Paint app

<u>use for javascript canvas</u>

```
from tkinter import *
import tkinter.font


class PaintApp:

    # Stores current drawing tool used
    drawing_tool = "line"

    # Tracks whether left mouse is down
    left_but = "up"

    # x and y positions for drawing with pencil
    x_pos, y_pos = None, None

    # Tracks x & y when the mouse is clicked and released
    x1_line_pt, y1_line_pt, x2_line_pt, y2_line_pt = None, None, None, None

    # ---------- CATCH MOUSE UP ----------

    def left_but_down(self, event=None):
        self.left_but = "down"

        # Set x & y when mouse is clicked
        self.x1_line_pt = event.x
        self.y1_line_pt = event.y

    # ---------- CATCH MOUSE UP ----------

    def left_but_up(self, event=None):
        self.left_but = "up"

        # Reset the line
        self.x_pos = None
        self.y_pos = None

        # Set x & y when mouse is released
        self.x2_line_pt = event.x
        self.y2_line_pt = event.y

        # If mouse is released and line tool is selected
        # draw the line
        if self.drawing_tool == "line":
            self.line_draw(event)
        elif self.drawing_tool == "arc":
            self.arc_draw(event)
        elif self.drawing_tool == "oval":
            self.oval_draw(event)
        elif self.drawing_tool == "rectangle":
            self.rectangle_draw(event)
        elif self.drawing_tool == "text":
            self.text_draw(event)

    # ---------- CATCH MOUSE MOVEMENT ----------

    def motion(self, event=None):

        if self.drawing_tool == "pencil":
            self.pencil_draw(event)

    # ---------- DRAW PENCIL ----------

    def pencil_draw(self, event=None):
        if self.left_but == "down":

            # Make sure x and y have a value
            if self.x_pos is not None and self.y_pos is not None:
                event.widget.create_line(self.x_pos, self.y_pos,                        event.x, event.y,
smooth=TRUE)

            self.x_pos = event.x
            self.y_pos = event.y

    # ---------- DRAW LINE ----------

    def line_draw(self, event=None):

        # Shortcut way to check if none of these values contain None
        if None not in (self.x1_line_pt, self.y1_line_pt, self.x2_line_pt, self.y2_line_pt):
            event.widget.create_line(self.x1_line_pt, self.y1_line_pt, self.x2_line_pt, self.y2_line_pt,
smooth=TRUE, fill="green")

    # ---------- DRAW ARC ----------

    def arc_draw(self, event=None):

        # Shortcut way to check if none of these values contain None
        if None not in (self.x1_line_pt, self.y1_line_pt, self.x2_line_pt,
self.y2_line_pt):

            coords = self.x1_line_pt, self.y1_line_pt, self.x2_line_pt,
self.y2_line_pt

            # start : starting angle for the slice in degrees
            # extent : width of the slice in degrees
            # fill : fill color if needed
            # style : can be ARC, PIESLICE, or CHORD
            event.widget.create_arc(coords, start=0, extent=150,
                         style=ARC)

    # ---------- DRAW OVAL ----------

    def oval_draw(self, event=None):
        if None not in (self.x1_line_pt, self.y1_line_pt, self.x2_line_pt,
self.y2_line_pt):

            # fill : Color option names are here http://wiki.tcl.tk/37701
            # outline : border color
            # width : width of border in pixels

            event.widget.create_oval(self.x1_line_pt, self.y1_line_pt,
self.x2_line_pt, self.y2_line_pt,
                                    fill="midnight blue",
                                    outline="yellow",
                                    width=2)

    # ---------- DRAW RECTANGLE ----------

    def rectangle_draw(self, event=None):
        if None not in (self.x1_line_pt, self.y1_line_pt, self.x2_line_pt,
self.y2_line_pt):

            # fill : Color option names are here http://wiki.tcl.tk/37701
            # outline : border color
            # width : width of border in pixels

            event.widget.create_rectangle(self.x1_line_pt, self.y1_line_pt,          self.x2_line_pt,
self.y2_line_pt,
                fill="midnight blue",
                outline="yellow",
                width=2)

    # ---------- DRAW TEXT ----------

    def text_draw(self, event=None):
        if None not in (self.x1_line_pt, self.y1_line_pt):
            # Show all fonts available
            print(tkinter.font.families())

            text_font = tkinter.font.Font(family='Helvetica',
            size=20, weight='bold', slant='italic')

            event.widget.create_text(self.x1_line_pt, self.y1_line_pt,
                                  fill="green",
                                  font=text_font,
                                  text="WOW")

    def __init__(self, root):
        drawing_area = Canvas(root)
        drawing_area.pack()
        drawing_area.bind("<Motion>", self.motion)
        drawing_area.bind("<ButtonPress-1>", self.left_but_down)
        drawing_area.bind("<ButtonRelease-1>", self.left_but_up)

root = Tk()

paint_app = PaintApp(root)

root.mainloop()
```