```python
from tkinter import *

window = Tk()

Label(window,text='화씨').grid(row=0,column=0)
Label(window,text='섭씨').grid(row=1,column=0)

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

Button(window, text="화씨->섭씨").grid(row=2,column=1)
window.mainloop()
```
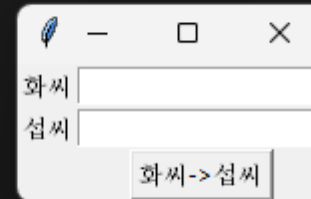
```python
from tkinter import*

window = Tk()

Label(window, text="너비").grid(row=0,column=0)
Label(window, text="높이").grid(row=1,column=0)

e1=Entry(window)
e2=Entry(window)

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

# Update the file path to the correct location of the image
# Update the file path to the correct location of the image
photo = PhotoImage(file=r"c:/Users/andycho/OneDrive/Desktop/2025 2학년 1학기/컴퓨터사고및응용/ard
window.photo = photo  # Keep a reference to the PhotoImage object
label = Label(window, image=photo)
label.grid(row=0,column=2,columnspan=2,rowspan=2)

Button(window, text="확대").grid(row=2,column=2)
Button(window,text='축소').grid(row=2,column=3)

window.mainloop()
```
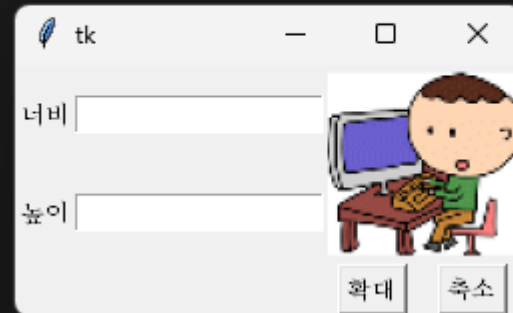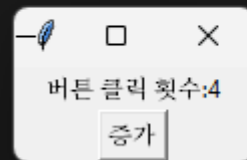
```python
from tkinter import*

window=Tk()

counter = 0

def clicked():
    global counter
    counter +=1
    label['text'] = '버튼 클릭 횟수:' + str(counter)

label = Label(window, text="아직 눌려지지 않음")
label.pack()
button = Button(window, text="증가", command=clicked).pack()

window.mainloop()
```
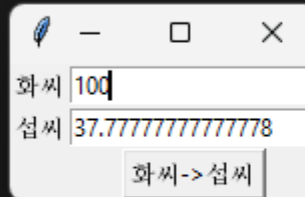
```python
from tkinter import*


def process():
    tf = float(e1.get())
    tc = (tf-32.0)*5.0/9.0
    e2.delete(0,END)
    e2.insert(0, str(tc))


window = Tk()

Label(window, text='화씨').grid(row=0,column=0)
Label(window, text='섭씨').grid(row=1,column=0)

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

Button(window, text='화씨->섭씨',command=process).grid(row=2,column=1)
window.mainloop()
```

```python
def reset():
    global answer
    answer = random.randint(1, 100)
    resultLabel['text'] = '다시해라'


window = Tk()
window.configure(bg='white')
window.title('숫자를 맞춰')


window.geometry('500x80')  # Corrected format
titleLabel = Label(window, text='숫자 게임에 온 걸 환영해', bg='white')
titleLabel.pack()


guessField = Entry(window)
guessField.pack(side="top")


tryButton = Button(window, text="시도", fg="green", bg="white", command=guessing)
tryButton.pack(side="left")


resetButton = Button(window, text="초기화", fg="red", bg="white", command=reset)
resetButton.pack(side="right")


resultLabel = Label(window, text="1 부터 100 사이의 숫자를 입력하시오.", bg='white')
resultLabel.pack(side="bottom")


window.mainloop()
```
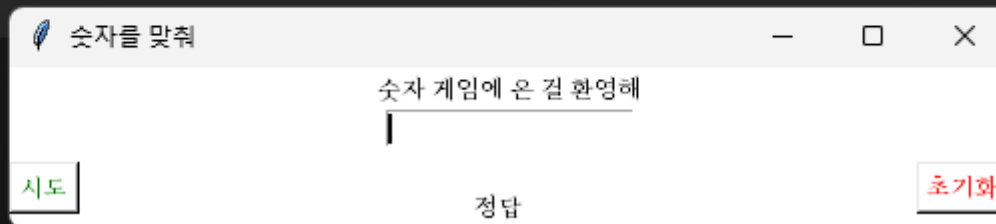
숫자를 맞춰

