

Python tkinter Combobox

Giới thiệu:

Combobox cho phép người dùng lựa chọn từ một danh sách các tùy chọn được xác định trước và cũng cung cấp tính linh hoạt để nhập các giá trị không có trong danh sách.

Thư viện ttk Combobox:

Combobox là một widget của ttk, vì vậy chúng ta phải import ttk, nếu không sẽ gặp lỗi sau:

`AttributeError: module 'tkinter' has no attribute 'Combobox'`

Sử dụng Combobox trong Tkinter để chọn tùy chọn hoặc thêm dữ liệu bởi người dùng & đọc, thêm và thiết lập tùy chọn mặc định:

```
import tkinter as tk
from tkinter import ttk

my_w = tk.Tk()
my_w.geometry("300x150") # Kích thước cửa sổ
my_w.title("www.8syncdev.com") # Thêm tiêu đề

months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun'] # Các
tùy chọn
cb1 = ttk.Combobox(my_w, values=months, width=7) #
Combobox
cb1.grid(row=1, column=1, padx=10, pady=20) # Thêm vào
grid
cb1.set('Apr') # Tùy chọn mặc định

my_w.mainloop() # Giữ cửa sổ mở
```

Giải thích:

- `cb1.set('Apr')`: Đặt một tùy chọn mặc định cho Combobox.
- Chúng ta có thể sử dụng chỉ số của tùy chọn `cb1.current(2)` để đặt nó làm lựa chọn mặc định.
- Để lấy giá trị được chọn, chúng ta sử dụng `cb1.get()`.
- Để lấy chỉ số của tùy chọn được chọn, chúng ta sử dụng `cb1.current()`.

Ví dụ:

```
import tkinter as tk
from tkinter import ttk

my_w = tk.Tk()
my_w.geometry("300x150") # Kích thước cửa sổ
my_w.title("www.8syncdev.com") # Thêm tiêu đề
```

```
def my_upd1():
    cb1.set('Apr') # Cập nhật lựa chọn thành 'Apr'
    l1.config(text=cb1.get() + ':' + str(cb1.current())) #
    Hiển thị giá trị và chỉ số

months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun']
cb1 = ttk.Combobox(my_w, values=months, width=7)
cb1.grid(row=1, column=1, padx=10, pady=20)

b1 = tk.Button(my_w, text="set('Apr')", command=lambda:
my_upd1())
b1.grid(row=1, column=2)

l1 = tk.Label(my_w, text='Month')
l1.grid(row=1, column=3)

my_w.mainloop()
```

Xóa lựa chọn của Combobox:

```
def my_upd1():
    # cb1.set('') # Xóa lựa chọn
    cb1.delete(0, 'end') # Xóa lựa chọn
    l1.config(text=cb1.get() + ':' + str(cb1.current())) #
    Hiển thị giá trị và chỉ số
```

Loại bỏ lựa chọn và từ danh sách giá trị:

```
def my_delete():
    my_new = []
    for opt in cb1['values']:
        if opt != cb1.get():
            my_new.append(opt)
    cb1['values'] = my_new
    cb1.delete(0, 'end')
```

Sự kiện thay đổi của Combobox:

```
def my_upd(*args):
    l1.config(text=sel.get() + " : " + str(cb1.current()))

sel = tk.StringVar()
cb1 = ttk.Combobox(my_w, values=months, width=7,
textvariable=sel)
cb1.grid(row=1, column=1, padx=10, pady=20)
```

```
l1 = tk.Label(my_w, text='Month')
l1.grid(row=1, column=2)

sel.trace('w', my_upd)
```

Thêm tùy chọn vào Combobox:

```
def my_insert():
    if e1.get() not in cb1['values']:
        cb1['values'] += (e1.get(),)
        e1.delete(0, 'end')

e1 = tk.Entry(my_w, bg='Yellow', width=10)
e1.grid(row=1, column=3)

b1 = tk.Button(my_w, text='Add', command=lambda:
my_insert())
b1.grid(row=1, column=4)
```

Ví dụ hoàn chỉnh và các tùy chọn khác:

```
# Code hoàn chỉnh và các tùy chọn khác được đề cập ở trên.
```