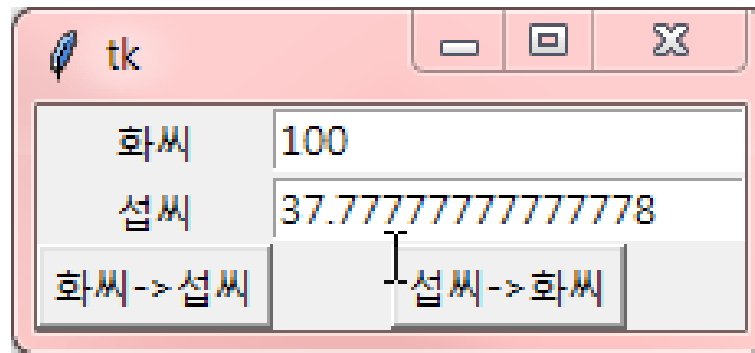




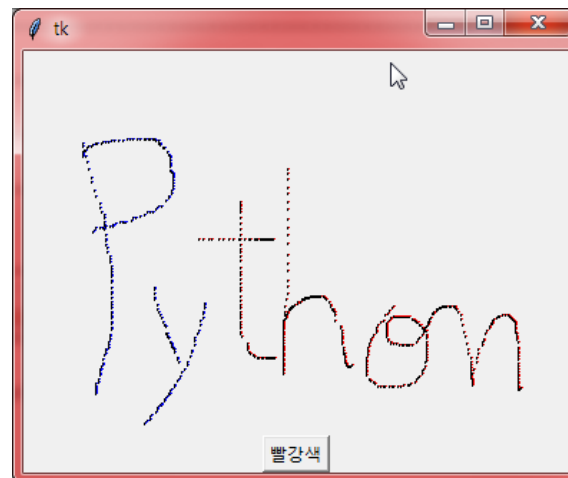
10장 TKINTER로 GUI 만들기

이번 장에서 만들 프로그램

(1) 온도 변환 프로그램을 GUI 버전으로 다시 제작해보자



(2) 마우스를 사용하여 화면에 그림을 그리는 프로그램을 작성해보자.



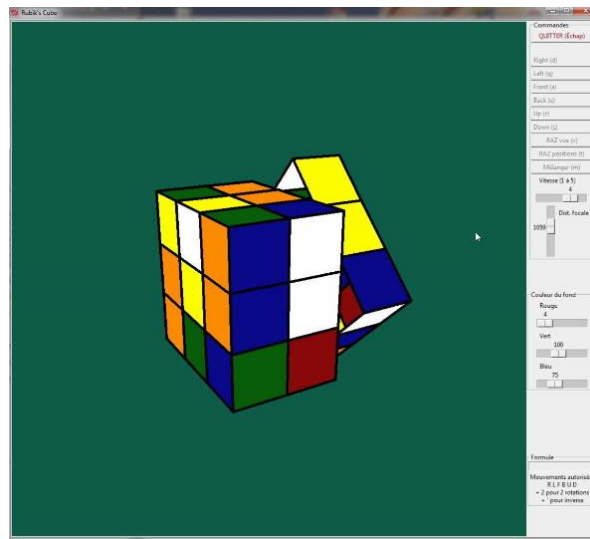
tkinter란?

