

Programming in Python

09 – Tk 라이브러리를 이용한 UI 구성



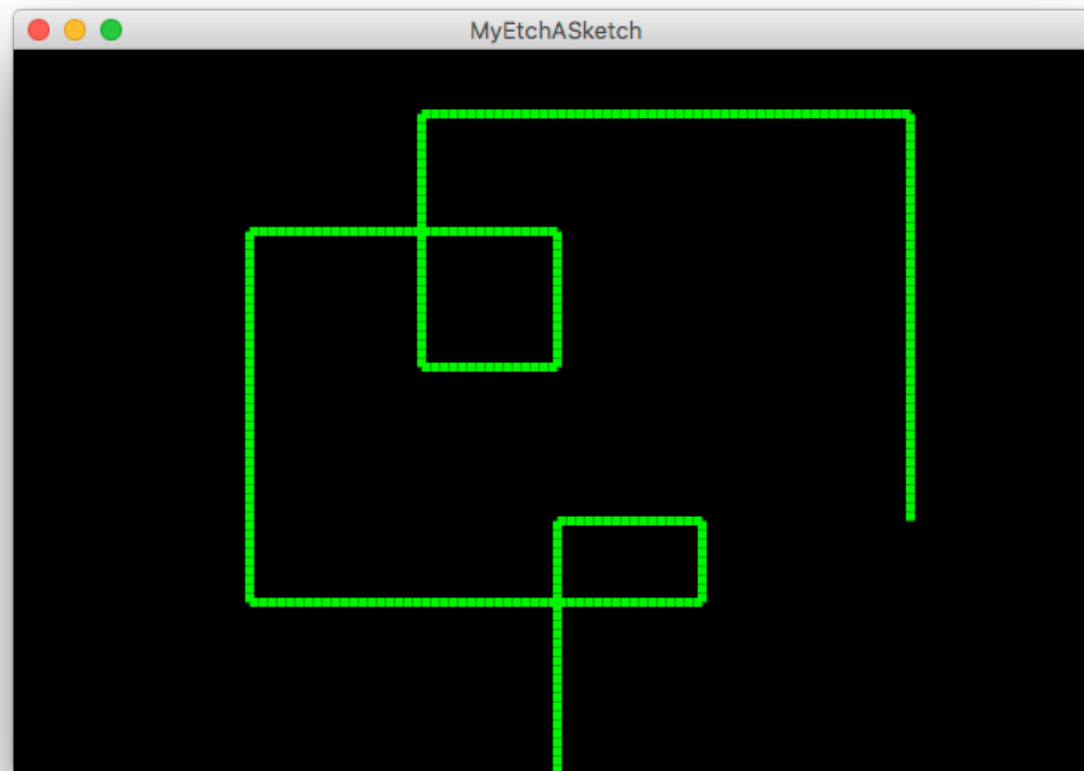
2016년 8월, 국민대학교 컴퓨터공학부

UI (User Interface) 의 구현

- Python 시스템 자체에는 그래픽 사용자 인터페이스에 관련한 내용이 없음
- 외부 라이브러리를 이용하여 윈도우 시스템을 포함하는 GUI (graphical user interface) 의 구성이 가능
- 이 장에서는 Tk 위젯 라이브러리 (많은 시스템에서 오래전부터 이용되어 오고 있음) 를 이용
 - [https://en.wikipedia.org/wiki/Tk_\(software\)](https://en.wikipedia.org/wiki/Tk_(software))
- Python 에서는 Tkinter 라는 이름으로 Tk 인터페이스를 포함하는 라이브러리를 제공
 - <https://wiki.python.org/moin/TkInter>

MyEtchASketch

- 키보드의 화살표 키를 눌러 위, 아래, 왼쪽, 오른쪽으로 이동하며 선을 그린다.
- 키보드의 “u” 키를 누르면 전체 화면이 지워진다.



