

```

#GUI_socket_client.py
# 섭씨 온도를 보내고 화씨 온도를 받아 표시하는 GUI 클라이언트 프로그램

from tkinter import *
from socket import *
import threading

#섭씨 온도를 서버로 전송하는 callback 함수
def calculate():
    global temp

    #섭씨 온도를 읽는다
    temp = float(entry1.get())
    #섭씨 온도를 서버로 전송
    sock.send(str(temp).encode()) #문자열로 변환하여 전송

#Thread handler
def handler(sock):
    #화씨 온도를 받아 화씨 온도 창에 표시한다
    #반드시 무한루프로 실행해야 계속 동작할 수 있다
    while True:
        #데이터를 수신하여 화씨 온도 창에 표시
        try: #수신 데이터가 없으면 예외 발생
            r_msg = sock.recv(1024) #메시지 수신
        except: #수신 데이터 없음
            pass
        else: #수신 데이터 표시
            #화씨 온도 창을 지운다
            entry2.delete(0, END)
            #새로운 화씨 온도를 표시한다
            entry2.insert(0, r_msg.decode())
            #섭씨 온도 창을 지운다
            entry1.delete(0, END)

sock = socket(AF_INET, SOCK_STREAM)
sock.connect(("localhost", 2500))

root = Tk()
#섭씨 온도 라벨과 입력 창
message_label = Label(text='Enter a temperature(C) ',font=('Verdana', 16))
entry1 = Entry(font=('Verdana', 16), width=5)

```

#화씨 온도 라벨과 입력 창

```
recv_label = Label(text='Temperature in F ',font=('Verdana', 16))
```

```
entry2 = Entry(font=('Verdana', 16), width=5)
```

#전송 버튼

```
calc_button = Button(text='전송', font=('Verdana', 12), command=calculate)
```

#위젯 배치

```
message_label.grid(row=0, column=0, sticky=W)
```

```
recv_label.grid(row=1, column=0, sticky=W)
```

```
entry1.grid(row=0, column=1)
```

```
entry2.grid(row=1, column=1)
```

```
calc_button.grid(row=0, column=2, padx=10, pady=10)
```

#데이터 수신을 위한 스레드 생성과 실행

```
cThread = threading.Thread(target=handler, args=(sock,))
```

```
cThread.daemon = True
```

```
cThread.start()
```

```
mainloop() # 항상 마지막
```