모두를 위한 파이썬 프로그래밍

11주 Graphic User Interface 다루기

tkinter

GUI (Graphical User Interface)

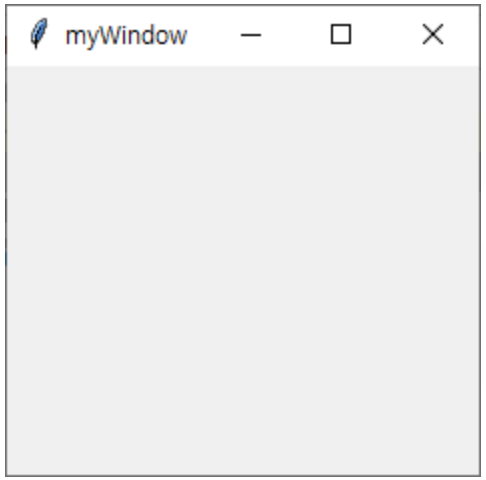
- 사용자가 컴퓨터와 정보를 교환할 때, 그래픽을 통해 작업할 수 있는 환경

tkinter:

- 파이썬의 스탠다드 GUI 패키지

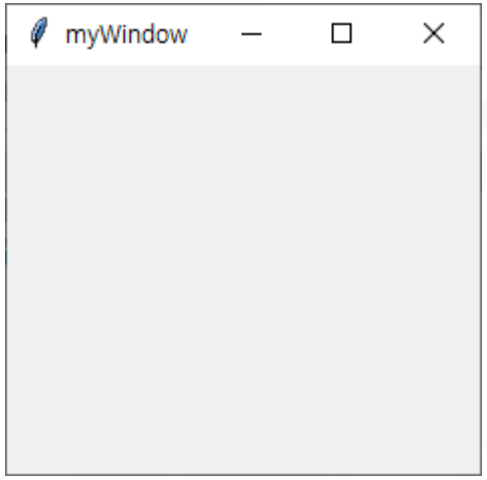
import와 기본 창 생성

```
from tkinter import *

window = None    # 빈 윈도우 객체 생성
window = Tk()    # Tk를 이용해 윈도우 생성
window.title('myWindow')    # 윈도우 타이틀에 이름지정
window.mainloop()    # 윈도우창을 계속 띄움
```



import와 기본 프로그램 구조

```
from tkinter import *

window = None

# 함수처리
def setupGUI():
    global window    # 전역 변수로 사용
    window = Tk()    # 윈도우 객체 생성
    window.title( 'myWindow' )


setupGUI()  # 윈도우 생성 함수 실행
window.mainloop()    # 윈도우 실행
```



tkinter 용어

- **윈도우** (Window): 스크린의 네모 모양의 창. 여러 개의 창을 생성가능
- **위젯** (Widget): 다양한 GUI 구성요소들을 부르는 용어
  - **레이블** (Label)
  - **버튼** (Button)
  - **메뉴**(Menu)
  - **엔트리** (Entry) : 값이 입력되는 위젯
- **레이아웃** (Layout): 윈도우나 프레임 안에 위젯의 위치
- **프레임** (Frame): 윈도우의 한 부분. GUI 요소들을 프레임 안에 모아서 위치할 수 있음
- **콜백** (Callback): GUI의 위젯과 연결되어 있는 특수한 함수. 사용자가 위젯과 상호작용할 때, 이 함수가 호출되어 사용자의 행동에 반응한다.

Label Widget

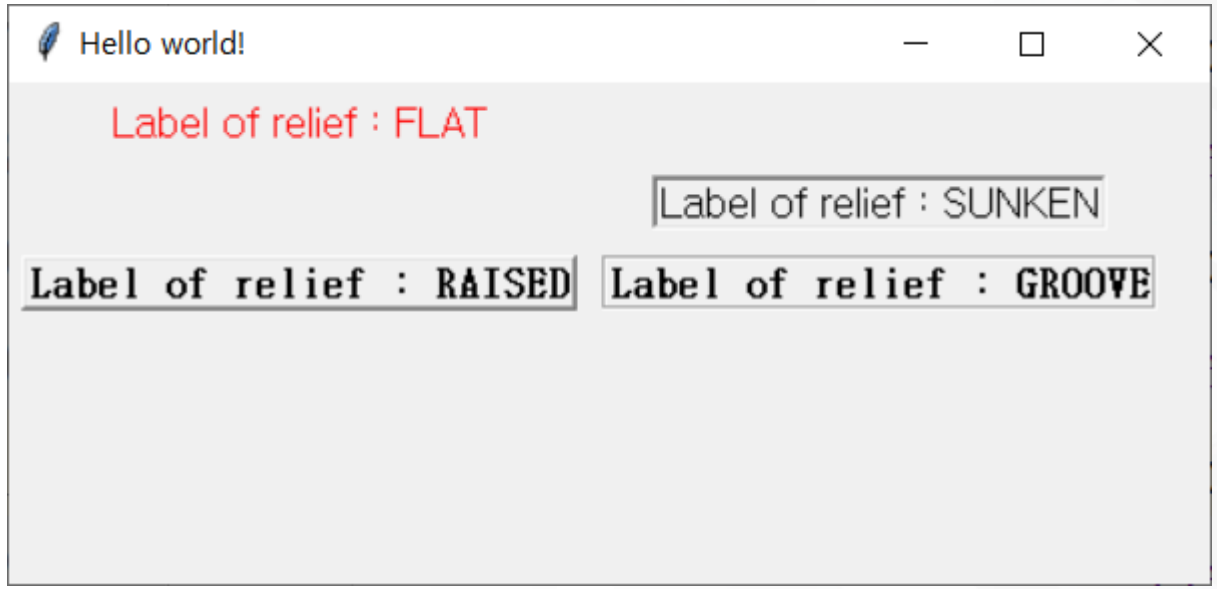
1

레이블 (Label) 위젯의 속성

사용자에게 **텍스트**나 **사진**을 보여준다.

```
label = Label(윈도우객체, bg='red', font='Arial', fg='black', text='label Text, …) # 라벨생성
label.pack() # 메인창 배치하기, grid()함수로 격자배치 가능
```