- tkinter는 파이썬에서 **그래픽 사용자 인터페이스(GUI: graphical user interface)**를 개발할 때 필요한 모듈

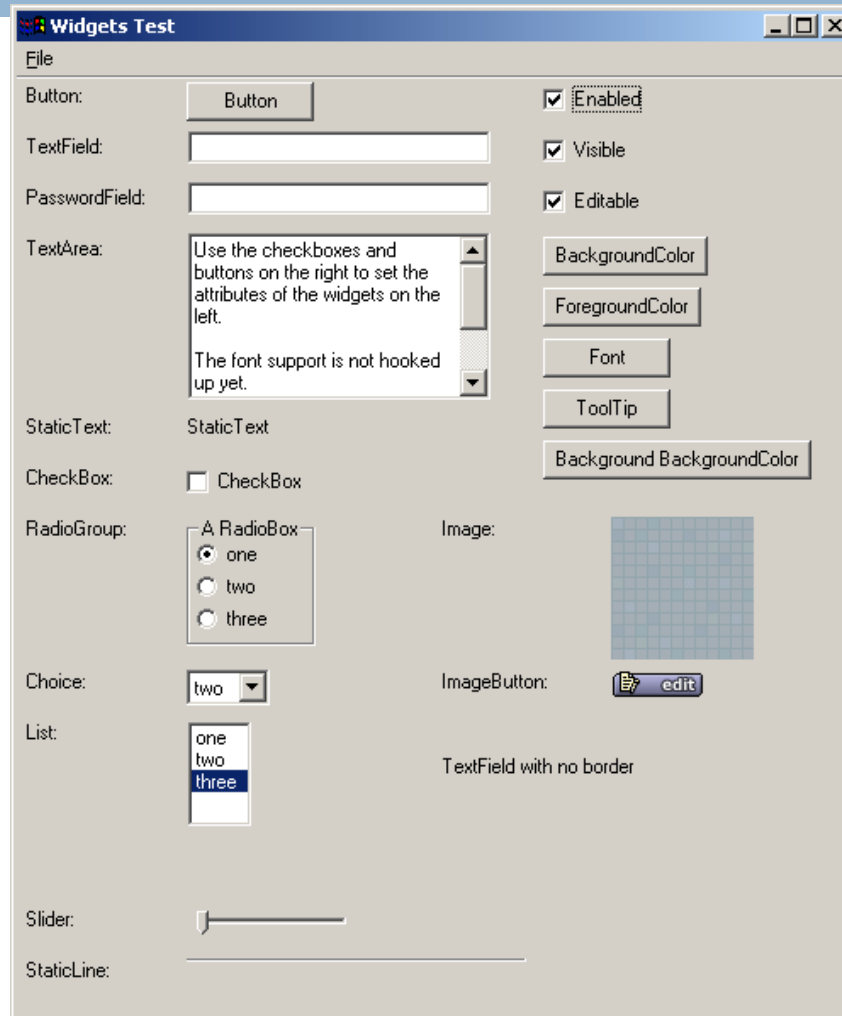


tkinter의 유래

- tkinter는 예전부터 유닉스 계열에서 사용되던 Tcl/Tk 위에 객체 지향 계층을 입힌 것이다. Tk는 John Ousterhout에 의하여 Tcl 스크립팅 언어를 위한 GUI 확장으로 개발



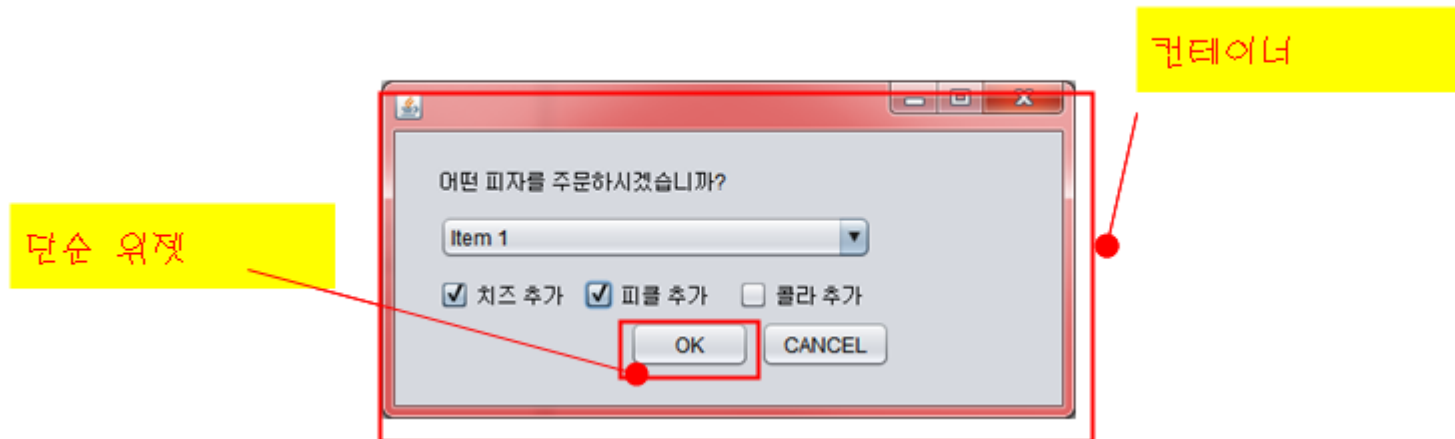
Tkinter의 위젯들



위젯	설명
Button	간단한 버튼으로 명령을 수행할 때 사용된다.
Canvas	화면에 무언가를 그릴 때 사용한다.
Checkbutton	2가지의 구별되는 값을 가지는 변수를 표현한다.
Entry	한 줄의 텍스트를 입력받는 필드이다.
Frame	컨테이너 클래스이다. 프레임은 경계선과 배경을 가지고 있다. 다른 위젯들을 그룹핑하는데 사용된다.
Label	텍스트나 이미지를 표시한다.
Listbox	선택 사항을 표시한다.
Menu	메뉴를 표시한다. 풀다운 메뉴나 팝업 메뉴가 가능하다.
Menubutton	메뉴 버튼이다. 풀다운 메뉴가 가능하다.
Message	텍스트를 표시한다. 레이블 위젯과 비슷하다. 하지만 자동적으로 주어진 크기로 텍스트를 축소할 수 있다.
Radiobutton	여러 값을 가질 수 있는 변수를 표시한다.
Scale	슬라이더를 끌어서 수치를 입력하는데 사용된다.
Scrollbar	캔버스, 엔트리, 리스트 박스, 텍스트 위젯을 위한 스크롤 바를 제공한다.
Text	형식을 가지는 텍스트를 표시한다. 여러 가지 스타일과 속성으로 텍스트를 표시할 수 있다.
Toplevel	최상위 윈도우로 표시되는 독립적인 컨테이너 위젯이다.
LabelFrame	경계선과 제목을 가지는 프레임 위젯의 변형이다.
PanedWindow	자식 위젯들을 크기조절이 가능한 패널로 관리하는 컨테이너 위젯이다.
Spinbox	특정한 범위에서 값을 선택하는 엔트리 위젯의 변형

