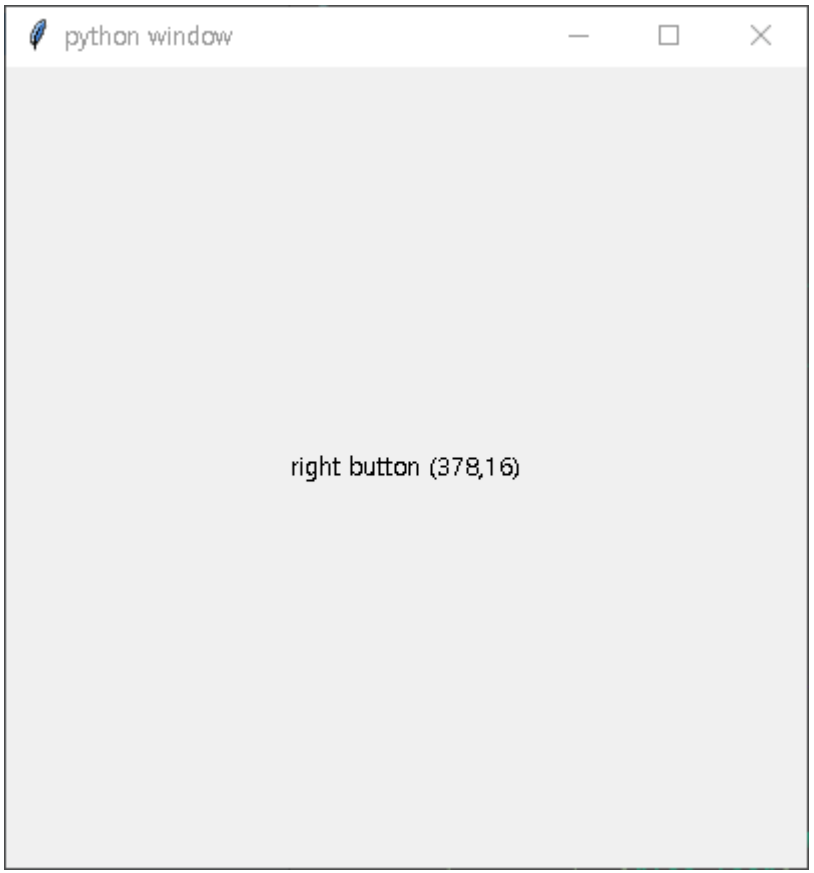
마우스 작동	관련 마우스 버튼	이벤트 코드
클릭할 때	모든 버튼 공통	〈Button〉
	왼쪽 버튼	〈Button-1〉
	가운데 버튼	〈Button-2〉
	오른쪽 버튼	〈Button-3〉
떼었을 때	모든 버튼 공통	〈ButtonRelease〉
	왼쪽 버튼	〈ButtonRelease-1〉
	가운데 버튼	〈ButtonRelease-2〉
	오른쪽 버튼	〈ButtonRelease-3〉
더블클릭할 때	모든 버튼 공통	〈Double-Button〉
	왼쪽 버튼	〈Double-Button-1〉
	가운데 버튼	〈Double-Button-2〉
	오른쪽 버튼	〈Double-Button-3〉
드래그할 때	왼쪽 버튼	〈B1-Motion〉
	가운데 버튼	〈B2-Motion〉
	오른쪽 버튼	〈B3-Motion〉

마우스 작동	이벤트 코드
마우스 커서가 위젯 위로 올라왔을 때	〈Enter〉
마우스 커서가 위젯에서 떠났을 때	〈Leave〉

버튼 이벤트 바인딩

- **bind(event, handler)**
- **핸들러 : 이벤트 발생 시 수행할 함수, 메소드**





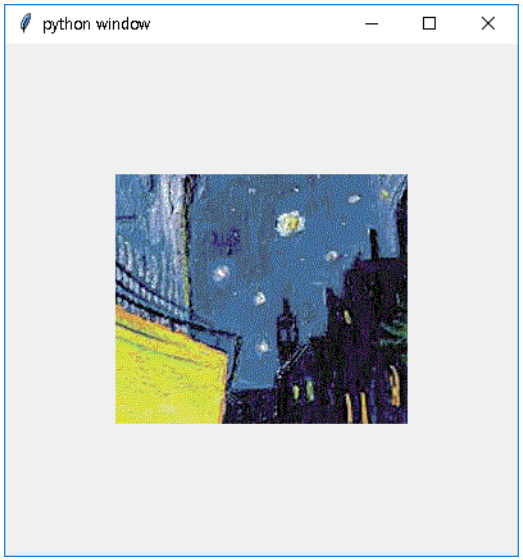
```python
# import tkinter module
from tkinter import *

# function
def my_func(event):
    txt = ""

    # click event : left / right
    if event.num == 1:
        txt = "left button ("
    elif event.num == 3:
        txt = "right button ("

    # click position
    pos_x = event.x
    pos_y = event.y
    txt += str(pos_x) + "," + str(pos_y) + ")"

    # label text
    la_txt.configure(text=txt)

# Tk instance
win = Tk()                          # root(base) window
win.title("python window")          # title
win.geometry("400x400")

# bind (base window, button)
win.bind("<Button>", my_func)

# label
la_txt = Label(win)
la_txt.pack(expand=True, anchor=CENTER)

win.mainloop()
```

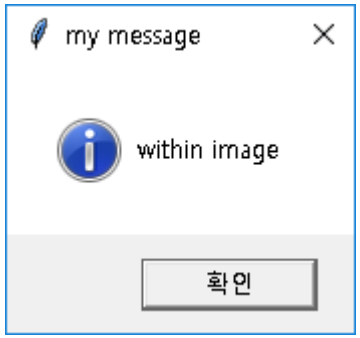
 # 이미 존재하는 label image 변경 시
photo = PhotoImage(file="파일명")
la_image.configure(image=photo)
la_image.image = photo

버튼 이벤트 바인딩

- 위젯.bind(event, handler)
- 이미지 Label에 마우스 이벤트 바인딩

```
1   # import tkinter module
2   from tkinter import *
3   from tkinter import messagebox
4
5   # function
6   def my_func(event):
7       messagebox.showinfo("my message", "within image")
8
9   # Tk instance
10  win = Tk()                          # root(base) window
11  win.title("python window")          # title
12  win.geometry("400x400")
13
14  # label
15  image = PhotoImage(file="cafe_part.gif")
16  la_img = Label(win, image=image)
17  la_img.pack(expand=True, anchor=CENTER)
18
19  # bind (image label, button)
20  la_img.bind("<Button>", my_func)
21
22  win.mainloop()
```





키보드 이벤트 바인딩

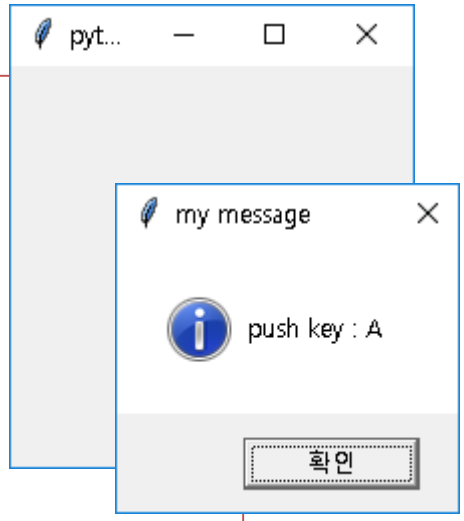
- **키보드 이벤트**

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

- **event.keycode : 눌러진 키 코드값**
- **chr(event.keycode) : 코드값을 문자로 변환**

키 이벤트 바인딩

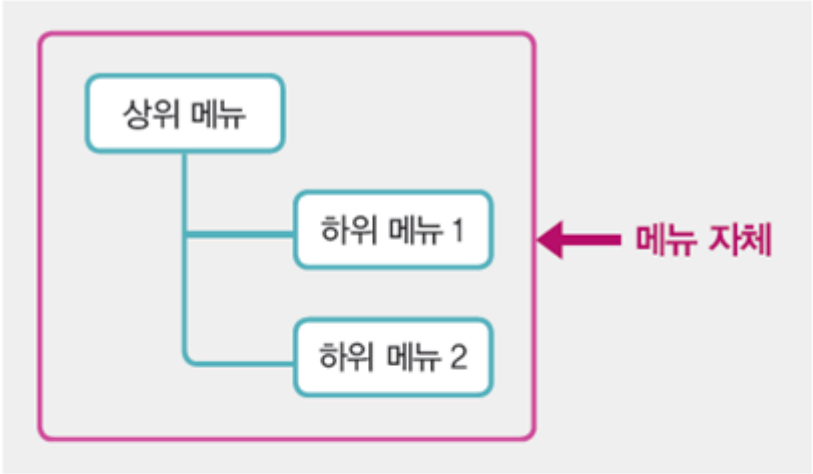
```python
# import tkinter module
from tkinter import *
from tkinter import messagebox

# function
def my_func(event):
    txt = "push key : " + chr(event.keycode)
    messagebox.showinfo("my message", txt)

# Tk instance
win = Tk()                          # root(base) window
win.title("python window")          # title

# bind (base window, button)
win.bind("<Key>", my_func)

win.mainloop()
```



