



BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP HỒ CHÍ MINH

Khoa: Công Nghệ Thông Tin



LAB REPORT 07

Student's ID :
Student's name : Hồ Phúc Lâm
Subject : PTHTDPT
Instructor : Nguyễn Thành Thái
Faculty : Công Nghệ Thông Tin
Completed Date : 02/10/2024

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

LAB 3.3 USER INTERFACE

LINK THAM KHẢO : <https://pythonguides.com/tkinter/>

1) Mục đích yêu cầu: làm quen với giao diện người dùng (UI : User Interface) dùng thư viện TKINTER)

2) Thực hiện:

- Cài đặt thêm thư viện TKINTER (dùng google Search Engine)
- Thực hiện Tkinter Fundamentals : GUI, Title, Exit Program, Windows size, label, Entry, Button,...
- Thực hiện Tkinter Widgets : Radiobutton, Checkbutton, Menu bar, Frame, Listbox, Progress bar widget, canvas tutorial, Tree view, Display data in textboxes, OptionMenu, textbox widget, Scrollbar, multiple windows tutorial, notebook widget, spinbox, messagebox, panel, scale,...
- Event Binding and Images: Tkinter Colors, Tkinter Image, Events, Tkinter Mainloop,

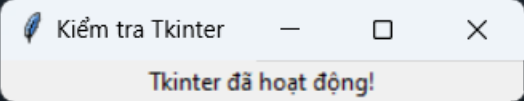
THỰC HIỆN LAB 3.3 USER INTERFACE

3.3.1. Cài đặt thư viện TKINTER

```
#22680401 - Ho Phuc Lam
#Lab3.3. user interface
pip install tk

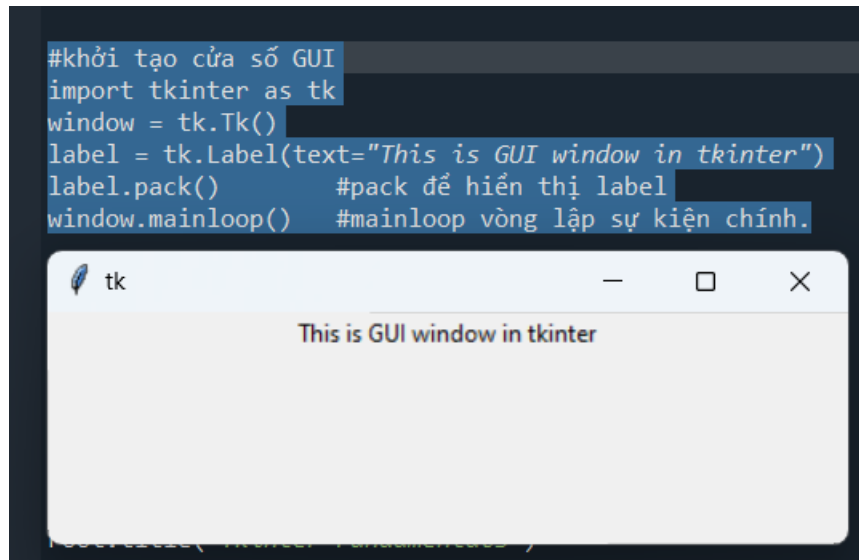
import tkinter as tk

#Kiểm tra thư viện
root = tk.Tk()
root.title("Kiểm tra Tkinter")
label = tk.Label(root, text="Tkinter đã hoạt động!")
label.pack()
root.mainloop()
```

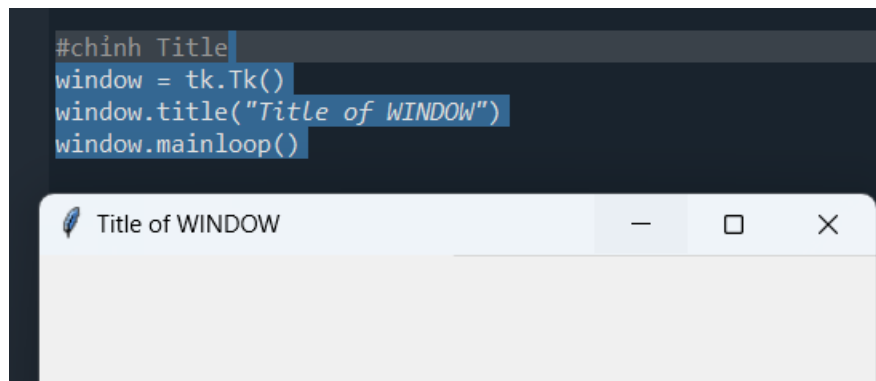


3.3.2. Thực hiện Tkinter Fundamentals

- thiết lập GUI



- Thiết lập title cho GUI

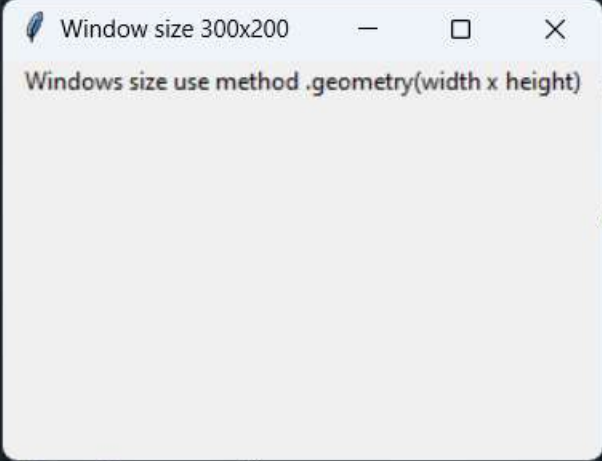


- Exit program – thoát chương trình



- **Windows size – kích thước cửa sổ**


```
5 # windows size - kích thước cửa sổ
6 window = tk.Tk()
7 window.title("Window size 300x200")
8 label = tk.Label(text="Windows size use method .geometry(width x height)")
9 label.pack()
10 window.geometry("300x200")
11 window.mainloop()
```



The screenshot shows a Tkinter window titled "Window size 300x200". The window contains a single label with the text "Windows size use method .geometry(width x height)". The window is styled with a standard Mac OS title bar (red, yellow, and green buttons).