속성(att.) 이름
background(bg) :배경색
borderwidth (bd) :테두리 두께
font : 글꼴명
foreground (fg) : 글자색
height / width (레이블크기)
image
justify (정렬)
padx (레이블 좌우 간격)
pady (레이블 위아래 간격)
relief (레이블 엣지각 모양)
text (내용)



레이블 (Label) 위젯의 속성

사용자에게 **텍스트**나 **사진**을 보여준다.

```
label = Label( parentWindow, att1 = att1val, att2 = att2val, …)
```

속성(att.) 이름	의미	예
background(bg)	배경색을 설정. 파이썬 지정색 'green' 또는 '#AAFF33'	`bg = 'green'`
borderwidth (bd)	레이블 테두리의 두께를 설정	`bd = 5`
font	폰트와 글자 크기, 굵기나 기울기를 설정	`font = "Futura 15 bold"`
foreground (fg)	글자의 색상을 설정.	`fg = "white"`
height / width	레이블의 높이/너비를 설정. (텍스트일 경우 텍스트 유닛, 이미지일 경우 픽셀)	`height = 2` `width = 2`
image	레이블이 이미지(PhotoImage 타입)을 보이도록 설정	`image = Pimg`
justify	여러 줄의 텍스트 정렬 (LEFT, RIGHT, CENTER)	`justify = CENTER`
padx	레이블의 가로(x축) 패딩(padding) 설정	`padx = 5`
pady	레이블의 세로(y축) 패딩 설정	`pady = 5`
relief	테두리 장식을 설정 (FLAT, SUNKEN, RAISED, GROOVE, RIDGE)	`Relief = RAISED`
text	레이블이 표시할 텍스트 설정	`text = "Hello World!"`

레이블(Label) 위젯 예제

```
from tkinter import *
window = None

def setupGUI():
    global window
    window = Tk()
    window.title('myWindow')
    title_label = Label(window, text='Korea University',
                        font='Broadway 25 bold',
                        relief=RAISED)
    title_label.grid(row=0, column=0, padx=5, pady=5)


setupGUI()
window.mainloop()
```

Button Widget

2

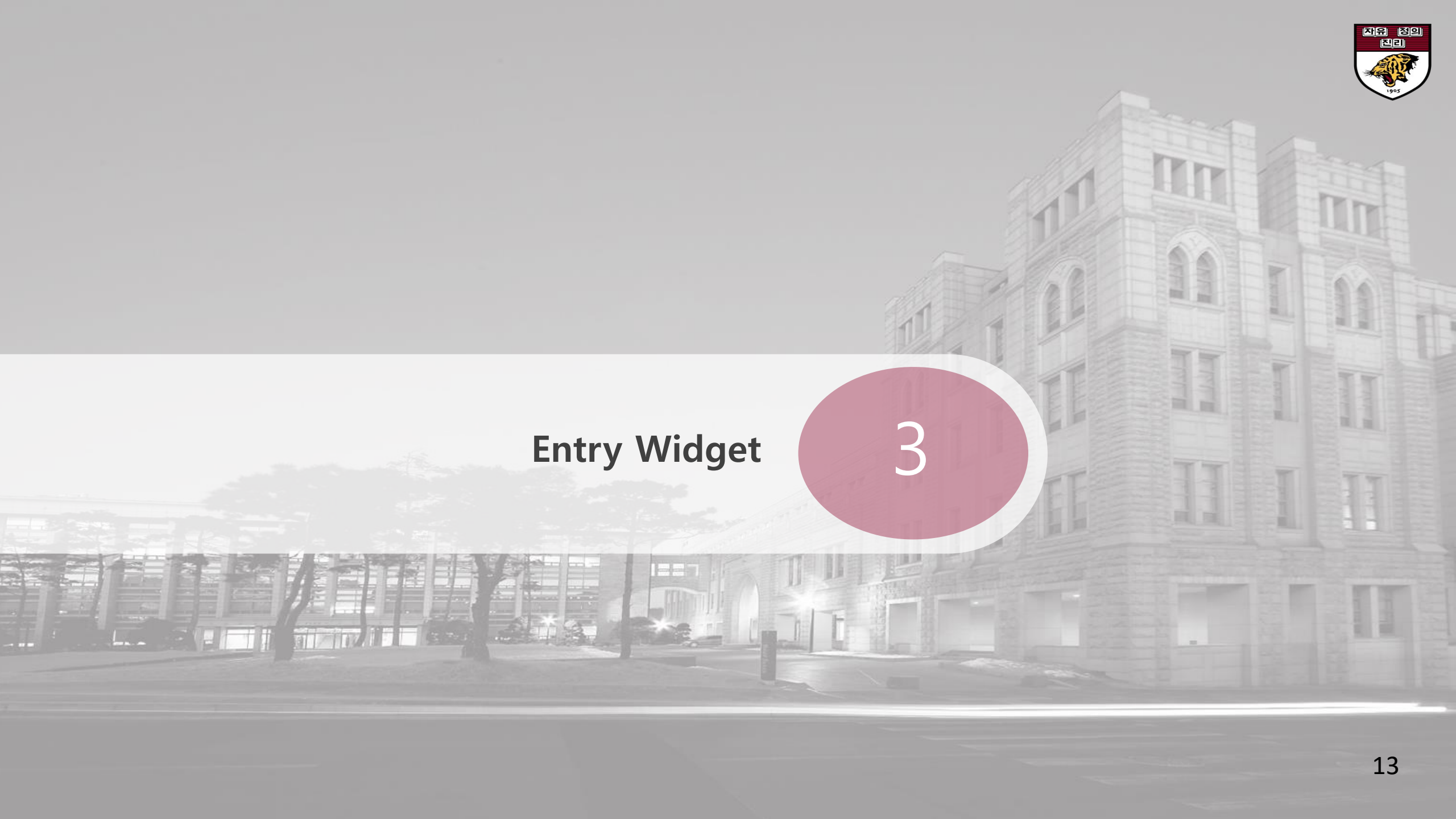
버튼 위젯(Button)의 속성

```
button = Button(<parentWindow>, att1 = att1val, att2 = att2val, …)
```