윈도우 메뉴

메뉴 생성

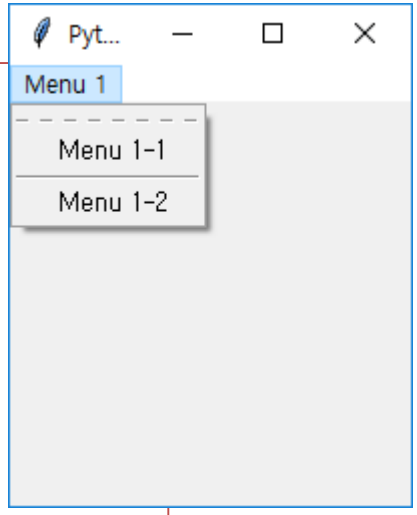
- 메뉴 구성 개념도




```
메뉴 자체=Menu(부모 윈도우)
부모 윈도우.config(menu=메뉴 자체)

상위 메뉴=Menu(메뉴 자체)
메뉴 자체.add_cascade(label="상위 메뉴텍스트, menu=상위 메뉴)
상위 메뉴.add_command(label="하위 메뉴1", command=함수1)
상위 메뉴.add_command(label="하위 메뉴2", command=함수2)
```

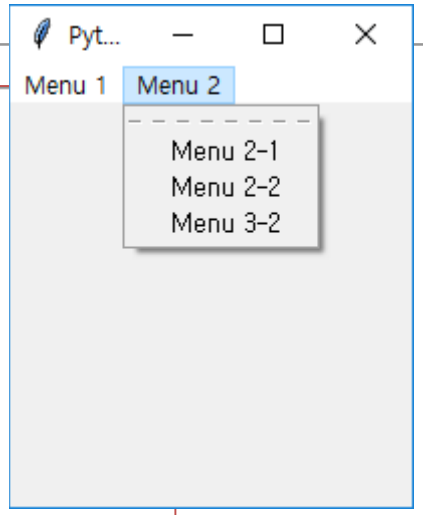
메뉴 생성

```python
from tkinter import *       # import module

# main window
win = Tk()
win.title("Python Programming")

# create main menu
main_menu = Menu(win)
win.config(menu=main_menu)

# create sub menu
menu1 = Menu(main_menu)

# main-sub menu cascade
main_menu.add_cascade(label="Menu 1", menu=menu1)

# add sub menu (open, exit)
menu1.add_command(label="Menu 1-1")
menu1.add_separator()
menu1.add_command(label="Menu 1-2")

# event waiting
win.mainloop()
```



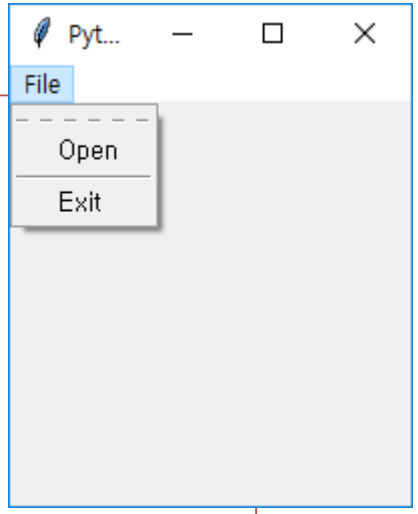
메뉴 생성

```
1   from tkinter import *       # import module
2
3   # main window
4   win = Tk()
5   win.title("Python Programming")
6
7   # create main menu
8   main_menu = Menu(win)
9   win.config(menu=main_menu)
10
11  # create sub menu
12  menu1 = Menu(main_menu)
13  menu2 = Menu(main_menu)
14
15  # main-sub menu cascade
16  main_menu.add_cascade(label="Menu 1", menu=menu1)
17  main_menu.add_cascade(label="Menu 2", menu=menu2)
18
19  # add sub menu (menu 1)
20  menu1.add_command(label="Menu 1-1")
21  menu1.add_separator()
22  menu1.add_command(label="Menu 1-2")
23
24  # add sub menu (menu 2)
25  menu2.add_command(label="Menu 2-1")
26  menu2.add_command(label="Menu 2-2")
27  menu2.add_command(label="Menu 3-2")
28
29  # event waiting
30  win.mainloop()
```



메뉴 생성

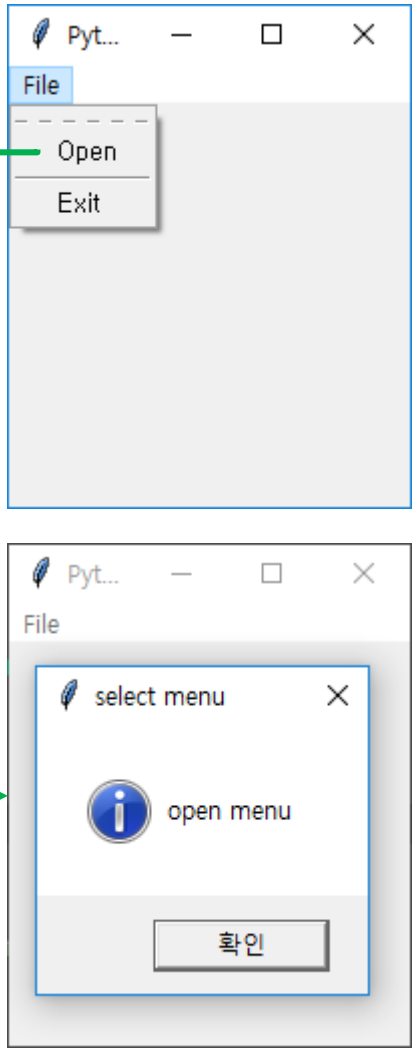
- [파일] - [열기], [종료] 메뉴 구성




```
1   from tkinter import *       # import module
2
3   # main window
4   win = Tk()
5   win.title("Python Programming")
6
7   # create main menu
8   main_menu = Menu(win)
9   win.config(menu=main_menu)
10
11  # create sub menu
12  menu1_file = Menu(main_menu)
13
14  # main-sub menu cascade
15  main_menu.add_cascade(label="File", menu=menu1_file)
16
17  # add sub menu (open, exit)
18  menu1_file.add_command(label="Open")
19  menu1_file.add_separator()
20  menu1_file.add_command(label="Exit")
21
22  # event waiting
23  win.mainloop()
```

메뉴 생성

- [파일] - [열기], [종료]




```python
from tkinter import *       # import module
from tkinter import messagebox

# func
def func_open():
    messagebox.showinfo("select menu", "open menu")

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


# main window
win = Tk()
win.title("Python Programming")

# create main menu
main_menu = Menu(win)
win.config(menu=main_menu)

# create sub menu
menu1_file = Menu(main_menu)

# main-sub menu cascade
main_menu.add_cascade(label="File", menu=menu1_file)

# add sub menu (open, exit)
menu1_file.add_command(label="Open", command=func_open)
menu1_file.add_separator()
menu1_file.add_command(label="Exit", command=func_exit)

# event waiting
win.mainloop()
```