- **Thêm label**

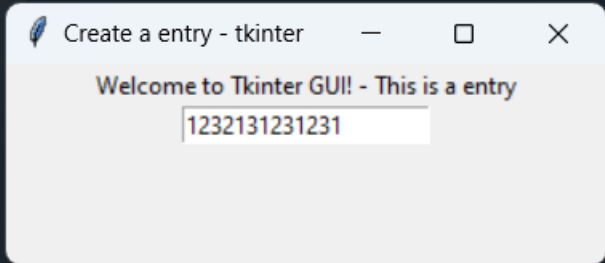
```
64
65 #label
66 window = tk.Tk()
67 window.title("Create a label - tkinter")
68 label = tk.Label(window, text="Welcome to Tkinter GUI! - This is a label")
69 label.pack()
70 window.geometry("300x100")
71 window.mainloop()
```



The screenshot shows a Tkinter window titled "Create a label - tkinter". The window contains a single label with the text "Welcome to Tkinter GUI! - This is a label". The window is styled with a standard Mac OS title bar (red, yellow, and green buttons).

- **Thêm Entry – ô nhập liệu**

```
3 #Entry - ô nhập liệu
4 window = tk.Tk()
5 window.title("Create a entry - tkinter")
6 label = tk.Label(window, text="Welcome to Tkinter GUI! - This is a entry")
7 label.pack()
8
9
10 entry = tk.Entry(window, width= 20)
11 entry.pack()
12
13 window.geometry("300x100")
14 window.mainloop()
15
16 user_input = entry.get()
```



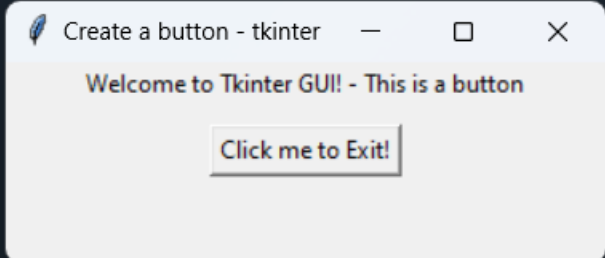
- **Button – tạo nút bấm**

```
#Button - tạo nút bấm
window = tk.Tk()
window.title("Create a button - tkinter")
label = tk.Label(window, text="Welcome to Tkinter GUI! - This is a button")
label.pack()

button = tk.Button(window, text="Click me to Exit!", command=exit_program)
button.pack(pady=10)

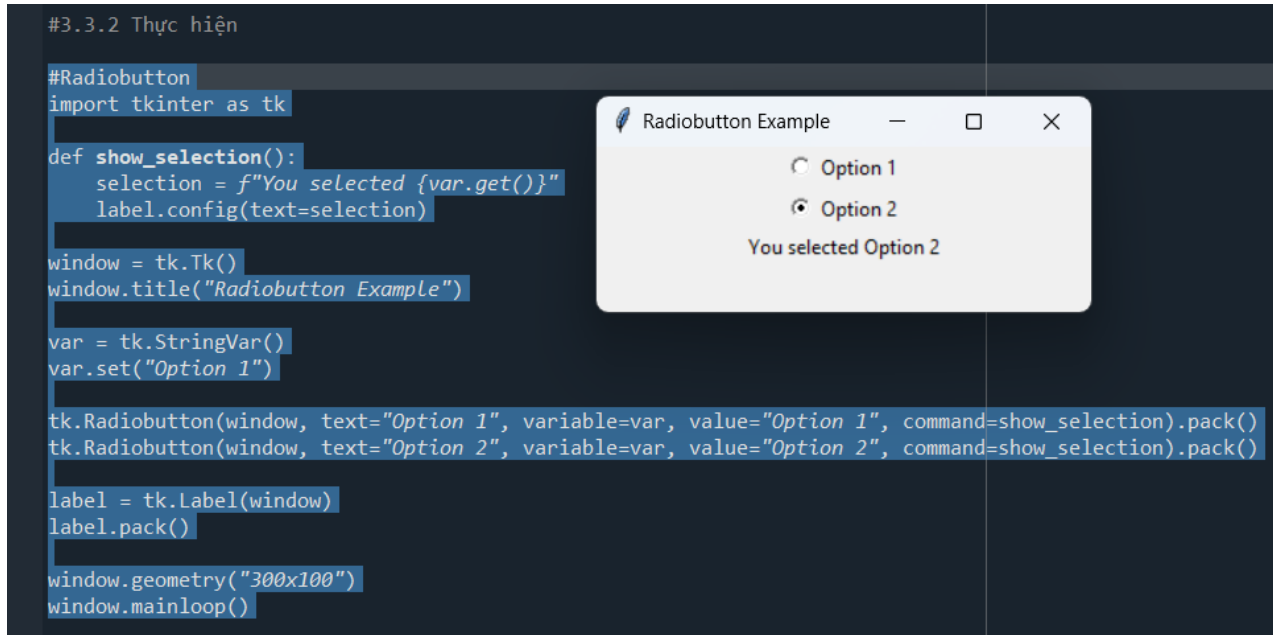
window.geometry("300x100")
window.mainloop()

"""
```