속성 이름	뜻/타입	예
background(bg)	배경색을 설정. 파이썬 지정색 'green' 또는 '#AAFF33'	`bg = 'green'`
borderwidth (bd)	레이블 테두리의 두께를 지정	`bd = 5`
command	콜백(callback) 함수를 설정	`command = myFunc`
font	폰트와 글자 크기, 굵기나 기울기를 설정	`font = "Futura 15 bold"`
foreground (fg)	글자의 색상을 설정	`fg = "white"`
height / width	레이블의 높이/너비를 설정 (텍스트일 경우 텍스트 유닛, 이미지일 경우 픽셀)	`height = 2` `width = 2`
image	레이블이 PhotoImage 형의이미지를 보이도록 설정	`image = Pimg`
justify	텍스트 정렬 (LEFT, RIGHT, CENTER)	`justify = CENTER`
padx	레이블의 가로(x축) 패딩(padding) 설정	`padx = 5`
pady	레이블의 세로(y축) 패딩 설정	`pady = 5`
relief	버튼모양 설정 (FLAT, SUNKEN, RAISED, GROOVE, RIDGE)	`Relief = RAISED`
text	레이블이 표시할 텍스트 설정	`text = "Hello World!"`

버튼(Button) 위젯과 콜백(callback) 함수 예제

```
from tkinter import *

window = None

def app_quit():
    window.destroy()

def setupGUI():
    global window
    window = Tk()
    window.title('myWindow')
    title_label = Label(window, text='Korea University',
                        font='Broadway 25 bold',
                        relief=RAISED)
    title_label.grid(row=0, column=0, padx=5, pady=5)

    quit_button = Button(window, text='Quit',
                         font='Broadway 25 bold',
                         command=app_quit)
    quit_button.grid(row=0, column=1, padx=3, pady=3)
    # title_label['text']='한국대학교'
    # quit_button['text']='종료'

setupGUI()
window.mainloop()
```



Entry Widget

3

엔트리(Entry) 위젯 속성

```
entry = Entry(<parentWindow>, att1 = att1val, att2 = att2val, …)
```

속성 이름	의미/타입	예
background(bg)	배경색을 설정. 파이썬 지정색 'green' 또는 '#AAFF33'	`bg = 'green'`
borderwidth (bd)	레이블 테두리의 두께를 설정	`bd = 5`
font	폰트와 글자 크기, 굵기나 기울기를 설정	`font = "Futura 15 bold"`
foreground (fg)	글자의 색상을 설정.	`fg = "white"`
width	레이블의 높이/너비를 설정. (텍스트일 경우 텍스트 유닛, 이미지일 경우 픽셀)	`height = 2` `width = 2`
justify	텍스트 정렬 (LEFT, RIGHT, CENTER)	`justify = CENTER`
relief	테두리 장식을 설정 (FLAT, SUNKEN, RAISED, GROOVE, RIDGE)	`Relief = RAISED`

엔트리(Entry) 위젯 메소드(method)

메소드 이름	뜻	예
get	엔트리 위젯의 현재 텍스트를 가져오기	`myEntry.get()`
delete	현재 텍스트의 문자를 주어진 인덱스에 따라서 삭제. 텍스트의 끝 인덱스를 의미하는 END 상수가 있다. **`myEntry.delete(2, 5)`** # 인덱스 2에서 인덱스4까지 문자 3개를 제거	`myEntry.delete(2, 5)` `myEntry.delte(0, END)`
font	엔트리 위젯의 폰트와 글자 크기, 굵기나 기울기를 설정	`font = "Broadway 15 bold"`
insert	문자열을 엔트리의 지정된 인덱스에 넣음 myEntry.insert(<index>, <string>)	`myEntry.insert(0, "Entry")`

엔트리(Entry) 위젯과 키 바인딩(key binding) 예제

```
from tkinter import *
window = None
entry_widget = None
title_label = None

def app_quit():
    window.destroy()

def setupGUI():
    global window
    global entry_widget
    global title_label
    window = Tk()
    window.title('myWindow')
    title_label = Label(window, text='Korea University',
                        font='Broadway 25 bold', relief=RAISED)
    title_label.grid(row=0, column=0, padx=5, pady=5)

    quit_button = Button(window, text='Quit', font='Broadway 25 bold',
                         command=app_quit)
    quit_button.grid(row=0, column=1, padx=3, pady=3)
    # title_label['text']='한국대학교'
    entry_widget = Entry(window, font='Broadway 25 bold', width=20)
    entry_widget.grid(row=1, column=0, padx=3, pady=3)
    entry_widget.bind('<Return>', entry_response)
setupGUI()
window.mainloop()
```