Dialog 연동

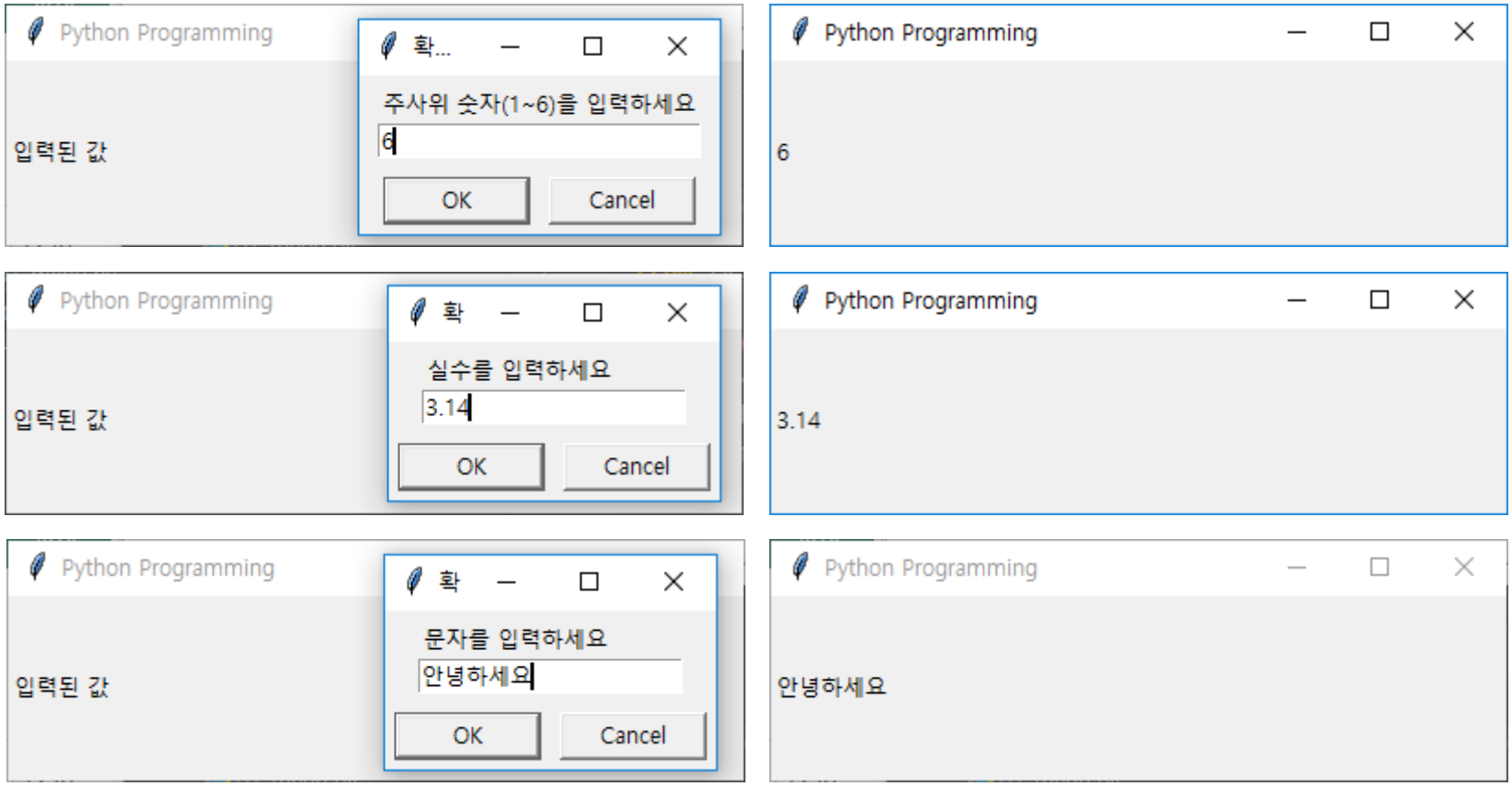
SimpleDialog

- 메인 window와 간단한 대화형 dialog 연결

```
1   # import module
2   from tkinter import *
3   from tkinter.simpledialog import *
4
5   # main window
6   win = Tk()
7   win.title("Python Programming")
8   win.geometry("400x100")
9
10  # label
11  la_var = Label(win, text="입력된 값")
12  la_var.pack(side=LEFT)
13
14  # dialog
15  value = askinteger("확대배수", "주사위 숫자(1~6)을 입력하세요", minvalue=1, maxvalue=6)
16
17  # show result
18  la_var.configure(text=str(value))
19
20  # event waiting
21  win.mainloop()
```

SimpleDialog

- tkinter.simpledialog 모듈 import 필요
- askinteger() : 정수를 입력 받는 다이얼로그
- askfloat() : 실수를 입력 받는 다이얼로그
- askstring() : 문자열을 입력 받는 다이얼로그



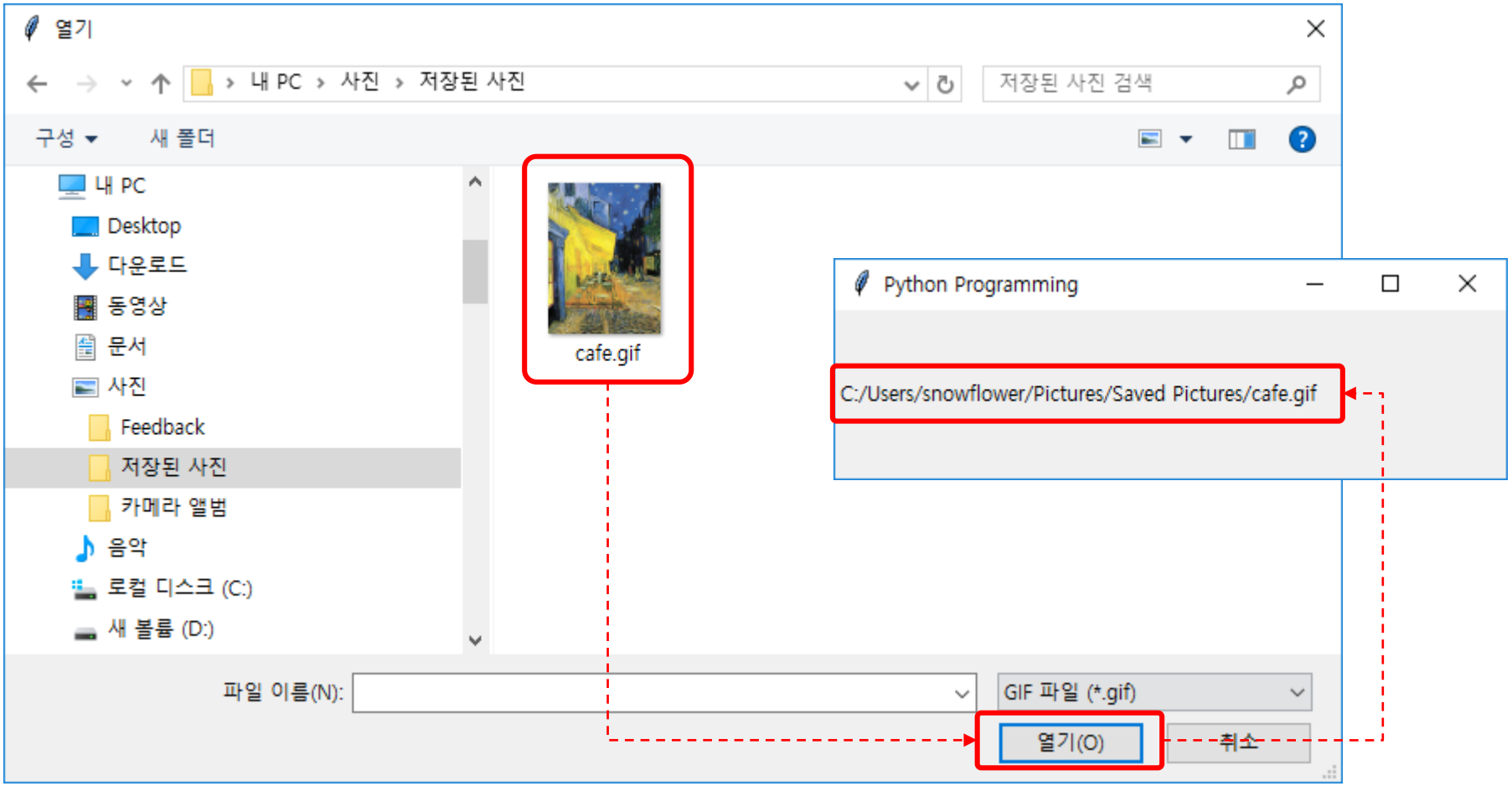
FileDialog

- **메인 window와 파일 dialog 연결**
  - tkinter.filedialog 모듈 import 필요
  - askopenfilename() : 파일 열기 다이얼로그 → 열기 → 경로 포함 파일명 return

```
1  # import module
2  from tkinter import *
3  from tkinter.filedialog import *
4
5  # main window
6  win = Tk()
7  win.title("Python Programming")
8  win.geometry("400x100")
9
10 # label
11 la_var = Label(win, text="선택된 파일명")
12 la_var.pack(side=LEFT)
13
14 # dialog
15 filename = askopenfilename(parent=win, filetypes=(("GIF 파일", "*.gif"), ("모든 파일", "*.*")))
16
17 # show result
18 la_var.configure(text=filename)
19
20 # event waiting
21 win.mainloop()
```

