Các Thuộc Tính của Tkinter: Các Khối Xây Dựng của Giao Diện Tkinter

Trong Tkinter, các widget có một số thuộc tính có thể được sử dụng để tùy chỉnh hình dạng và hành vi của chúng.

Các thuộc tính này có thể được thiết lập khi widget được tạo, hoặc chúng có thể được thay đổi động bất kỳ lúc nào.

Ví dụ về Các Thuộc Tính

Trong đoạn mã sau, khi tạo nút bt1, chúng tôi đã sử dụng một số thuộc tính và gán giá trị cho chúng:

```
bt1 = tk.Button(my_w, text='Tôi là một Nút',
bg='lightgreen', command=lambda: my_upd())
```

Trong đó:

- text: Chuỗi để hiển thị trên nút
- bg: Màu nền của nút
- command: Xử lý sự kiện khi nút được nhấp

Sử dụng các thuộc tính trên, đầu ra của cửa sổ của chúng tôi với nút là như sau:

Tạo Nút với một tập hợp các thuộc tính

Danh Sách Thuộc Tính Phổ Biến

Dưới đây là một số thuộc tính của widget Tkinter phổ biến được sử dụng rộng rãi:

- text: Văn bản được hiển thị trên widget.
- background: Màu nền của widget.
- foreground: Màu văn bản của widget.
- font: Phông chữ được sử dụng để hiển thị văn bản trên widget.
- relief: Loại đường biên xung quanh widget.
- state: Trạng thái của widget, có thể là "normal", "disabled", hoặc "active".
- cursor: Con trỏ hiển thị khi chuột di chuyển qua widget.
- image: Hình ảnh được hiển thị trên widget.
- command: Hàm được gọi khi widget được nhấp.

Danh Sách Thuộc Tính của Nút

Dưới đây là một danh sách đầy đủ các thuộc tính của một widget Nút, chúng ta có thể sử dụng chúng dựa trên yêu cầu của chúng ta:

```
['activebackground', 'activeforeground', 'anchor', 'background', 'bd', 'bg',
'bitmap', 'borderwidth', 'command', 'compound', 'cursor', 'default',
```

```
'disabledforeground', 'fg', 'font', 'foreground', 'height',
'highlightbackground', 'highlightcolor', 'highlightthickness', 'image',
'justify', 'overrelief', 'padx', 'pady', 'relief', 'repeatdelay',
'repeatinterval', 'state', 'takefocus', 'text', 'textvariable', 'underline',
'width', 'wraplength']
```

Cấu Hình giá trị của Thuộc Tính

Để đặt giá trị của một thuộc tính, bạn có thể sử dụng phương thức `configure()`. Phương thức này nhận tên thuộc tính và giá trị dưới dạng các đối số từ khóa. Ví dụ, đoạn mã sau đặt thuộc tính `text` của một widget `Button` thành "Nhấn vào tôi!":

```
button.configure(text="Nhấn vào tôi!")
```

Chúng ta cũng có thể đặt nhiều thuộc tính cùng một lúc bằng cách sử dụng các tên thuộc tính và giá trị cho phương thức `configure()`. Ví dụ, đoạn mã sau đặt các thuộc tính `text`, `background`, và `foreground` của một widget `Button`:

```
button.configure(text="Nhấn vào tôi!", background="red",
foreground="white")
```

Hoặc chúng ta có thể tạo một từ điển của các thuộc tính với các giá trị của chúng và sử dụng như vậy.

```
set1 = {'fg': 'red', 'font': ['Arial', 18, 'bold'], 'text':
'King & Queen', 'bg': 'lightgreen'}
button.config(set1)
```

Lấy Giá Trị của Thuộc Tính

Chúng ta có thể hiển thị giá trị của một thuộc tính cụ thể như sau:

```
print(button1['width'])  # Giá trị của thuộc tính = 20
```

Sử dụng `cget()`:

```
print(button.cget('bg'))  # Sử dụng khóa hoặc thuộc tính để
lấy giá trị
```

Liệt Kê Tất Cả Các Giá Trị của Tất Cả Các Thuộc Tính của Một Widget

```
for options in button1.config():
    print(options + ": " + str(button1[options]))
```

Kiểm tra tính sẵn có của thuộc tính cho widget

```
if "bitmap" in l2.config():
    print("Thuộc tính có sẵn")
else:
    print("Thuộc tính không có sẵn")
```

Ví dụ I: Bật hoặc Tắt một Nút

```
import tkinter as tk

my_w = tk.Tk()
my_w.geometry('350x150')
my_w.title('www.8syncdev.com')

def my_upd():
    if bt2['state'] == 'disabled':
        bt2['state'] = 'normal'  # Bật nút
        bt1['text'] = 'Tắt'  # Cập nhật văn bản
    else:
        bt2['state'] = 'disabled'  # Tắt nút
        bt1['text'] = 'Bật'  # Cập nhật văn bản

bt1 = tk.Button(my_w, text='Bật',

 font=22, bg='lightyellow', command=lambda: my_upd())
bt1.grid(row=0, column=0, padx=50, pady=20)

bt2 = tk.Button(my_w, text='Gửi', font=22,
state='disabled', activebackground='lightgreen')
bt2.grid(row=0, column=1, padx=20, pady=20)

my_w.mainloop()
```

Ví dụ II: Thêm một tập hợp các thuộc tính vào các nút

```
import tkinter as tk

my_w = tk.Tk()
my_w.geometry("450x150")

students = {1: 'King', 2: 'Queen', 3: 'Jack', 4: 'Ron',
```

```
            5: 'Alex', 6: 'Ravi', 7: 'Mike', 8: 'Teek', 9:
'Pink', 10: 'Geek'}
set1 = {'fg': 'red', 'bg': 'lightgreen'}  # Tập hợp các
thuộc tính với các giá trị
col = 0  # Giá trị cột khởi đầu
for j in students:
    e = tk.Button(my_w, text=students[j])
    e.grid(row=1, column=col, padx=4, pady=20)
    if j % 3 == 0:  # Chia hết cho 3
        e.config(set1)  # Cập nhật các thuộc tính
    col += 1  # Tăng giá trị cột
my_w.mainloop()
```

Trên đây là một số ví dụ về cách sử dụng và tùy chỉnh các thuộc tính của các widget trong Tkinter. Bằng cách tận dụng các thuộc tính này, bạn có thể tạo ra các giao diện người dùng phong phú và linh hoạt.