추가된 코드:

```
def entry_response(event):
    global entry_widget
    global title_label
    text = entry_widget.get()
    entry_widget.delete(0, END)
    title_label['text'] = text
```

마우스 키 바인딩(Mouse key binding) 이벤트 처리

```
entry_widget.bind( '<Button-3>', 실행할 함수명)
```

- 마우스 이벤트 기본 처리
  - 키보드 및 마우스를 누르는 것을 이벤트(Event)라 함. mainloop( ) 함수는 이러한 이벤트가 발생하기를 기다리는 함수임

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〉

마우스 작동		이벤트 코드
마우스 커서가 위젯 위로 올라왔을 때		〈Enter〉
마우스 커서가 위젯에서 떠났을 때		〈Leave〉
드래그할 때	왼쪽 버튼	〈B1-Motion〉
	가운데 버튼	〈B2-Motion〉
	오른쪽 버튼	〈B3-Motion〉

키보드 바인딩(key binding) 이벤트 처리

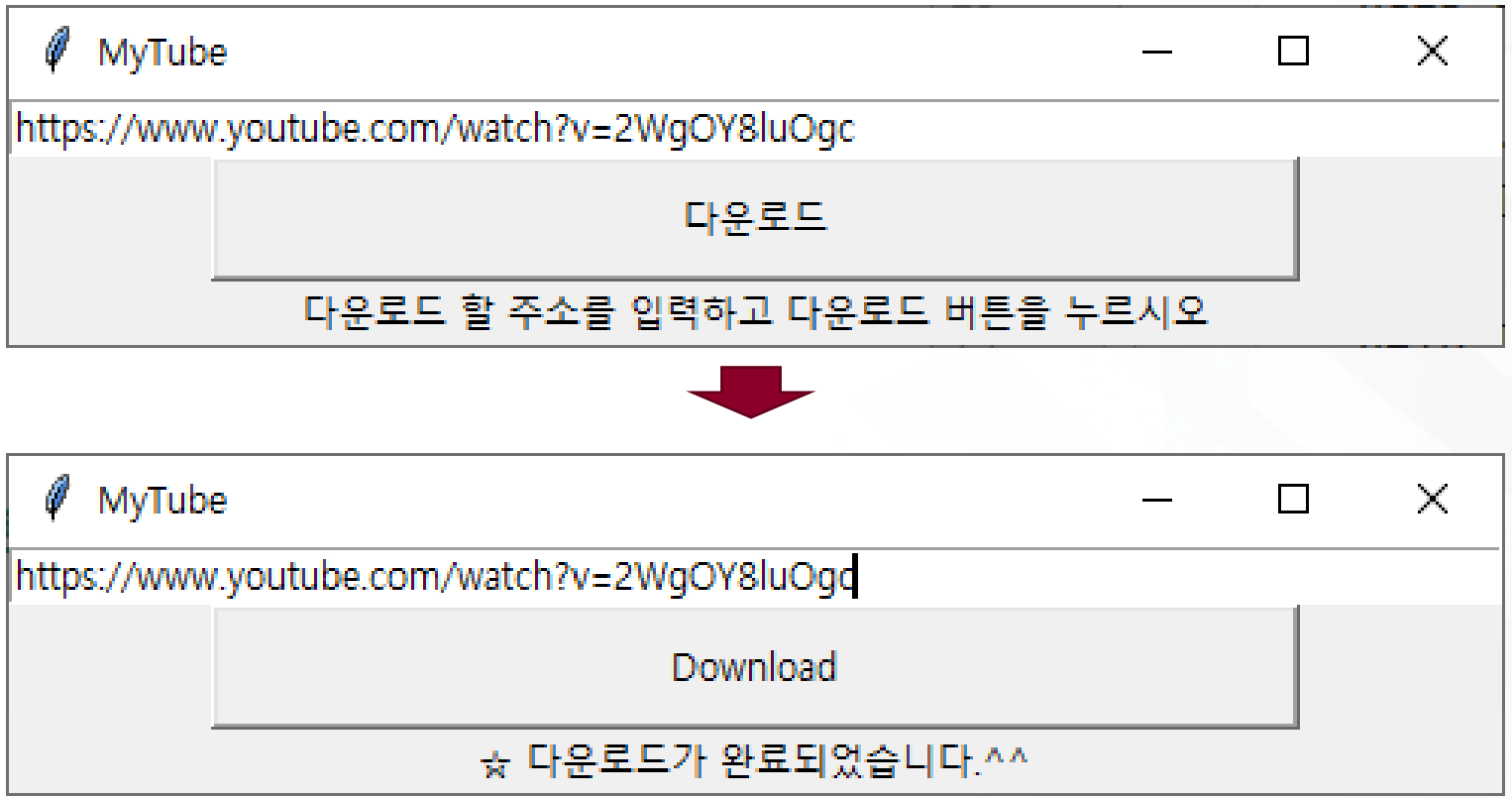
```
entry_widget.bind('<Return>', 실행할 함수명)
```

- 키보드 이벤트 기본 처리
  - 위젯에서 키보드가 눌리면 발생하는 이벤트이며 대표적인 것은 <Key> 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

응용 예제

- MyTube 동영상 다운로드 프로그램을 GUI로 구성하고 실행파일로 생성해보자
- 다음과 같은 안내 문구를 적용한다.