Tk 라이브러리의 이용

Python 3.x

```
from tkinter import *
```

Python 2.x

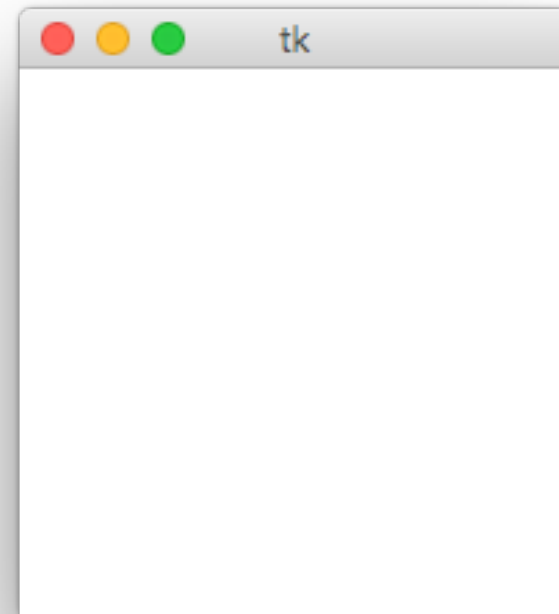
```
from Tkinter import *
```

윈도우의 생성

```
>>> from tkinter import *
```

```
>>> window = Tk()
```

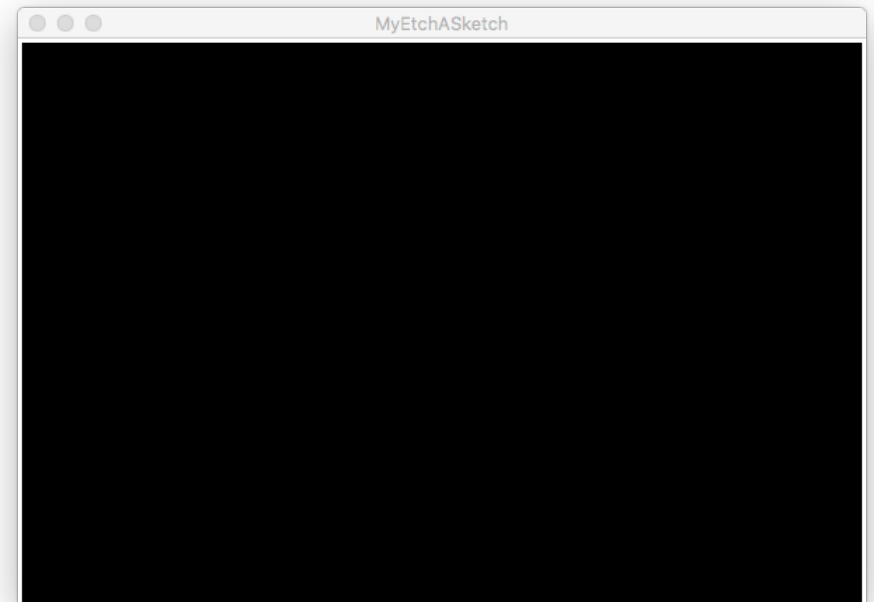
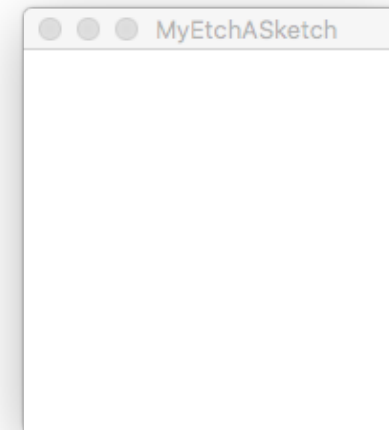
```
>>> ctrl-D (ctrl-Z)
```



캔버스 (Canvas)

```
>>> window.title("MyEtchASketch")  
"
```

```
>>> canvas = Canvas(bg="black", height=400, width=600)  
>>> canvas.pack()
```



선 그리기 (create_line) 와 지우기 (delete)

```
>>> canvas.create_line(200, 100, 500, 300, width=5, fill="green")
```