숫자 게임에 온 걸 환영해

시도   정답   초기화

```python
import random
from tkinter import *

window = Tk()
window.title("가위 바위 보 게임")
Label(window, text='선택하세요', font=('Helvetica', 16)).pack()

frame = Frame(window)
frame.pack()

scissors_image = PhotoImage(file=r"C:\Users\andycho\OneDrive\Desktop\2025 2학년 1학기\컴퓨터사고및응용\archive\9주차\가위.png")
paper_image = PhotoImage(file=r"C:\Users\andycho\OneDrive\Desktop\2025 2학년 1학기\컴퓨터사고및응용\archive\9주차\보.png")
rock_image = PhotoImage(file=r"C:\Users\andycho\OneDrive\Desktop\2025 2학년 1학기\컴퓨터사고및응용\archive\9주차\묵.png")

def decide(human):
    computer = random.choice(["가위", "바위", "보"])
    if computer == "바위":
        computer_image.config(image=rock_image)
    elif computer == "보":
        computer_image.config(image=paper_image)
    else:
        computer_image.config(image=scissors_image)
    if (computer == "바위" and human == "보") or (computer == "보" and human == "가위") or (computer == "가위" and human == "바위"):
        result = "인간 승리!"
    elif computer == human:
        result = "비겼습니다."
    else:
        result = "컴퓨터 승리!"
    output.config(text=f"인간: {human}  컴퓨터: {computer}  결과: {result}")

def pass_s():
    decide("가위")

def pass_r():
    decide("바위")

def pass_p():
    decide("보")

rock = Button(frame, image=rock_image, command=pass_r)
rock.pack(side='left')
paper = Button(frame, image=paper_image, command=pass_p)
paper.pack(side='left')
scissors = Button(frame, image=scissors_image, command=pass_s)
scissors.pack(side='left')

computer_image = Label(window, image=rock_image)
computer_image.pack()

Label(window, text="컴퓨터는 다음을 선택하였다.", font=("Helvetica", 16)).pack()

output = Label(window, text="", font=("Helvetica", 16))
output.pack()

window.mainloop()
```