4

팝업창과 다이얼로그 박스(대화 상자)

팝업창과 대화 박스 예제 ①

새로운 파이썬 스크립트에서 작성하세요.

```
from tkinter import *
from tkinter import messagebox

window = None


def setupGUI():
    global window
    window = Tk()
    window.title('Dialog Window')

    title_label = Label(window, text='Message Box Samples',
                        font='Arial 14 bold', width='30')
    title_label.grid(row=0, column=1, padx=10, pady=10)


setupGUI()
window.mainloop()
```

messagebox는 tkinter에서 팝업창을 사용할 수 있게 해주는 모듈입니다.

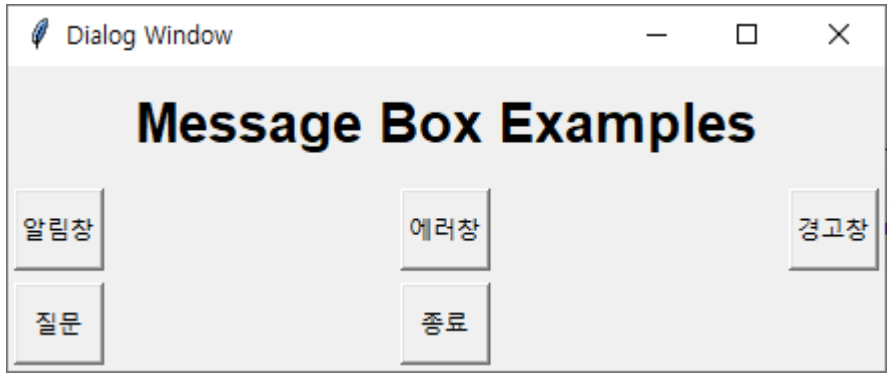
팝업창과 대화 박스 예제 ②

추가된 코드:

```
def infoBox():
    messagebox.showinfo("알림", "알림창 내용")

def errorBox():
    messagebox.showerror("에러", "에러창 내용")

def warningBox():
    messagebox.showwarning("경고", "경고창 내용")
```


```
# ============== 알림창 팝업 버튼 ==================
infoButton = Button(rootWin, text = "알림창", relief = RAISED, width = 5, height = 2, command = infoBox)
infoButton.grid(row = 1, column = 0, padx = 3, pady = 3)

errorButton = Button(rootWin, text = "에러창", relief = RAISED, width = 5, height = 2, command = errorBox)
errorButton.grid(row = 1, column = 1, padx = 3, pady = 3)

warningButton = Button(rootWin, text = "경고창", relief = RAISED, width = 5, height = 2, command =
warningBox)
warningButton.grid(row = 1, column = 2, padx = 3, pady = 3)
```

팝업창과 대화 박스 예제 ③



추가된 함수:

```
def quit():
    ans = messagebox.askokcancel('종료', '프로그램을 종료하시겠습니까?')
    if ans: # ans is True
        window.destroy()


def yes_no_dialog():
    ans = messagebox.askyesno('질문', '당신은 고대인 입니까?')
    if ans: # ans is True
        messagebox.showinfo('--;', '저는 현대인 입니다.')
    else:
        messagebox.showinfo('^^;', '저는 고대인 입니다.')
```

