제10강 Tkinter

학습 목차

- Python GUI
- Tkinter
  - Window
  - Label
  - Button
  - Entry
  - Text
  - Checkbox
  - Radiobutton
  - Listbox
  - Combobox

Python GUI Options

- **Tkinter**
  - 파이썬 내장 표준 GUI. IDLE 이 Tkinter 로 제작됨.
  - 간단한 application 을 만들기에 적합.

- **PyQt5**
  - 가장 최신 기능을 탑재.
  - 상용 application 개발에 적합.

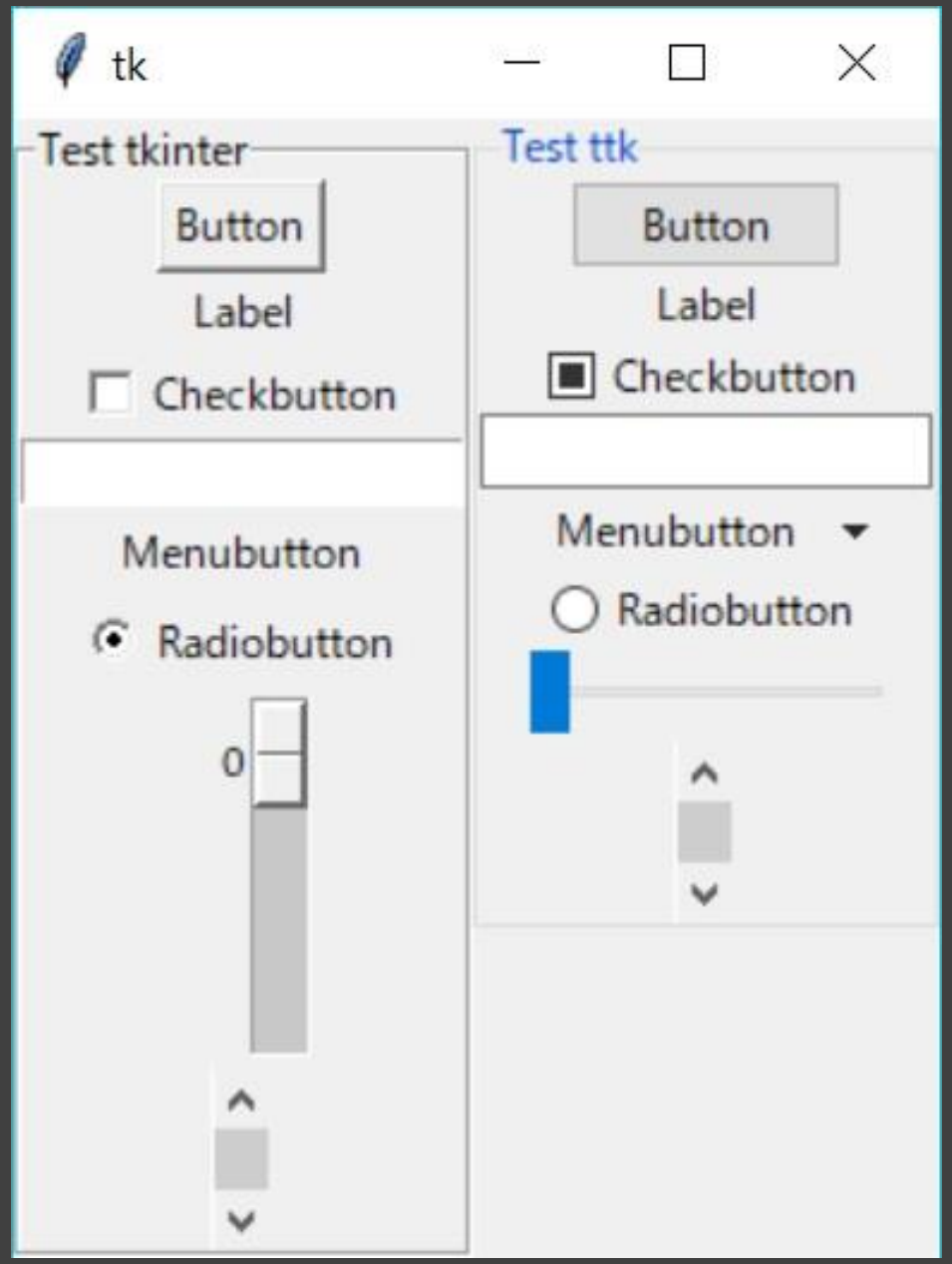
- **Kivy**
  - 모바일 application 에 특화.