FileDialog

- 메인 window와 파일 dialog 연결



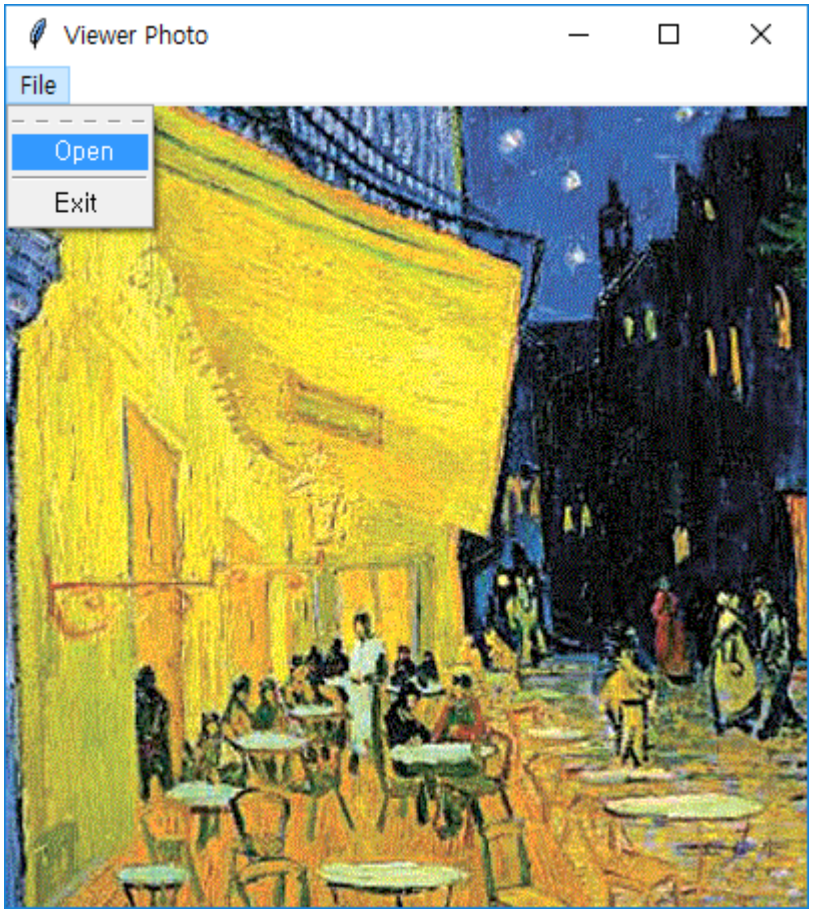
이미지 열기 프로그램 (1)

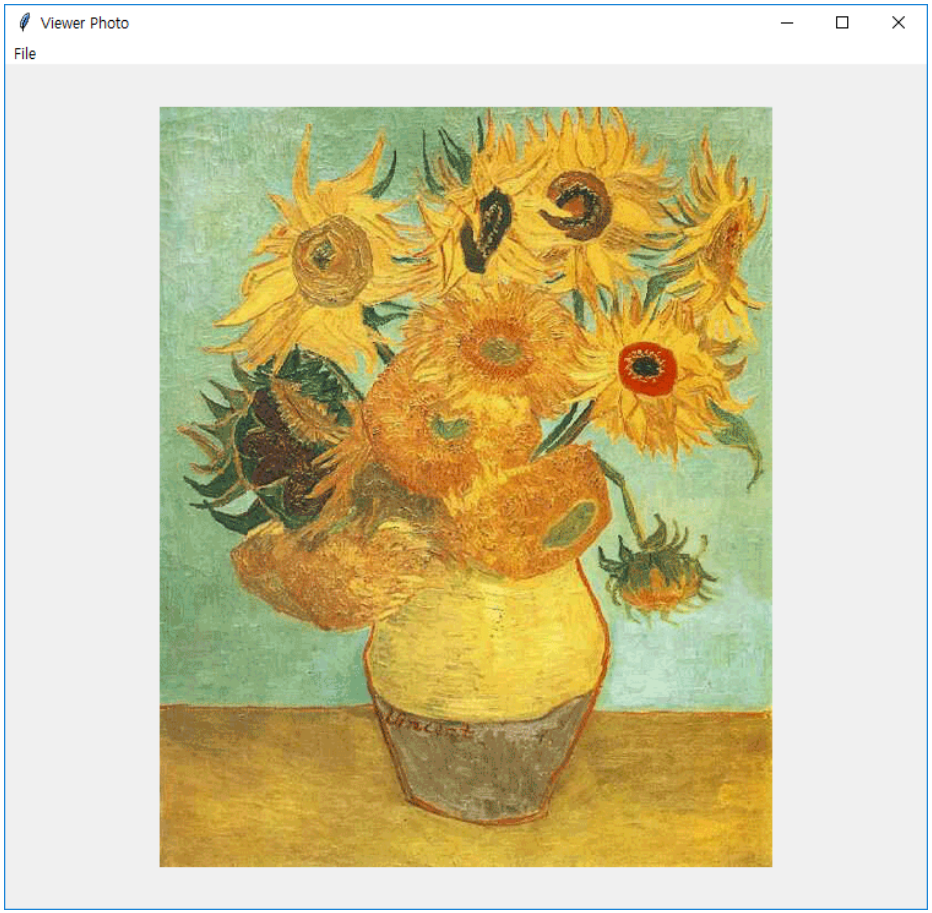
```
# import module
from tkinter import *
from tkinter.filedialog import *

# func
def func_open():
    filename = askopenfilename(parent=win, filetypes=(("GIF 파일", "*.gif"), ("모든 파일", "*.*")))
    photo = PhotoImage(file=filename)
    la_image.configure(image=photo)
    la_image.image = photo

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

# main window
win = Tk()
win.title("Viewer Photo")
win.geometry("400x400")

```

이미지 열기 프로그램 (2)