Problems   Output   Debug Console   **Terminal**   Ports   Query Results

가위 바위 보 게임

선택하세요

컴퓨터는 다음을 선택하였다.

```python
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', '.', '=', '+', ''
]

def click(key):
    if key == "C":
        display.delete(0, END)
    elif key == "=":
        try:
            result = eval(display.get())
            display.delete(0, END)
            display.insert(END, str(result))
        except Exception:
            display.delete(0, END)
            display.insert(END, "Error")
    else:
        display.insert(END, key)

row_index = 1
col_index = 0

for button_text in button_list:
    if button_text != "":
        Button(window, text=button_text, width=5, command=lambda bt=button_text: click(bt))\
            .grid(row=row_index, column=col_index)
    col_index += 1
    if col_index > 4:
        row_index += 1
        col_index = 0

window.mainloop()
```
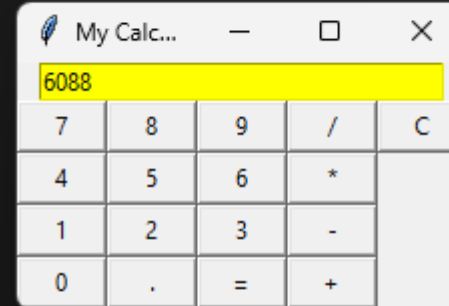
```python
from tkinter import*

WIDTH =  600
HEIGHT = 200
def displayRect():
    canvas.create_rectangle(10,10,WIDTH-10,HEIGHT-10)

def displayOval():
    canvas.create_oval(10, 10, WIDTH - 10, HEIGHT - 10, fill="yellow")

def displayArc():
    canvas.create_arc(10, 10, WIDTH - 10, HEIGHT - 10, start=0, extent=120, width=10, fill='blue')

def displayPolygon():
    canvas.create_polygon(10, 10, WIDTH - 10, HEIGHT - 10, 200, 90, 300, 160)

def displayLine():
    canvas.create_line(10, 10, WIDTH - 10, HEIGHT - 10, fill='green')

def clearCanvas():
    canvas.delete("all")

window=Tk()
canvas = Canvas(window,width=WIDTH,height=HEIGHT,bg='white')
canvas.pack()
frame=Frame(window)
frame.pack()

btRectangle = Button(frame, text="Rectangle", command=displayRect)
btRectangle.grid(row=1, column=0)

btOval = Button(frame, text="Oval", command=displayOval)
btOval.grid(row=1, column=1)

btArc = Button(frame, text="Arc", command=displayArc)
btArc.grid(row=1, column=2)

btPolygon = Button(frame, text="Polygon", command=displayPolygon)
btPolygon.grid(row=1, column=3)

btLine = Button(frame, text="Line", command=displayLine)
btLine.grid(row=1, column=4)

btClear = Button(frame, text="Clear", command=clearCanvas)
btClear.grid(row=1, column=5)

window.mainloop()
```
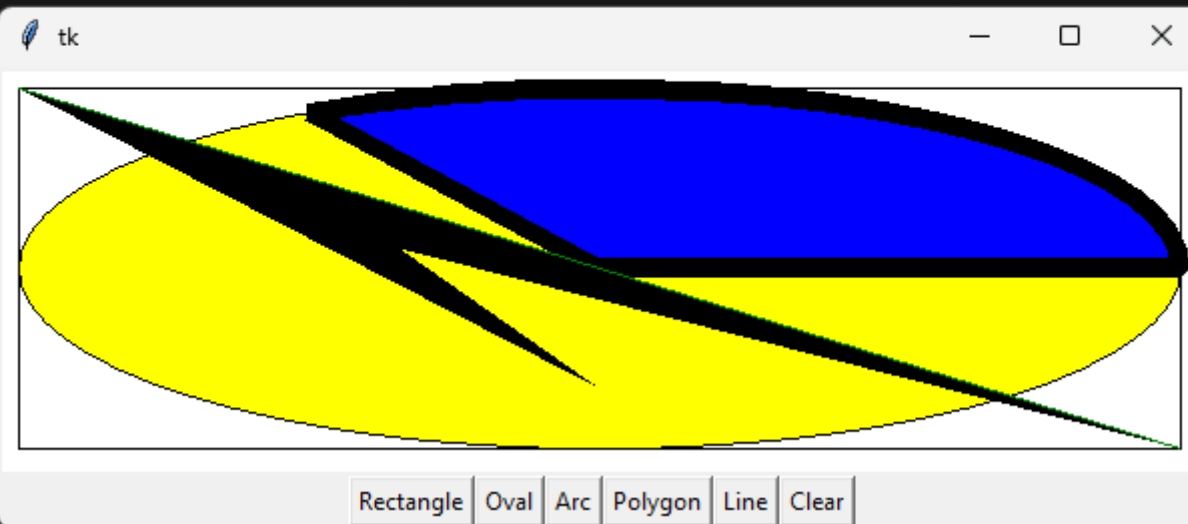
```python
from tkinter import *
from tkinter.colorchooser import askcolor

