

# The Canvas Widget

Thomas Schwarz, SJ

# Canvas Widget

- A simple, but powerful widget that allows you to draw and even animate
- Canvases are objects
- Have their own coordinate system

# Canvas Widget

- Coordinates uses graphics coordinate conventions
  - x coordinate from left to right
  - y coordinate from top to bottom



# Canvas Widget

- Creating a canvas widget:
  - Specify parent window, height, and width
  - Then display it
  - Can use Canvas methods to create elements in a canvas
    - `create_rectangle`, `create_oval`, `create_line`,  
`create_polygon`, `create_image`, `create_text`, `create_arc`
  - Can set many options such as boundary, background color, ...

# Canvas Widget

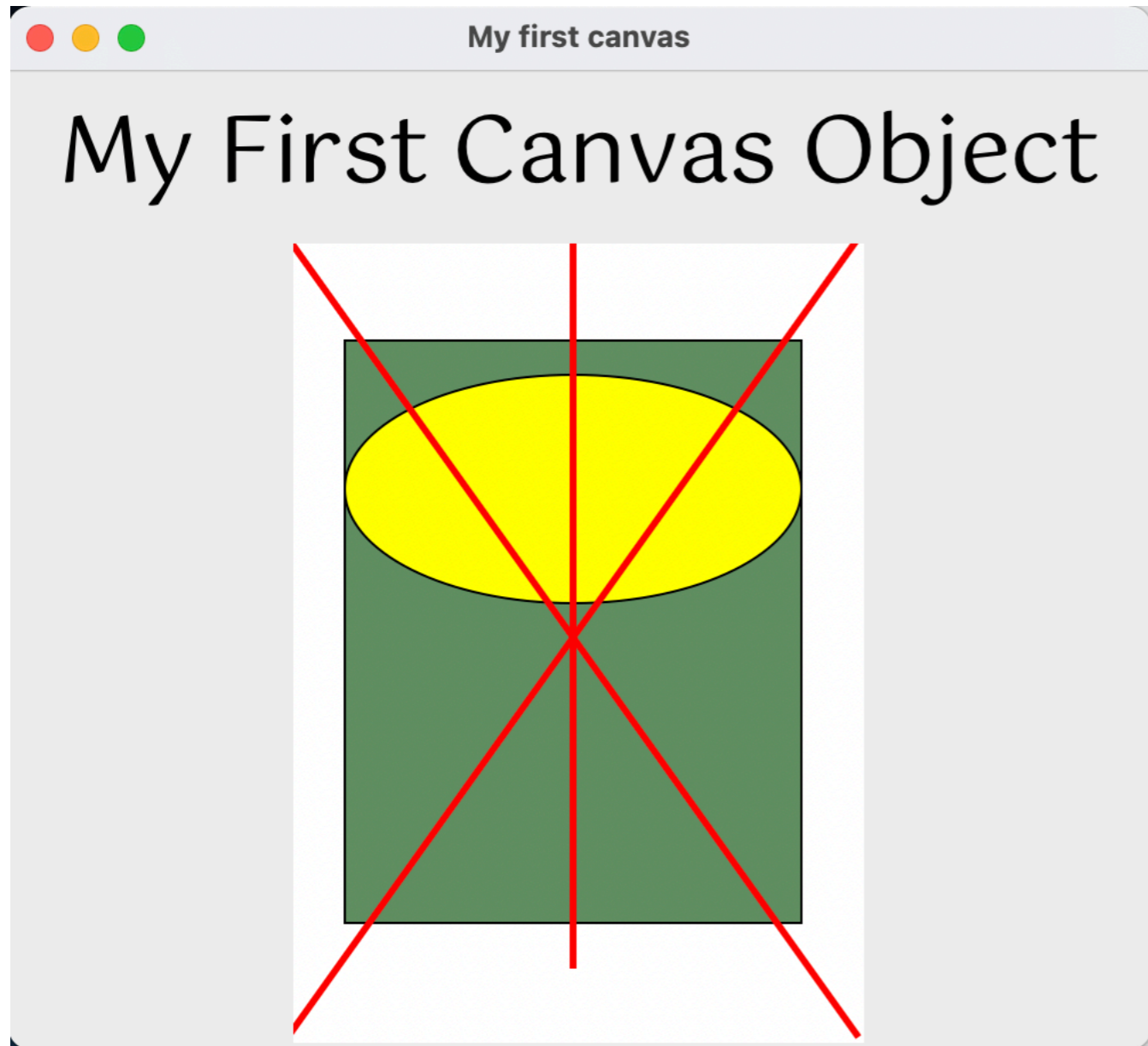
- Canvas objects defined by canvas coordinates:
  - Upper-left, Lower-right
- Canvas objects have colors:
  - Either use names: “Red”, “Yellow”, ...
  - Or use RGB-values with leading hashtag
    - “#2328f7”
    - Hexadecimal system: “#ffffff” is white, “#000000” is black