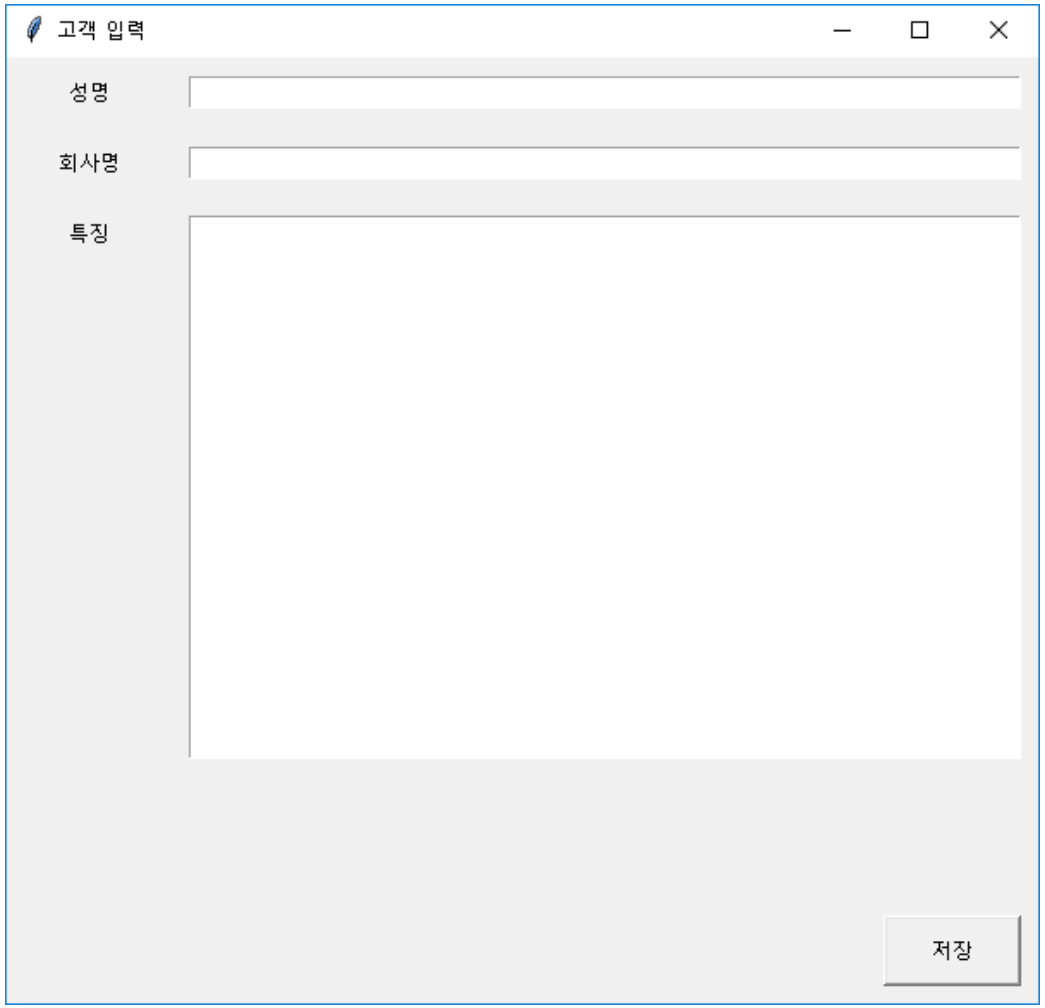
```
21  # main menu
22  main_menu = Menu(win)
23  win.config(menu=main_menu)
24
25  # create sub menu
26  menu1_file = Menu(main_menu)
27
28  # main-sub menu cascade
29  main_menu.add_cascade(label="File", menu=menu1_file)
30
31  # add sub menu (open, exit)
32  menu1_file.add_command(label="Open", command=func_open)
33  menu1_file.add_separator()
34  menu1_file.add_command(label="Exit", command=func_exit)
35
36  # image label
37  la_image = Label(win)
38  la_image.pack(expand=True, anchor=CENTER)
39
40  # event waiting
41  win.mainloop()
```

이미지 열기 프로그램



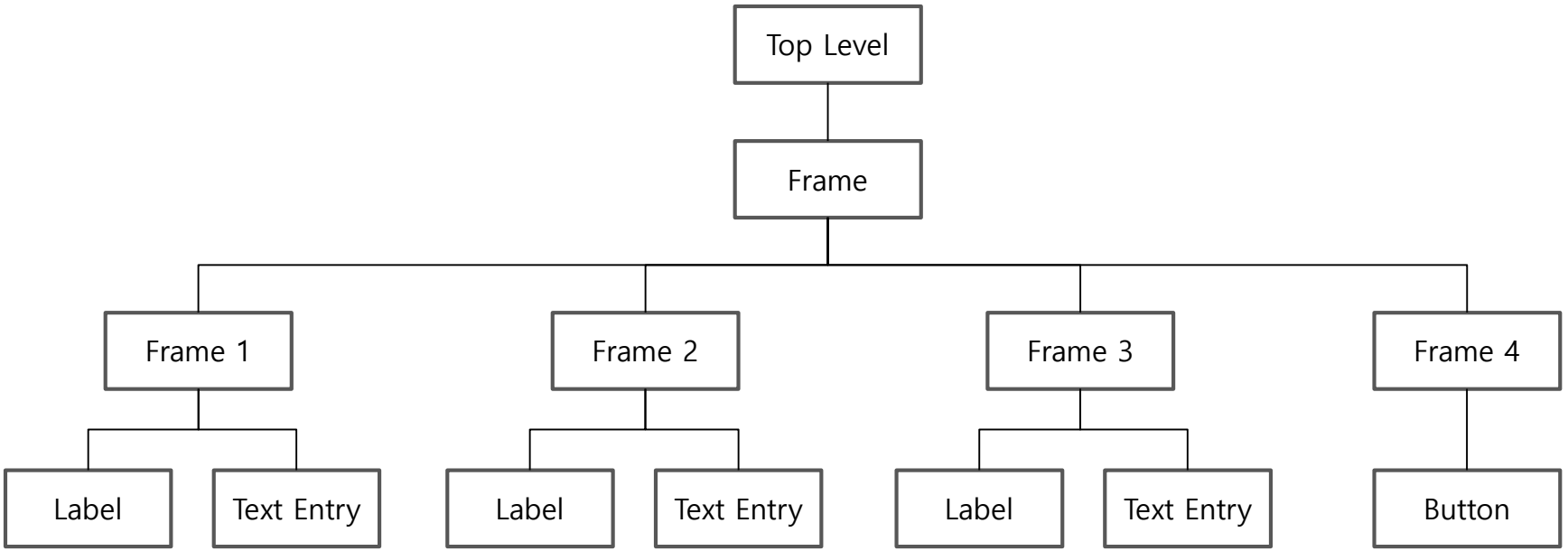


Frame



Frame

- **Frame : 위젯들을 그룹화하는 컨테이너 역할 → my_frame class 생성**
  - frame1 : 성명 label, entry(textbox)
  - frame2 : 회사명 label, entry(textbox)
  - frame3 : 특징 label, entry(textbox)
  - frame4 : 저장 button



입력화면 예시 (1)

```
1   # import tkinter module
2   from tkinter import *
3
4   # frame class
5   class my_frame(Frame):
6       def __init__(self, master):
7           Frame.__init__(self, master)
8
9           self.master = master
10          self.master.title("고객 입력")
11          self.pack(fill=BOTH, expand=True)
12
13          # 성명
14          frame1 = Frame(self)
15          frame1.pack(fill=X)
16
17          la_name = Label(frame1, text="성명", width=10, bg="green")
18          la_name.pack(side=LEFT, padx=10, pady=10)
19
20          txt_name = Entry(frame1)
21          txt_name.pack(fill=X, padx=10, expand=True)
22
```

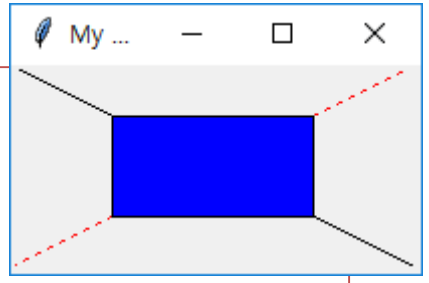
입력화면 예시 (2)

```
23          # 회사
24          frame2 = Frame(self)
25          frame2.pack(fill=X)
26
27          la_comp = Label(frame2, text="회사명", width=10, bg="blue")
28          la_comp.pack(side=LEFT, padx=10, pady=10)
29
30          txt_comp = Entry(frame2)
31          txt_comp.pack(fill=X, padx=10, expand=True)
32
33          # 특징
34          frame3 = Frame(self)
35          frame3.pack(fill=BOTH, expand=True)
36
37          la_cmt = Label(frame3, text="특징", width=10, bg="red")
38          la_cmt.pack(side=LEFT, anchor=N, padx=10, pady=10)
39
40          txt_cmt = Text(frame3)
41          txt_cmt.pack(fill=X, pady=10, padx=10)
42
43          # 저장
44          frame4 = Frame(self)
45          frame4.pack(fill=X)
46          btn_save = Button(frame4, text="저장", width=10, height=2)
47          btn_save.pack(side=RIGHT, padx=10, pady=10)
48
49      # Tk instance
50      win = Tk()
51      win.geometry("600x550+100+100")      # size:600x550, x_pos:100, y_pos:100
52      app = my_frame(win)
53      win.mainloop()
```

