Shapes and Graphics

Michael C. Hackett

Assistant Professor, Computer Science

Community College *of* Philadelphia

Lecture Topics

- Canvases
- Drawing Lines
- Drawing Rectangles
- Drawing Circles, Ovals and Arcs/Slices
- Drawing Polygons
- Drawing Text

Colors/Fonts

- Global Variable Names — Brown
- Local Variable Names — Lt Blue
- Literals — Blue
- Keywords — Orange
- Operators/Punctuation — Black
- Functions — Purple
- Parameters — Gold
- Comments — Gray
- Modules — Pink
- Object/Class Names — Green

Source Code — **Consolas**
Output — `Courier New`

Creating a Canvas

- A *canvas* is a widget that allows the drawing of lines and shapes.
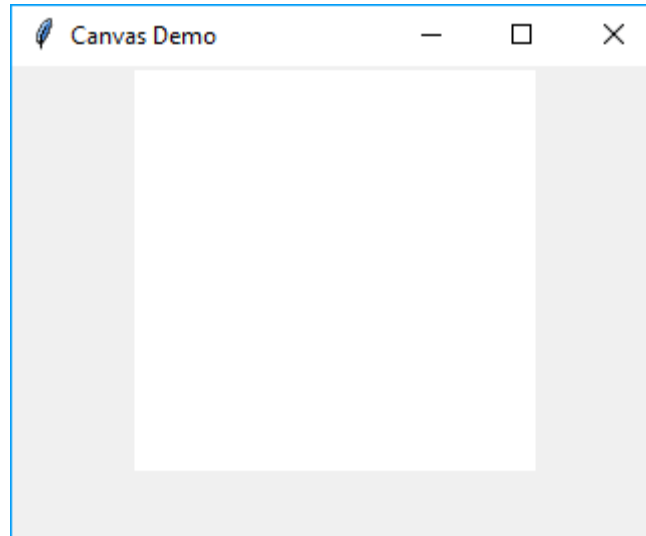
Creating a Canvas

- When creating a canvas, you must specify:
  - The window or frame it belongs to.
  - The canvas' width and height.

- You may optionally set the canvas' background color.
  - The background will be transparent if no background is set.

```
canvas = tkinter.Canvas(test_window,
                        width=200,
                        height=200,
                        background="white")
```

Creating a Canvas

```python
import tkinter

def main() :
    #Creates the window
    test_window = tkinter.Tk()
    #Creates the Canvas widget.
    canvas = tkinter.Canvas(test_window,
                            width=200,
                            height=200,
                            background="white")


    #Packs the canvas onto the window
    canvas.pack()
    #Enters the main loop, displaying the window
    #and waiting for events
    tkinter.mainloop()

main()
```
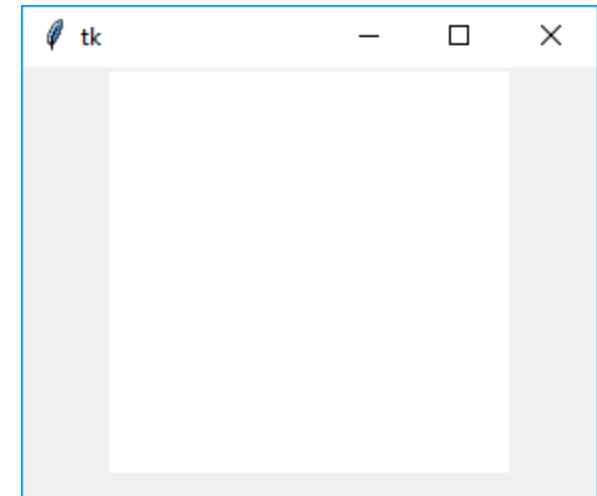
Tk Named Colors

