**Hướng dẫn lập trình giao diện GUI bằng Tkinter trong ngôn ngữ Python**

GUI là viết tắt của cụm từ Graphical User Interface, có nghĩa là giao diện đồ họa người dùng. Thuật ngữ này được dùng để ám chỉ cách người dùng tương tác với các thiết bị máy tính thông qua hình ảnh, thay vì sử dụng các câu lệnh phức tạp.

Lập trình giao diện người dùng với Tkinter

Sau đây VniTeach sẽ hướng dẫn lập trình giao diện GUI bằng Tkinter trong ngôn ngữ Python

**1. Tạo giao diện đầu tiên**

#Thêm thư viện tkinter

from tkinter import \*

#Tạo một cửa sổ mới

window = Tk()

#Thêm tiêu đề cho cửa sổ

window.title('Welcome to VniTeach app')

#Đặt kích thước của cửa sổ

window.geometry('350x200')

#Lặp vô tận để hiển thị cửa sổ

window.mainloop()

Dòng cuối cùng gọi  mainloop hàm. Hàm này gọi vòng lặp vô tận của cửa sổ, vì vậy cửa sổ sẽ chờ bất kỳ tương tác người dùng nào cho đến khi chúng ta đóng nó.

Nếu bạn quên gọi  mainloop hàm, sẽ không có gì xuất hiện cho người dùng.

**2. Làm việc với label**

Đối tượng này dùng để hiển thị văn bản hoặc hình ảnh.

**Cú pháp**

w = Label( master, option, … )

**Tham số**

* **master** − Tên của cửa sổ chứa nhãn.
* **options** − Đây là danh sách một số tùy chọn thường dùng cho nhãn, tùy chọn này có thể sử dụng cặp từ khóa và giá trị.

|  |  |
| --- | --- |
| **TT** | **Tùy chọn và giải thích** |
| 1 | **anchor** Tùy chọn này dùng để căn lề của văn bản, mặc định anchor=CENTER, nghĩa là văn bản sẽ căn giữa. |
| 2 | **bg** Màu nền của nhãn (label). |
| 3 | **bitmap** Đặt tùy chọn này bằng một bitmap hoặc đối tượng hình ảnh và nhãn sẽ hiển thị đồ họa đó. |
| 4 | **bd** Quy định kích thước của đường viền (border) bao quanh nhãn, mặc định là 2 pixels. |
| 5 | **cursor** Dùng để thiết lập loại con trỏ chuột (*arrow, dot,…*). |
| 6 | **font** Nếu bạn đang hiển thị văn bản trong nhãn này (với tùy chọn văn bản hoặc văn bản có thể thay đổi, tùy chọn phông chữ chỉ định loại phông chữ mà văn bản sẽ được hiển thị. |
| 7 | **fg** Nếu bạn đang hiển thị văn bản hoặc một bitmap trong nhãn này, tùy chọn này chỉ định màu của văn bản. Nếu bạn đang hiển thị một bitmap, đây là màu sẽ xuất hiện tại vị trí của các bit 1 trong bitmap. |
| 8 | **height** Quy định độ cao của nhãn. |
| 9 | **image** Dùng để hiển thị ảnh tĩnh trong nhãn. |
| 10 | **justify** Quy định nhiều dòng của văn bản sẽ canh lề LEFT, CENTER (mặc định) hoặc RIGHT. |
| 11 | **padx** Thiết lập khoảng trống phía trước và sau (padding) của văn bản, mặc định là 1. |
| 12 | **pady** Thiết lập khoảng trống phía trên và phía dưới (padding) của văn bản, mặc định là 1. |
| 13 | **relief** Chỉ định sự xuất hiện của đường viền quanh nhãn, mặc định là FLAT; có nhiều giá trị khác. |
| 14 | **text** Dùng để hiển thị nội dung của một hoặc nhiều dòng văn bản chứa trong nhãn. Sử dụng quy ước (“\n”) để ngắt dòng. |
| 15 | **textvariable** Để bổ sung văn bản được hiển thị trong tiện ích nhãn thành một biến điều khiển của lớp StringVar, hãy đặt tùy chọn này cho biến đó. |
| 16 | **underline** Gạch chân chữ viết cho một số ký tự, bắt đầu đếm từ 0, có thể thiết lập n. Mặc định underline=-1, nghĩa là không gạch chân chữ viết. |
| 17 | **width** Quy định chiều rộng của nhãn (đơn vị số ký tự), nếu tùy chọn này không thiết lập thì nhãn sẽ thay đổi kích thước cho phù hợp với số lượng ký tự của văn bản mà nó chứa. |
| 18 | **wraplength** Thiết lập số lượng ký tự cho mỗi dòng, mặc định bằng 0 nghĩa là dòng sẽ tự động ngắt (chỉ gồm có duy nhất một dòng). |