1

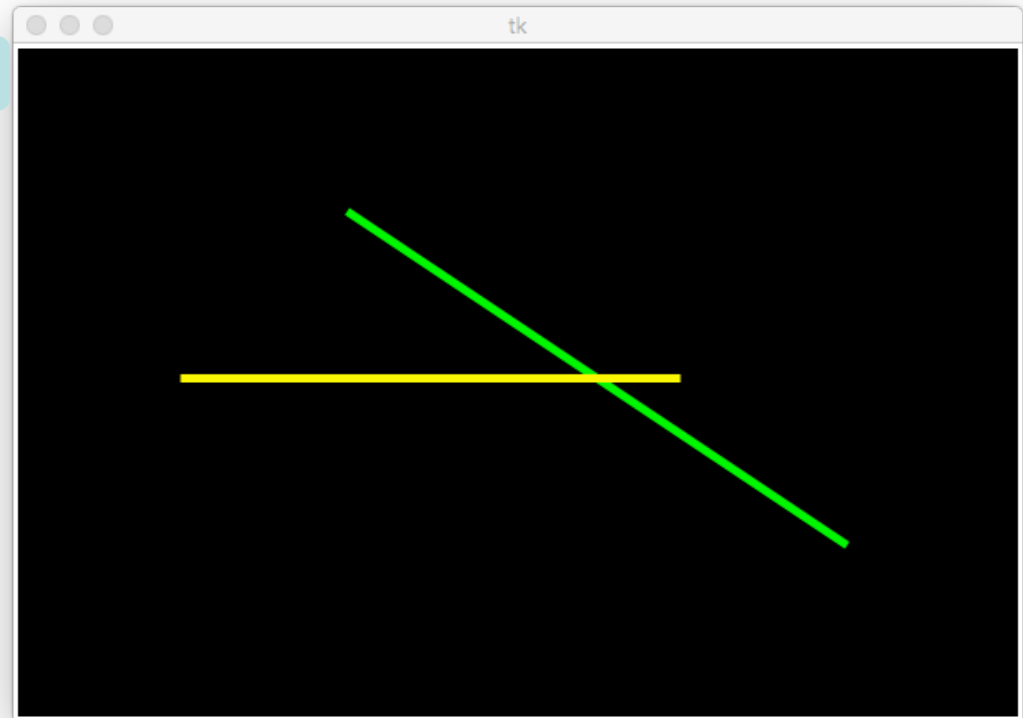
```
>>> canvas.create_line(100, 200, 400, 200, width=5, fill="yellow")
```

2

```
>>> canvas.delete(1)
```

```
>>> canvas.delete(ALL)
```

(0, 0)



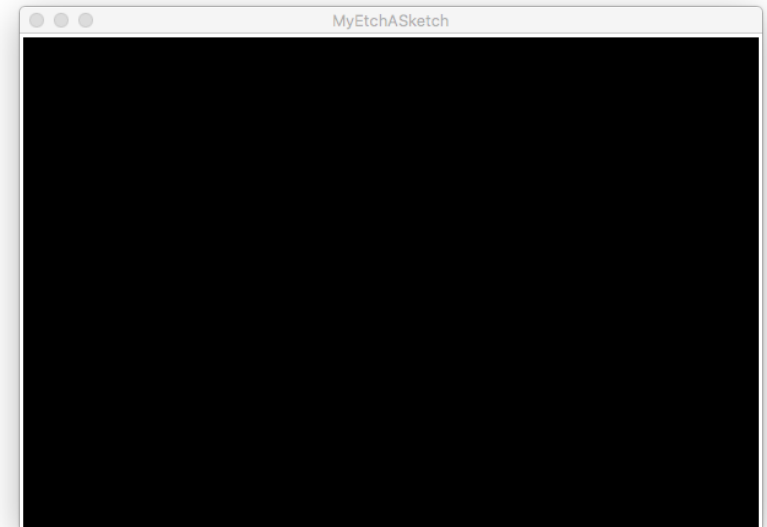
(width, height)

키보드 입력 처리하기 (bind)

```
>>> def line(event):  
...     type(event)  
...     print(event)  
...     canvas.create_line(100, 200, 400, 200, width=5, fill="yellow")  
...  
>>> window.bind("a", line)  
'4330924304line'
```

```
>>> def erase(event):  
...     canvas.delete(ALL)  
...  
>>> window.bind("u", erase)  
'4331198976erase'
```

위쪽 방향키: "<Up>"
아래쪽 방향키: "<Down>"
왼쪽 방향키: "<Left>"
오른쪽 방향키: "<Right>"



프로그래밍

```
from tkinter import *

canvas_height = 400
canvas_width = 600
canvas_colour = "black"

p1_x = canvas_width / 2
p1_y = canvas_height
p1_colour = "green"
line_width = 5
line_length = 5

def p1_move_N(event):
    pass

def p1_move_S(event):
    pass

def p1_move_E(event):
    pass

def p1_move_W(event):
    pass

def erase_all(event):
    pass
```

myEtchASketch.py

```
#### main

window = Tk()
window.title("MyEtchASketch")
canvas = Canvas(bg=canvas_colour, height=canvas_height,
                width=canvas_width, highlightthickness=0)
canvas.pack()

# key map
window.bind("<Up>", p1_move_N)
window.bind("<Down>", p1_move_S)
window.bind("<Left>", p1_move_W)
window.bind("<Right>", p1_move_E)
window.bind("u", erase_all)

window.mainloop()
```