단순 위젯과 컨테이너 위젯

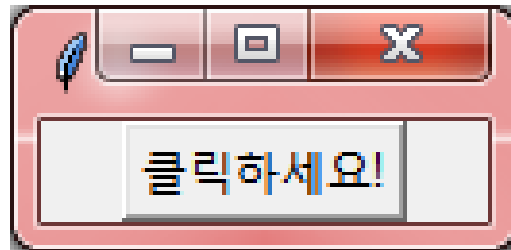
- 단순 위젯: Button, Canvas, Checkbutton, Entry, Label, Message 등이 여기에 속한다.
- 컨테이너 컴포넌트: 다른 컴포넌트를 안에 포함할 수 있는 컴포넌트로서 Frame, Toplevel, LabelFrame, PanedWindow 등이 여기에 속한다.



버튼이 있는 윈도우를 생성해보자

- 하나의 버튼이 있는 윈도우를 생성해보자

```
from tkinter import *  
  
window = Tk()  
button = Button(window, text="클릭하세요!")  
button.pack()  
  
window.mainloop()
```



엔트리와 레이블 위젯도 사용해보자.

```
from tkinter import *

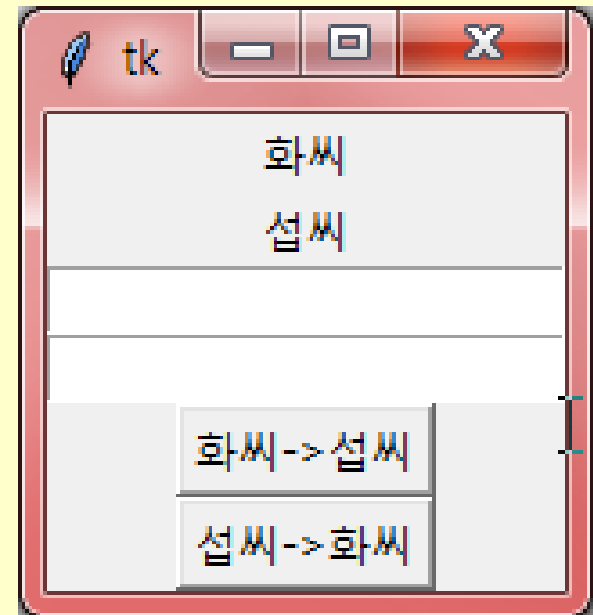
window = Tk()

l1 = Label(window, text="화씨")
l2 = Label(window, text="섭씨")
l1.pack()
l2.pack()

e1 = Entry(window)
e2 = Entry(window)
e1.pack()
e2.pack()

b1 = Button(window, text="화씨->섭씨")
b2 = Button(window, text="섭씨->화씨")
b1.pack()
b2.pack()

window.mainloop( )
```



배치 관리자

- 압축(pack) 배치 관리자
- 격자(grid) 배치 관리자
- 절대(place) 배치 관리자



격자 배치 관리자

```
from tkinter import *

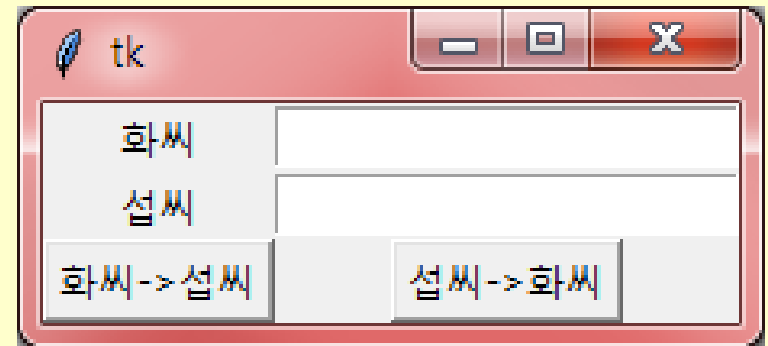
window = Tk()

l1 = Label(window , text="화씨")
l2 = Label(window, text="섭씨")
l1.grid(row=0, column=0)
l2.grid(row=1, column=0)

e1 = Entry(window)
e2 = Entry(window)
e1.grid(row=0, column=1)
e2.grid(row=1, column=1)

b1 = Button(window, text="화씨->섭씨")
b2 = Button(window, text="섭씨->화씨")
b1.grid(row=2, column=0)
b2.grid(row=2, column=1)

window.mainloop()
```