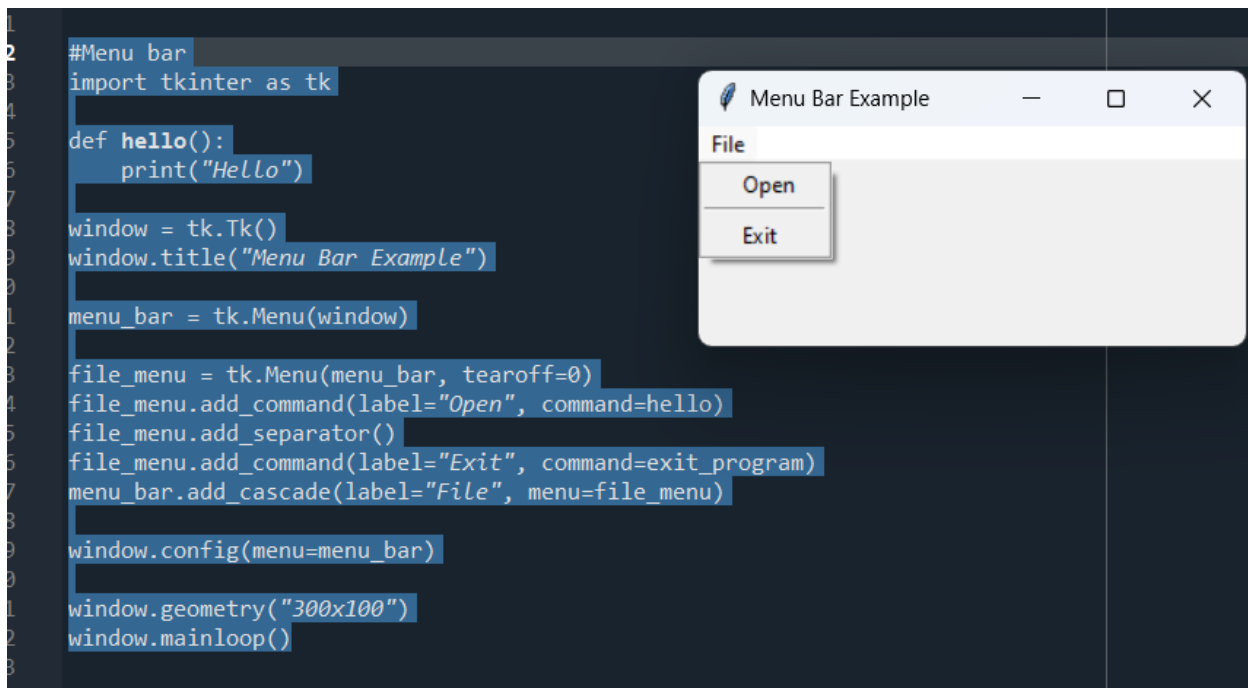


3.3.3. Thực hiện Tkinter Widgets

- Radiobutton



- Menu bar



- Frame

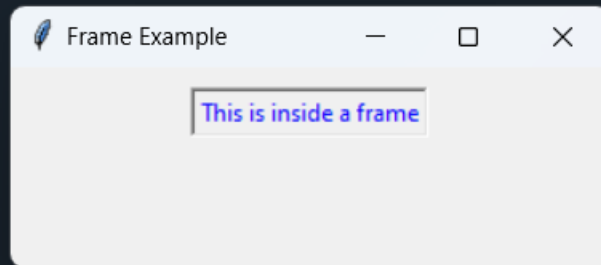
```
#Frame
import tkinter as tk

window = tk.Tk()
window.title("Frame Example")

frame = tk.Frame(window, relief=tk.SUNKEN, borderwidth=2)
frame.pack(padx=10, pady=10)

label = tk.Label(frame, text="This is inside a frame", fg="blue")
label.pack()

window.geometry("300x100")
window.mainloop()
```



- Listbox

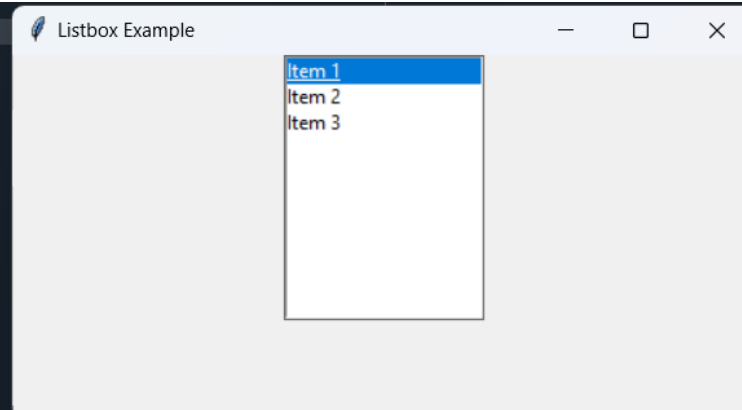
```
#Listbox
import tkinter as tk

window = tk.Tk()
window.title("Listbox Example")

listbox = tk.Listbox(window)
listbox.pack()

for item in ["Item 1", "Item 2", "Item 3"]:
    listbox.insert(tk.END, item)

window.geometry("300x100")
window.mainloop()
```



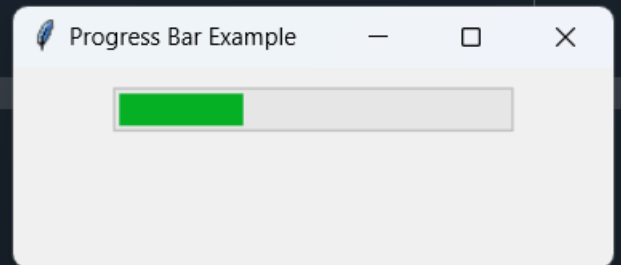
- **Progress bar widget**

```
#Progress bar widget
import tkinter as tk
from tkinter.ttk import Progressbar

window = tk.Tk()
window.title("Progress Bar Example")

progress = Progressbar(window, orient=tk.HORIZONTAL, length=200, mode='determinate')
progress.pack(pady=10)
progress.start(10) # Chạy thanh tiến trình

window.geometry("300x100")
window.mainloop()
```



