37장

그래픽 사용자 인터페이스를 위한 라이브러리

37장 그래픽 사용자 인터페이스를 위한 라이브러리

37.1 그래픽 사용자 인터페이스를 위한 라이브러리

37.1 그래픽 사용자 인터페이스를 위한 라이브러리

그래픽 사용자 인터페이스(Graphical User Interface, GUI) 라이브러리 – 사용자와 운영 체제가 서로그래픽을 사용하는 위젯(widget)이라는 제어 요소를 통해 상호작용할 수 있게 하는 메서드와 클래스를 제공한다.

- 위젯 – 버튼, 스크롤바, 메뉴, 창, 그리기 캔버스, 프로그래스바, 대화 창 등이 있으며, 위젯의 목표는 더 나은 사용자 경험을 제공하는 것이다.
- tkinter – 표준 GUI 라이브러리. https://docs.python.org/3.7/library/tkinter.html

GUI 애플리케이션 개발 세 단계

- 보여줄 창의 크기, 위치, 제목을 결정해 설정하기
- 화면을 구성할 '요소'인 메뉴, 버튼 등의 위젯을 창에 추가하기
- 버튼 클릭, 메뉴 아이템 선택 등의 이벤트에 대해 각 위젯이 어떤 동작을 할지 선택하기. 이벤트 핸들러(event handler)를 작성해서 위젯의 동작을 구현할 수 있다. 이벤트 핸들러는 함수 형태로, 사용자가 특정 위젯과 상호작용할 때 프로그램이 어떻게 동작할지 기술한다.

37.2 tkinter 라이브러리를 사용해 프로그램 설정하기

37.2 tkinter 라이브러리를 사용해 프로그램 설정하기

```
import tkinter ---- tkinter 라이브러리를 불러옴

window = tkinter.Tk() ---- 새 객체를 만들고 window라는 변수에 대입
window.geometry("800x200") ---- 창 크기 변경
window.title("첫 번째 GUI") ---- 창에 제목 추가
window.configure(background="grey") ---- 창 배경색 변경
window.mainloop() ---- 프로그램 시작
```

코드 37-1 창 만들기

» 셀프 체크 37.2

37.3 위젯 추가하기

37.3 위젯 추가하기

위젯을 추가하려면 코드가 두 줄 필요하다.

- 한 줄은 위젯 객체를 만들고, 다른 한 줄은 창에서 그 객체가 차지할 위치를 지정한다.
- 첫 번째 줄은 창(window)을 부모로 하는 버튼을 만들어서 btn이라는 변수와 연관시킨다. 두 번째 줄은 창에서 버튼의 위치를 지정한다(이때 pack을 사용하는데, 아무 인자도 넘기지 않으면 창의 맨 위에서 아래 방향으로 하나씩 위젯을 배치한다).

```
btn = tkinter.Button(window)
btn.pack()
```

37.3 위젯 추가하기

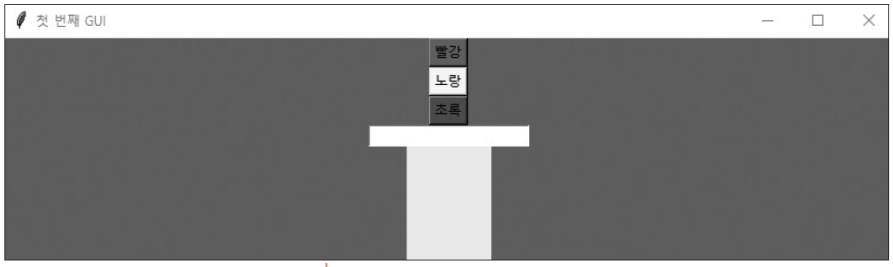

```
import tkinter

window = tkinter.Tk()
window.geometry("800x200")
window.title("첫 번째 GUI")
window.configure(background="grey")

red = tkinter.Button(window, text="빨강", bg="red")
red.pack()

yellow = tkinter.Button(window, text="노랑", bg="yellow")
yellow.pack()

green = tkinter.Button(window, text="초록", bg="green")
green.pack()

textbox = tkinter.Entry(window)
textbox.pack()

colorlabel = tkinter.Label(window, height="10", width="10")
colorlabel.pack()

window.mainloop()
```

---- window를 부모로 빨간 배경에 '빨강'이라는 글씨가 적힌 버튼을 새로 만듦

---- 방금 정한 특성을 지닌 버튼을 창에 추가

---- 텍스트를 입력할 수 있는 박스를 만듦

---- 높이가 10인 레이블을 만듦



코드 37-2
창에 위젯 추가하기

37.3 위젯 추가하기

표준 tkinter 라이브러리가 제공하는 위젯들

위젯 이름	설명	위젯 이름	설명
Button	버튼	Menubutton	메뉴를 보여줌
Canvas	모양을 그림	OptionMenu	팝업 메뉴를 보여줌
Checkbutton	체크할 수 있는 옵션이 표시된 버튼(하나 이상 선택 가능)	PanedWindow	크기를 변경할 수 있는 창
Entry	사용자가 텍스트를 입력할 수 있는 텍스트 필드	Radiobutton	옵션을 라디오 버튼 형태로 표시하는 버튼(여러 옵션 중에 한 가지만 선택 가능)
Frame	안에 다른 위젯이 들어갈 수 있는 컨테이너	Scale	슬라이더(긴 막대에 범위와 눈금을 표시하고, 움직일 수 있는 작은 막대로 범위 내의 값 중 하나를 선택함)
Label	텍스트나 이미지를 보여줌	Scrollbar	다른 위젯에 스크롤바를 추가함
LabelFrame	여백을 추가하는 컨테이너	Spinbox	Entry와 비슷하지만, 미리 정해진 텍스트 중에 한 가지를 선택할 수만 있음
Listbox	목록을 보여줌	Text	여러 줄로 나뉜 텍스트를 보여줌
Menu	메뉴 커맨드를 보여줌(Menubutton 내부에 들어감)	TopLevel	다른 창을 표시하기 위한 컨테이너

셀프 체크 37.3

37.4 이벤트 핸들러 추가하기

37.4 이벤트 핸들러 추가하기

사용자가 위젯과 상호작용하면 프로그램이 어떤 작업을 수행할지 지정하는 코드를 작성하자.