# Canvas Widget

```
import tkinter as tk

class My_first:
    def __init__(self):
        self.top = tk.Tk()
        self.top.title("My first graphics application")
        self.define_widgets()
        self.top.mainloop()
    def define_widgets(self):
        my_labell1 = tk.Label( text="Canvas Object", justify=tk.CENTER, padx=20)
        my_labell1.pack(side="top")
        my_canvas = tk.Canvas(self.top, width = 250, height = 350)
        my_canvas.pack(side="bottom")
        my_canvas.create_rectangle(25, 45, 225, 300, fill = "#639164")
        my_canvas.create_oval(25, 60, 225, 160, fill="Yellow")
        my_canvas.create_line(125,0, 125, 320, fill = "Red", width=3)
        my_canvas.create_line(0,0,250, 350, fill="Red", width = 3)
        my_canvas.create_line(250,0,0,350,fill="Red", width = 3)
mf = My_first()
```

# Canvas Widget



- Canvas in white 250 x 350
- Oval has bounding box 25, 60, 225, 160

# Canvas Widget

```
import tkinter as tk
```

```
class My_first_canvas:  
    def __init__(self):  
        self.top = tk.Tk()  
        self.top.title("My first graphics application")  
        self.define_widgets()  
        self.top.mainloop()  
  
    def define_widgets(self):  
        my_labell1 = tk.Label( text="Canvas Object", justify=tk.CENTER, padx=20)  
        my_labell1.pack(side="top")  
        my_canvas = tk.Canvas(self.top, width = 250, height = 350)  
        my_canvas.pack(side="bottom")  
        my_canvas.create_rectangle(25, 45, 225, 300, fill = "#639164")  
        my_canvas.create_oval(25, 60, 225, 160, fill="Yellow")  
        my_canvas.create_line(125,0, 125, 320, fill = "Red", width=3)  
        my_canvas.create_line(0,0,250, 350, fill="Red", width = 3)  
        my_canvas.create_line(250,0,0,350,fill="Red", width = 3)  
  
mf = My_first_canvas()
```

A label on top of the application



# Canvas Widget

```
import tkinter as tk

class My_first_canvas:
    def __init__(self):
        self.top = tk.Tk()
        self.top.title("My first graphics application")
        self.define_widgets()
        self.top.mainloop()

    def define_widgets(self):
        my_labell1 = tk.Label( text="Canvas Object", justify=tk.CENTER, padx=20)
        my_labell1.pack(side="top")
        my_canvas = tk.Canvas(self.top, width = 250, height = 350)
        my_canvas.pack(side="bottom")
        my_canvas.create_rectangle(25, 45, 225, 300, fill = "#639164")
        my_canvas.create_oval(25, 60, 225, 160, fill="Yellow")
        my_canvas.create_line(125,0, 125, 320, fill = "Red", width=3)
        my_canvas.create_line(0,0,250, 350, fill="Red", width = 3)
        my_canvas.create_line(250,0,0,350,fill="Red", width = 3)

mf = My_first_canvas()
```

Create a canvas of size 250x350

# Canvas Widget

```
import tkinter as tk
```

```
class My_first_canvas:
```

```
    def __init__(self):
```

```
        self.top = tk.Tk()
```

```
        self.top.title("My first canvas")
```

```
        self.define_widgets()
```

```
        self.top.mainloop()
```

```
    def define_widgets(self):
```

```
        my_labell1 = tk.Label( text="My First Canvas Object", justify=tk.CENTER, padx=20,
```

```
font=('Gotu',40))
```

```
        my_labell1.pack(side="top")
```

```
        my_canvas = tk.Canvas(self.top, width = 250, height = 350, bg='white')
```

```
        my_canvas.pack(side="bottom")
```

```
        my_canvas.create_rectangle(25, 45, 225, 300, fill = "#639164")
```

```
        my_canvas.create_oval(25, 60, 225, 160, fill="Yellow")
```

```
        my_canvas.create_line(125,0, 125, 320, fill = "Red", width=3)
```

```
        my_canvas.create_line(0,0,250, 350, fill="Red", width = 3)
```

```
        my_canvas.create_line(250,0,0,350,fill="Red", width = 3)
```

```
mf = My_first_canvas()
```

**Create a number of elements**

# Canvas Widget

```
import tkinter as tk

class My_first_canvas:
    def __init__(self):
        self.top = tk.Tk()
        self.top.title("My first canvas")
        self.define_widgets()
        self.top.mainloop()
    def define_widgets(self):
        my_labell1 = tk.Label( text="My First Canvas Object", justify=tk.CENTER, padx=20,
font=('Gotu',40))
        my_labell1.pack(side="top")
        my_canvas = tk.Canvas(self.top, width = 250, height = 350, bg='white')
        my_canvas.pack(side="bottom")
        my_canvas.create_rectangle(25, 45, 225, 300, fill = "#639164")
        my_canvas.create_oval(25, 60, 225, 160, fill="Yellow")
        my_canvas.create_line(125,0, 125, 320, fill = "Red", width=3)
        my_canvas.create_line(0,0,250, 350, fill="Red", width = 3)
        my_canvas.create_line(250,0,0,350,fill="Red", width = 3)
mf = My_first_canvas()
```

A hashtag color