- **Canvas tutorial**

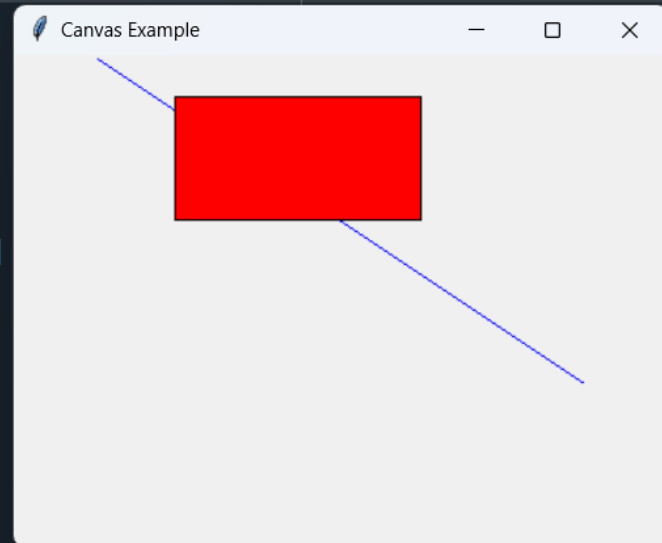
```
#canvas tutorial
import tkinter as tk

window = tk.Tk()
window.title("Canvas Example")

canvas = tk.Canvas(window, width=300, height=200)
canvas.pack()

canvas.create_line(0, 0, 300, 200, fill="blue")
canvas.create_rectangle(50, 25, 200, 100, fill="red")

window.geometry("400x300")
window.mainloop()
```



- Treeview

```
#tree view
#import tkinter as tk
from tkinter import ttk

window = tk.Tk()
window.title("Treeview Example")

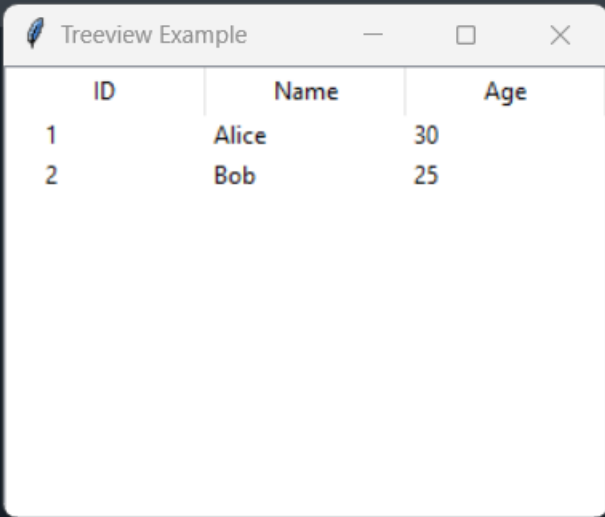
tree = ttk.Treeview(window)
tree.pack()

tree["columns"] = ("one", "two")
tree.column("#0", width=100)
tree.column("one", width=100)
tree.column("two", width=100)

tree.heading("#0", text="ID")
tree.heading("one", text="Name")
tree.heading("two", text="Age")

tree.insert("", "end", text="1", values=("Alice", 30))
tree.insert("", "end", text="2", values=("Bob", 25))

window.mainloop()
```



ID	Name	Age
1	Alice	30
2	Bob	25

- Option menu

```
#Option menu
import tkinter as tk

def show_selected():
    label.config(text=f"Selected: {var.get()}")

window = tk.Tk()
window.title("OptionMenu Example")

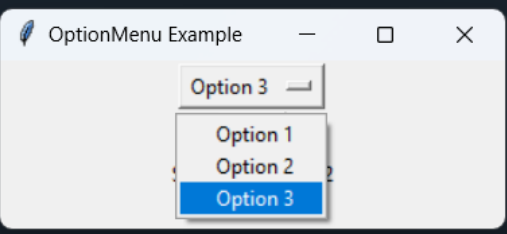
var = tk.StringVar(window)
var.set("Option 1")

option_menu = tk.OptionMenu(window, var, "Option 1", "Option 2", "Option 3")
option_menu.pack()

button = tk.Button(window, text="Show", command=show_selected)
button.pack()

label = tk.Label(window)
label.pack()

window.geometry("300x100")
window.mainloop()
```



- #Display data in textboxes

```
#Display data in textboxes
import tkinter as tk

def display_data():
    # Lấy dữ liệu từ Entry và hiển thị trong TextBox
    data = entry.get()
    text_box.delete(1.0, tk.END) # Xóa nội dung trước đó
    text_box.insert(tk.END, data) # Chèn dữ liệu vào TextBox

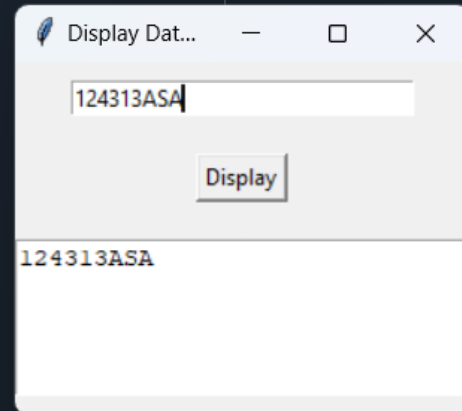
# Tạo cửa sổ giao diện
window = tk.Tk()
window.title("Display Data in TextBox")

# Tạo ô Entry để nhập dữ liệu
entry = tk.Entry(window, width=30)
entry.pack(pady=10)

# Nút bấm để hiển thị dữ liệu
button = tk.Button(window, text="Display", command=display_data)
button.pack(pady=10)

# Tạo TextBox để hiển thị nhiều dòng dữ liệu
text_box = tk.Text(window, height=5, width=30)
text_box.pack(pady=10)

window.mainloop()
```