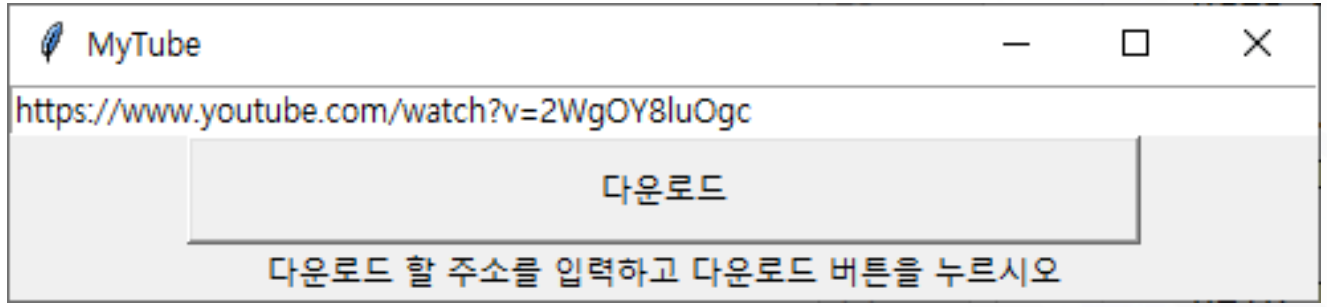
팝업창과 대화 박스 예제 ④

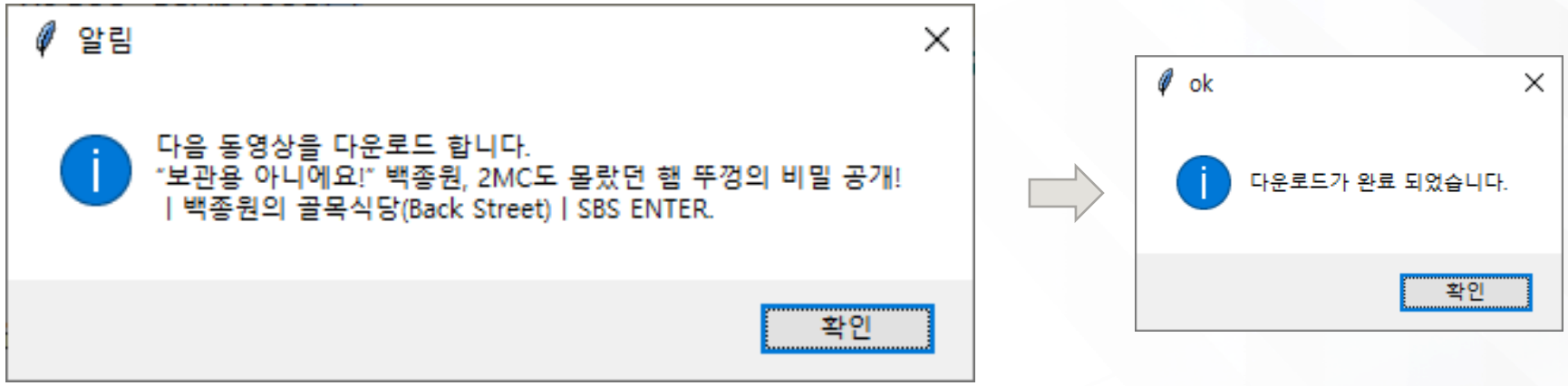
SsetupGUI함수에 추가된 코드:

```
quit_button = Button(window, text='종료', width=5, height=2, command=quit)
quit_button.grid(row=2, column=1, padx=3, pady=3)

yes_no_button = Button(window, text='질문', width=5, height=2, command=yes_no_dialog)
yes_no_button.grid(row=2, column=0, padx=3, pady=3)
```

연습문제

- MyTube 동영상 다운로드 프로그램을 GUI로 구성하고 실행파일로 생성해 봅시다
- 다운로드 버튼을 클릭하면 다운로드할 동영상 제목을 알려주고 다운이 완료되면 다음과 같은 대화상자를 표시해 봅시다





전자 계산기 만들기

5

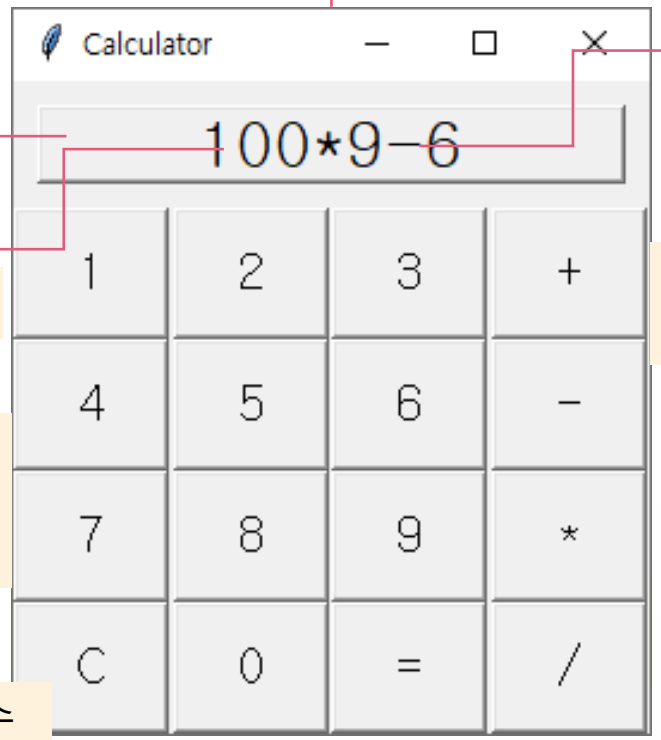
계산기 구성하기

```
display_label = tk.Label(window, textvariable=equation)
```

window

display_label

값이 변화는 textvariable
equation = tk.StringVar()



수식(eval)
expression

숫자버튼 클릭시 실행할 함수
def press(num)

```
# 입력한 문자 연결하기
expression += str(num)
equation.set(expression)
```

버튼 btn0 ~ btn9
높이5, 너비2
command=lambda: press(1)

Clear 버튼 클릭시 수행한 함수

```
equation.set('0')
```

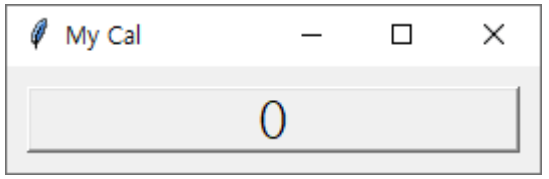
=버튼 클릭시 실행할 함수
def result()

```
equation.set(eval(expression))
expression = ''
```

계산기 – 출력창 만들기

새로운 파이썬 스크립트에서 작성하세요.

```
import tkinter as tk

window = None
display_label = None
equation = None

def main_GUI():
    setup_GUI()
    window.mainloop()

def setup_GUI():
    global window
    global display_label
    global equation

    window = tk.Tk()
    window.title('My Cal')

    equation = tk.StringVar()
    equation.set('0')

    display_label = tk.Label(window, textvariable=equation, relief=tk.RAISED, width=15, font='Futura 20')
    display_label.grid(row=0, column=0, columnspan=4, padx=10, pady=10)

main_GUI()
```

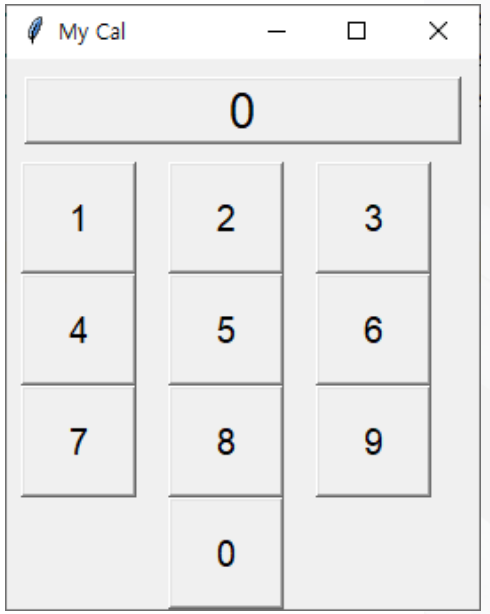


프로그램의 기본 뼈대를 만들어 줍니다.