Canvas

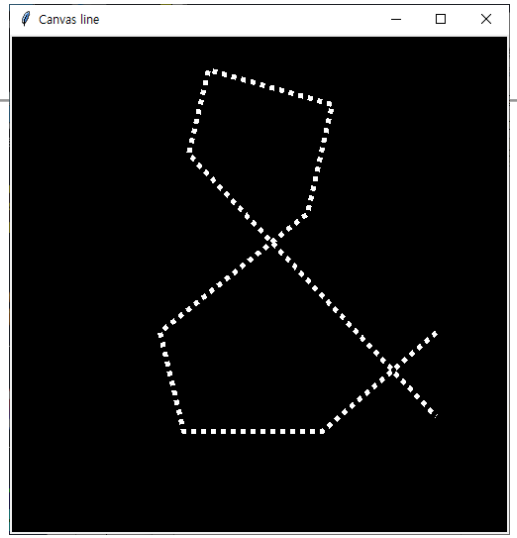
Canvas

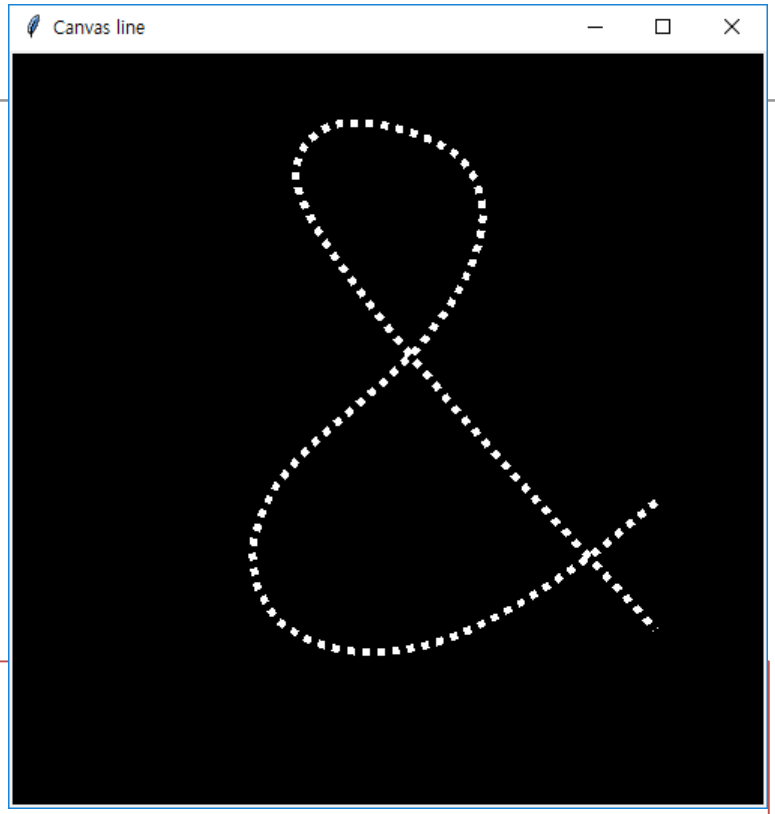
- **Canvas : 그래프를 그리거나, 다른 것들을 그릴 수 있는 위젯**




```
1   from tkinter import *
2
3   # main window
4   win = Tk()
5   win.title("My Canvas")
6
7   # canvas
8   drw = Canvas(win, width=200, height=100)
9   drw.pack()
10
11  # canvas line
12  drw.create_line(0, 0, 200, 100)
13  drw.create_line(0, 100, 200, 0, fill="red", dash=(4))
14
15  # canvas rectangle
16  drw.create_rectangle(50, 25, 150, 75, fill="blue")
17
18  win.mainloop()
```

Canvas

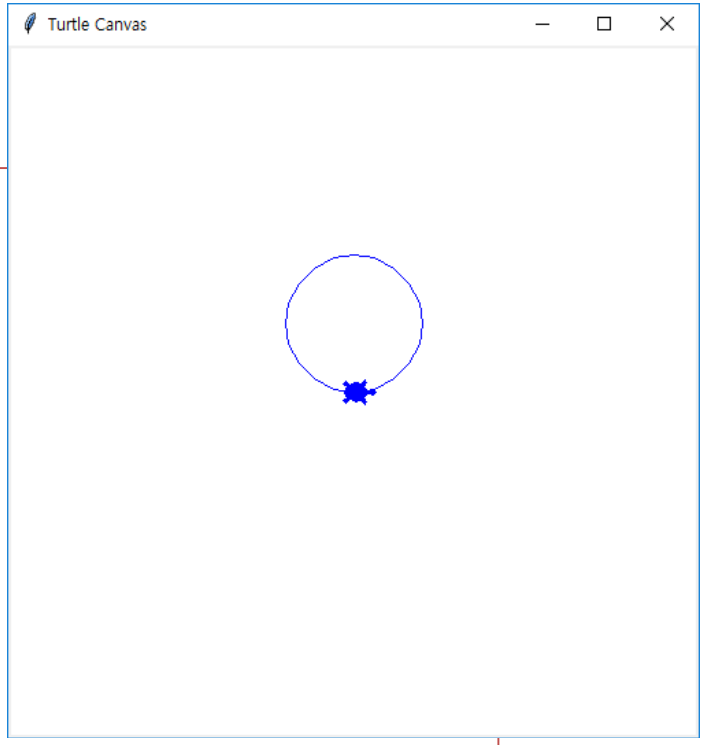





```
1   from tkinter import *
2
3   # main window
4   win = Tk()
5   win.title("Canvas line")
6
7   # canvas
8   drw = Canvas(win, width=500, height=500, bg="black")
9   drw.pack()
10
11  # canvas line
12  Q = [430,300, 315,400, 175,400, 151,300, 300,180, 325,70, 200,35, 180,120, 430,385]
13  drw.create_line(Q, smooth="true", dash=(4), fill="white", width=5)
14
15  win.mainloop()
```

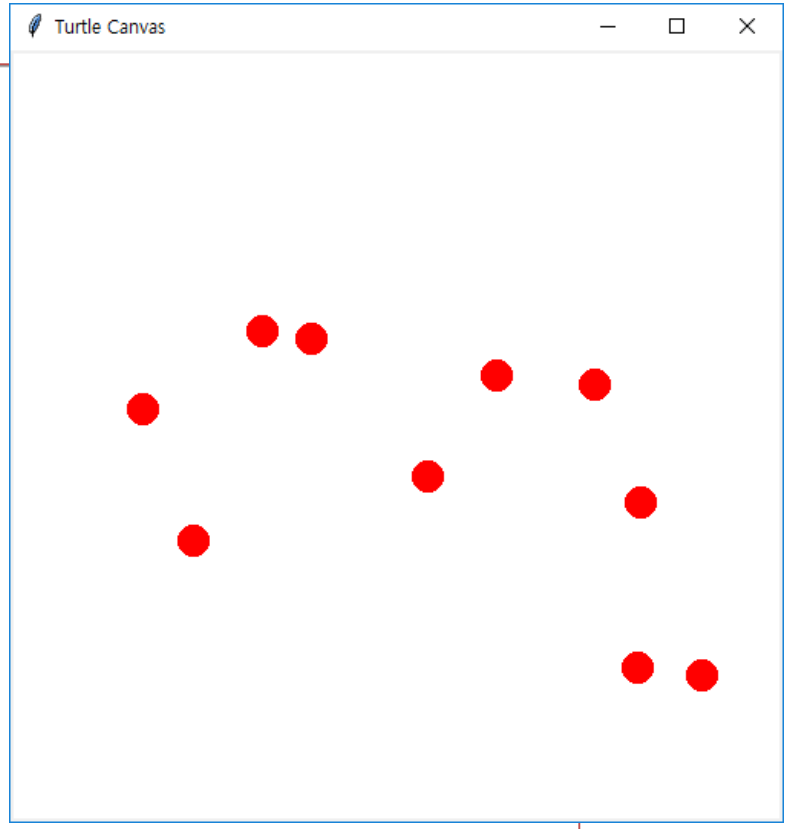
Canvas - turtle

```
1   from tkinter import *
2   import turtle
3
4   # main window
5   win = Tk()
6   win.title("Turtle Canvas")
7
8   # canvas
9   drw = Canvas(win, width=500, height=500)
10  drw.pack()
11
12  # turtle
13  t = turtle.RawTurtle(drw)
14  t.shape("turtle")
15  t.color("blue")
16  t.circle(50)
17
18  win.mainloop()
```