Coordinate System

- Shapes will be drawn on a canvas using coordinates.

```
canvas = tkinter.Canvas(test_window, width=200, height=200, background="white")
```
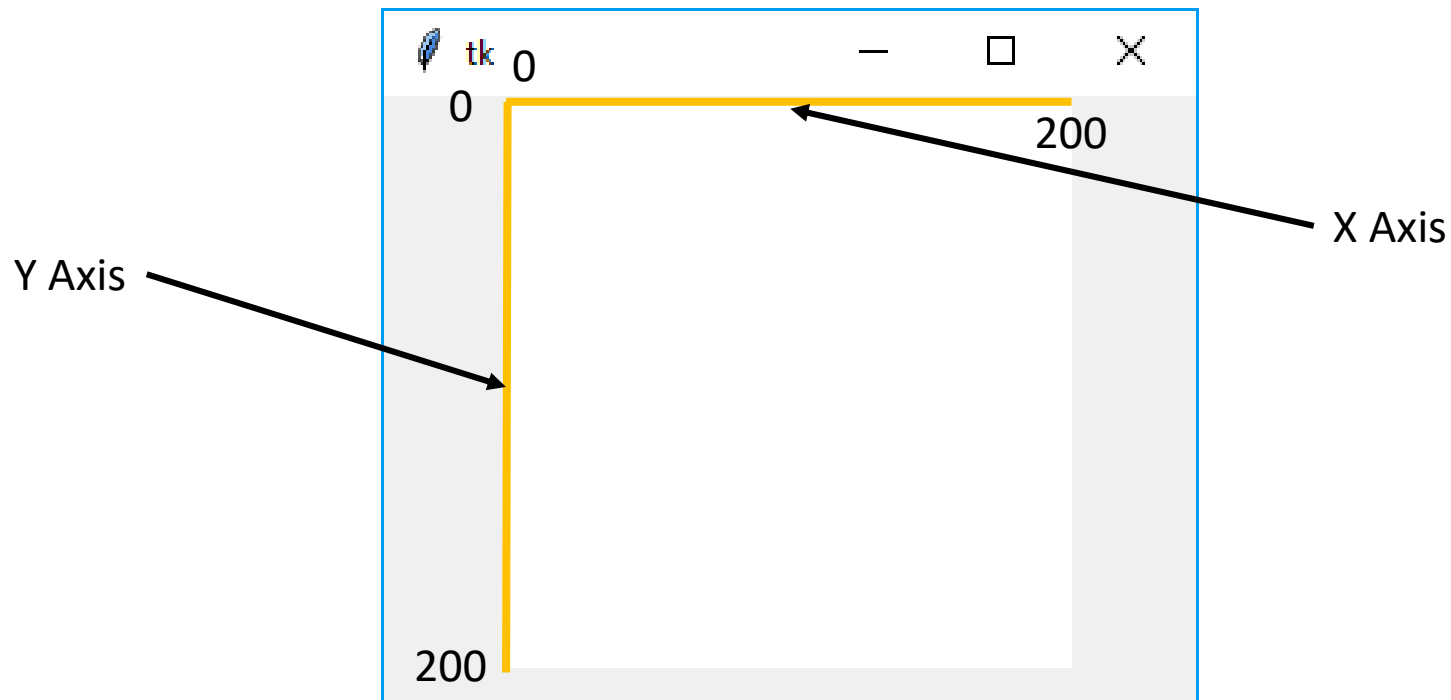
Drawing Lines

- The canvas object's create_line function will draw a straight line.

- A pair of coordinates is required.

```
canvas = tkinter.Canvas(test_window,
                        width=200,
                        height=200,
                        background="white")

canvas.create_line(50, 90, 160, 170)
```


Line Example
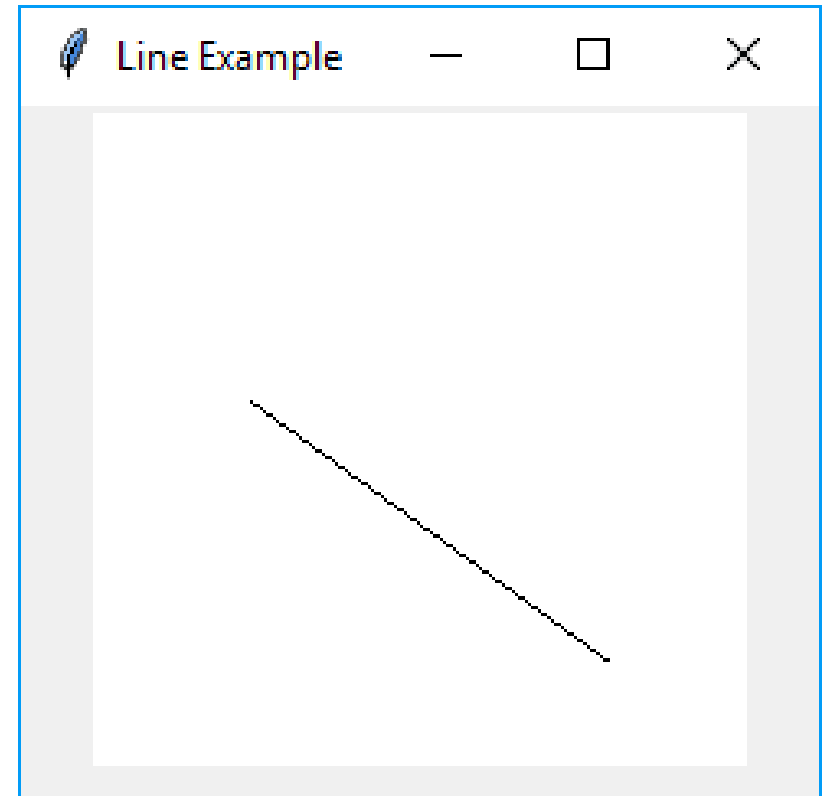
Drawing Lines


```
canvas.create_line(50, 90, 160, 170)
```