계산기 예제 – 숫자버튼 생성

setup_GUI() 함수 안에 숫자 버튼들을 만들어줍니다.

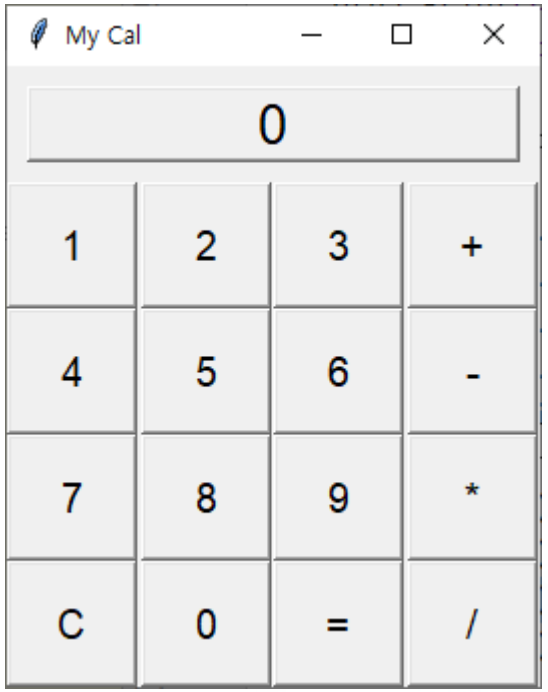
```
btn1 = tk.Button(window, text='1', width=5, height=2, font='Arial 15')
btn2 = tk.Button(window, text='2', width=5, height=2, font='Arial 15')
btn3 = tk.Button(window, text='3', width=5, height=2, font='Arial 15')
btn4 = tk.Button(window, text='4', width=5, height=2, font='Arial 15')
btn5 = tk.Button(window, text='5', width=5, height=2, font='Arial 15')
btn6 = tk.Button(window, text='6', width=5, height=2, font='Arial 15')
btn7 = tk.Button(window, text='7', width=5, height=2, font='Arial 15')
btn8 = tk.Button(window, text='8', width=5, height=2, font='Arial 15')
btn9 = tk.Button(window, text='9', width=5, height=2, font='Arial 15')
btn0 = tk.Button(window, text='0', width=5, height=2, font='Arial 15')

btn1.grid(row=1, column=0)
btn2.grid(row=1, column=1)
btn3.grid(row=1, column=2)
btn4.grid(row=2, column=0)
btn5.grid(row=2, column=1)
btn6.grid(row=2, column=2)
btn7.grid(row=3, column=0)
btn8.grid(row=3, column=1)
btn9.grid(row=3, column=2)
btn0.grid(row=4, column=1)
```



계산기 예제 – 연산자 버튼 생성

setup_GUI() 함수 안에 연산자 버튼들을 만들어줍니다.

```
clearBtn = tk.Button(window, text='C', width=5, height=2, font='Arial 15')
resultBtn = tk.Button(window, text='=', width=5, height=2, font='Arial 15')
addBtn = tk.Button(window, text='+', width=5, height=2, font='Arial 15')
subBtn = tk.Button(window, text='-', width=5, height=2, font='Arial 15')
mulBtn = tk.Button(window, text='*', width=5, height=2, font='Arial 15')
divBtn = tk.Button(window, text='/', width=5, height=2, font='Arial 15')

clearBtn.grid(row=4, column=0)
resultBtn.grid(row=4, column=2)
addBtn.grid(row=1, column=3)
subBtn.grid(row=2, column=3)
mulBtn.grid(row=3, column=3)
divBtn.grid(row=4, column=3)
```



계산기 예제 – 숫자 버튼을 클릭할 때 실행할 함수 생성

import 선언문 아래 전역 변수인 expression 변수를 추가합니다.

```
expression = ''# 수식작성용 변수 생성

def press1():
    global expression
    expression = expression + '1'
    print(expression)
    equation.set(expression)
def press2():
    global expression
    expression = expression + '2'
    equation.set(expression)
def press3():
    global expression
    expression = expression + '3'
    equation.set(expression)
def press4():
    global expression
    expression = expression + '4'
    equation.set(expression)
def press5():
    global expression
    expression = expression + '5'
    equation.set(expression)
def press6():
    global expression
    expression = expression + '6'
    equation.set(expression)
def press7():
    global expression
    expression = expression + '7'
    equation.set(expression)
def press8():
    global expression
    expression = expression + '8'
    equation.set(expression)
def press9():
    global expression
    expression = expression + '9'
    equation.set(expression)
def press0():
    global expression
    expression = expression + '0'
    equation.set(expression)
```

계산기 예제 – 연산 버튼을 클릭할 때 실행할 함수 생성


```
def press_add():
    global expression
    expression = expression + '+'
    equation.set(expression)
def press_sub():
    global expression
    expression = expression + '-'
    equation.set(expression)
def press_mul():
    global expression
    expression = expression + '*'
    equation.set(expression)
def press_div():
    global expression
    expression = expression + '/'
    equation.set(expression)
def clearDisplay():
    global expression
    display_label['text'] = '0'
    equation.set('0')
```