Canvas - circle

```python
from tkinter import *
import turtle
import random

# main window
win = Tk()
win.title("Turtle Canvas")

# canvas
drw = Canvas(win, width=500, height=500)
drw.pack()

# circle list
a = []
for i in range(10):
    circle = turtle.RawTurtle(drw)
    circle.color("red")
    circle.shape("circle")
    circle.penup()
    circle.speed(0)
    circle.goto(x=random.randint(-200,200), y=random.randint(-200,200))

    a.append(circle)

win.mainloop()
```

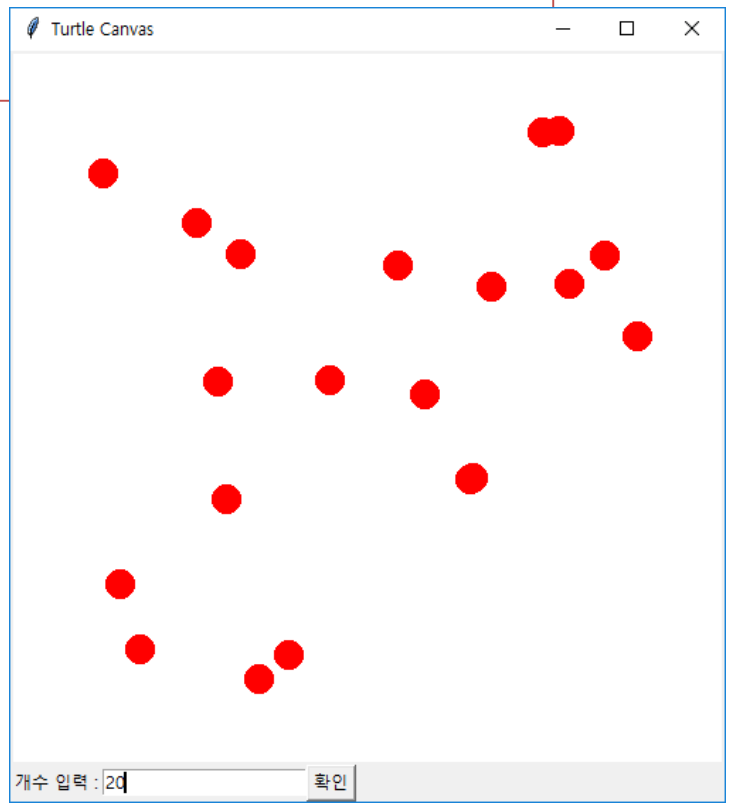


Canvas – input circle

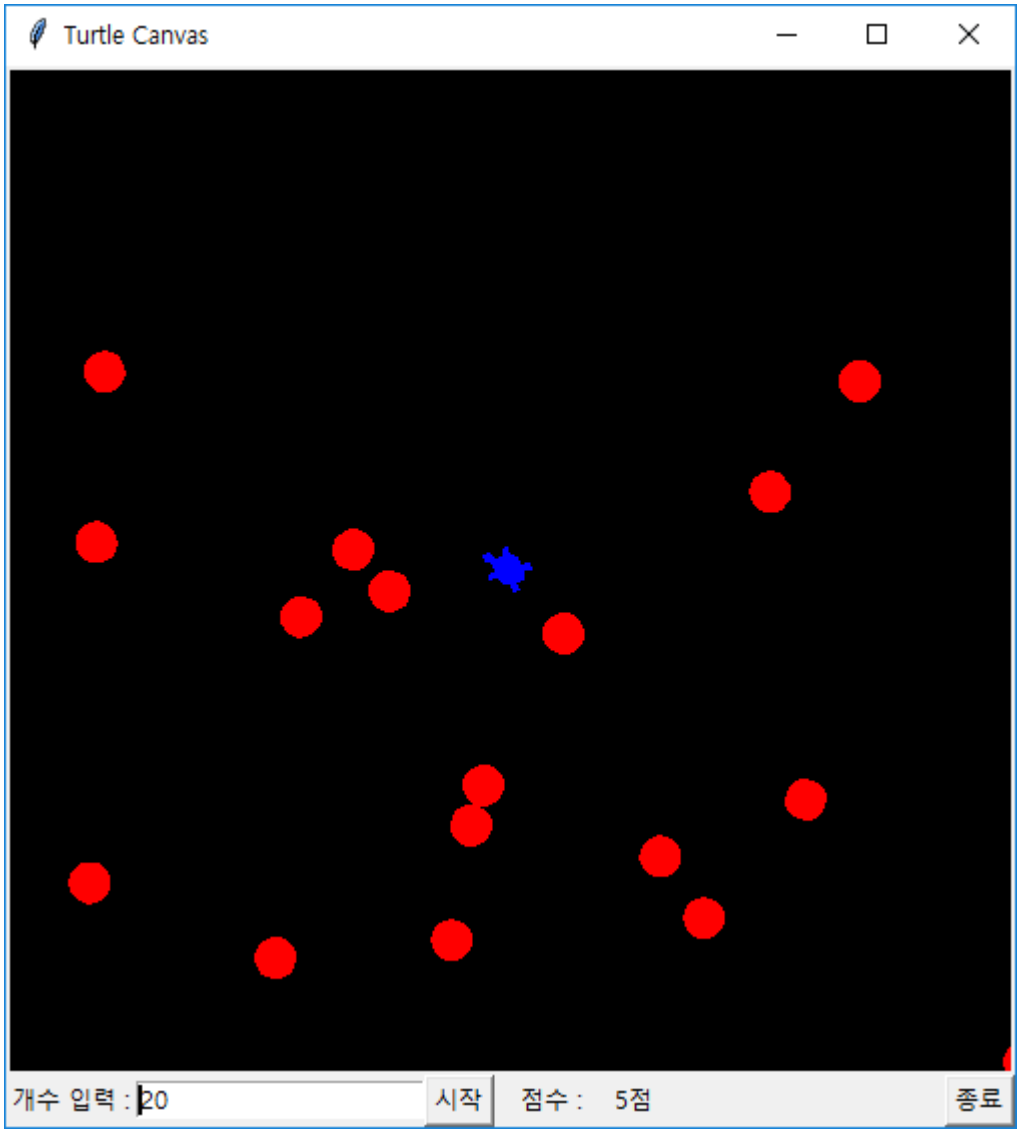
```
from tkinter import *
import turtle
import random

def create_circle():
    num = int(in_num.get())

    # circle list
    for i in range(num):
        circle = turtle.RawTurtle(drw)
        circle.color("red")
        circle.shape("circle")
        circle.penup()
        circle.goto(x=random.randint(-200, 200), y=random.randint(-200, 200))

        a.append(circle)

# main window
win = Tk()
win.title("Turtle Canvas")

a = []

# canvas
drw = Canvas(win, width=500, height=500)
drw.pack()

```

Canvas – input circle

```
28  # label
29  la_txt = Label(win, text="개수 입력 :")
30  la_txt.pack(side=LEFT)
31
32  # text box
33  in_num = Entry(win)
34  in_num.pack(side=LEFT)
35
36  # button
37  btn_ok = Button(win, text="확인", command=create_circle)
38  btn_ok.pack(side=LEFT)
39
40  win.mainloop()
```