Ví dụ đơn giản

from tkinter import \*

window = Tk()

window.title("Welcome to VniTeach app")

#Thêm label có nội dung Hello, font chữ

lbl = Label(window, text="Hello", font=("Arial Bold", 50))

#Xác định vị trí của label

lbl.grid(column=0, row=0)

window.mainloop()

**2. Làm việc với Button**

Nút bấm có thể hiển thị văn bản hoặc hình ảnh, chúng ta có thể thiết lập sự kiện khi chúng ta click vào nút bấm như tự động gọi hàm chẳng hạn.

**Cú pháp**

w = Button( master, option = value, ... )

**Tham số**

* **master** − Tên của cửa sổ chứa nút bấm.
* **options** − Đây là danh sách một số tùy chọn thường dùng cho nút bấm, tùy chọn này có thể sử dụng cặp từ khóa và giá trị.

|  |  |
| --- | --- |
| **TT** | **Tùy chọn và giải thích** |
| 1 | **activebackground** Màu nền của nút bấm khi con trỏ chuột ở phía trên nút bấm |
| 2 | **activeforeground** Màu chữ của nút bấm khi con trỏ chuột ở phía trên nút bấm |
| 3 | **bd** Độ rộng của đường biên của nút bấm, mặc định là 2. |
| 4 | **bg** Màu nền của nút bấm. |
| 5 | **command** Hàm hoặc phương thức sẽ được gọi khi click vào nút bấm. |
| 6 | **fg** Màu chữ của nút bấm. |
| 7 | **font** Phông chữ của văn bản trên nút bấm. |
| 8 | **height** Độ cao của nút bấm (số dòng) hoặc pixels đối với hình ảnh. |
| 9 | **highlightcolor** Màu nổi bật khi nút bấm có focus. |
| 10 | **image** Hình ảnh sẽ hiển thị trên nút bấm thay vì văn bản. |
| 11 | **justify** Quy định nhiều dòng của văn bản sẽ canh lề LEFT, CENTER (mặc định) hoặc RIGHT. |
| 12 | **padx** Thiết lập khoảng trống phía trước và sau (padding) của văn bản, mặc định là 1. |
| 13 | **pady** Thiết lập khoảng trống phía trên và phía dưới (padding) của văn bản, mặc định là 1. |
| 14 | **relief** Chỉ định sự xuất hiện của đường viền quanh nút bấm. Một số giá trị như: SUNKEN, RAISED, GROOVE và RIDGE. |
| 15 | **state** Thiết lập thành DISABLED để chuyển sang nút bấm màu xám (bị ẩn) và không phản hồi. Có giá trị là ACTIVE khi con trỏ chuột ở trên nút bấm. Mặc định là NORMAL. |
| 16 | **underline** Mặc định là -1, nghĩa là văn bản trên nút bấm sẽ không gạch chân. Nếu không âm, ký tự văn bản tương ứng sẽ được gạch chân. |
| 17 | **width** Độ rộng của nút bấm là số ký tự (nếu hiển thị văn bản) hoặc pixels (nếu hiển thị hình ảnh). |
| 18 | **wraplength** Thiết lập số lượng ký tự cho mỗi dòng, mặc định bằng 0 nghĩa là dòng sẽ tự động ngắt (chỉ gồm có duy nhất một dòng). |

**Phương thức**

|  |  |
| --- | --- |
| **TT** | **Phương thức và giải thích** |
| 1 | **flash()** Làm cho nút nhấp nháy nhiều lần giữa màu đang hoạt động và màu bình thường. Để nút ở trạng thái ban đầu. Bỏ qua nếu nút bị tắt. |
| 2 | **invoke()** Gọi lệnh gọi lại của nút và trả về những gì mà hàm đó trả về. Không có tác dụng nếu nút bị tắt hoặc không có lệnh gọi lại. |

**Ví dụ:**

from tkinter import \*

window = Tk()

window.title("Welcome to VniTeach app")

window.geometry('350x200')

lbl = Label(window, text="Hello")

lbl.grid(column=0, row=0)