이벤트 핸들러 함수의 구현

```
def p1_move_N(event):  
    global p1_y  
    canvas.create_line(p1_x, p1_y, p1_x, (p1_y - line_length), width=line_width, fill=p1_colour)  
    p1_y = p1_y - line_length  
  
def p1_move_S(event):  
    global p1_y  
    canvas.create_line(p1_x, p1_y, p1_x, (p1_y + line_length), width=line_width, fill=p1_colour)  
    p1_y = p1_y + line_length  
  
def p1_move_E(event):  
    global p1_x  
    canvas.create_line(p1_x, p1_y, (p1_x + line_length), p1_y, width=line_width, fill=p1_colour)  
    p1_x = p1_x + line_length  
  
def p1_move_W(event):  
    global p1_x  
    canvas.create_line(p1_x, p1_y, (p1_x - line_length), p1_y, width=line_width, fill=p1_colour)  
    p1_x = p1_x - line_length  
  
def erase_all(event):  
    canvas.delete(ALL)
```

코드의 개선 - Exercise

p1_move(x, y) 라는 함수를 새로 만들어, 아래와 같이 이용하여 코드의 불필요한 반복을 피한다.

```
def p1_move_N(event):  
    p1_move(0, -line_length)
```

Exercise - 정답

```
def p1_move_N(event):  
    p1_move(0, -line_length)
```

```
def p1_move_S(event):  
    p1_move(0, line_length)
```

```
def p1_move_E(event):  
    p1_move(line_length, 0)
```

```
def p1_move_W(event):  
    p1_move(-line_length, 0)
```

```
def p1_move(x, y):  
    global p1_x  
    global p1_y  
    canvas.create_line(p1_x, p1_y, p1_x + x, p1_y + y, width=line_width, fill=p1_colour)  
    p1_x += x  
    p1_y += y
```

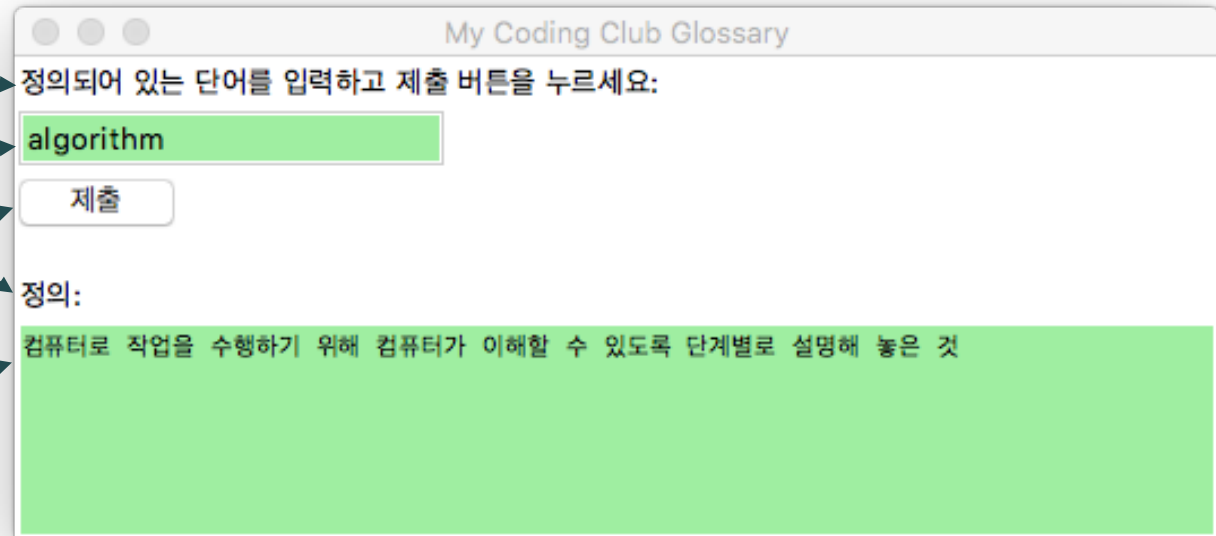
버튼과 텍스트 입출력을 포함하는 GUI

레이블 (Label)

엔트리 (Entry)

버튼 (Button)

텍스트 (Text)