- **WxPython**

Tkinter

- tkinter 모듈
  - 기본 모듈

- tkinter.ttk 모듈
  - 조금 더 “이쁘다” ?
  - 스타일링이 가능
  - 기본 모듈 위젯들을 덮어씀 - 일부는 사용 방법이 호환되지 않음.



기본 구조

- 윈도우 만들고,
- Widget 생성.
- 위젯 배치
  - pack(), place(), grid() 등을 사용

pack 파라미터

이름	의미	기본값	속성
side	특정 위치로 공간 할당	top	top, bottom, left, right
anchor	할당된 공간 내에서 위치 지정	center	center, n, e, s, w, ne, nw, se, sw
fill	할당된 공간에 대한 크기 맞춤	none	none, x, y, both
expand	미사용 공간 확보	False	Boolean
ipadx	위젯에 대한 x 방향 내부 패딩	0	상수
ipady	위젯에 대한 y 방향 내부 패딩	0	상수
padx	위젯에 대한 x 방향 외부 패딩	0	상수
pady	위젯에 대한 y 방향 외부 패딩	0	상수

https://076923.github.io/posts/Python-tkinter-10/

로깅 준비

```
import logging
logging.basicConfig(
    level=logging.INFO,
    format='%(filename)s(%(lineno)d): %(asctime)s - %(levelname)s - %(message)s'
)

from logging import info as info, debug as debug, warning as warning
```

Main Window

```
from tkinter import *
from tkinter.ttk import *


window = Tk()
window.title("My Tkinter App")
window.geometry("800x800+10+10")
window.resizable(False, False)

# window.geometry("+%d+%d" % (window.winfo_screenwidth()/2, window.winfo_screenheight()/2))
# window.resizable(False, False)


window.mainloop()
```

bind

- 위젯에 대한 이벤트 발생을 처리

```
def stop(event=None):
    window.quit()

window.bind("<Escape>", stop)
```

참고: https://docs.python.org/3/library/tkinter.html#bindings-and-events
https://python-course.eu/tkinter/events-and-binds-in-tkinter.php
https://web.archive.org/web/20190515021108id_/http://infohost.nmt.edu/tcc/help/pubs/tkinter/web/key-names.html

Label

- 텍스트 출력

```
label = Label(master=window, text='Hello, Tkinter~', font=('Segoe UI', 10))
label.pack()
# label.place(x=100, y=100)
```

Button

```
def rotate():
    text = label.cget('text')
    text = text[1:] + text[0]
    label.config(text=text)

button = Button(master=window, text='Rotate', command=rotate)
button.pack()
```

Entry

```
def update_text(event=None):
    label.config(text=entry.get())

entry = Entry(master=window, font=('Segoe UI', 10))
entry.bind('<Return>', update_text)
entry.pack()
```

Text

- 라인 기반 처리
- 위치: “라인.컬럼” “라인.컬럼 + n chars”
  - “3.0” – 3번째 라인의 첫번째 칼럼.
  - “4.1+4chars” – 4번째 라인 두번째 칼럼에서 4글자 뒤의 위치
  - END: 마지막 끝 위치.

```
from tkinter.scrolledtext import ScrolledText

wiki_python = '''Python is a high-level.....'''*2

text = ScrolledText(master=window, font=('Segoe UI', 10))
text.delete("1.0", END)
text.insert(END, wiki_python)
text.pack()
```

Text

- 일부분에 대한 tag 기능
  - tag_config, tag_remove

```
text.tag_config('found', background='yellow', foreground='red')
text.tag_add('found', '3.0', '3.0+16chars')
```

