

Graphical User Interfaces I

Thomas Schwarz, SJ
Marquette University

Graphics

- How do you interact with an application with Graphical User Interface?
 - Users click on elements
 - Application responds

Graphics Programming

- Application programming is event-based programming
 - You create the application with a set of widgets
 - You program how to react to events such as:
 - Mouse clicks
 - Keyboard events
 - ...

Information on TkInter

- Programming with TkInter is frustrating
 - Error messages tend to be cryptic
 - Finding errors is difficult
- Best manual to my knowledge
 - <https://infohost.nmt.edu/tcc/help/pubs/tkinter/tkinter.pdf>
- Most functions have a very large list of named parameters, so you better use it

How to create an application

- Import tkinter (comes with Python 3)
 - Create a root window
 - Call mainloop on it
 - A lot goes on under the hood
 - Window created with buttons for magnification, resizing, and stopping
 - mainloop starts waiting for events

How to create an application

- Can create application via a class
- Can create application via functions
- Minimum: create root window + call `mainloop()` to start event queue

How to create an application

```
from tkinter import *  
  
class My_first:  
    def __init__(self):  
        self.top = Tk()  
        self.top.mainloop()  
  
mf = My_first()
```

```
from tkinter import *  
  
root_window = Tk()  
root_window.mainloop()
```

Creating Widgets

- We now add widgets to the window
 - Two steps:
 - Creating the widget
 - Place widget
 - Three methods, which cannot be mixed
 - `pack()`
 - `grid()`
 - `place()`

Creating Widgets: Labels

- Labels are used to print something static
- Constructor needs the root window and the contents
- Contents can be of different types:
 - Text
 - Images
 - Images are created from gif or png files using photoimage
 - Warning: Need to guarantee that photoimage is not garbage collected

Label Widget

- Imperative version:

```
import tkinter as tk

root_window = tk.Tk()
root_window.title("My first graphics application")
my_label1 = tk.Label(root_window,
                     text="Python",
                     justify=tk.LEFT,
                     padx=20,
                     pady=50)

my_label1.pack(side="left")
logo = tk.PhotoImage(file="python_milwaukee.gif")
my_label2 = tk.Label(root_window, image=logo)
my_label2.pack(side="right")

root_window.mainloop()
```

Label Widget

- Imperative version:

```
import tkinter as tk

root_window = tk.Tk()
root_window.title("My first graphics application")
my_label1 = tk.Label(root_window, text="Python", justify="left", padx=20, pady=50)
my_label1.pack(side="left")
logo = tk.PhotoImage(file="python_milwaukee.gif")
my_label2 = tk.Label(root_window, image=logo)
my_label2.pack(side="right")

root_window.mainloop()
```



Label Widget

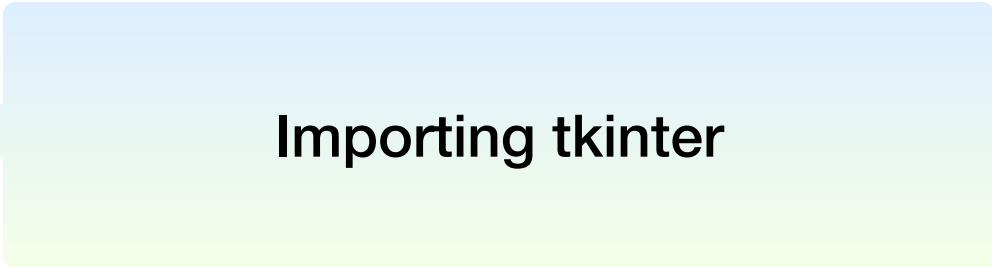
- Imperative version:

```
import tkinter as tk

root_window = tk.Tk()
root_window.title("My first graphics application")
my_label1 = tk.Label(root_window,
                    text="Python",
                    justify=tk.LEFT,
                    padx=20,
                    pady=50)

my_label1.pack(side="left")
logo = tk.PhotoImage(file="python_milwaukee.gif")
my_label2 = tk.Label(root_window, image=logo)
my_label2.pack(side="right")

root_window.mainloop()
```