# 설정 상수
DEFAULT_PEN_SIZE = 1.0
DEFAULT_COLOR = "black"
CANVAS_WIDTH = 600
CANVAS_HEIGHT = 400

class PaintApp:
    def __init__(self, root):
        self.root = root
        self.root.title("Tkinter 그림판")

        self.pen_size = DoubleVar(value=DEFAULT_PEN_SIZE)
        self.color = DEFAULT_COLOR
        self.mode = "pen"
        self.old_x = None
        self.old_y = None

        self.create_widgets()
        self.bind_events()

    def create_widgets(self):
        # 버튼들
        Button(self.root, text="펜", command=self.use_pen).grid(row=0, column=0, sticky=W)
        Button(self.root, text="브러쉬", command=self.use_brush).grid(row=0, column=1, sticky=W)
        Button(self.root, text="색상선택", command=self.choose_color).grid(row=0, column=2, sticky=W)
        Button(self.root, text="지우개", command=self.use_eraser).grid(row=0, column=3, sticky=W)
        Button(self.root, text="모두삭제", command=self.clear_canvas).grid(row=0, column=4, sticky=W)

        # 선 굵기 조절
        Scale(self.root, variable=self.pen_size, from_=1, to=10, orient=VERTICAL).grid(row=1, column=5, sticky=N)

        # 캔버스
        self.canvas = Canvas(self.root, bg="white", width=CANVAS_WIDTH, height=CANVAS_HEIGHT)
        self.canvas.grid(row=1, column=0, columnspan=5)

    def bind_events(self):
        self.canvas.bind("<B1-Motion>", self.p
        self.canvas.bind("<ButtonRelease-1>",

    # 모드 설정 함수들
    def use_pen(self):
        self.mode = "pen"

    def use_brush(self):
        self.mode = "brush"

    def choose_color(self):
        color = askcolor(color=self.color)[1]
        if color:
            self.color = color

    def use_eraser(self):
        self.mode = "erase"

    def clear_canvas(self):
        self.canvas.delete("all")

    def paint(self, event):
        fill_color = "white" if self.mode ==
        width = self.pen_size.get() * (3 if se

        if self.old_x and self.old_y:
            self.canvas.create_line(
                self.old_x, self.old_y, event.
                width=width, fill=fill_color,
                capstyle=ROUND, smooth=True
            )
        self.old_x, self.old_y = event.x, even

    def reset(self, event):
        self.old_x, self.old_y = None, None
```
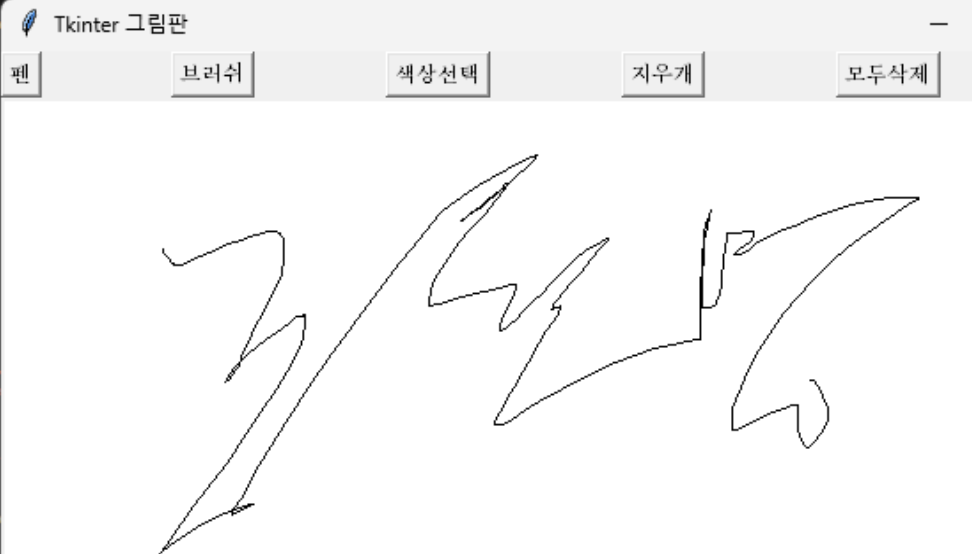
C:\Users\andycho\OneDrive\
exe "c:/Users/andycho/OneDri
"