- #textbox widget

```
#textbox widget
import tkinter as tk

# Tạo hàm để hiển thị nội dung từ TextBox
def get_text():
    content = text_box.get("1.0", tk.END) # Lấy dữ liệu từ dòng 1, ký tự 0 đến cuối
    print("Nội dung TextBox:", content)

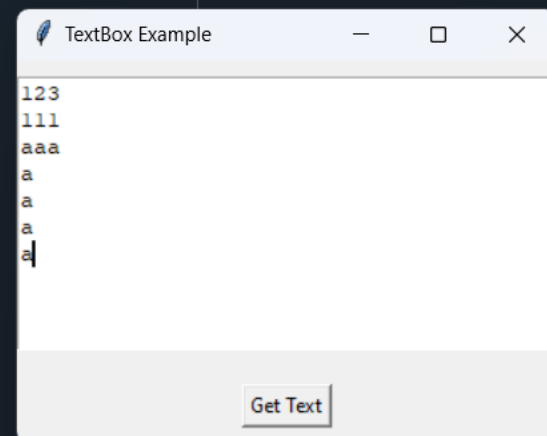
# Tạo cửa sổ giao diện
window = tk.Tk()
window.title("TextBox Example")

# Tạo TextBox để nhập hoặc hiển thị nhiều dòng văn bản
text_box = tk.Text(window, height=10, width=40)
text_box.pack(pady=10)

# Tạo nút bấm để lấy nội dung từ TextBox
button = tk.Button(window, text="Get Text", command=get_text)
button.pack(pady=10)

window.mainloop()

#Scrollbar
```



- #Scrollbar

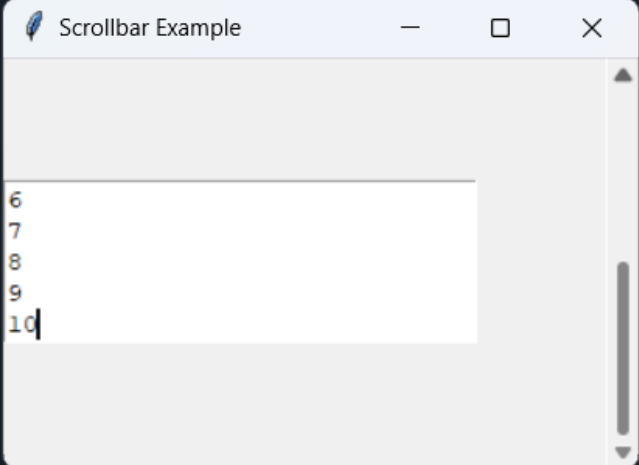
```
#Scrollbar
import tkinter as tk

window = tk.Tk()
window.title("Scrollbar Example")

text = tk.Text(window, height=5, width=30)
scrollbar = tk.Scrollbar(window)
scrollbar.pack(side=tk.RIGHT, fill=tk.Y)