Importing tkinter

Label Widget

- Imperative version:

```
import tkinter as tk

root_window = tk.Tk()
root_window.title("My first graphics application")
my_label1 = tk.Label(root_window,
                    text="Python",
                    justify=tk.LEFT,
                    padx=20,
                    pady=50)

my_label1.pack(side="left")
logo = tk.PhotoImage(file="python_milwaukee.gif")
my_label2 = tk.Label(root_window, image=logo)
my_label2.pack(side="right")

root_window.mainloop()
```

Creating a window

Label Widget

- Imperative version:

```
import tkinter as tk

root_window = tk.Tk()
root_window.title("My first graphics application")
my_label1 = tk.Label(root_window,
                    text="Python",
                    justify=tk.LEFT,
                    padx=20,
                    pady=50)

my_label1.pack(side="left")
logo = tk.PhotoImage(file="python_milwaukee.gif")
my_label2 = tk.Label(root_window, image=logo)
my_label2.pack(side="right")

root_window.mainloop()
```

Create a title bar

Label Widget

- Imperative version:

```
import tkinter as tk
```

```
root_window = tk.Tk()
```

```
root_window.title("My first graphics application")
```

```
my_label1 = tk.Label(root_window,
```

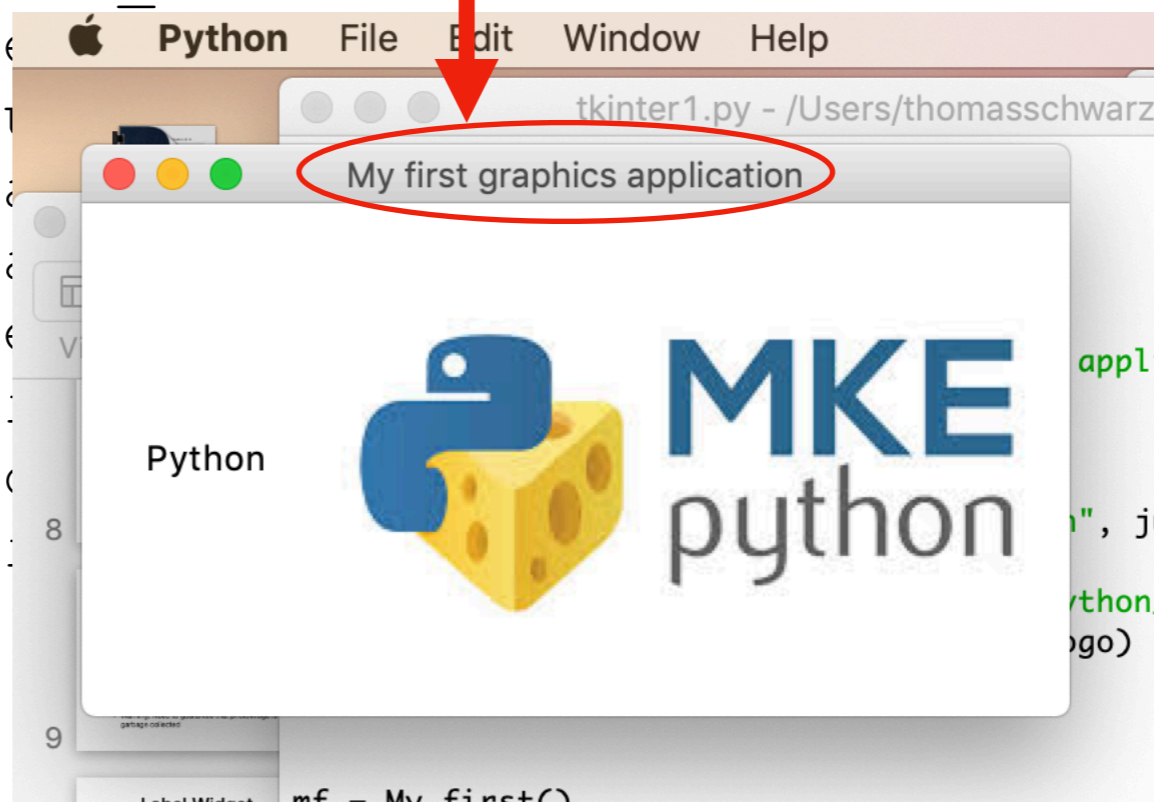
```
my_label1.pack(side="left")
```

```
logo = tk.PhotoImage(file="python.png")
```

```
my_label2 = tk.Label(root_window,
```

```
my_label2.pack(side="right")
```

```
root_window.mainloop()
```