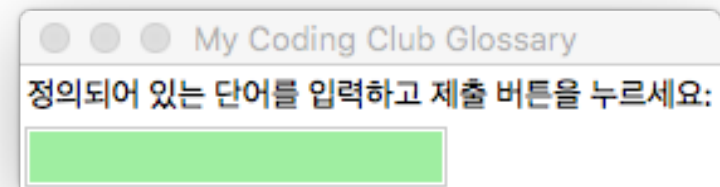
레이블과 엔트리 생성

```
>>> from Tkinter import *
>>> window = Tk()
>>> window.title("My Coding Club Glossary")
""
>>> Label(window, text="정의되어 있는 단어를 입력하고 제출 버튼을 누르세요:") \
    .grid(row=0, column=0, sticky=W)
>>> entry = Entry(window, width=20, bg="light green")
>>> entry.grid(row=1, column=0, sticky=W)
```

아래와 같이 하면 안될까?

```
entry = Entry(window, width=20, bg="light green").grid(row=1, column=0, sticky=W)
```

```
>>> type(entry)
<type 'instance'>
```



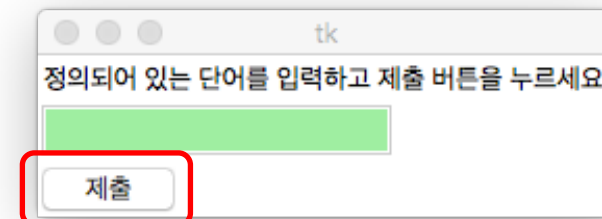
버튼 추가

```
>>> def click():  
...     pass
```

...

```
>>> Button(window, text="제출", width=5, command=click) \  
...     .grid(row=2, column=0, sticky=W)
```

이 부분이 없었다면?

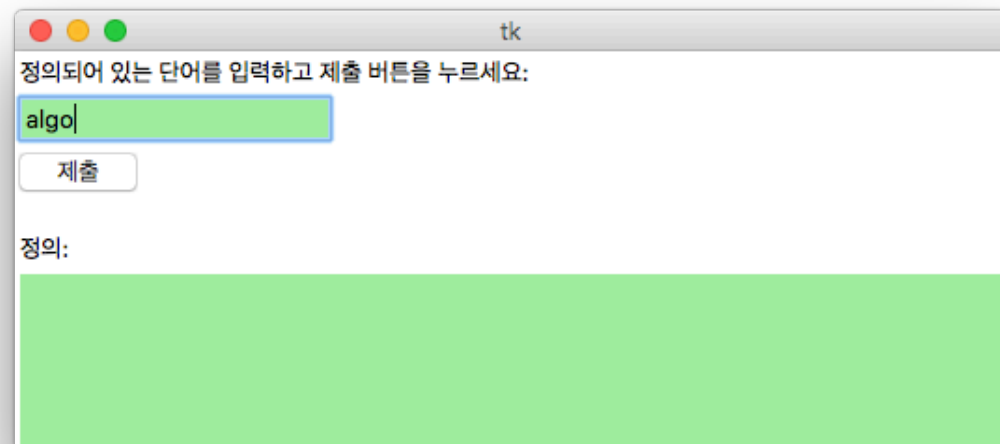


텍스트 상자 추가

```
>>> Label(window, text="\n정의:").grid(row=3, column=0, sticky=W)
```

```
>>> output = Text(window, width=75, height=6, wrap=WORD, background="light green")
```

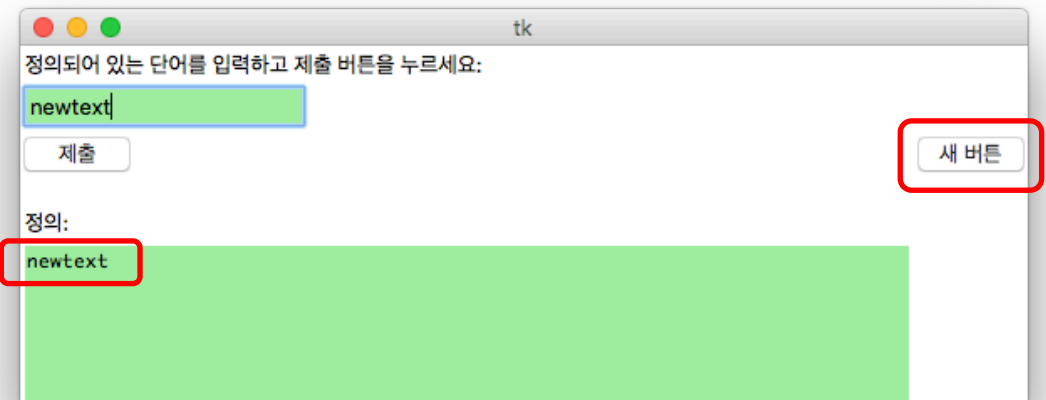
```
>>> output.grid(row=4, column=0, sticky=W)
```