#Thêm một nút nhấn Click Me

btn = Button(window, text="Click Me", bg="orange", fg="red")

#Thiết lập vị trí của nút nhấn có màu nền và màu chữ

btn.grid(column=1, row=0)

window.mainloop()

Xử lý sự kiện khi nhấn nút Click Me

from tkinter import \*

window = Tk()

window.title("Welcome to VniTeach app")

window.geometry('350x200')

lbl = Label(window, text="Hello")

lbl.grid(column=0, row=0)

#Hàm khi nút được nhấn

def clicked():

lbl.configure(text="Button was clicked !!")

#Gọi hàm clicked khi nút được nhấn

btn = Button(window, text="Click Me", command=clicked)

btn.grid(column=1, row=0)

window.mainloop()

Không nhất thiết tên hàm để xử lý sự kiện click là clicked như trên, ví dụ như sau:

# !/usr/bin/python3

from tkinter import \*

from tkinter import messagebox

top = Tk()

top.geometry("100x100")

def helloCallBack():

msg = messagebox.showinfo( "Hello Python", "Hello World")

B = Button(top, text = "Hello", command = helloCallBack)

B.place(x = 50,y = 50)

top.mainloop()

**3. Làm việc với Textbox**

from tkinter import \*

window = Tk()

window.title("Welcome to VniTeach app")

window.geometry('350x200')

lbl = Label(window, text="Hello")

lbl.grid(column=0, row=0)

#Tạo một Textbox

txt = Entry(window,width=10)

#Vị trí xuất hiện của Textbox

txt.grid(column=1, row=0)

#Đặt vị trí con trỏ tại Textbox

txt.focus()

#Hàm xử lý khi nút được nhấn

def clicked():

res = "Welcome to " + txt.get()

lbl.configure(text= res)

btn = Button(window, text="Click Me", command=clicked)

btn.grid(column=2, row=0)

#Để tắt chức năng nhập của Textbox bằng state

#txt = Entry(window,width=10, state='disabled')

window.mainloop()

**4. Làm việc với Combobox**

from tkinter import \*

from tkinter.ttk import \*

window = Tk()

window.title("Welcome to VniTeach app")

window.geometry('350x200')

#Tạo hộp chọn Combobox

combo = Combobox(window)

#Các giá trị của hộp chọn

combo['values']= (1, 2, 3, 4, 5, "Text")

#Thiết lập giá trị được chọn

combo.current(1) #set the selected item

combo.grid(column=0, row=0)

#Lấy giá trị của hộp chọn bằng combo.get()

window.mainloop()

**5. Làm việc với Checkbox**

from tkinter import \*

from tkinter.ttk import \*

window = Tk()

window.title("Welcome to VniTeach app")

window.geometry('350x200')

#Thiết lập trạng thái của Checkbox

chk\_state = BooleanVar()

chk\_state.set(True) #set check state

#Tạo Checkbox có trạng thái đã tích chọn

chk = Checkbutton(window, text='Choose', var=chk\_state)

chk.grid(column=0, row=0)

window.mainloop()

**6. Làm việc với Radio**

from tkinter import \*

from tkinter.ttk import \*

window = Tk()

window.title("Welcome to LikeGeeks app")

selected = IntVar()

rad1 = Radiobutton(window,text='First', value=1, variable=selected)

rad2 = Radiobutton(window,text='Second', value=2, variable=selected)

rad3 = Radiobutton(window,text='Third', value=3, variable=selected)

def clicked():

print(selected.get())

btn = Button(window, text="Click Me", command=clicked)

rad1.grid(column=0, row=0)

rad2.grid(column=1, row=0)

rad3.grid(column=2, row=0)

btn.grid(column=3, row=0)

window.mainloop()

**7. Làm việc với ScrolledText**

from tkinter import \*

from tkinter import scrolledtext

window = Tk()

window.title("Welcome to VniTeach app")

window.geometry('350x200')

txt = scrolledtext.ScrolledText(window,width=40,height=10)

txt.grid(column=0,row=0)

window.mainloop()

**8. Làm việc với Messagebox**

– Hộp thoại thông báo

from tkinter import \*

from tkinter import messagebox

window = Tk()

window.title("Welcome to VniTeach app")

window.geometry('350x200')

def clicked():

messagebox.showinfo('Message title', 'Message content')

btn = Button(window,text='Click here', command=clicked)

btn.grid(column=0,row=0)

window.mainloop()