Checkbutton

```
def cb_clicked():
    info(f'checkbutton updated {checked.get()}')

checked = BooleanVar(value=True)
checkbutton = Checkbutton(master=window, text='CheckButton', command=cb_clicked, variable=checked)
checkbutton.pack()
```

Radiobutton

```
def rb_clicked():
    info(f'checkbutton updated {color.get()}')

color = StringVar(value='red')
Radiobutton(master=window, text='RED', value='red', command=rb_clicked, variable=color).pack()
Radiobutton(master=window, text='GREEN', value='green', command=rb_clicked, variable=color).pack()
Radiobutton(master=window, text='BLUE', value='blue', command=rb_clicked, variable=color).pack()
```

Combobox vs Listbox

- **Listbox**
  - 여러 개의 값을 보여줌.
  - 여러 아이템을 동시에 선택할 수 있음.

- **Combobox**
  - 하나의 값만을 보여줌.
  - 여러 아이템을 동시에 선택할 수 없음.
  - 새로운 아이템을 직접 추가할 수 있음.

Listbox

```
def item_selected(event=None):
    # curselection() is tuple
    info(f'{pattern_listbox.curselection()} is selected')
    for index in pattern_listbox.curselection():
        info(f'{pattern_listbox.get(index)}')

selected = StringVar(value='Hello')
# selecte mode - SINGLE, BROWSE, MULTIPLE
pattern_listbox = Listbox(master=window, selectmode=SINGLE, height=3, listvariable=selected)
pattern_listbox.insert(1, 'Python')
pattern_listbox.insert(2, r'\d\d')
pattern_listbox.insert(END, '[a-z]+')
pattern_listbox.insert(END, '[0-9]+')
pattern_listbox.select_set(1)
pattern_listbox.pack()

pattern_listbox.bind('<<ListboxSelect>>', item_selected)
```

Combobox

```
def combo_selected(event=None):
    info(f'combo item {combo.get()} is selected')

combo_var = StringVar(value='USA')
combo = Combobox(master=window, values=['korea', 'japan', 'china'], textvariable=combo_var)
combo.set('russia')
combo.current(0)
combo.pack()

combo.bind('<<ComboboxSelected>>', combo_selected)
```

Combobox – 새 아이템 추가

```
def enter_combo_item(event=None):
    combo['values'] += (combo.get(), )

combo.bind('<Return>', enter_combo_item)
```

Combobox - readonly

```
# combo box readonly
combo.configure(state='readonly')
```

Frame

```
top_frame = Frame(master=window)
top_frame.pack()
label = Label(master=top_frame, text='Hello, Tkinter~'.center(50, '*'), width=50, font=('Segoe UI', 10))
label.pack(side=LEFT, fill=BOTH, expand=True)
# label.place(x=100, y=100)

def rotate():
    text = label.cget('text')
    text = text[1:] + text[0]
    label.config(text=text)

button = Button(master=top_frame, text='Rotate', width=30, command=rotate)
button.pack(side=LEFT, fill=BOTH, expand=TRUE)
```

LabelFrame

```
option_frame = LabelFrame(text='Options')
option_frame.pack(side=TOP)


def cb_clicked():
    info(f'checkbutton updated {checked.get()}')

checked = BooleanVar(value=True)
checkbutton = Checkbutton(master=option_frame, text='CheckButton', command=cb_clicked, variable=checked, width=30)
checkbutton.pack(side=LEFT)



def rb_clicked():
    info(f'checkbutton updated {color.get()}')

color = StringVar(value='red')
Radiobutton(master=option_frame, text='RED', value='red', command=rb_clicked, variable=color, width=10).pack(side=LEFT)
Radiobutton(master=option_frame, text='GREEN', value='green', command=rb_clicked, variable=color, width=10).pack(side=LEFT)
Radiobutton(master=option_frame, text='BLUE', value='blue', command=rb_clicked, variable=color, width=10).pack(side=LEFT)
```

```
from tkinter.scrolledtext import ScrolledText

text_frame = LabelFrame(text='Text')
text_frame.pack()

wiki_python = '' ''*2
text = ScrolledText(master=text_frame, font=('Segoe UI', 10))
text.delete("1.0", END)
text.insert(END, wiki_python)
text.pack()
```