버튼 커맨드 테스트

```
>>> def newclick():  
...     entered_text = entry.get()  
...     output.delete(0.0, END)  
...     output.insert(END, entered_text)  
...  
>>> b = Button(window, text="새 버튼", width=5, command=newclick)  
>>> b.grid(row=2, column=1, sticky=E)
```

- Entry.get() - 텍스트 엔트리 위젯으로부터 입력한 텍스트를 수집
- Text.delete() - 줄번호.문자위치로부터 줄번호.문자위치까지를 삭제
- Text.insert() - 줄번호.문자위치에 새로운 텍스트를 삽입

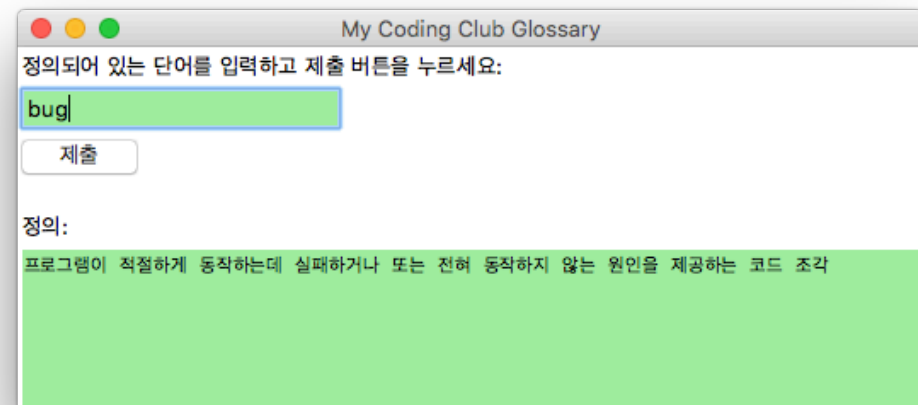


사전의 정의와 프로그래밍 - Exercise

```
my_glossary = {  
    'algorithm': '컴퓨터로 작업을 수행하기 위해 컴퓨터가 이해할 수 있도록 단계별로 설명해 놓은 것',  
    'bug': '프로그램이 적절하게 동작하는데 실패하거나 또는 전혀 동작하지 않는 원인을 제공하는 코드 조각',  
    'function': '재사용할 수 있는 코드 조각',  
}
```

myGlossary.py

```
# -*- coding: utf-8 -*-  
  
from tkinter import *  
...  
def click():  
    pass  
  
...  
window.mainloop()
```

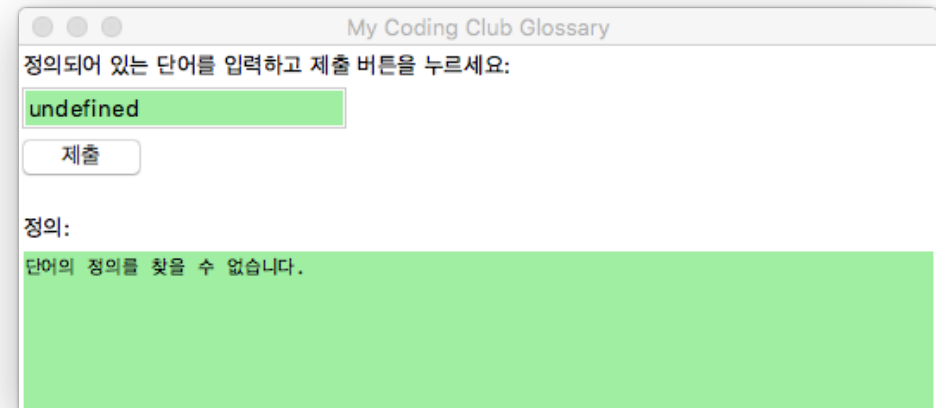


Exercise - 정답

```
def click():  
    entered_text = entry.get()  
    output.delete(0.0, END)  
    definition = my_glossary[entered_text]  
    output.insert(END, definition)
```

오른쪽 그림에서처럼,
정의되지 않은 단어가 입력되면
특정한 안내문을 출력하도록 프로그래밍 하려면?

1. try ... except
2. if
3. get



발전 과제

- myEtchASketch.py 를 수정하여
 - 특정 키를 누르면 그려지는 라인의 색상을 변경하도록 해 본다.
- myGlossary.py 를 수정하여
 - 4지선다형 문제가 출제되고, 그것을 맞추는 게임을 만들어 본다.