

제11강 Tkinter

학습 목차

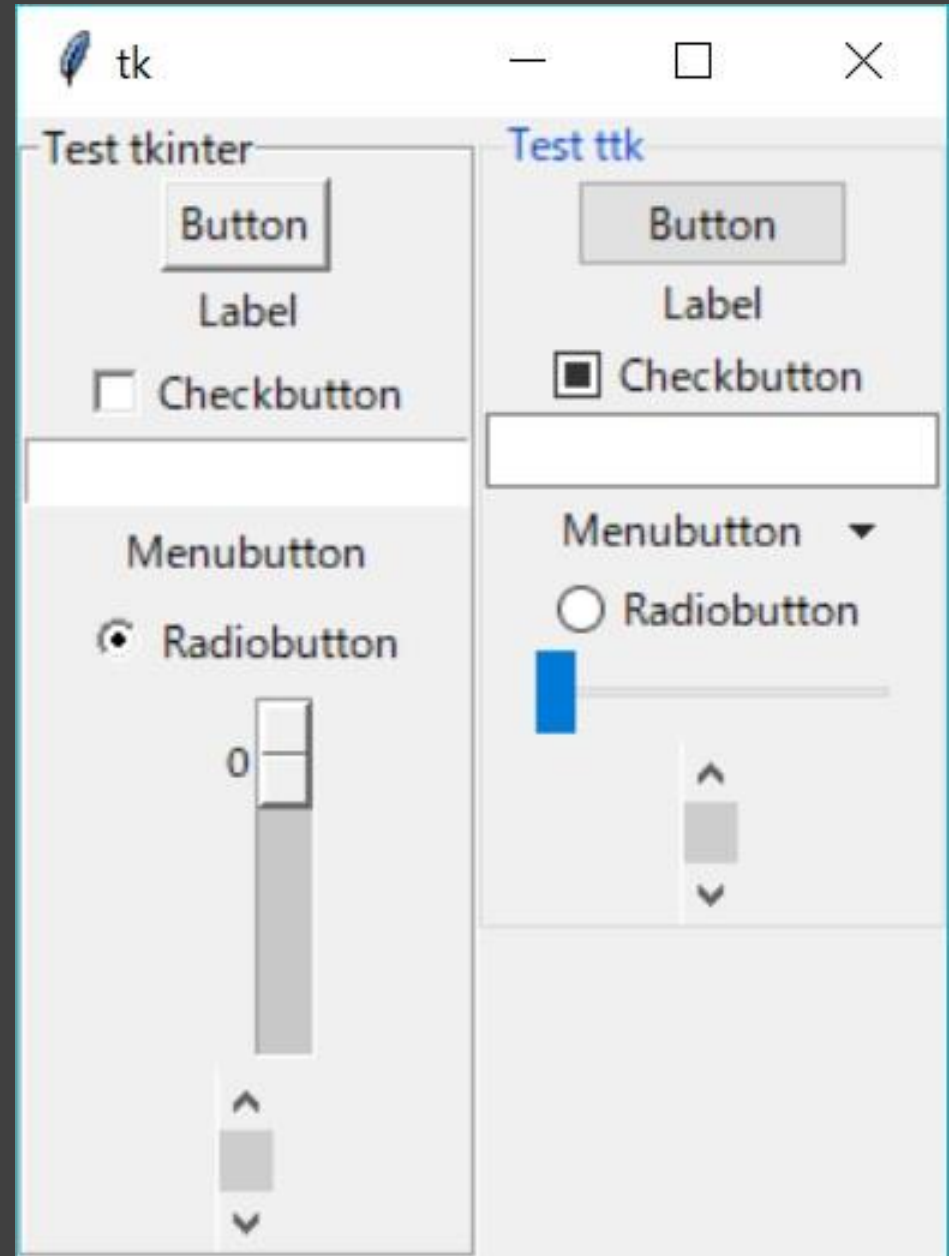
- Python GUI
- Tkinter
 - Window
 - Label
 - Button
 - Entry
 - Text
 - Checkbox
 - Radiobutton
 - Listbox
 - Combobox

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/>

Main Window

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

window = Tk()
window.title("My Tkinter App")
window.geometry("320x240+1920+800")
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(window, text="HELLO, TKINTER")  
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(window, text="Hello", command=rotate)  
button.pack()
```

Entry

```
def change_text(event=None):  
    new_text = entry.get()  
    label.config(text=new_text.center(100, ' '))
```

```
entry = Entry(width=100)  
entry.bind('<Return>', change_text)  
entry.pack(side=LEFT)
```

Frame

```
first_line_frame = Frame(window)
first_line_frame.pack()

label = Label(first_line_frame, text='hello, tkinter')
label.pack(side=LEFT)

def rotate():
    text = label.cget('text')
    text = text[1:] + text[0]
    label.config(text=text)
    # label.config(text=text.center(100, ' '))

button = Button(first_line_frame, text="Hello", command=rotate)
button.pack(side=RIGHT)
```

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
wikipedia_python = '''blarblarblar '''

# text = ScrolledText(text_frame, font=("Consolas", 10))
text = ScrolledText(text_frame, width=50, height=10, font=("Consolas", 10))
text.delete("1.0", END)
text.insert(END, wikipedia_python)
text.pack(fill=BOTH, expand=True)
```