Tkinter 그림판

| 펜 | 브러쉬 | 색상선택 | 지우개 | 모두삭제 |

```python
     window = Tk()
     canvas=Canvas(window, width=600, height=400)
     canvas.pack()

     class Ball():
         def __init__(self, color, size):
             self.id=canvas.create_oval(0,0,size,size,fill=color)
             self.dx=random.randint(1,10)
             self.dy=random.randint(1,10)
     #move는 이미 파이썬에서 만들어놓은 함수없다.
         def move(self):
             canvas.move(self.id, self.dx, self.dy)
             x0, y0, x1, y1 = canvas.coords(self.id)
             if y1 > canvas.winfo_height() or y0 < 0:
                 self.dy = -self.dy
             if x1 > canvas.winfo_width() or x0<0:
                 self.dx = -self.dx

     '''ball1=Ball('blue', 60)
     ball2=Ball('green',100)
     ball3=Ball('orange',80)

     while True:
         ball1.move()
         ball2.move()
         ball3.move()
         window.update()
         time.sleep(0.05)'''

     colors = ["red", "orange", "yellow", "green", "blue", "indigo", "violet"]
     ballList= [Ball(random.choice (colors), 60) for _ in range(30)]
     while True:
         for ball in ballList:
             ball.move()
         window.update()
         time.sleep(1/180)



     window.mainloop()
```
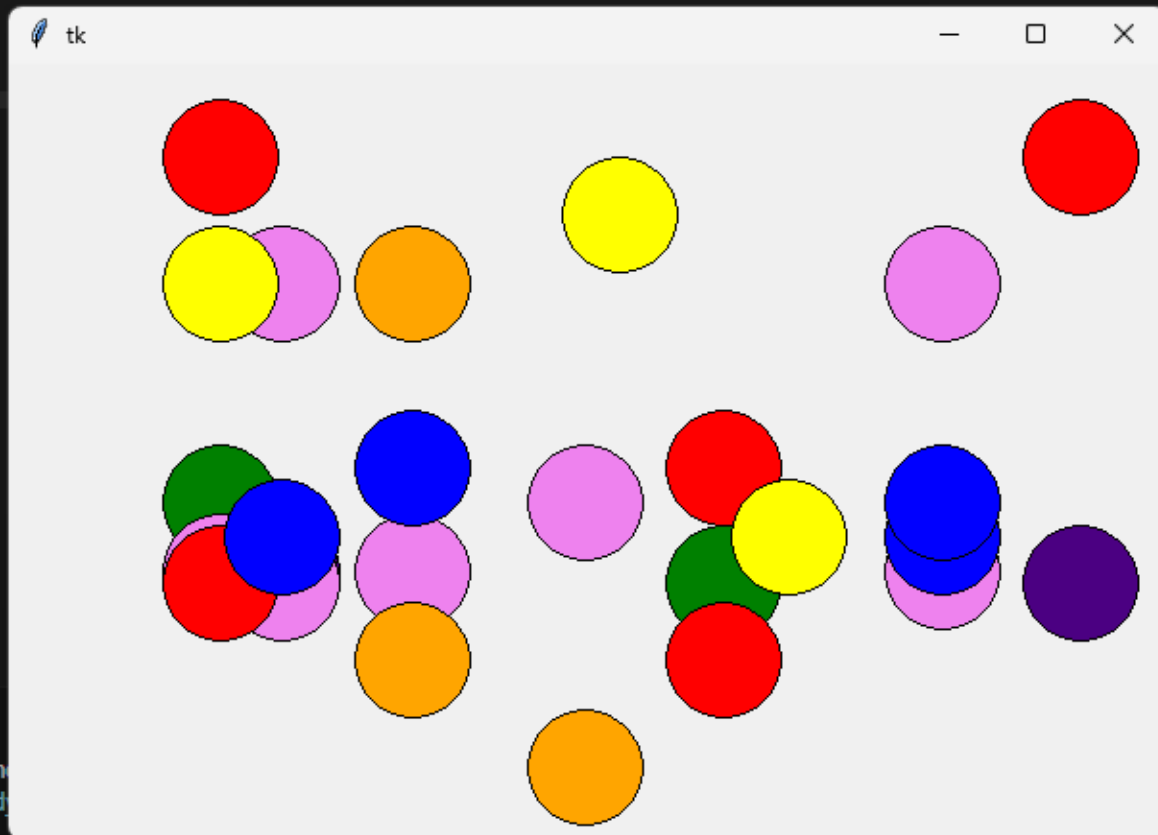
```python
import tkinter as tk

def increase():
    global count
    count += 1
    lbl.config(text=str(count))

def decrease():
    global count
    count -= 1
    lbl.config(text=str(count))

# 메인 윈도우 생성
root = tk.Tk()
root.title("tk")

# 초기 카운터 값
count = 0

# (선택) 버튼들을 감쌀 프레임을 하나 만들어도 됩니다.
frame = tk.Frame(root)
frame.pack(padx=5, pady=5)

# pack 만 사용해서 왼쪽부터 차례대로 붙이기
btn_dec = tk.Button(frame, text="감소", command=decrease)
btn_dec.pack(side='left', padx=5)

lbl = tk.Label(frame, text=str(count), width=5, anchor='center')
lbl.pack(side='left', padx=5)

btn_inc = tk.Button(frame, text="증가", command=increase)
btn_inc.pack(side='left', padx=5)


root.mainloop()
```
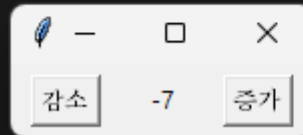
```python
def on_cancel():
    # 입력란을 깨끗하게 비웁니다
    ent_id.delete(0, tk.END)
    ent_pw.delete(0, tk.END)

# 메인 윈도우 생성
root = tk.Tk()
root.title("tk")
root.resizable(False, False)  # 창 크기 고정

# ─────────── 위젯 생성 및 배치 ───────────
# 1행: 아이디 레이블 + 엔트리
lbl_id = tk.Label(root, text="아이디")
lbl_id.grid(row=0, column=0, padx=5, pady=5, sticky="e")

ent_id = tk.Entry(root)
ent_id.grid(row=0, column=1, padx=5, pady=5)

# 2행: 패스워드 레이블 + 엔트리 (입력 내용 숨김)
lbl_pw = tk.Label(root, text="패스워드")
lbl_pw.grid(row=1, column=0, padx=5, pady=5, sticky="e")

ent_pw = tk.Entry(root, show="*")
ent_pw.grid(row=1, column=1, padx=5, pady=5)

# 3행: 로그인 • 취소 버튼
btn_login = tk.Button(root, text="로그인", width=10, command=on_login)
btn_login.grid(row=2, column=0, padx=5, pady=10)

btn_cancel = tk.Button(root, text="취소", width=10, command=on_cancel)
btn_cancel.grid(row=2, column=1, padx=5, pady=10)

# ─────────── 그리드 컬럼 너비 조정 ───────────
root.grid_columnconfigure(0, weight=0)  # 첫 번째 컬럼(레이블)은 최소 크기
root.grid_columnconfigure(1, weight=1)  # 두 번째 컬럼(입력창+버튼)은 늘어날 수 있도록

# 이벤트 루프 시작
root.mainloop()
```
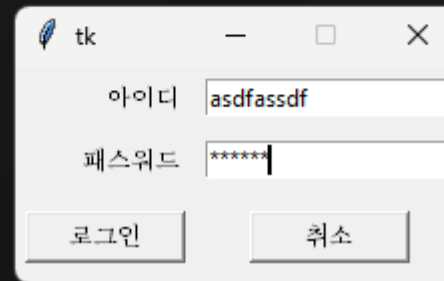
```python
import tkinter as tk

# ―――― 콜백 함수 ――――
def grow(event):
    # 현재 사각형 좌표 가져오기
    x1, y1, x2, y2 = canvas.coords(rect)
    # 우측·하단으로 10px씩 확장
    canvas.coords(rect, x1, y1, x2 + 10, y2 + 10)

def shrink(event):
    x1, y1, x2, y2 = canvas.coords(rect)
    # 최소 크기(너비·높이 20px) 이상일 때만 축소
    if (x2 - x1) > 20 and (y2 - y1) > 20:
        canvas.coords(rect, x1, y1, x2 - 10, y2 - 10)

# ―――― 메인 윈도우 설정 ――――
root = tk.Tk()
root.title("사각형 크기 조절")

# ―――― 캔버스 생성 ――――
canvas = tk.Canvas(root, width=400, height=300, bg="white")
canvas.pack(padx=10, pady=10)

# ―――― 사각형 그리기 ――――
# (x1, y1) = (50, 50), (x2, y2) = (150, 100)
rect = canvas.create_rectangle(50, 50, 150, 100, fill="skyblue")

# ―――― 마우스 이벤트 바인딩 ――――
canvas.bind("<Button-1>", grow)   # 왼쪽 클릭 → 키우기
canvas.bind("<Button-3>", shrink) # 오른쪽 클릭 → 줄이기

root.mainloop()
```