text.pack(side=tk.LEFT)
scrollbar.config(command=text.yview)
text.config(yscrollcommand=scrollbar.set)

window.mainloop()
```



- #multiple windows tutorial

```
#multiple windows tutorial
import tkinter as tk
from tkinter import Toplevel

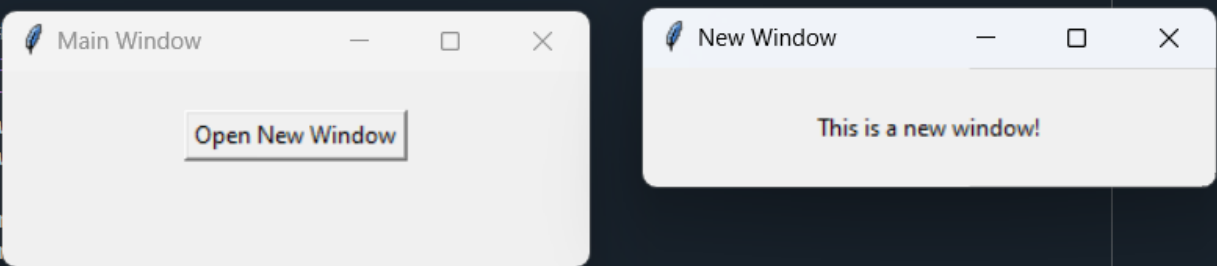
# Hàm mở cửa sổ mới
def open_new_window():
    # Tạo cửa sổ phụ
    new_window = Toplevel(window)
    new_window.title("New Window")

    # Thêm nội dung vào cửa sổ phụ
    label = tk.Label(new_window, text="This is a new window!")
    label.pack(padx=20, pady=20)

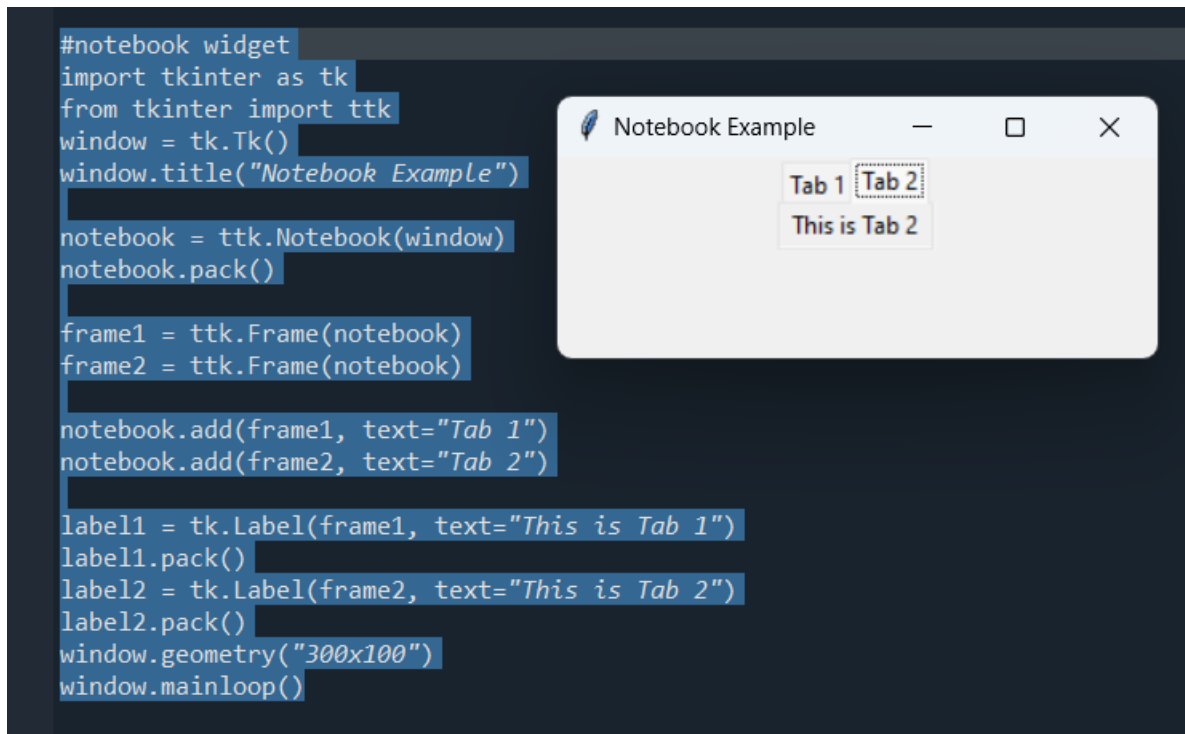
# Tạo cửa sổ chính
window = tk.Tk()
window.title("Main Window")

# Nút để mở cửa sổ phụ
button = tk.Button(window, text="Open New Window", command=open_new_window)
button.pack(padx=20, pady=20)

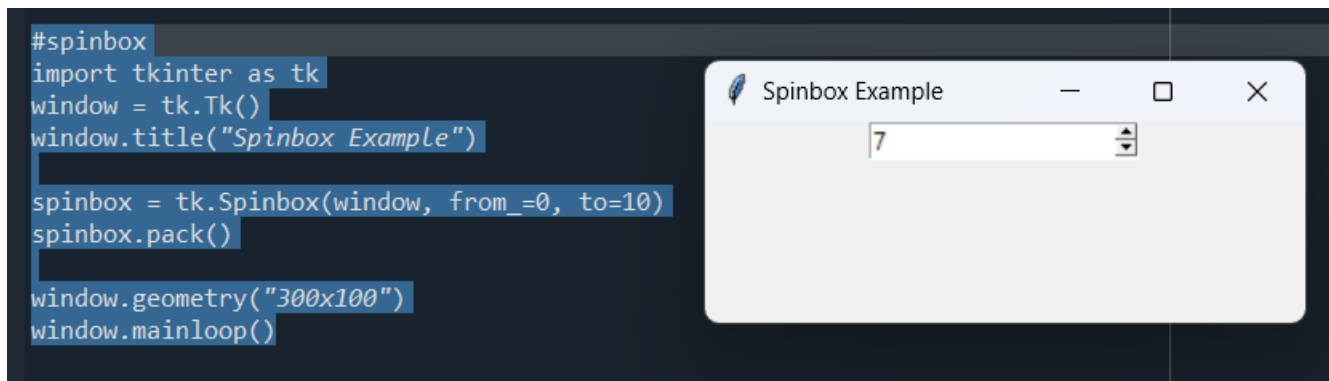
window.geometry("300x100")
window.mainloop()
```