Drawing Lines

- Supplying additional coordinates will continue the line.

```
canvas.create_line(120, 10,
                   70, 90,
                   130, 50,
                   70, 180)
```

Drawing Rectangles and Squares

- The canvas object's create_rectangle function will draw rectangles/squares.

- Two pairs of coordinates are required.
  - The fill argument is optional.

```python
canvas = tkinter.Canvas(test_window,
                        width=200,
                        height=200,
                        background="white")


canvas.create_rectangle(40, 40,
                        100, 100,
                        fill="red")
```

Drawing Rectangles and Squares

- Lines and Shapes are drawn in the order they are created.
  - This allows drawing on top of previously drawn lines and shapes.

```
canvas.create_rectangle(40, 40,
                        100, 100,
                        fill="red")


canvas.create_rectangle(80, 80,
                        130, 110,
                        fill="blue")
```

Drawing Circles and Ovals

- The canvas object's create_oval function will draw circles/ovals.

- Two pairs of coordinates are required.
  - The fill argument is optional.

```
canvas = tkinter.Canvas(test_window,
                        width=200,
                        height=200,
                        background="white")


canvas.create_oval(20, 20,
                   120, 120,
                   fill="blue")
```
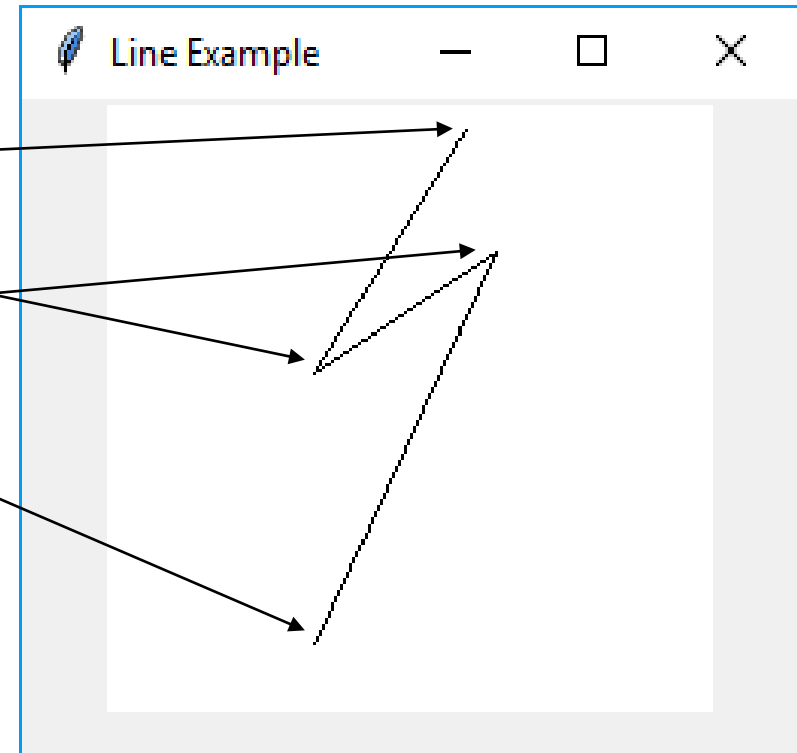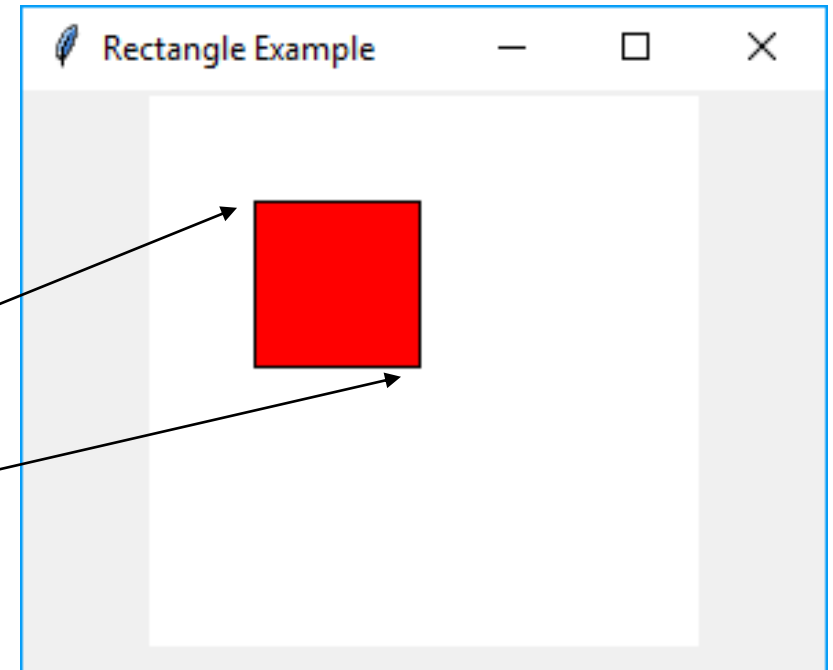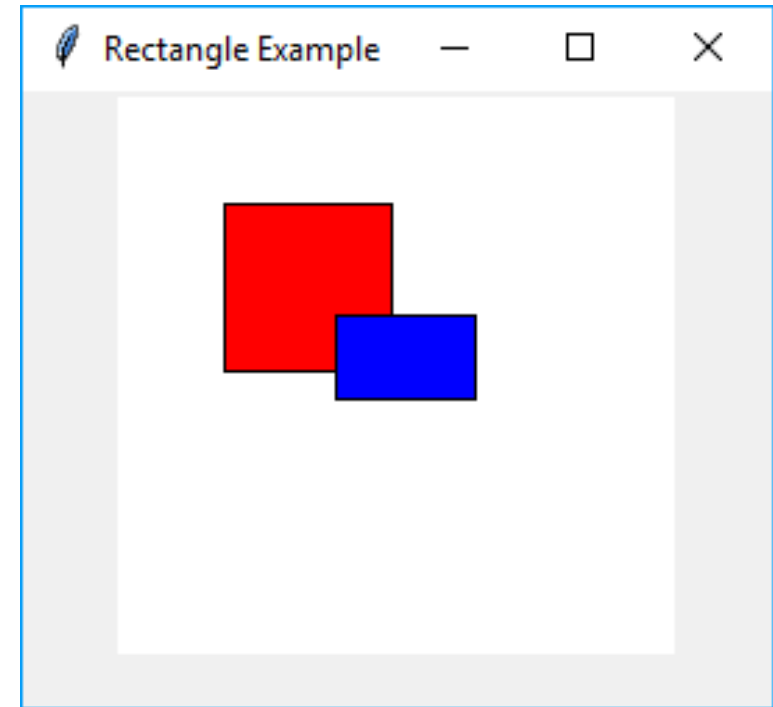
Drawing Circles and Ovals

- Lines and Shapes are drawn in the order they are created.
  - This allows drawing on top of previously drawn lines and shapes.

```
canvas.create_oval(20, 20,
                   120, 120,
                   fill="blue")


canvas.create_oval(80, 60,
                   180, 130,
                   fill="green")
```