- 사용자가 마우스를 움직이거나, 마우스 버튼을 눌렀다 떼거나, 키보드의 키를 누르고 뗄 때마다 이벤트(event)가 발생한다.
- GUI 프로그램은 이런 이벤트를 처리하는 함수를 만들어서 사용자와의 상호작용을 구현한다.

위젯마다 따로 이벤트 핸들러를 지정할 수 있다.

- tkinter의 경우 위젯마다 기본 이벤트 처리 방식을 따로 제공하며, 위젯마다 다른 이름으로 기본 동작을 부른다.
- 기본적으로는 위젯을 생성할 때 파라미터 이름을 지정해 처리 방식을 지정할 수 있다.
- 하지만 변경이 필요한 경우 widget.configure() 함수를 통해 처리 방식을 변경할 수 있다.

37.4 이벤트 핸들러 추가하기

» 버튼 위젯의 경우 버튼 클릭 시 실행할 함수를 커맨드(command)라고 부른다. 버튼을 생성하면서 커맨드를 지정할 수 있다.

```
import tkinter

def change_color(): ---- 발생할 수 있는 이벤트를 표현하는 함수
    window.configure(background="white") ---- 함수가 창의 배경색을 바꿈

window = tkinter.Tk()
window.geometry("800x200")
window.title("첫 번째 GUI")
window.configure(background="grey")

white = tkinter.Button(window, text="누르세요", command=change_color)
white.pack()

window.mainloop()
```

어떤 동작과 연결된 버튼. command에 함수 이름을 지정해서 동작과 연결함

코드 37-3
버튼 클릭 이벤트 핸들러

37.4 이벤트 핸들러 추가하기

» 사용자가 텍스트 박스에 입력한 수를 읽어서 그 숫자부터 0까지 카운트다운을 수행하는 프로그램

```
import tkinter
import time

def countdown(): ---- 이벤트 핸들러 함수
    countlabel.configure(background="white") ---- 레이블의 색을 흰색으로 바꿈
    howlong = int(textbox.get()) ---- 텍스트 박스의 값을 가져와서 int로 변환
    for i in range(howlong,0,-1): ---- 텍스트 박스에 입력한 수부터 0까지 내려가는 루프 시작
        countlabel.configure(text=i) ---- 레이블의 텍스트를 루프 변수의 값으로 설정
        window.update() ---- 창을 갱신해서 변경한 레이블 값을 표시하게 만듦
        time.sleep(1) ---- time 라이브러리를 사용해 1초간 일시 중단
    countlabel.configure(text="끝!") ---- 값이 0이 되면 레이블 텍스트를 변경

window = tkinter.Tk()
window.geometry("800x600")
window.title("첫 번째 GUI")
window.configure(background="grey")
```

37.4 이벤트 핸들러 추가하기

```
lbl = tkinter.Label(window, text="몇 초를 카운트다운 하겠습니까?") ---- 사용자에게 사용법을 알려주기 위한 레이블
lbl.pack()
textbox = tkinter.Entry(window) ---- 사용자에게서 수를 입력받기 위한 텍스트 박스
textbox.pack()
count = tkinter.Button(window, text="카운트다운!", command=countdown) ----
count.pack()                                                          카운트다운을 시작하기 위한 버튼
                                                                      소스코드 맨 앞에서 정의한 함수를
                                                                      버튼 command에 연결함
countlabel = tkinter.Label(window, height="10", width="10") ---- 카운트다운이 되는 동안 변하는 숫자를 보여 주는 레이블
countlabel.pack()

window.mainloop()
```

코드 37-4
텍스트 박스에서 값을 읽어서 그 값부터 시작하는 카운트다운을 수행하는 프로그램

37.5 요약

37.5 요약

- GUI 프로그램은 창 안에서 작업이 이뤄진다.
- 창 안에 위젯이라고 불리는 GUI 요소를 추가할 수 있다.
- 사용자가 위젯과 상호작용할 때 어떤 작업을 수행할지 정의하는 함수를 추가할 수 있다.

37.5 요약

» (Q37.1) 사람들의 이름, 전화번호, 이메일 주소를 주소록에 저장하는 프로그램을 작성하라.창에는 이름, 전화번호, 이메일을 입력 받는 텍스트 박스 세 개, 주소를 추가하는 버튼, 모든 주소를 보는 버튼, 마지막으로 레이블이 있어야 한다. 사용자가 추가 버튼을 클릭하면 프로그램은 세가지 텍스트 박스에서 값을 읽어와서 주소록에 정보를 추가한다. 사용자가 보기 버튼을 클릭하면 프로그램은 주소록에 저장돼 있는 모든 정보를 읽어서 레이블에 표시한다.