Text

- 일부분에 대한 tag 기능
 - tag_configure, tag_remove

```
text.tag_config('found', background='yellow', foreground='red')  
text.tag_add('found', '3.0+3chars', '3.0+10chars')
```

Checkbutton

```
def cb_clicked():  
    print(f'{ignore_case.get()}')  
  
ignore_case = IntVar()  
case_checkbutton = Checkbutton(window, text='Ignore Case', command=cb_clicked, variable=ignore_case)  
ignore_case.set(0)
```

Radiobutton

```
found_color = StringVar(value='yellow')  
# found_color.set('yellow')  
Radiobutton(window, text="Green", value="green", variable=found_color).pack()  
Radiobutton(window, text="Yellow", value="yellow", variable=found_color).pack()
```


Variable callback

```
def gender_updated(var, index, mode):  
    print(f'{var=} {mode=} {index=}')  
    print(f'{gender.get()=}')  
  
gender = StringVar(value='male')  
Radiobutton(window, text='Male', value='male', variable=gender).pack()  
Radiobutton(window, text='Female', value='female', variable=gender).pack()  
gender.trace_add('write', gender_updated)
```

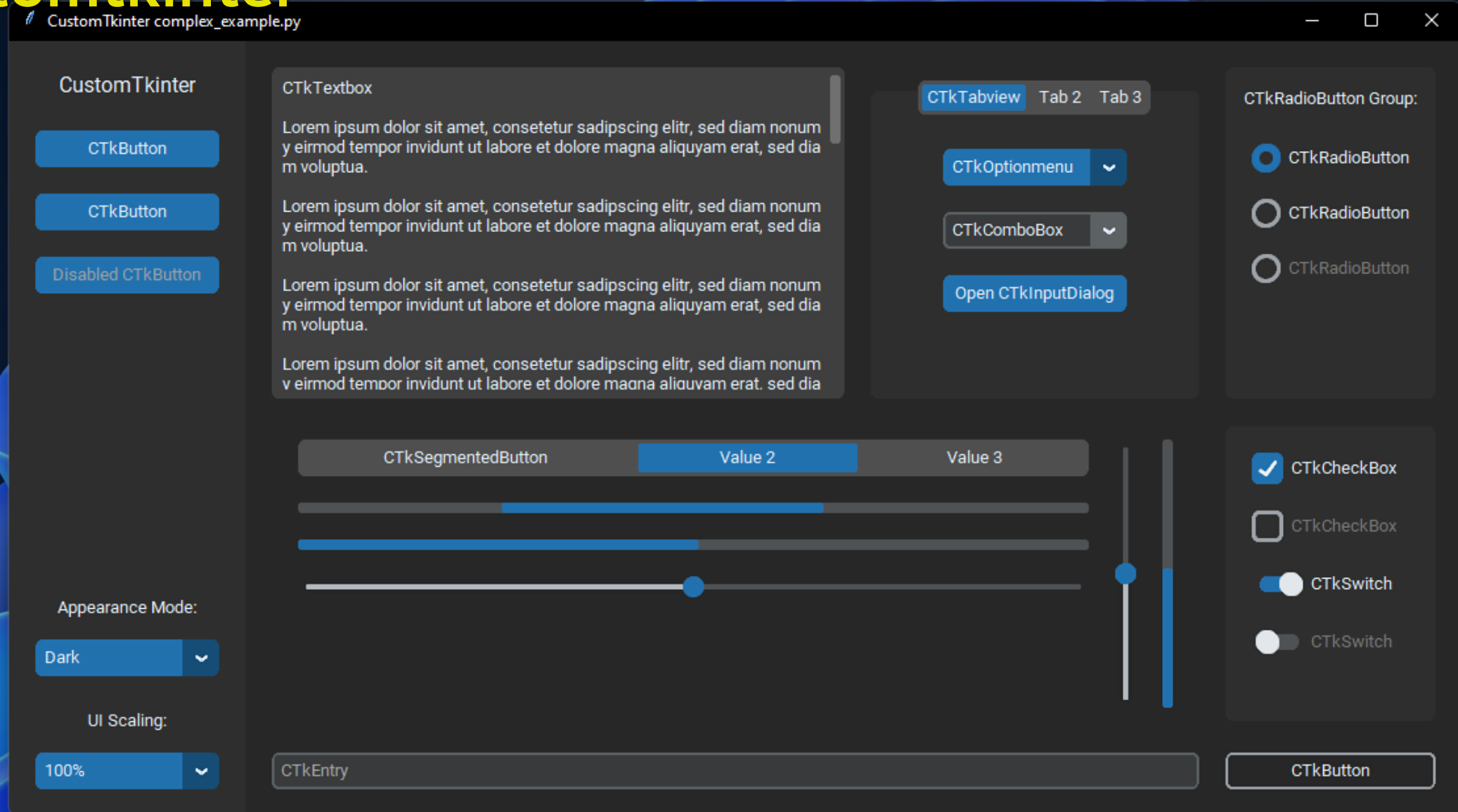
정규식 패턴 탐색

```
word = Entry(width=40)
def search_word(event=None):
    print(f'{word.get()}')
    pattern = re.compile(word.get())
    input_text = text.get("1.0", END)
    lines = input_text.splitlines()

    text.tag_config("found", background="yellow", foreground="red")
    text.tag_remove("found", "1.0", END)
    for i, line in enumerate(lines):
        for mo in pattern.finditer(line):
            print(mo)
            text.tag_add("found", f"{i+1}.0+{mo.span()[0]}chars", f"{i+1}.0+{mo.span()[1]}chars")

word.bind('<Return>', search_word)
word.pack()
```