Label Widget

- Imperative version:

```
import tkinter as tk

root_window = tk.Tk()
root_window.title("My first graphics application")
my_label1 = tk.Label(root_window,
                     text="Python",
                     justify=tk.LEFT,
                     padx=20,
                     pady=50)

my_label1.pack(side="left")
logo = tk.PhotoImage(file="python_milwaukee.gif")
my_label2 = tk.Label(root_window, image=logo)
my_label2.pack(side="right")

root_window.mainloop()
```

Create the text label:
text
justification
padding on x, y

Label Widget

- Imperative version:

```
import tkinter as tk

root_window = tk.Tk()
root_window.title("My first graphics application")
my_label1 = tk.Label(root_window,
                     text="Python",
                     justify=tk.LEFT,
                     padx=20,
                     pady=50)

my_label1.pack(side="left")
logo = tk.PhotoImage(file="python_milwaukee.gif")
my_label2 = tk.Label(root_window, image=logo)
my_label2.pack(side="right")

root_window.mainloop()
```

Need to pack:
Can pick where:
"left", "right", "top", "bottom"

Label Widget

- Imperative version:

```
import tkinter as tk

root_window = tk.Tk()
root_window.title("My first graphics application")
my_label1 = tk.Label(root_window,
                    text="python",
                    justify=tk.LEFT,
                    padx=20,
                    pady=50)

my_label1.pack(side="left")
logo = tk.PhotoImage(file="python_milwaukee.gif")
my_label2 = tk.Label(root_window, image=logo)
my_label2.pack(side="right")

root_window.mainloop()
```

Create a photo-image
needs to be gif or png

Important to give it a name or garbage
collection might take it away before it is used

Label Widget

- **Class method:** The whole application is in a

```
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_label1 = tk.Label(text="Python")
```

```
        my_label1.pack(side="left")
```

```
        self.logo = tk.PhotoImage(file="python_milwaukee.png")
```

```
        my_label2 = tk.Label(image=self.logo)
```

```
        my_label2.pack(side="right")
```

```
mf = My_first()
```

Label Widget

- **Class method:** The whole application is in a

```
import tkinter as tk
```

Constructor calls on widget construction

```
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_label1 = tk.Label(text="Python")
```

```
        my_label1.pack(side="left")
```

```
        self.logo = tk.PhotoImage(file="python_milwaukee.png")
```

```
        my_label2 = tk.Label(image=self.logo)
```

```
        my_label2.pack(side="right")
```

```
mf = My_first()
```

Label Widget

- Class method: The whole application is in a

```
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_label1 = tk.Label(text="Python")
```

```
        my_label1.pack(side="left")
```

```
        self.logo = tk.PhotoImage(file="python_milwaukee.png")
```

```
        my_label2 = tk.Label(image=self.logo)
```

```
        my_label2.pack(side="right")
```

```
mf = My_first()
```

Need to safeguard the image or it might vanish.