사각형 크기 조절
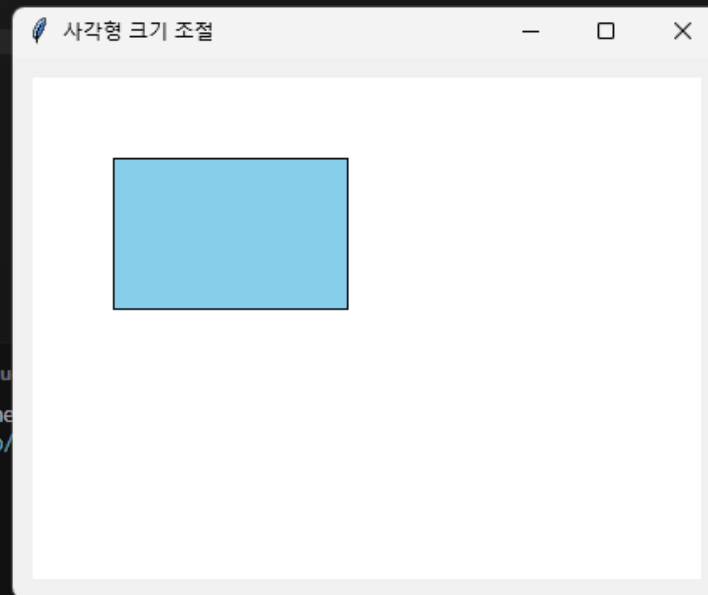
```python
def on_key_press(event):
    if event.keysym == 'Left':
        dx = -10
    elif event.keysym == 'Right':
        dx = 10
    elif event.keysym == 'Up':
        dy = -10
    elif event.keysym == 'Down':
        dy = 10
    # 사각형 이동
    canvas.move(rect, dx, dy)

# --- 메인 윈도우 설정 ---
root = tk.Tk()
root.title("화살표 키로 사각형 이동")

# --- 캔버스 생성 ---
canvas = tk.Canvas(root, width=400, height=300, bg="white")
canvas.pack(padx=10, pady=10)

# --- 사각형 그리기 ---
# (x1, y1) = (50, 50), (x2, y2) = (150, 100)
rect = canvas.create_rectangle(50, 50, 150, 100, fill="lightgreen")

# --- 키 바인딩 ---
# 캔버스에 포커스를 맞춰야 키 입력을 받을 수 있습니다.
canvas.focus_set()
canvas.bind("<KeyPress>", on_key_press)

# --- 이벤트 루프 시작 ---
root.mainloop()
```