Drawing Arcs/Slices

90°

180°   0° / 360°

270°
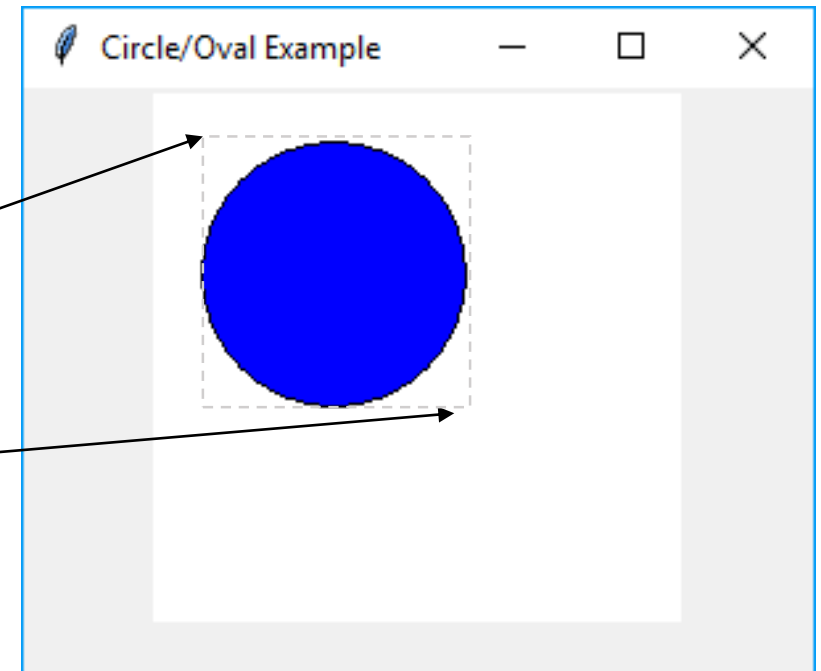