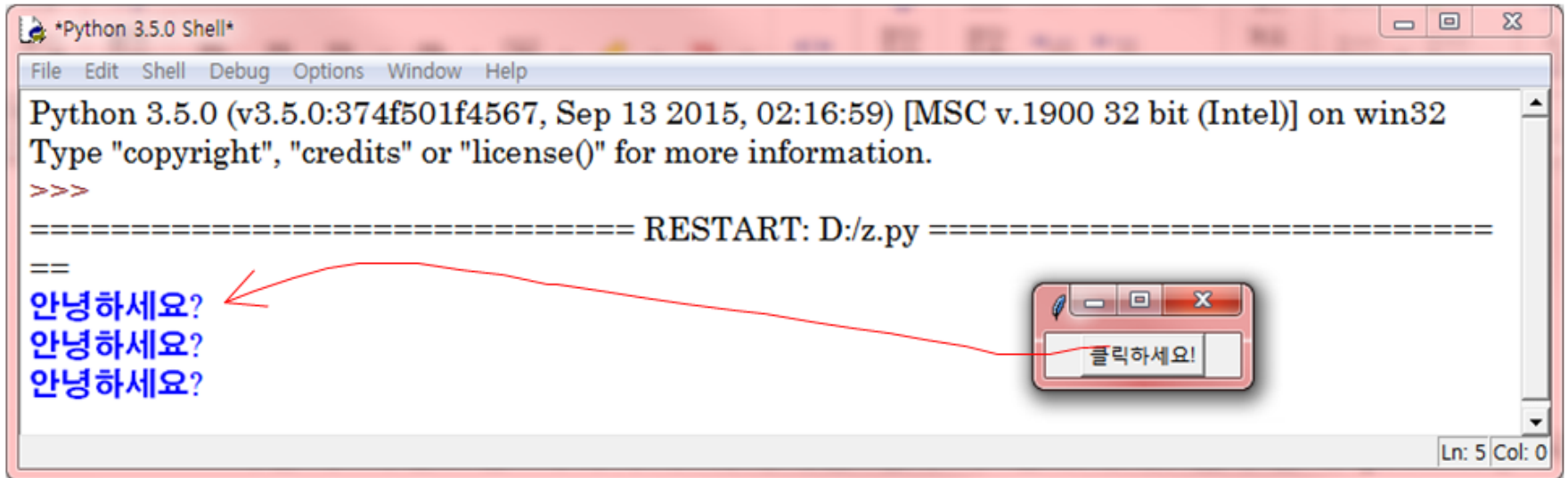


버튼 이벤트 처리

```
from tkinter import *  
  
def process():  
    print("안녕하세요!")  
  
window = Tk()  
button = Button(window, text="클릭하세요!", command=process)  
button.pack()  
window.mainloop()
```



버튼 이벤트 처리



버튼 이벤트 처리

```
from tkinter import *
```

```
def process():  
    e2.insert(0, "100")
```

```
window = Tk()
```

```
l1 = Label(window , text="화씨")
```

```
l2 = Label(window, text="섭씨")
```

```
l1.grid(row=0, column=0)
```

```
l2.grid(row=1, column=0)
```

```
e1 = Entry(window)
```

```
e2 = Entry(window)
```

```
e1.grid(row=0, column=1)
```

```
e2.grid(row=1, column=1)
```

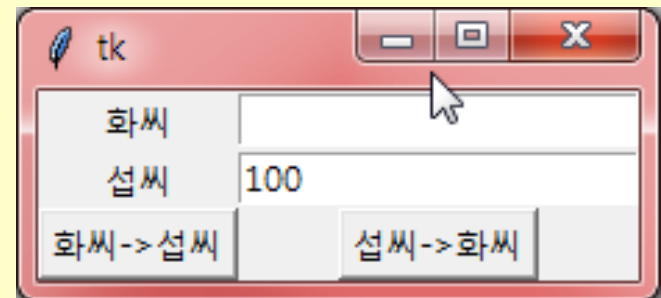
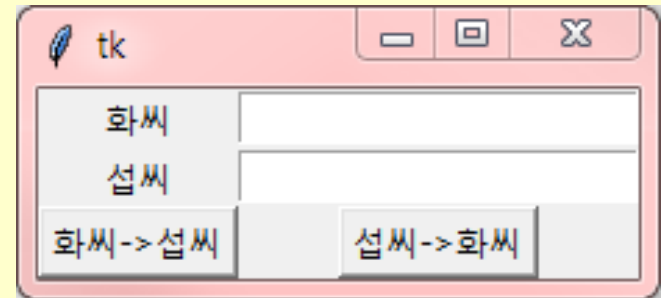
```
b1 = Button(window, text="화씨->섭씨", command=process)
```

```
b2 = Button(window, text="섭씨->화씨")
```

```
b1.grid(row=2, column=0)
```

```
b2.grid(row=2, column=1)
```

```
window.mainloop()
```



```
from tkinter import *
```

```
def process():  
    temperature = float(e1.get())  
    mytemp = (temperature-32)*5/9  
    e2.insert(0, str(mytemp))
```