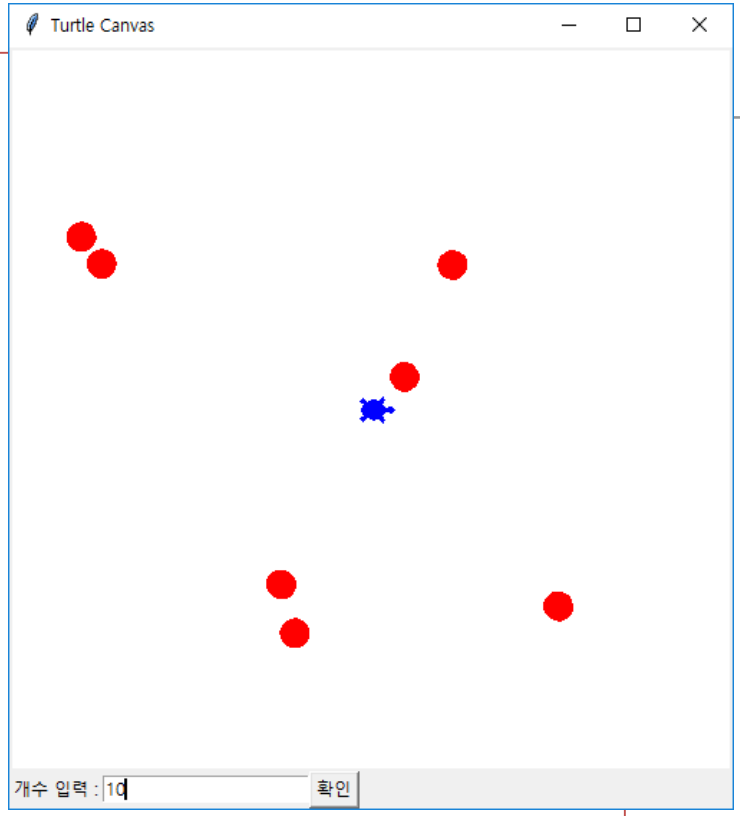
Turtle Game



Turtle Game – circle moving

```
1   from tkinter import *
2   import turtle
3   import random
4
5   def play():
6       for mv_a in a:
7           mv_a.right(random.randint(0,100))
8           mv_a.forward(40)
9
10      scr.ontimer(play, 500)
11
12  def create_circle():
13      num = int(in_num.get())
14
15      # circle list
16      for i in range(num):
17          circle = turtle.RawTurtle(drw)
18          circle.color("red")
19          circle.shape("circle")
20          circle.penup()
21          circle.speed(0)
22          circle.goto(x=random.randint(-200, 200), y=random.randint(-200, 200))
23
24          a.append(circle)
25
26
```

```
27  # main window
28  win = Tk()
29  win.title("Turtle Canvas")
30
31  a = []
32
33  # canvas
34  drw = Canvas(win, width=500, height=500)
35  drw.pack()
36
37  # label
38  la_txt = Label(text="개수 입력 :")
39  la_txt.pack(side=LEFT)
40
41  # text box
42  in_num = Entry()
43  in_num.pack(side=LEFT)
44
45  # button
46  btn_ok = Button(text="확인", command=create_circle)
47  btn_ok.pack(side=LEFT)
48
49  # turtle
50  t = turtle.RawTurtle(drw)
51  t.shape("turtle")
52  t.color("blue")
53  t.speed(0)
54  t.penup()
55
56  scr = t.getscreen()
57  play()
58
59  win.mainloop()
```



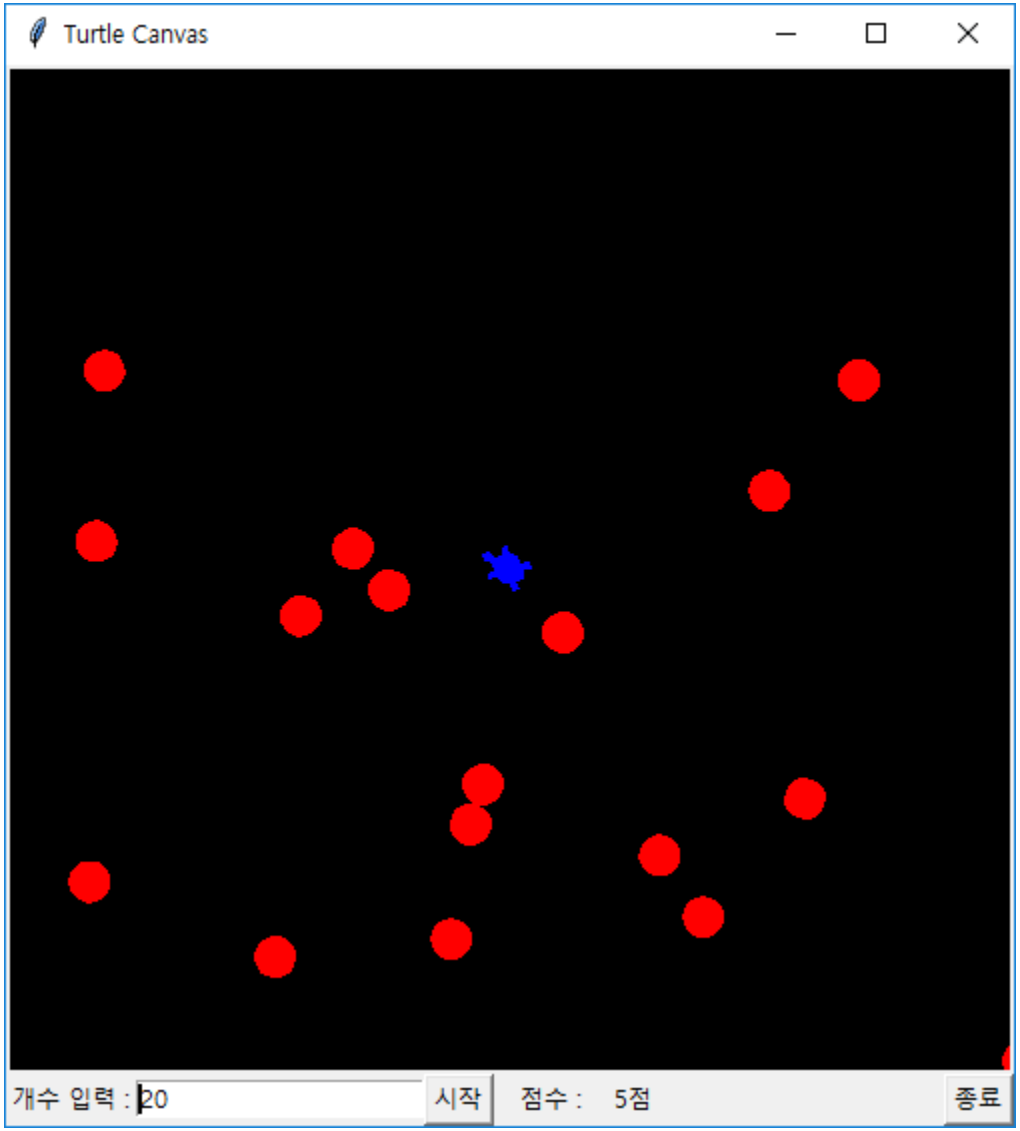
Turtle Game – key play

```
from tkinter import *
import turtle
import random

def turtle_left(event):
    t.left(30)

def turtle_right(event):
    t.right(30)

def turtle_go(event):
    t.forward(30)

def play():
    global cnt

    for mv_a in a:
        mv_a.right(random.randint(0,100))
        mv_a.forward(40)

        if t.distance(mv_a) < 30:
            cnt += 1
            str_cnt = str(cnt) + "점"
            la_result.configure(text=str_cnt)

    scr.ontimer(play, 300)
```

```
29      def create_circle():
30          num = int(in_num.get())
31
32          # circle list
33          for i in range(num):
34              circle = turtle.RawTurtle(drw)
35              circle.color("red")
36              circle.shape("circle")
37              circle.penup()
38              circle.speed(0)
39              circle.goto(x=random.randint(-200, 200), y=random.randint(-200, 200))
40
41              a.append(circle)
42
43
44      # main window
45      win = Tk()
46      win.title("Turtle Canvas")
47
48      a = []
49      cnt = 0
50
51      # canvas
52      drw = Canvas(win, width=500, height=500)
53      drw.pack()
54
55      # label
56      la_txt1 = Label(win, text="개수 입력 :")
57      la_txt1.pack(side=LEFT)
58
59      # text box
60      in_num = Entry(win)
61      in_num.pack(side=LEFT)
62
```

```
63  # button
64  btn_ok = Button(win, text="시작", command=create_circle)
65  btn_ok.pack(side=LEFT)
66
67  # label
68  la_txt2 = Label(win, text="점수 :")
69  la_txt2.pack(side=LEFT, padx=10)
70
71  # result
72  la_result = Label(win)
73  la_result.pack(side=LEFT)
74
75  # button
76  btn_exit = Button(win, text="종료", command=quit)
77  btn_exit.pack(side=RIGHT)
78
79  # turtle
80  t = turtle.RawTurtle(drw)
81  t.shape("turtle")
82  t.color("blue")
83  t.speed(0)
84  t.penup()
85
86  scr = t.getscreen()
87  scr.bgcolor("black")
88  play()
89
90  win.bind("<Left>", turtle_left)
91  win.bind("<Right>", turtle_right)
92  win.bind("<Up>", turtle_go)
93
94  win.mainloop()
```

Turtle Game – key play



Any Questions...
Just Ask!