Label Widget

- Class method: The whole application is in a

```
import tkinter as tk
```

```
class My_first:
```

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

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

```
        self.top.title("My
```

```
        self.define_widgets
```

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

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

```
        my_label1 = tk.Label
```

```
        my_label1.pack(side="left")
```

```
        self.logo = tk.PhotoImage(file="python_milwaukee.png")
```

```
        my_label2 = tk.Label(image=self.logo)
```

```
        my_label2.pack(side="right")
```

```
mf = My_first()
```

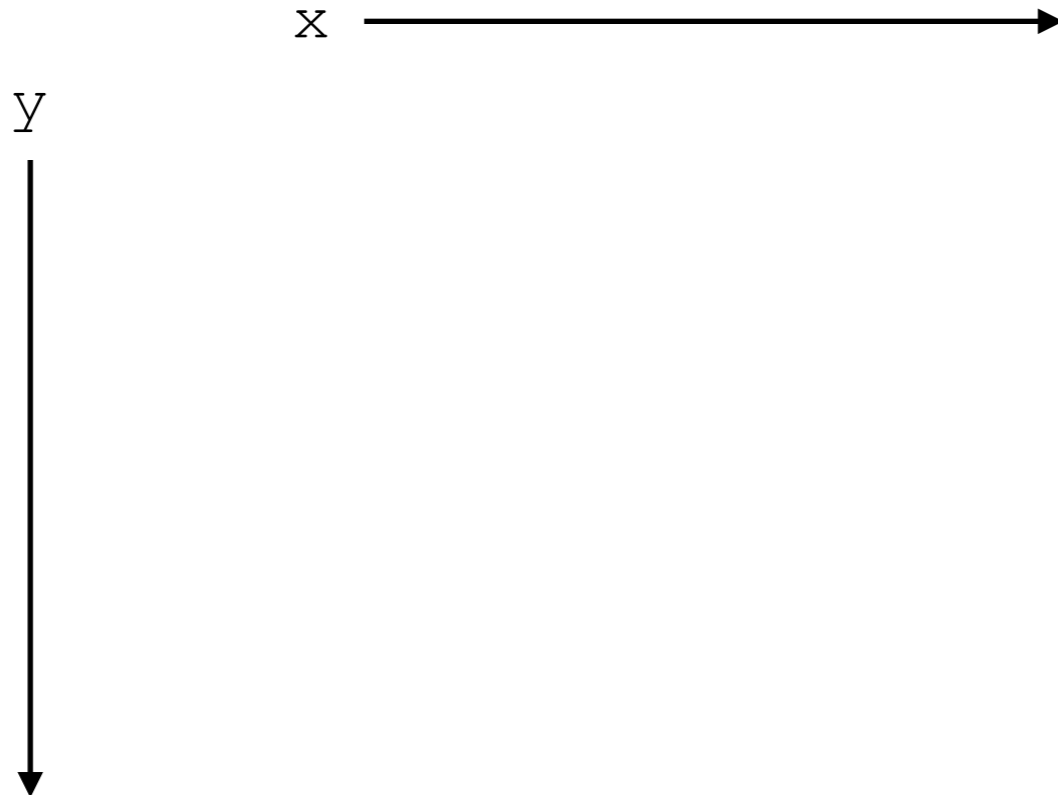


Canvas Widget

- Canvas widget allows generic graphics possibilities
- Can be used for simple animation
- 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