– Hộp thoại cảnh báo và lỗi

#Hộp thoại cảnh báo

messagebox.showwarning('Message title', 'Message content')

#Hộp thoại báo lỗi

messagebox.showerror('Message title', 'Message content')

– Hộp thoại câu hỏi

from tkinter import messagebox

res = messagebox.askquestion('Message title','Message content')

res = messagebox.askyesno('Message title','Message content')

res = messagebox.askyesnocancel('Message title','Message content')

res = messagebox.askokcancel('Message title','Message content')

res = messagebox.askretrycancel('Message title','Message content')

**9. Làm việc với SpinBox**

from tkinter import \*

window = Tk()

window.title("Welcome to VniTeach app")

window.geometry('350x200')

#Tạo spinbox có chiều rộng 5, giá trị từ 0 đến 100

spin = Spinbox(window, from\_=0, to=100, width=5)

spin.grid(column=0,row=0)

window.mainloop()

– Liệt kê các giá trị của spinbox

#spinbox chỉ gồm có 3 giá trị là 3, 5 và 10

spin = Spinbox(window, values=(3, 5, 10), width=5)

– Đặt giá trị mặc định cho spinbox

var =IntVar()

#Đặt giá trị mặc định là 25

var.set(25)

spin = Spinbox(window, from\_=0, to=100, width=5, textvariable=var)

**10. Làm việc với trình chọn tệp và thư mục**

from tkinter import filedialog

file = filedialog.askopenfilename(filetypes = (("Text files","\*.txt"),("all files","\*.\*")))

dir = filedialog.askdirectory()

Chỉ định thư mục ban đầu cho hộp thoại tệp bằng cách chỉ định  initialdir như thế này

from tkinter import filedialog

from os import path

file = filedialog.askopenfilename(initialdir= path.dirname(\_\_file\_\_))

**11. Làm việc với Menu**

from tkinter import \*

from tkinter import Menu

window = Tk()

window.title("Welcome to VniTeach app")

menu = Menu(window)

new\_item = Menu(menu)

new\_item.add\_command(label='New')

new\_item.add\_separator()

new\_item.add\_command(label='Edit')

menu.add\_cascade(label='File', menu=new\_item)

window.config(menu=menu)

window.mainloop()

Sự kiện khi người dùng click chọn một bảng chọn nào đó:

#Thực hiện hàm clicked khi người dùng chọn New

new\_item.add\_command(label='New', command=clicked)

**12. Làm việc với các Tabs**

Để tạo một điều khiển tab, có một vài bước.

* Đầu tiên, chúng ta tạo một điều khiển tab bằng cách sử dụng lớp  Notebook.
* Tạo một tab bằng cách sử dụng  Frame lớp.
* Thêm tab đó vào điều khiển tab.
* Đóng gói điều khiển tab để nó hiển thị trong cửa sổ.

from tkinter import \*

from tkinter import ttk

window = Tk()

window.title("Welcome to VniTeach app")

tab\_control = ttk.Notebook(window)

tab1 = ttk.Frame(tab\_control)

tab\_control.add(tab1, text='First')

tab\_control.pack(expand=1, fill='both')

window.mainloop()

**Thêm Widgets vào Notebook**

Sau khi tạo các tab, bạn có thể đặt các widget bên trong các tab này bằng cách gán thuộc tính cha cho tab mong muốn.

from tkinter import \*

from tkinter import ttk

window = Tk()

window.title("Welcome to LikeGeeks app")

tab\_control = ttk.Notebook(window)

tab1 = ttk.Frame(tab\_control)

tab2 = ttk.Frame(tab\_control)

tab\_control.add(tab1, text='First')

tab\_control.add(tab2, text='Second')

lbl1 = Label(tab1, text= 'label1')

lbl1.grid(column=0, row=0)

lbl2 = Label(tab2, text= 'label2')

lbl2.grid(column=0, row=0)

tab\_control.pack(expand=1, fill='both')

window.mainloop()

Kết quả

**Thêm khoảng cách cho các widget (Padding)**

Bạn có thể thêm các khoảng cách trước và sau của mỗi tab bằng cách sử dụng các thuộc tính padx và  pady .

lbl1 = Label(tab1, text= 'label1', padx=5, pady=5)