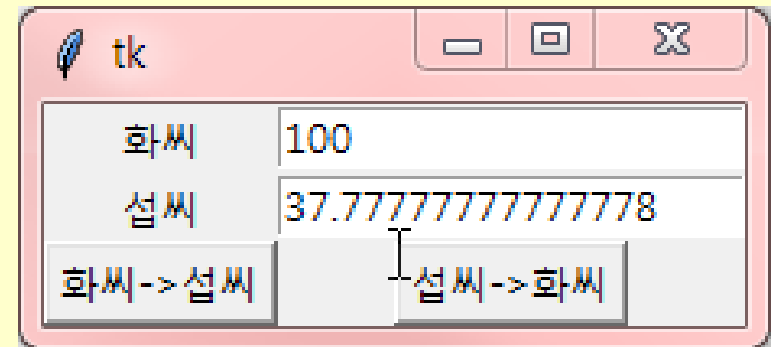
```
window = Tk()
```

```
l1 = Label(window , text="화씨")  
l2 = Label(window, text="섭씨")  
l1.grid(row=0, column=0)  
l2.grid(row=1, column=0)
```

```
e1 = Entry(window)  
e2 = Entry(window)  
e1.grid(row=0, column=1)  
e2.grid(row=1, column=1)
```

```
b1 = Button(window, text="화씨->섭씨", command=process)  
b2 = Button(window, text="섭씨->화씨")  
b1.grid(row=2, column=0)  
b2.grid(row=2, column=1)
```

```
window.mainloop()
```



```
from tkinter import *
```

```
def process():  
    temperature = float(e1.get())  
    mytemp = (temperature-32)*5/9  
    e2.insert(0, str(mytemp))
```

```
window = Tk()
```

```
l1 = Label(window , text="화씨", font='helvetica 16 italic')
```

```
l2 = Label(window, text="섭씨", font='helvetica 16 italic')
```

```
l1.grid(row=0, column=0)
```

```
l2.grid(row=1, column=0)
```

```
e1 = Entry(window, bg="yellow", fg="white")
```

```
e2 = Entry(window, bg="yellow", fg="white")
```

```
e1.grid(row=0, column=1)
```

```
e2.grid(row=1, column=1)
```

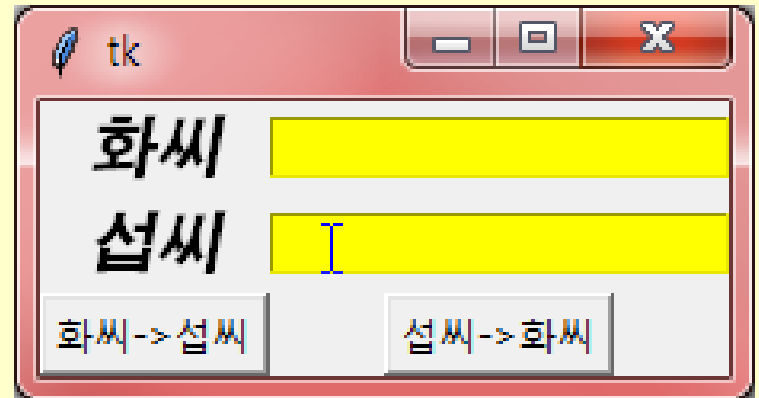
```
b1 = Button(window, text="화씨->섭씨", command=process)
```

```
b2 = Button(window, text="섭씨->화씨")
```

```
b1.grid(row=2, column=0)
```

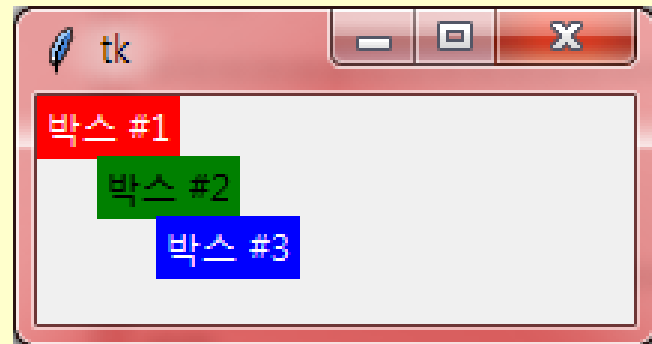
```
b2.grid(row=2, column=1)
```

```
window.mainloop()
```