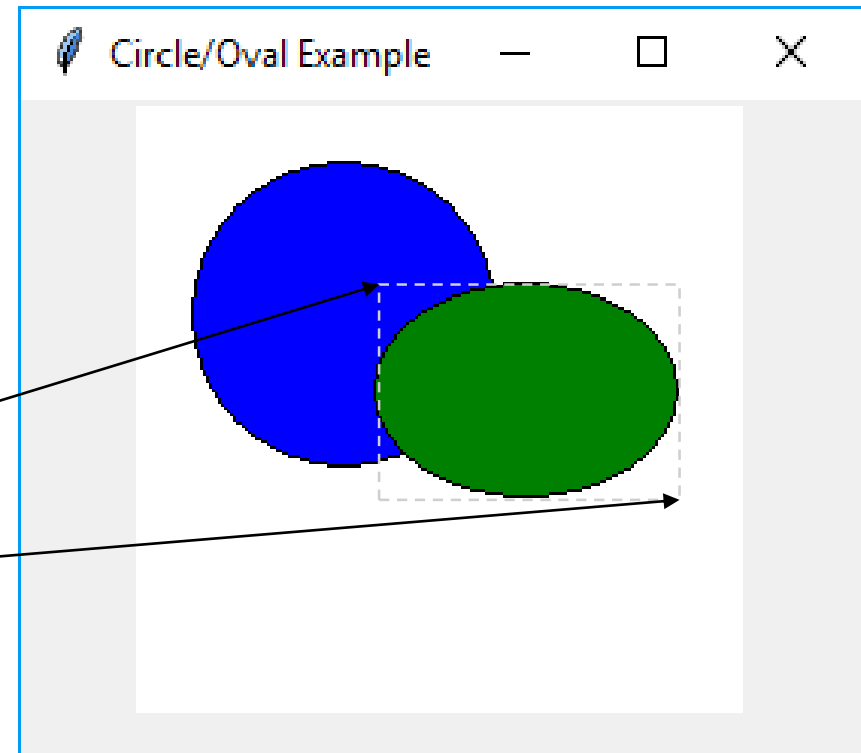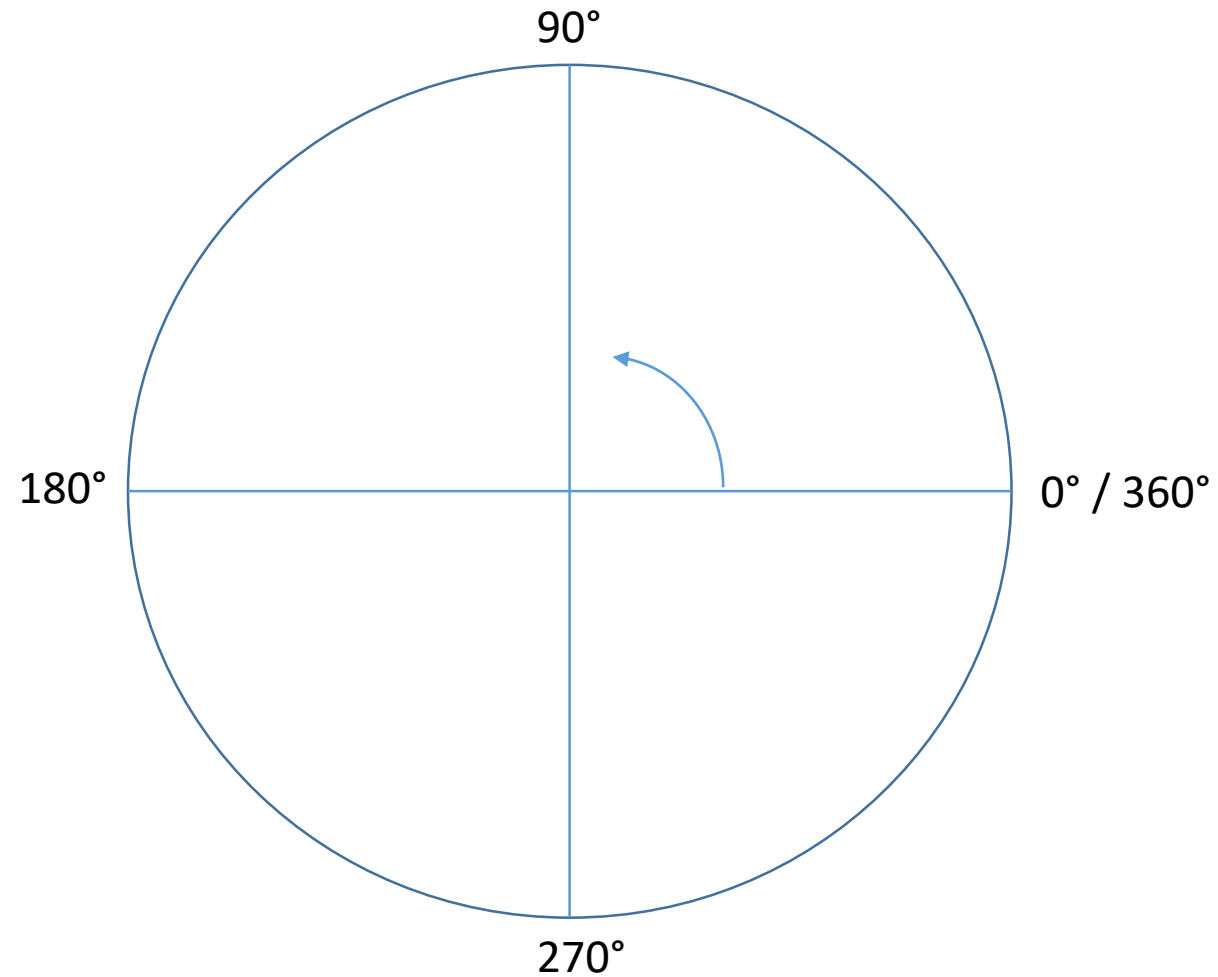
Drawing Arcs/Slices