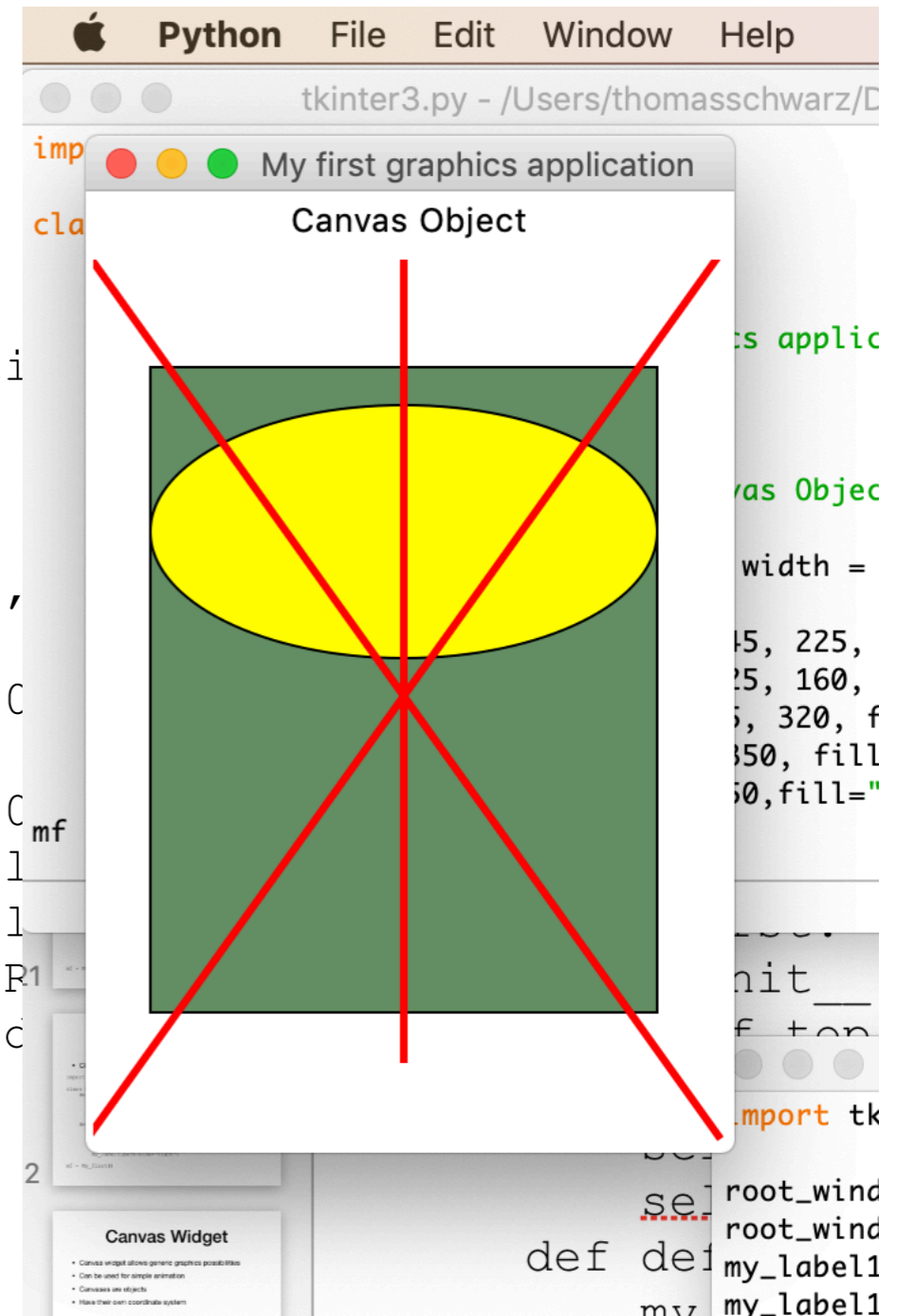
```
import tkinter as tk

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

```
import tkinter as tk

class My_first:
    def __init__(self):
        self.top = tk.Tk()
        self.top.title("My first graphics applicati
        self.define_widgets()
        self.top.mainloop()
    def define_widgets(self):
        my_labell1 = tk.Label( text="Canvas Object",
        my_labell1.pack(side="top")
        my_canvas = tk.Canvas(self.top, width = 250
        my_canvas.pack(side="bottom")
        my_canvas.create_rectangle(25, 45, 225, 300)
        my_canvas.create_oval(25, 60, 225, 160, fill="yellow")
        my_canvas.create_line(125,0, 125, 320, fill="red")
        my_canvas.create_line(0,0,250, 350, fill="red")
        my_canvas.create_line(250,0,0,350,fill="red")
mf = My_first()
```



Canvas Widget

A label on top of the application

```
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_labell = tk.Label( text="Canvas Object", justify=tk.CENTER, padx=20)
        my_labell.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

```
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()
```

Create a canvas of size 250x350

Canvas Widget

Create a number of elements

```
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

```
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()
```

A hashtag color