customtkinter



변환

```
from customtkinter import *
```

```
Tk = CTk
```

```
Frame = CTkFrame
```

```
Label = CTkLabel
```

```
Button = CTkButton
```

```
Entry = CTkEntry
```

```
ScrolledText = CTkTextbox
```

```
Checkbutton = CTkCheckBox
```

```
Radiobutton = CTkRadioButton
```

```
set_default_color_theme("dark-blue") # Themes: "blue" (standard), "green", "dark-blue"
```

```
set_appearance_mode("dark") # 'system', 'light'
```

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

```
history_frame = LabelFrame(text='History')
history_frame.pack(fill=X)

history_listbox = Listbox(history_frame, selectmode='single', height=5)
history_listbox.insert(0, 'Python')
history_listbox.insert(1, '\d\d')
history_listbox.insert(END, '[a-zA-z]+')
history_listbox.pack(side=LEFT, fill=X, expand=True)
history_listbox.bind('<<ListboxSelect>>', select_pattern)

scrollbar = Scrollbar(history_frame)
scrollbar.pack(side=RIGHT, fill=Y)

history_listbox.configure(yscrollcommand=scrollbar.set)
scrollbar.configure(command=history_listbox.yview)
```

Combobox

```
# 콤보박스 생성

combobox = Combobox(first_frame, width=100, height=5, values=['abc', 'def', 'xyz'])
combobox.set('test') # 현재 텍스트 설정
combobox.current(1) # second 아이템 선택
combobox.get() # 현재 텍스트 확인
combobox['values'] += ('newdata', ) # 새로운 아이템 추가, 튜플 더하기
combobox.bind('<<ComboboxSelected>>', select_item) # 선택에 대한 바인딩
combobox.bind('<Return>', enter_item) # 엔터 키 입력에 대한 바인딩
combobox.pack(fill=BOTH, expand=True, padx=5, pady=5)
```

Menu

```
menu = Menu()

menu_File = Menu(menu, tearoff=False) # tearoff : menu 분리
menu_File.add_command(label="New", accelerator='Ctrl+N', command=create_new_file)
menu_File.add_command(label="Open", accelerator='Ctrl+O')
menu_File.add_command(label="Save File", accelerator='Ctrl+S', state='disable')
menu_File.add_separator()
menu_File.add_command(label="Quit", accelerator='Ctrl+Q', command=stop)
menu.add_cascade(label="File", underline=True, menu=menu_File) # udnerline bug in win 10

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=0, menu=menu_Colors) # udnerline bug in win 10

window.bind_all('<Control-q>', stop)
window.config(menu=menu)
```


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('OkCancel', 'Ok or Cancel?')
print(answer)
answer = Messagebox.askquestion('Question', 'OK?')
print(answer)
answer = Messagebox.askretrycancel('RETRY CANCEL', 'Retry or Cancel?')
print(answer)
answer = Messagebox.askyesno('YES NO', 'Yes or No ?')
print(answer)
answer = Messagebox.askyesnocancel('YES NO Cancel', 'Yes? No? Cancel?') # cancel : answer = None
print(answer)
```

filedialog

```
from tkinter import filedialog
```

```
def open_file():  
    # file_name = filedialog.askopenfilename(title='Select a text file', filetypes=(("text files", "*.txt"), ("all files", "*.*")))   
    file_names = filedialog.askopenfilenames(title='Select text files', filetypes=(("text files (.txt)", "*.txt"), ("all files", "*.*")))   
    print(file_names)  
    pass  
  
def save_file_as():  
    file_name = filedialog.asksaveasfilename(title='Save file as...', filetypes=(("text file (.txt)", "*.txt"), ("all files", "*.*")))   
    print(file_name)  
    pass
```

<https://docs.python.org/ko/3/library/dialog.html>

참고

- 공식 Reference

- <https://docs.python.org/3/library/tk.html>

- 책

- <https://tkdocs.com/book.html>