- The canvas object's create_arc function will draw arcs/slices.
    - Two pairs of coordinates are required.
    - The start argument specifies the start degrees
    - The extent argument specifies degrees from the start
    - The fill argument is optional.

```
canvas = tkinter.Canvas(test_window,
                        width=200,
                        height=200,
                        background="white")


canvas.create_arc(10, 10,
                  190, 190,
                  start=45,
                  extent=30,
                  fill="red")
```
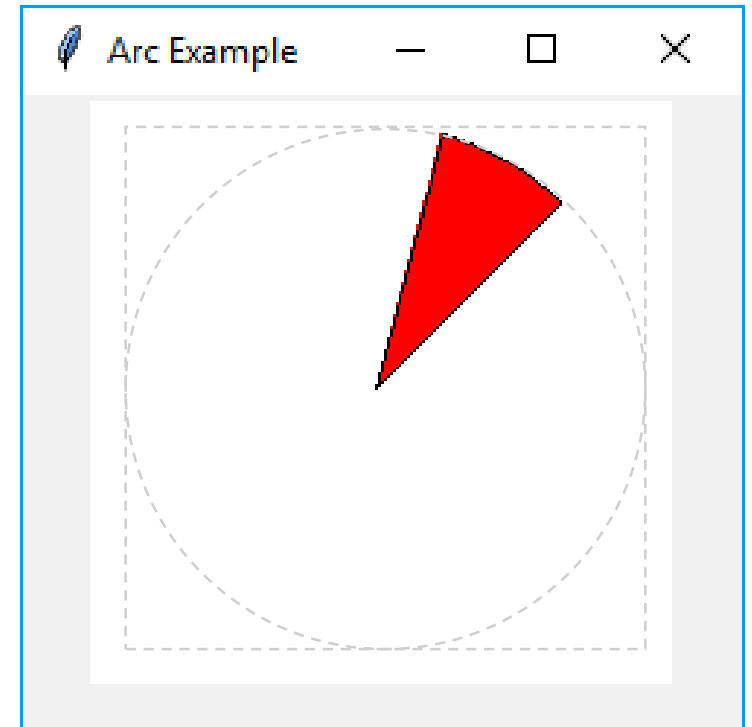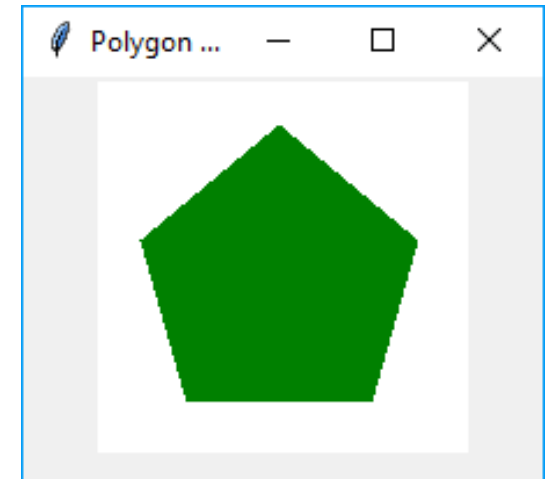
Drawing Polygons

