

# Overview

---

In this lesson, you will learn how to draw **polygons directly on the Tk canvas** using the turtle library. Polygons allow you to connect multiple line segments together to form a closed shape that can be **filled**, unlike regular canvas lines which only create outlines.

## Important Information

---

Polygons are one of the most flexible drawing tools on the canvas. They are similar to lines, but with an important difference:

- **Lines** connect points and draw only an outline
- **Polygons** connect points, automatically close the shape, and can be filled with color

### Canvas Coordinates in Turtle Graphics

When using the turtle canvas:

- $(0, 0)$  is the **center of the canvas**
- Positive **x** values move to the right
- Negative **x** values move to the left
- Positive **y** values move **down**
- Negative **y** values move **up**

This is different from standard Tkinter canvases, but turtle adjusts the coordinate system so the center is  $(0, 0)$ .

### Creating a Polygon

Polygons are created using:

```
canvas.create_polygon(x1, y1, x2, y2, x3, y3)
```

Each pair of numbers represents a point:

- $(x_1, y_1)$  → first corner
- $(x_2, y_2)$  → second corner
- $(x_3, y_3)$  → third corner
- Additional points can be added as needed

The canvas:

- Connects the points in order
- Automatically connects the last point back to the first
- Creates a closed shape

### How Polygons Compare to Lines

Lines:

```
canvas.create_line(x1, y1, x2, y2, x3, y3)
```

- Connect multiple segments
- Do **not** fill
- Only draw outlines

Polygons:

```
canvas.create_polygon(x1, y1, x2, y2, x3, y3)
```

- Connect the same way as lines
- Automatically close the shape
- Can be filled with color

### Outline Color

You can set the polygon's outline color using **outline**:

```
canvas.create_polygon(  
    x1, y1, x2, y2, x3, y3,  
    outline="black"  
)
```

If no fill color is provided, the polygon will not be filled.

### Border Thickness

You can control the thickness of the polygon's outline using **width**:

```
canvas.create_polygon(  
    x1, y1, x2, y2, x3, y3,  
    width=4  
)
```

### Dashed Outlines

You can create dashed polygon outlines using **dash**:

```
canvas.create_polygon(  
    x1, y1, x2, y2, x3, y3,  
    dash=(8, 4)  
)
```

Dashed settings affect only the outline, not the fill.

### Combining Settings

All polygon settings can be combined in a single call:

```
canvas.create_polygon(  
    -50, -50,  
    50, -50,  
    0, 50,  
    outline="blue",  
    fill="lightblue",  
    width=3  
)
```

## Set Up

---

Create a new Python file called [canvas\\_polygons.py](#).

## Copy, Change, Challenge

---

### Copy

Copy and run the following code.

```
import turtle  
  
screen = turtle.Screen()  
canvas = screen.getcanvas()  
  
canvas.create_polygon(  
    -50, -50,  
    50, -50,  
    0, 50  
)  
  
turtle.done()
```

You should see a triangle centered on the canvas.

### Change

Modify the polygon so that:

- The shape has more than three points
- The polygon is filled with a color
- The outline color is different

Run the program again and observe the result.

### Challenge

Create multiple polygons so that:

- One polygon is centered at  $(0, 0)$
- One polygon extends into negative x and y values
- One polygon clearly demonstrates filling (outline + fill)