

Python Tkinter Canvas

Canvas trong Python Tkinter

Canvas là một phần của thư viện Tkinter của Python, cho phép tạo đồ họa bằng cách sử dụng các hình dạng như đường, hình chữ nhật, hình tròn, đa giác, cung, v.v. Nó cũng có thể chứa các loại widget khác.

Kích thước của Canvas

- **Chiều cao (height):** Độ dài theo chiều dọc của Canvas.
- **Chiều rộng (width):** Độ dài theo chiều ngang của Canvas.

Tọa độ X và Y

- **X:** Tọa độ ngang (hoành độ).
- **Y:** Tọa độ dọc (tung độ).

Các loại hình dạng có thể tạo trên Canvas

1. Tạo Text

```
import tkinter as tk

my_w = tk.Tk()
my_c = tk.Canvas(my_w, width=350, height=150)
my_c.pack()

my_c.create_text(175, 40, fill='#c0c0c0', font="Times 22
bold", text="Welcome to 8syncdev.com")

my_w.mainloop()
```

2. Tạo Đường thẳng (Line)

```
import tkinter as tk

my_w = tk.Tk()
my_c = tk.Canvas(my_w, width=100, height=100)
my_c.pack()

x1 = 0
y1 = 50
x2 = 90
y2 = 50

my_c.create_line(x1, y1, x2, y2, fill="#ff00ff")
```

```
my_w.mainloop()
```

3. Tạo Hình chữ nhật (Rectangle)

```
import tkinter as tk

my_w = tk.Tk()
my_c = tk.Canvas(my_w, width=200, height=200)
my_c.pack()

my_c.create_rectangle(80, 80, 110, 110, fill='#c0c0c0')

my_w.mainloop()
```

4. Tạo Hình oval (Oval)

```
import tkinter as tk

my_w = tk.Tk()
my_c = tk.Canvas(my_w, width=150, height=150)
my_c.pack()

my_c.create_oval(25, 25, 125, 125, fill='#c0c0c0')

my_w.mainloop()
```

5. Tạo Hình tròn (Circle)

```
import tkinter as tk

my_w = tk.Tk()
my_c = tk.Canvas(my_w, width=200, height=200)
my_c.pack()

def my_circle(my_canvas, x, y, r):
    my_id = my_canvas.create_oval(x-r, y-r, x+r, y+r,
    fill='#c0c0c0')
    return my_id

my_circle(my_c, 60, 60, 15)

my_w.mainloop()
```

6. Tạo Hình ảnh (Image)

```
import tkinter as tk
from PIL import Image, ImageTk

my_w = tk.Toplevel()
my_c = tk.Canvas(my_w, width=200, height=200)
my_c.pack()

f_name = tk.PhotoImage(file='icon-dwn.png')
my_img = my_c.create_image(50, 50, image=f_name)

my_w.mainloop()
```

7. Tạo Cung (Arc)

```
import tkinter as tk

my_w = tk.Tk()
my_c = tk.Canvas(my_w, width=150, height=150)
my_c.pack()

my_c.create_arc(10, 10, 140, 140, start=15, extent=340,
fill='#c0c0c0')

my_w.mainloop()
```

8. Tạo Đa giác (Polygon)

```
import tkinter as tk

my_w = tk.Tk()
my_c = tk.Canvas(my_w, width=150, height=150)
my_c.pack()

my_c.create_polygon(5, 40, 15, 120, 130, 70, 35, 5,
fill='#c0c0c0')

my_w.mainloop()
```

9. Tạo Bitmap

```
import tkinter as tk

my_w = tk.Tk()
```

```
my_c = tk.Canvas(my_w, width=200, height=150)
my_c.pack()

my_c.create_bitmap(50, 50, bitmap='question')

my_w.mainloop()
```

Thêm Bitmap vào Canvas

Các tùy chọn

- **fill**: Màu sắc để tô màu cho hình dạng.
- **width**: Độ rộng của đường viền của hình dạng.
- **outline**: Màu của viền (đối với hình chữ nhật, oval, v.v.).
- **dash**: Vẽ đường kẻ nét đứt thay vì nét liền.
- **stipple**: Mẫu tô cho hình dạng thay vì màu tô tẹt.
- **state**: Trạng thái của hình dạng: bình thường, vô hiệu hóa hoặc ẩn.
- **activefill**: Màu khi hình dạng được nhấn chuột.
- **activeoutline**: Màu viền khi hình dạng được nhấn chuột.
- **activedash**: Giống như dash khi hình dạng được nhấn chuột.
- **activewidth**: Giống như width khi hình dạng được nhấn chuột.
- **disableddash**: Khi hình dạng bị vô hiệu hóa (state=disabled).