절대 위치 배치 관리자

```
from tkinter import *  
  
window = Tk()  
  
w = Label(window, text="박스 #1", bg="red", fg="white")  
w.place(x=0, y=0)  
w = Label(window, text="박스 #2", bg="green", fg="black")  
w.place(x=20, y=20)  
w = Label(window, text="박스 #3", bg="blue", fg="white")  
w.place(x=40, y=40)  
  
window.mainloop()
```



이미지 표시 프로그램

```
from tkinter import *

def change_img():
    path = inputBox.get()
    img = PhotoImage(file = path)
    imageLabel.configure(image = img)
    imageLabel.image = img

window = Tk()

photo = PhotoImage(file="wl.gif")
imageLabel = Label(window, image=photo)
imageLabel.pack()

inputBox = Entry(window)
inputBox.pack()

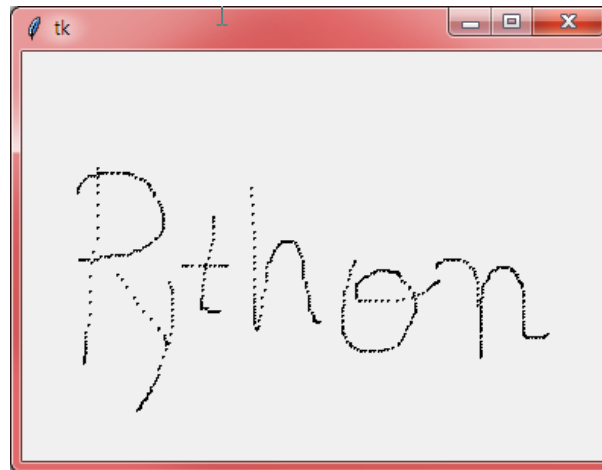
button = Button(window, text='Submit', command=change_img)
button.pack()

window.mainloop()
```



MyPaint 프로그램 #1

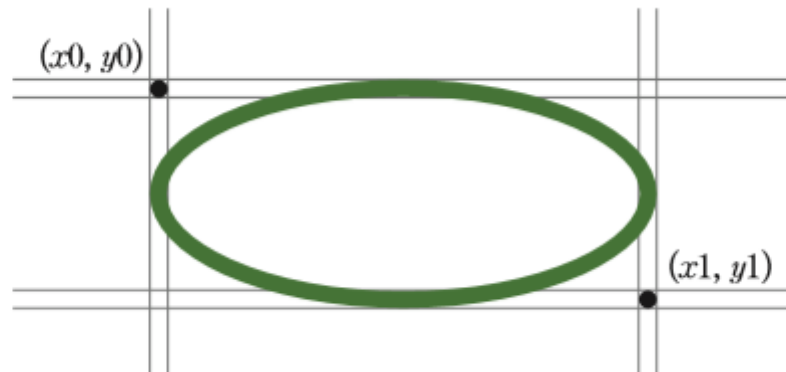
- 다음과 같이 마우스를 움직여서 화면에 그림을 그리는 윈도우의 그림판과 비슷한 프로그램을 작성해보자.



캔버스 위젯

- tkinter에서 그림을 그리려면 **캔버스(canvas)**라는 위젯이 필요하다. Canvas 위젯을 사용하면 많은 그래픽 기능을 사용할 수 있다.

```
window = Tk()  
...  
canvas = Canvas(window, width=300, height=200)  
canvas.create_oval(x0, y0, x1, y1, option, ...)
```



MyPaint 프로그램 #2

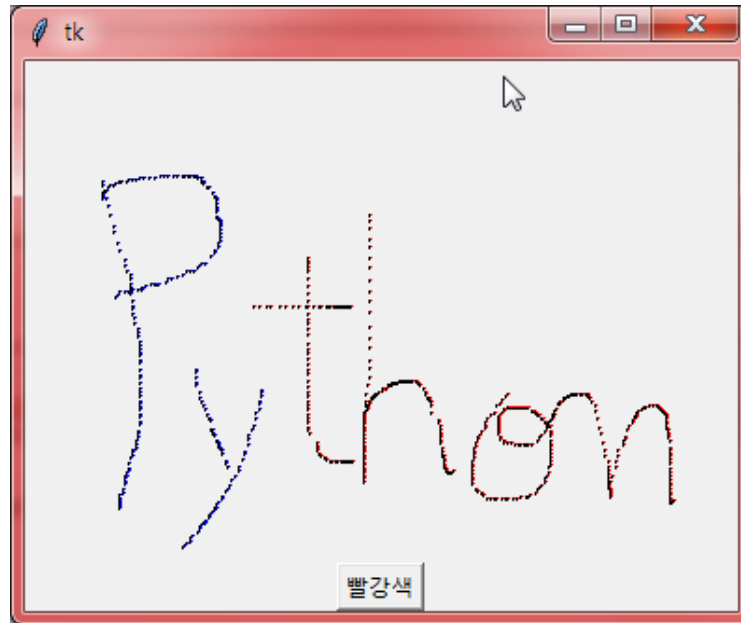
```
from tkinter import *

def paint(event):
    x1, y1 = ( event.x-1 ), ( event.y+1 )
    x2, y2 = ( event.x-1 ), ( event.y+1 )
    canvas.create_oval( x1, y1, x2, y2, fill = "black")

window = Tk()
canvas = Canvas(window)
canvas.pack()
canvas.bind("<B1-Motion>", paint)
window.mainloop()
```