- The canvas object's create_polygon function will draw regular and irregular shapes.
    - Undetermined pairs of coordinates are required.

    - The fill argument is optional.
        - Shape will be filled black if not provided
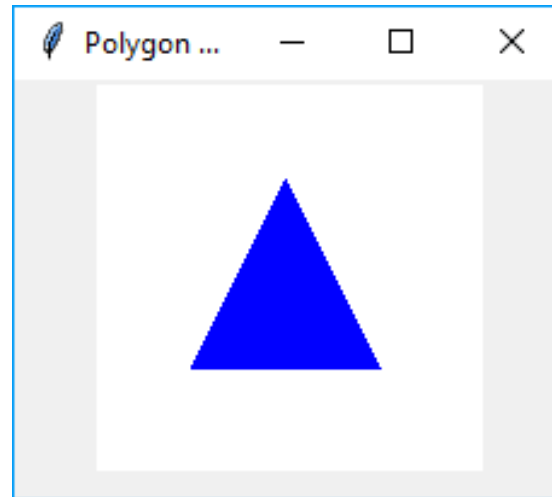
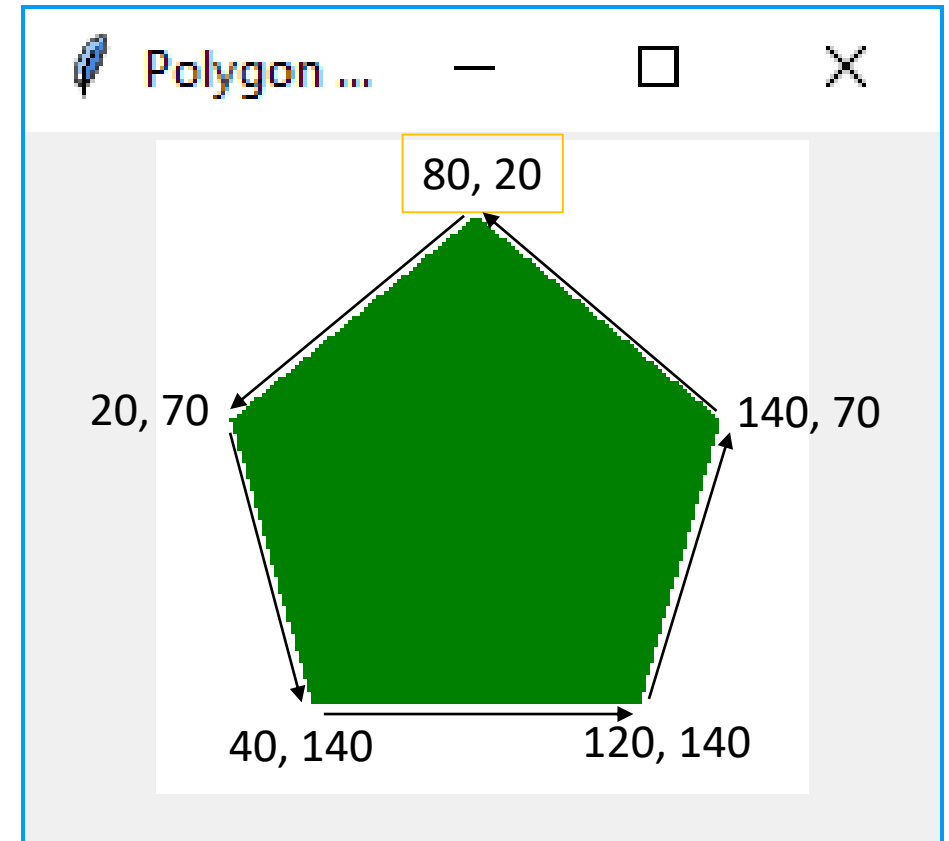# Drawing Polygons

```
canvas.create_polygon(40, 120,
                       120, 120,
                       80, 40,
                       40, 120,
                       fill="blue")
```
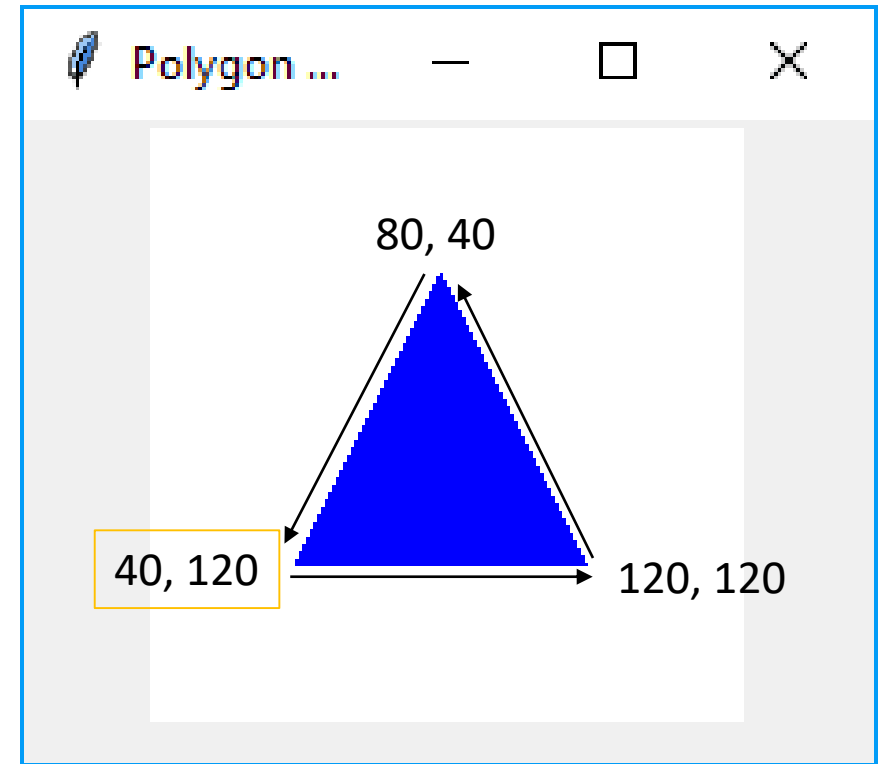
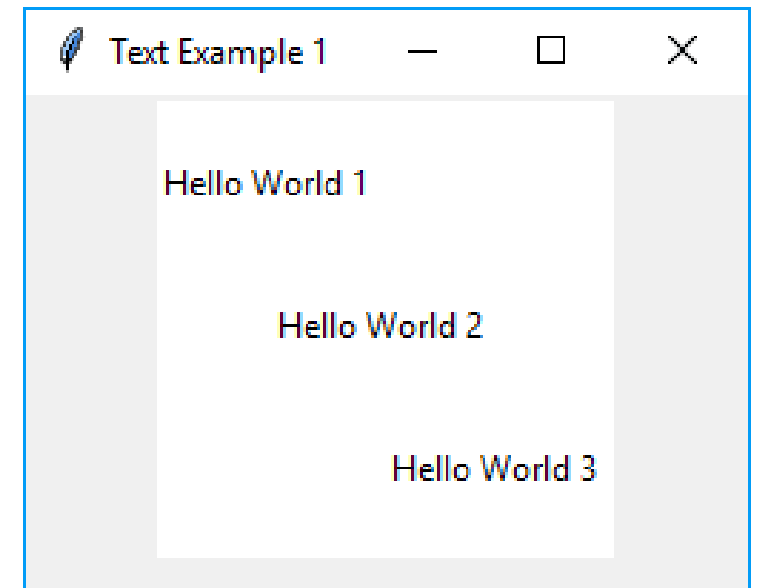# Drawing Polygons

```
canvas.create_polygon(80, 20,
                      20, 70,
                      40, 140,
                      120, 140,
                      140, 70,
                      80, 20,
                      fill="green")
```

Drawing Text

- The canvas object's create_text function will draw text.
  - One pair of coordinates is required.
    - The **center** of the text.

  - The text argument supplies the text to be drawn.

  - A fill argument is optional.
    - Will set the text's color.
    - Text will be black if not provided

Drawing Text

```
canvas.create_text(40, 30,
                   text="Hello World 1")


canvas.create_text(80, 80,
                   text="Hello World 2")


canvas.create_text(120, 130,
                   text="Hello World 3")
```