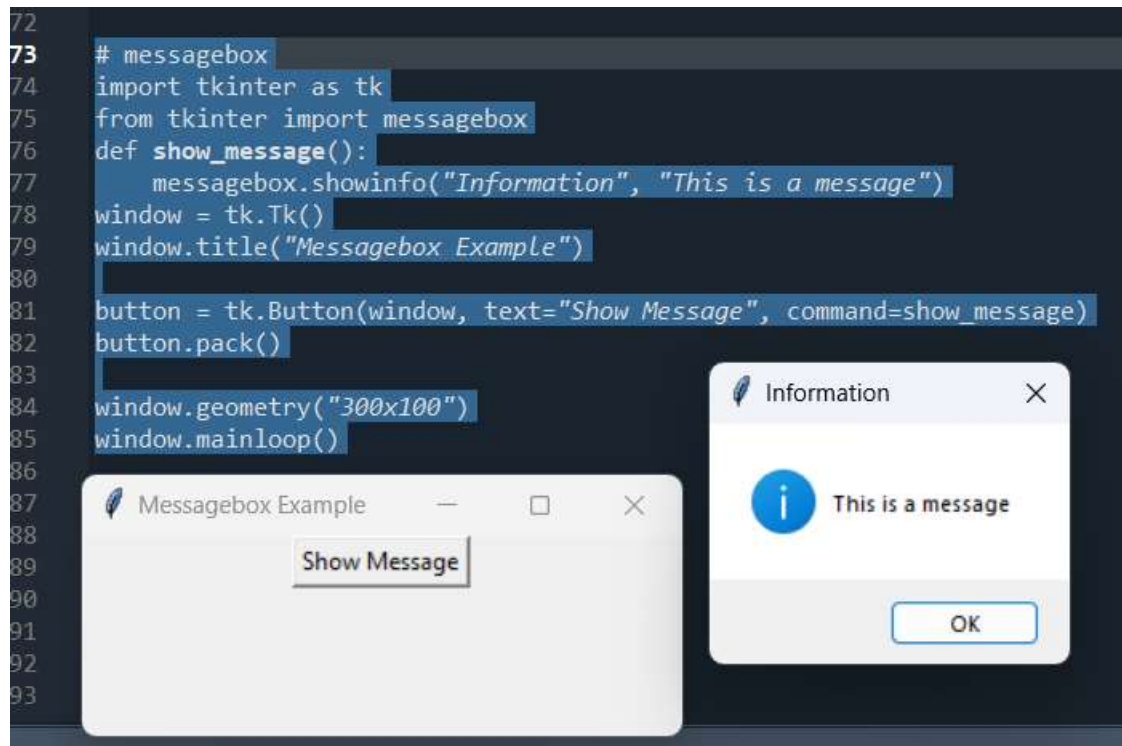
- **#notebook widget**



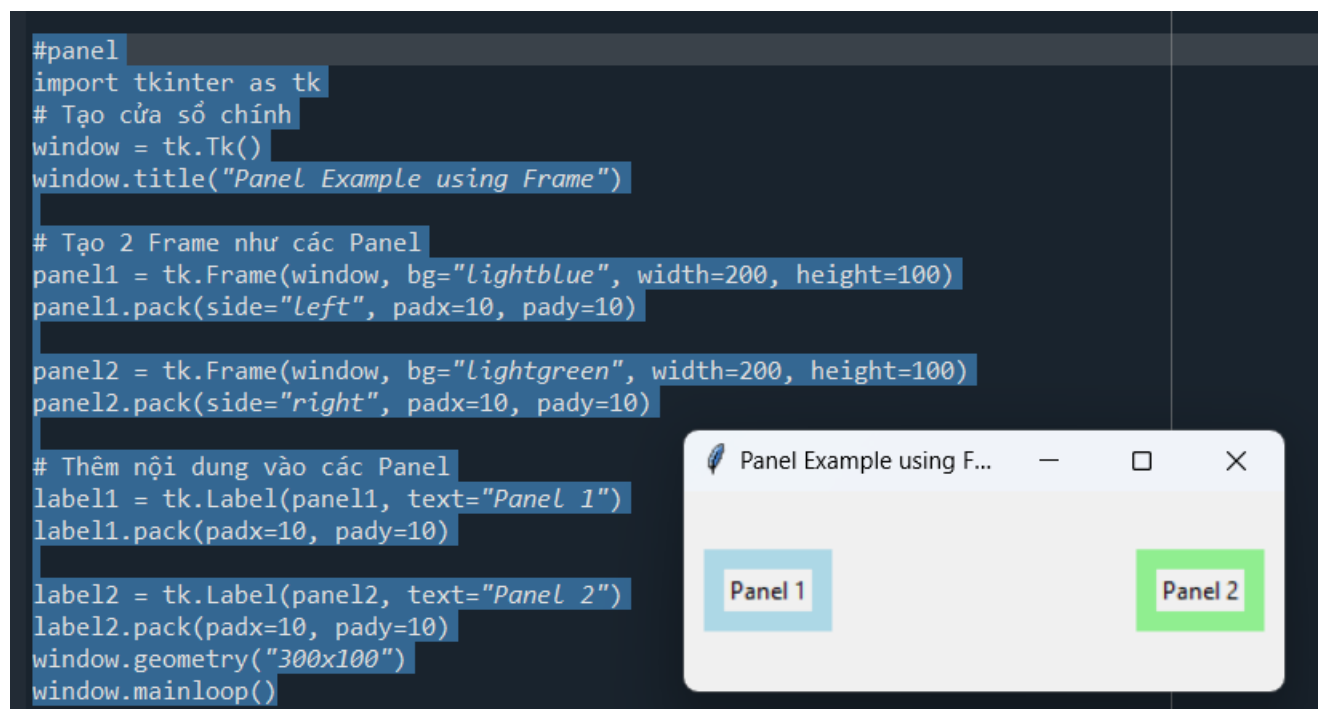
- **#spinbox**



- # messagebox



- #panel

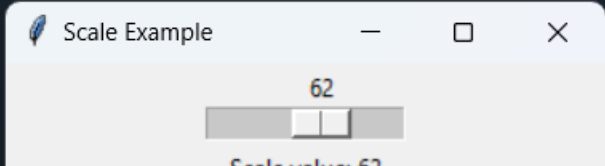


- #scale

```
#scale
import tkinter as tk
def show_value(val):
    label.config(text=f"Scale value: {val}")
window = tk.Tk()
window.title("Scale Example")

scale = tk.Scale(window, from_=0, to=100, orient=tk.HORIZONTAL, command=show_value)
scale.pack()

label = tk.Label(window)
label.pack()
window.geometry("300x100")
window.mainloop()
```



3.3.4. Event Binding and Images:

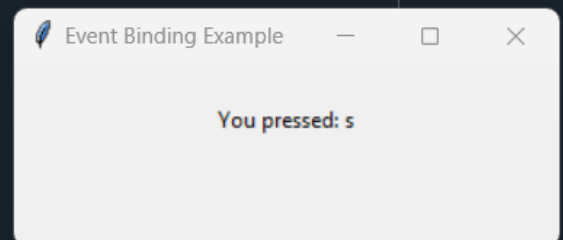
- Event Binding trong Tkinter cho phép bạn gắn kết một sự kiện cụ thể (như nhấn phím, nhấp chuột) với một hàm xử lý. Khi sự kiện xảy ra, hàm xử lý sự kiện sẽ được gọi.

```
#3.3.4. Event Binding and Images:
import tkinter as tk
# Hàm xử lý khi nhấn phím
def on_key_press(event):
    label.config(text=f"You pressed: {event.char}")

# Tạo cửa sổ giao diện
window = tk.Tk()
window.title("Event Binding Example")

# Tạo nhãn để hiển thị kết quả
label = tk.Label(window, text="Press any key")
label.pack(pady=20)

# Bind sự kiện "nhấn phím" với hàm xử lý
window.bind("<Key>", on_key_press)
window.geometry("300x100")
window.mainloop()
```

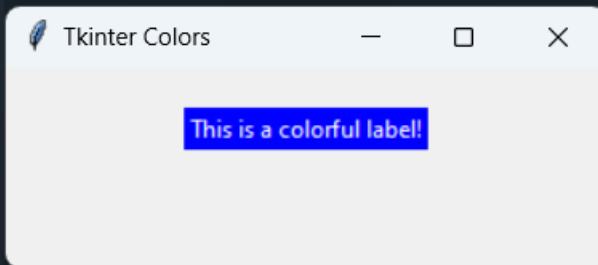


- Tkinter Colors:

```
#Tkinter Colors
import tkinter as tk
# Tạo cửa sổ giao diện
window = tk.Tk()
window.title("Tkinter Colors")

# Tạo nhãn với màu nền và màu chữ
label = tk.Label(window, text="This is a colorful label!", bg="blue", fg="white")
label.pack(padx=20, pady=20)

window.geometry("300x100")
window.mainloop()
```