Các tùy chọn khác

- **disabledfill**
- **disabledoutline**
- **disabledoutlinestipple**
- **disabledstipple**
- **disabledwidth**
- **relief**: Các giá trị có thể là flat, groove, raised, ridge, solid, hoặc sunken.

Tất cả các mục trên Canvas

```
import tkinter as tk

my_w = tk.Tk()
width, height = 410, 410
d = str(width) + "x" + str(height + 40)
my_w.geometry(d)

c1 = tk.Canvas(my_w, width=width-10, height=height-10,
bg='lightgreen')
c1.grid(row=1, column=0, padx=5, pady=10)
```

```

c1.create_text(290, 50, fill='#c0c0c0', font='Time 16
bold', text='welcome to 8syncdev')
my_rect = c1.create_rectangle

(60, 20, 180, 80)
c1.create_oval(180, 90, 390, 200, fill='gray', dash=(25, 5,
1, 10), activedash=(50, 10), stipple='gray75', width=5,
activestipple='gray25')
c1.create_line(10, 10, 50, 50, width=1, arrow='last')
c1.create_arc(50, 85, 180, 240, start=45, extent=135,
fill="red")
c1.create_polygon(35, 160, 70, 390, 180, 310, 260, 350,
200, 250, fill='yellow')
f_name = tk.PhotoImage(file='D:\\top2.png')
my_img = c1.create_image(315, 375, image=f_name)

my_w.mainloop()

```

Sử dụng Canvas như Hình nền

Dưới đây là một ví dụ về cách sử dụng Canvas như hình nền, thay đổi hình ảnh background:

```

import tkinter as tk

path_image = 'your_path_here/' # Đường dẫn đến thư mục lưu
trữ hình ảnh

my_w = tk.Tk()

bg = tk.PhotoImage(file=path_image + 'bg2.png')
c1 = tk.Canvas(my_w, width=1000, height=500)
c1.grid(row=0, column=0, columnspan=4, rowspan=4,
sticky='nw', padx=0)
c1.create_image(0, 0, image=bg, anchor='nw')

my_w.mainloop()

```

Canvas dọc để sử dụng như Button

```

import tkinter as tk
import tkinter.font as tkfont

my_w = tk.Tk()
my_w.geometry("400x250")
font = tkfont.nametofont("TkDefaultFont")

my_str = "8syncdev" # Chuỗi hiển thị trên Button

```

```

height = font.measure(my_str) + 10 # Sử dụng làm chiều cao
của Canvas
width = font.metrics()['linespace'] + 10 # Sử dụng làm
chiều rộng của Canvas

canvas = tk.Canvas(my_w, height=height, width=width,
background="SystemButtonFace", borderwidth=2,
relief='raised')
canvas.create_text((4, 4), angle='90', anchor='ne',
text=my_str, fill='SystemButtonText', font=font)
canvas.grid(row=0, column=0, padx=20, pady=20)

l1_str = tk.StringVar(value='Welcome') # Thông điệp khi
nhấn chuột vào Canvas
l1 = tk.Label(my_w, textvariable=l1_str, font=('Times', 20,
'normal'))
l1.grid(row=0, column=1, padx=5, pady=20)

def my_task(): # Thực hiện khi chuột được thả ra
    canvas.configure(relief='sunken')
    l1_str.set('Hi and welcome to 8syncdev')

canvas.bind("<ButtonPress-1>", lambda x: my_task()) # Khi
chuột được nhấn
canvas.bind("<ButtonRelease-1>", lambda x:
canvas.configure(relief='raised')) # Khi chuột được thả ra

my_w.mainloop()

```