List 스크롤바

```
history_frame = LabelFrame(text='Pattern History')
history_frame.pack()

def item_selected(event=None):
    info(f'{pattern_listbox.curselection()} is selected')
    for index in pattern_listbox.curselection():
        info(f'{pattern_listbox.get(index)}')

selected = StringVar(value='Hello')
# selecte mode - SINGLE, BROWSE, MULTIPLE
pattern_listbox = Listbox(master=history_frame, selectmode=MULTIPLE, width=80, height=10, listvariable=selected)
pattern_listbox.insert(1, 'Python')
pattern_listbox.insert(2, r'\d\d')
pattern_listbox.insert(END, '[a-z]+')
pattern_listbox.insert(END, '[0-9]+')
pattern_listbox.select_set(1)
pattern_listbox.pack(side=LEFT)

pattern_listbox.bind('<<ListboxSelect>>', item_selected)

list_scroll_bar = Scrollbar(master=history_frame, command=pattern_listbox.yview)
list_scroll_bar.pack(side=RIGHT, fill=Y)
pattern_listbox.configure(yscrollcommand=list_scroll_bar.set)
```

MessageBox

```
import tkinter.messagebox as Messagebox

Messagebox.showinfo('Info', 'This is information.')
Messagebox.showwarning('Warn', 'This is warning.')
Messagebox.showerror('Error', 'This is error.')

answer = Messagebox.askokcancel('OK Cancel', 'Ok or Cancel?')
info(answer)
answer = Messagebox.askquestion('Question', 'Ok?')
info(answer)
answer = Messagebox.askretrycancel('Retry Cancel', 'Retry or Cancel?')
info(answer)
answer = Messagebox.askyesno('Yes No', 'Yes or No?')
info(answer)
answer = Messagebox.askyesnocancel('Yes No Cancel', 'Yes? No? or Cancel?')
info(answer)
```

Menu

```
menu = Menu()

menu_File = Menu(menu, tearoff=False)
menu_File.add_command(label='New', accelerator='Ctrl+N', command=create_new_file)
menu_File.add_command(label='Open', accelerator='Ctrl+O', command=open_file)
menu_File.add_command(label="Save", accelerator='Ctrl+S', state='disabled')
menu_File.add_command(label="Save as...", accelerator='Shift+S', command=save_file_as)
menu_File.add_separator()
menu_File.add_command(label='Quit', accelerator='Ctrl+Q')
menu.add_cascade(label='File', underline=True, menu=menu_File)

window.config(menu=menu)
```

Menu - RadioButton

```
menu_Colors = Menu(menu, tearoff=False)
found_color = StringVar(value='yellow')
menu_Colors.add_radiobutton(label='Yellow', value='yellow', variable=found_color)
menu_Colors.add_radiobutton(label='Green', value='green', variable=found_color)
menu.add_cascade(label="Colors", underline=1, menu=menu_Colors)


# trace variable change
def color_updated(var, index, mode):
    info(f'{var=} {index=} {mode=}')
    info(f'{found_color.get()}')

found_color.trace_add('write', color_updated)
```