0으로 초기화

```
def resultPress():
    try:
        global expression
        total = str(eval(expression))
        equation.set(total)
        expression = ''
    except:
        equation.set('  error  ')
        expression = ''
```

eval() 함수는 문자열을 파이썬 코드로 인식하여 실행한 값을 반환하는 함수입니다.

```
Python Console
>>> eval("1+2+3")
6
```

계산기 예제 – 버튼을 클릭할 때 실행할 함수 연결

버튼의 command 속성에 실행할 함수 연결

```
btn1 = tk.Button(window, text='1', width=5, height=2, font='Arial 15', command=press1)
btn2 = tk.Button(window, text='2', width=5, height=2, font='Arial 15', command=press2)
btn3 = tk.Button(window, text='3', width=5, height=2, font='Arial 15', command=press3)
btn4 = tk.Button(window, text='4', width=5, height=2, font='Arial 15', command=press4)
btn5 = tk.Button(window, text='5', width=5, height=2, font='Arial 15', command=press5)
btn6 = tk.Button(window, text='6', width=5, height=2, font='Arial 15', command=press6)
btn7 = tk.Button(window, text='7', width=5, height=2, font='Arial 15', command=press7)
btn8 = tk.Button(window, text='8', width=5, height=2, font='Arial 15', command=press8)
btn9 = tk.Button(window, text='9', width=5, height=2, font='Arial 15', command=press9)
btn0 = tk.Button(window, text='0', width=5, height=2, font='Arial 15', command=press0)

clearBtn = tk.Button(window, text='C', width=5, height=2, font='Arial 15' , command=clearDisplay)
resultBtn = tk.Button(window, text='=', width=5, height=2, font='Arial 15', command=resultPress)
addBtn = tk.Button(window, text='+', width=5, height=2, font='Arial 15'   , command=press_add)
subBtn = tk.Button(window, text='-', width=5, height=2, font='Arial 15'   , command=press_sub)
mulBtn = tk.Button(window, text='*', width=5, height=2, font='Arial 15'   , command=press_mul)
divBtn = tk.Button(window, text='/', width=5, height=2, font='Arial 15'   , command=press_div)
```

업그레이드하기 – 중복 함수의 최적화

clearDisplay(), resultPress()함수를 제외한 모든 숫자, 연산자 버튼을 하나의 람다 함수에 연결

```
def press1():
    global expression
    expression = expression + '1'
    print(expression)
    equation.set(expression)
def press2():
    global expression
    expression = expression + '2'
    equation.set(expression)
def press3():
    global expression
    expression = expression + '3'
    equation.set(expression)
def press4():
    global expression
    expression = expression + '4'
    equation.set(expression)
def press5():
    global expression
    expression = expression + '5'
    equation.set(expression)
```

```
def press6():
    global expression
    expression = expression + '6'
    equation.set(expression)
def press7():
    global expression
    expression = expression + '7'
    equation.set(expression)
def press8():
    global expression
    expression = expression + '8'
    equation.set(expression)
def press9():
    global expression
    expression = expression + '9'
    equation.set(expression)
def press0():
    global expression
    expression = expression + '0'
    equation.set(expression)
```

```
def press_add():
    global expression
    expression = expression + '+'
    equation.set(expression)
def press_sub():
    global expression
    expression = expression + '-'
    equation.set(expression)
def press_mul():
    global expression
    expression = expression + '*'
    equation.set(expression)
def press_div():
    global expression
    expression = expression + '/'
    equation.set(expression)
```

```
def press(num):
    global expression
    expression = expression + str(num)
    equation.set(expression)
```

업그레이드하기 – 람다함수를 사용하여 버튼과 함수 연결

clearDisplay(), resultPress()함수를 제외한 모든 숫자, 연산자 버튼을 하나의 람다 함수에 연결

```
# 버튼 commnad 속성에서는 press(1)과 같은 방법으로 함수를 호출할 수 없다.
btn1 = tk.Button(window, text='1', width=5, height=2, font='Arial 15', command=press(1))

# 아래와 같이 람다함수를 사용하여 함수이 가능함.
btn1 = tk.Button(window, text='1', width=5, height=2, font='Arial 15', command=lambda: press(1))
btn2 = tk.Button(window, text='2', width=5, height=2, font='Arial 15', command=lambda: press(2))
btn3 = tk.Button(window, text='3', width=5, height=2, font='Arial 15', command=lambda: press(3))
btn4 = tk.Button(window, text='4', width=5, height=2, font='Arial 15', command=lambda: press(4))
btn5 = tk.Button(window, text='5', width=5, height=2, font='Arial 15', command=lambda: press(5))
btn6 = tk.Button(window, text='6', width=5, height=2, font='Arial 15', command=lambda: press(6))
btn7 = tk.Button(window, text='7', width=5, height=2, font='Arial 15', command=lambda: press(7))
btn8 = tk.Button(window, text='8', width=5, height=2, font='Arial 15', command=lambda: press(8))
btn9 = tk.Button(window, text='9', width=5, height=2, font='Arial 15', command=lambda: press(9))
btn0 = tk.Button(window, text='0', width=5, height=2, font='Arial 15', command=lambda: press(0))
addBtn = tk.Button(window, text='+', width=5, height=2, font='Arial 15', command=lambda: press('+'))
subBtn = tk.Button(window, text='-', width=5, height=2, font='Arial 15', command=lambda: press('-'))
mulBtn = tk.Button(window, text='*', width=5, height=2, font='Arial 15', command=lambda: press('*'))
divBtn = tk.Button(window, text='/', width=5, height=2, font='Arial 15', command=lambda: press('/'))
```

Thank you