MyPaint 프로그램 #3

- 앞의 MyPaint 프로그램에서 색상을 변경할 수 있도록 하여보자. 캔버스 위젯 아래에 버튼 "빨강색"을 추가하고 이 버튼을 누르면 색상이 빨강색으로 변경되게 하자.



```
from tkinter import *

mycolor = "blue"

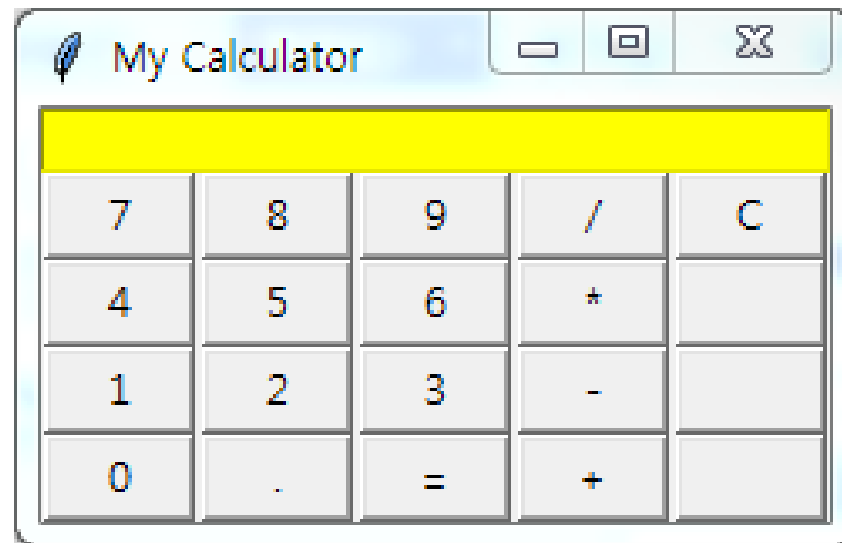
def paint(event):
    x1, y1 = ( event.x-1 ), ( event.y+1 )
    x2, y2 = ( event.x-1 ), ( event.y+1 )
    canvas.create_oval( x1, y1, x2, y2, fill = mycolor)

def change_color():
    global mycolor
    mycolor="red"

window = Tk()
canvas = Canvas(window)
canvas.pack()
canvas.bind("<B1-Motion>", paint)
button = Button(window, text="빨강색", command=change_color)
button.pack()
window.mainloop()
```

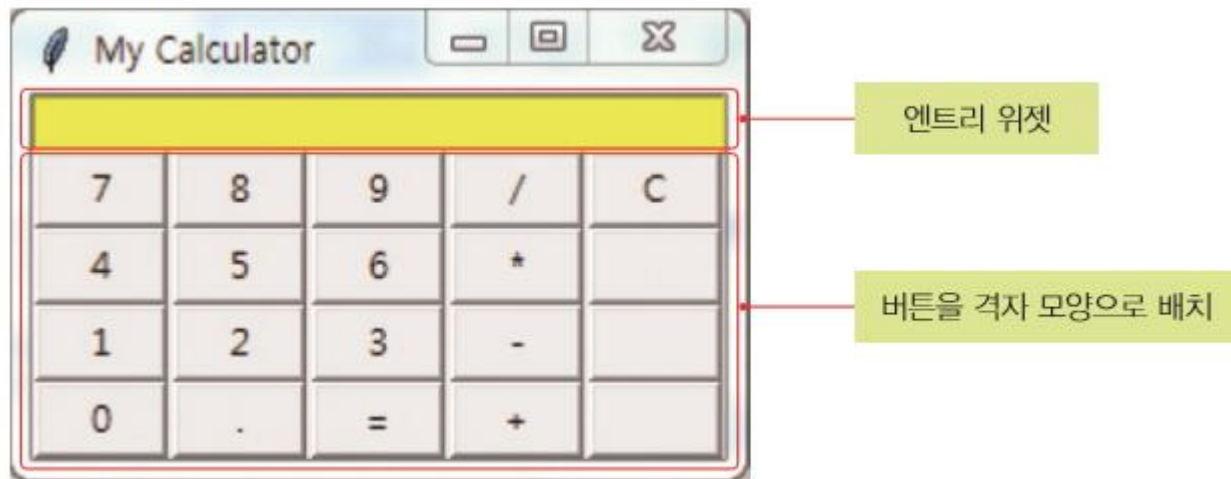
계산기 프로그램 #1

- 우리는 다음과 같은 계산기를 작성해보자. 약간 어려운 내용도 있어서 입문자인 경우에는 건너뛰어도 된다.