bind_all

```
window.bind_all('<Control-q>', lambda e: window.quit())
```

filedialog

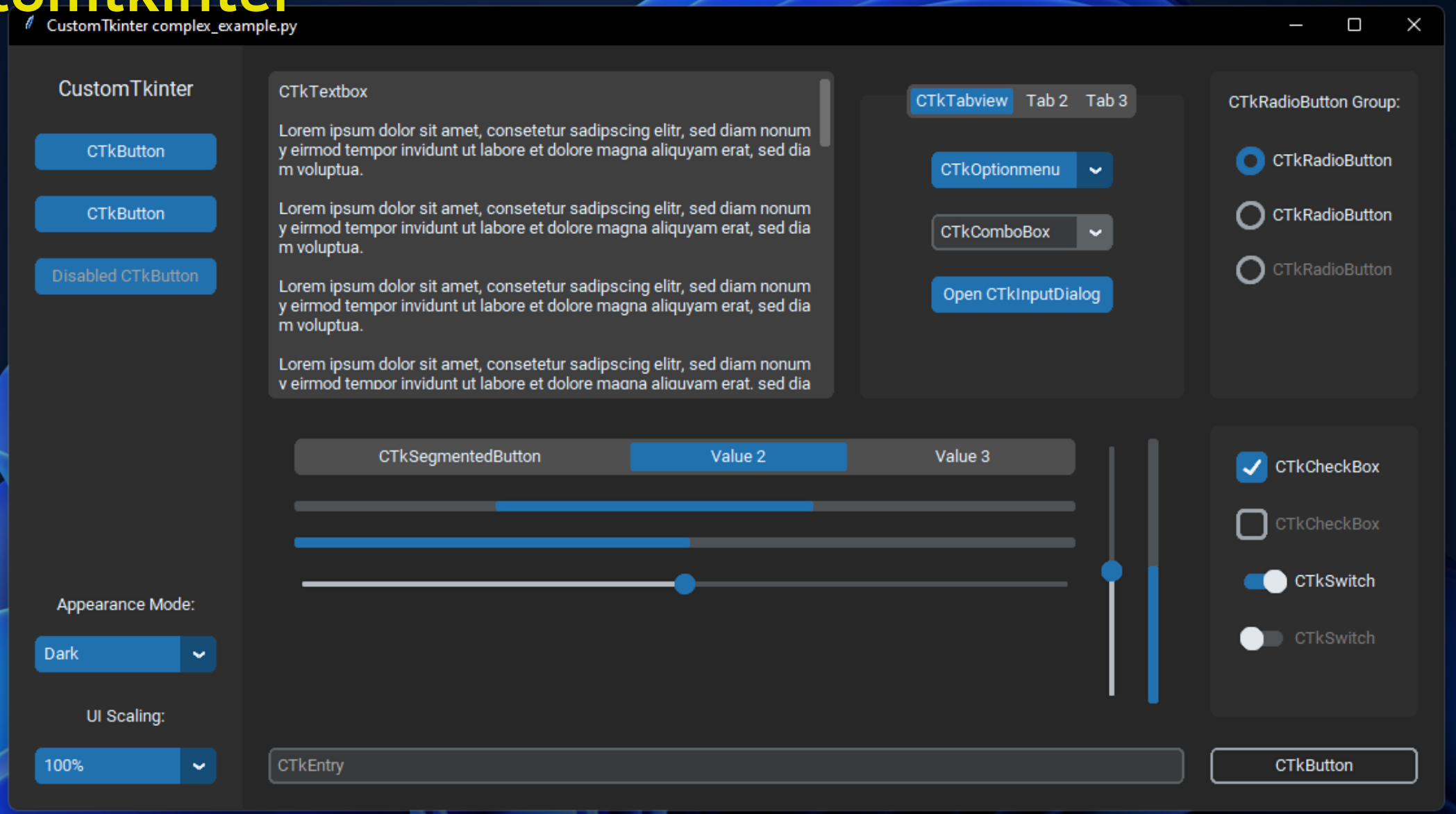
```
from tkinter import filedialog

def open_file():
    # file_name = filedialog.askopenfilename(title='Select a text file', filetypes=((".txt files", "*.txt"), ("all files", "*.*")))
    file_names = filedialog.askopenfilenames(title='Select files', filetypes=(("text files (.txt)", "*.txt"), ("all files", "*.*")))
    info(file_names)



def save_file_as():
    file_name = filedialog.asksaveasfilename(title='Save file as...', filetypes=(("text file (.txt)", "*.txt"), ("all files", "*.*")), defaultextension='txt')
    info(file_name)
```

customtkinter



참고

- 공식 Reference
  - https://docs.python.org/3/library/tk.html

- 책
  - https://tkdocs.com/book.html

- 기타
  - https://www.pythontutorial.net/tkinter/
  - https://076923.github.io/posts/Python-tkinter-1/