- Tkinter Image

```
#Tkinter Image
import tkinter as tk
# Tạo cửa sổ giao diện
window = tk.Tk()
window.title("Tkinter Image Example")

# Tải hình ảnh (hỗ trợ định dạng PNG, GIF)
image = tk.PhotoImage(file="tuan07.png")

# Hiển thị hình ảnh trong nhãn (Label)
label = tk.Label(window, image=image)
label.pack(padx=20, pady=20)

window.geometry("300x100")
window.mainloop()
```



- Events

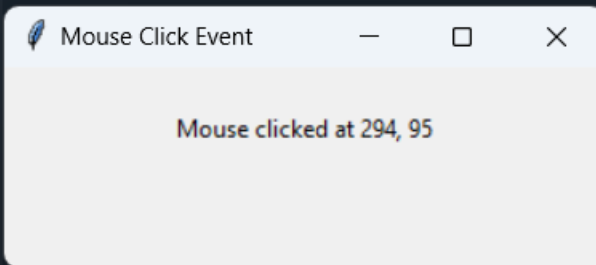

```
#Events
import tkinter as tk
# Hàm xử lý khi nhấp chuột
def on_click(event):
    label.config(text=f"Mouse clicked at {event.x}, {event.y}")

# Tạo cửa sổ giao diện
window = tk.Tk()
window.title("Mouse Click Event")

# Tạo nhãn để hiển thị vị trí nhấp chuột
label = tk.Label(window, text="Click anywhere")
label.pack(pady=20)

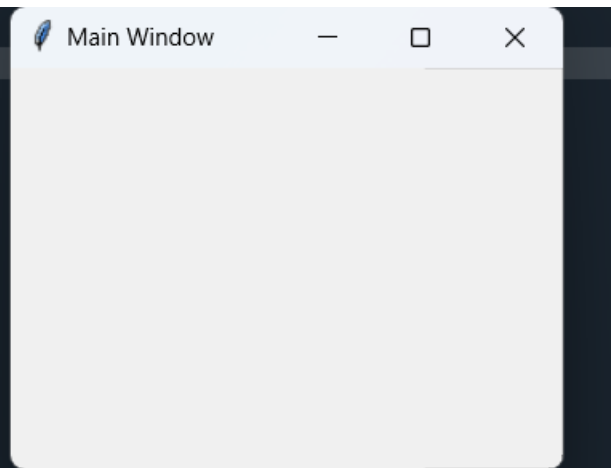
# Bind sự kiện click chuột
window.bind("<Button-1>", on_click) # Button-1 là click chuột trái

window.geometry("300x100")
window.mainloop()
```



- Tkinter Mainloop

```
#Tkinter Mainloop
window = tk.Tk()
window.title("Main Window")
window.mainloop()
```



LAB 4 IMAGE (làm trong 2 tuần)

1) Mục đích yêu cầu :

- +Củng cố kiến thức cơ bản về IMAGE, các khái niệm về hệ màu, độ phân giải,...
- +Tiếp cận thư viện xử lý image như PIL, opencv,... các API hỗ trợ,...
- +Chuyển dữ liệu ảnh vào trong cấu trúc NumPy và ngược lại
- +Tải và hiển thị hình ảnh bằng API Matplotlib, OpenCV và Keras
- +Tiến hành các thao tác cơ bản đối với hình ảnh bằng thư viện Pillow và NumPy và lưu nó vào hệ thống cục bộ của bạn.
- +Đọc hình ảnh dưới dạng mảng trong Keras API và OpenCV

2) Tài liệu tham khảo (mở link sau xem hướng dẫn)

<https://www.pluralsight.com/guides/importing-image-data-into-numpy-arrays>

3) Thực hiện:

- Tìm hiểu và cài đặt thư viện Pillow, kiểm tra phiên bản thư viện cài đặt
- Thực hiện việc nạp 1 ảnh dùng phương pháp 1 : Pillow Library và phương pháp 2 : Matplotlib Library
- Chuyển đổi dữ liệu ảnh sang Numpy Array và ngược lại, sau đó hiển thị dữ liệu dưới dạng cấu trúc Array
- Chuyển đổi ảnh sang dạng mức xám và lưu lại kết quả ở các định dạng khác nhau (vd png, gif, peg,...)
- Định dạng lại kích thước ảnh (resizing)
- Viết một chương trình hoàn chỉnh bao gồm các chức năng ở trên, các chức năng trong chương trình viết dạng thủ tục, được gọi khi cần.

4) Công cụ hỗ trợ : Python programming language

a) Spyder IDE hoặc PYTHON commandline

b) Cài đặt các thư viện hỗ trợ :

Câu lệnh cài đặt : `pip install <gói cài đặt>`

Vd : `pip install opencv-python`

`pip install simpleaudio`

`pip install matplotlib`

- download file mẫu *.jpg và lưu trong thư mục
- Mở notepad viết chương trình *.py lưu trong một thư mục {thường là chung với thư mục của file audio}
- Chạy thử code : `>python <tên file>.py`
- Lưu các bài tập trong thư mục, nén và nộp (cuối giờ thực hành)

#4.1. Tìm hiểu và cài đặt thư viện Pillow

```
557
558 #4.1. Tìm hiểu và cài đặt thư viện Pillow
559 pip install Pillow
560 import PIL
561 print(PIL.__version__)
562
563
```

Console 1/A X

```
In [80]: pip install Pillow
Requirement already satisfied: Pillow in c:\users\computer\anaconda3\lib\site-packages (10.3.0)
Note: you may need to restart the kernel to use updated packages.

In [81]: import PIL
...: print(PIL.__version__)
10.3.0
```