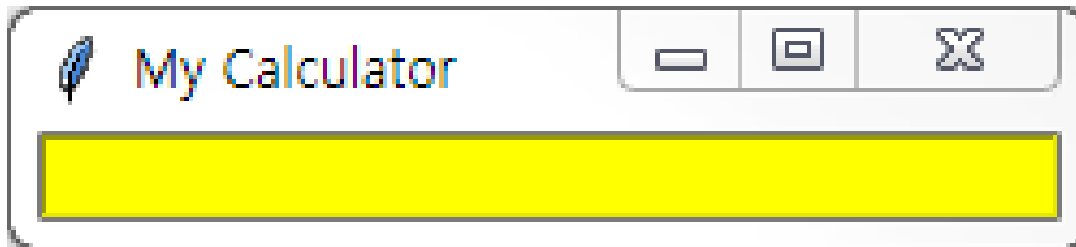
사용자 인터페이스 작성

- 계산기는 격자 배치 관리자를 사용 하면 될 것이다. 그리고 버튼과 엔트리 위젯만 있으면 된다.



계산기 프로그램 #2

```
from tkinter import *  
  
window = Tk()  
window.title("My Calculator")  
display = Entry(window, width=33, bg="yellow")  
display.grid(row=0, column=0, columnspan=5)
```



계산기 프로그램 #3

```
button_list = [  
    '7', '8', '9', '/', 'C',  
    '4', '5', '6', '*', '  
    '1', '2', '3', '-', '  
    '0', '.', '=', '+', ' ' ]
```

```
row_index = 1
```

```
col_index = 0
```

```
for button_text in button_list:
```

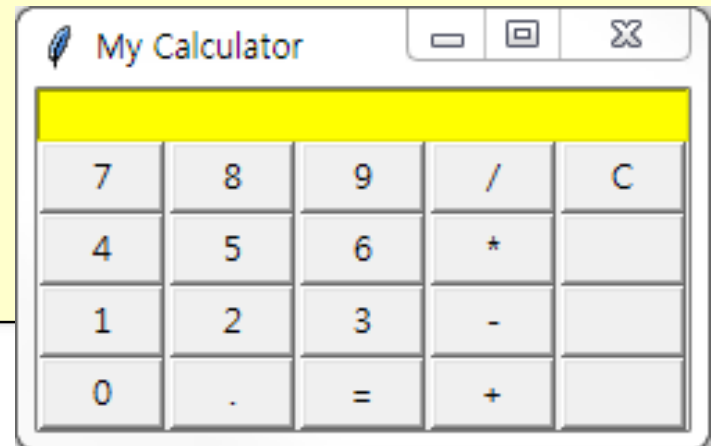
```
    Button(window, text=button_text, width=5).grid(row=row_index,  
column=col_index)
```

```
    col_index += 1
```

```
    if col_index > 4:
```

```
        row_index += 1
```

```
        col_index = 0
```



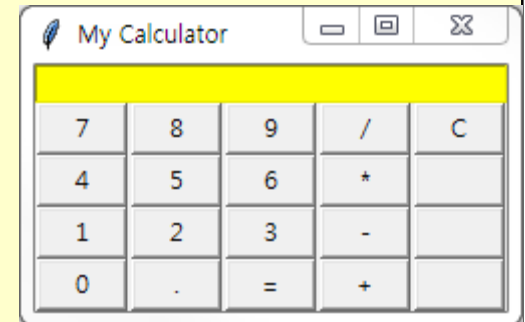
계산기 프로그램 #4

```
from tkinter import *

window = Tk()
window.title("My Calculator")
display = Entry(window, width=33, bg="yellow")
display.grid(row=0, column=0, columnspan=5)

button_list = [
    '7', '8', '9', '/', 'C',
    '4', '5', '6', '*', ' ',
    '1', '2', '3', '-', ' ',
    '0', '.', '=', '+', ' ' ]

row_index = 1
col_index = 0
for button_text in button_list:
    Button(window, text=button_text, width=5).grid(row=row_index,
column=col_index)
    col_index += 1
    if col_index > 4:
        row_index += 1
        col_index = 0
```



Q & A