화살표 키로 사각형 이동

```python
import tkinter as tk
import random

# —— 메인 윈도우 설정 ——
root = tk.Tk()
root.title("랜덤 사각형 10개")

# —— 캔버스 생성 ——
WIDTH, HEIGHT = 600, 400
canvas = tk.Canvas(root, width=WIDTH, height=HEIGHT, bg="white")
canvas.pack(padx=10, pady=10)

# —— 랜덤 사각형 10개 그리기 ——
for _ in range(10):
    # 위치 (x1,y1) 랜덤
    x1 = random.randint(0, WIDTH  - 50)
    y1 = random.randint(0, HEIGHT - 50)
    # 크기 (너비·높이) 랜덤 (20~100px)
    x2 = x1 + random.randint(20, 100)
    y2 = y1 + random.randint(20, 100)
    # 색상 랜덤 (hex 코드)
    color = "#{:02x}{:02x}{:02x}".format(
        random.randint(0, 255),
        random.randint(0, 255),
        random.randint(0, 255)
    )
    # 사각형 그리기
    canvas.create_rectangle(x1, y1, x2, y2, fill=color, outline="")

# —— 이벤트 루프 시작 ——
root.mainloop()
```

랜덤 사각형 10개

Problems   Output

PS C:\Users\andycho\OneDrive\Desktop\2025 2학년 1학기> & C:/Users/andycho/anaconda3/python