# Amit Divekar | Practical 6

## Geometric Transformations and Basic Operations

In [1]:
```
import matplotlib.pyplot as plt
from sympy import Point, Line, Triangle, Polygon, RegularPolygon, pi, Matrix
import numpy as np
```

### Q1. Write a Python program using SymPy to demonstrate the translation of a point in two-dimensional geometry. Given a point P(2, 3), translate it by the vector (4, −2)

In [2]:
```
from sympy import Point
P = Point(2, 3)
P_trans = P.translate(4, -2)
print("Original Point:", P)
print("Translated Point:", P_trans)
```

```
Original Point: Point2D(2, 3)
Translated Point: Point2D(6, 1)
```

### Q2. Write a Python program using SymPy to rotate a line in the 2D plane. Given a line joining the points (1,1) and (4,4), rotate it through 90 degrees

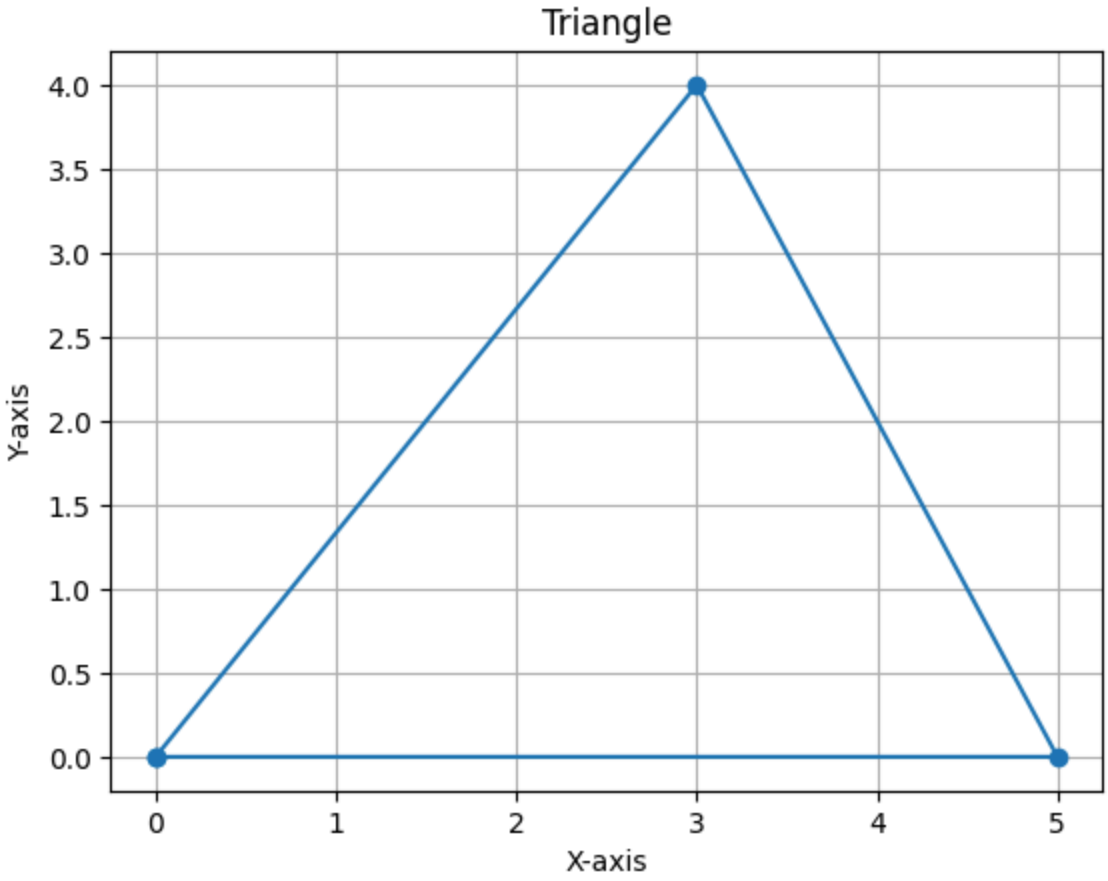
In [3]:
```
from sympy import Line, Point, pi
L = Line(Point(1, 1), Point(4, 4))
L_rot = L.rotate(pi/2)
print("Original Line:", L)
print("Rotated Line:", L_rot)
```

```
Original Line: Line2D(Point2D(1, 1), Point2D(4, 4))
Rotated Line: Line2D(Point2D(-1, 1), Point2D(-4, 4))
```

### Q3. Write a Python program to plot a triangle using SymPy and Matplotlib by specifying its three vertices and displaying it graphically.

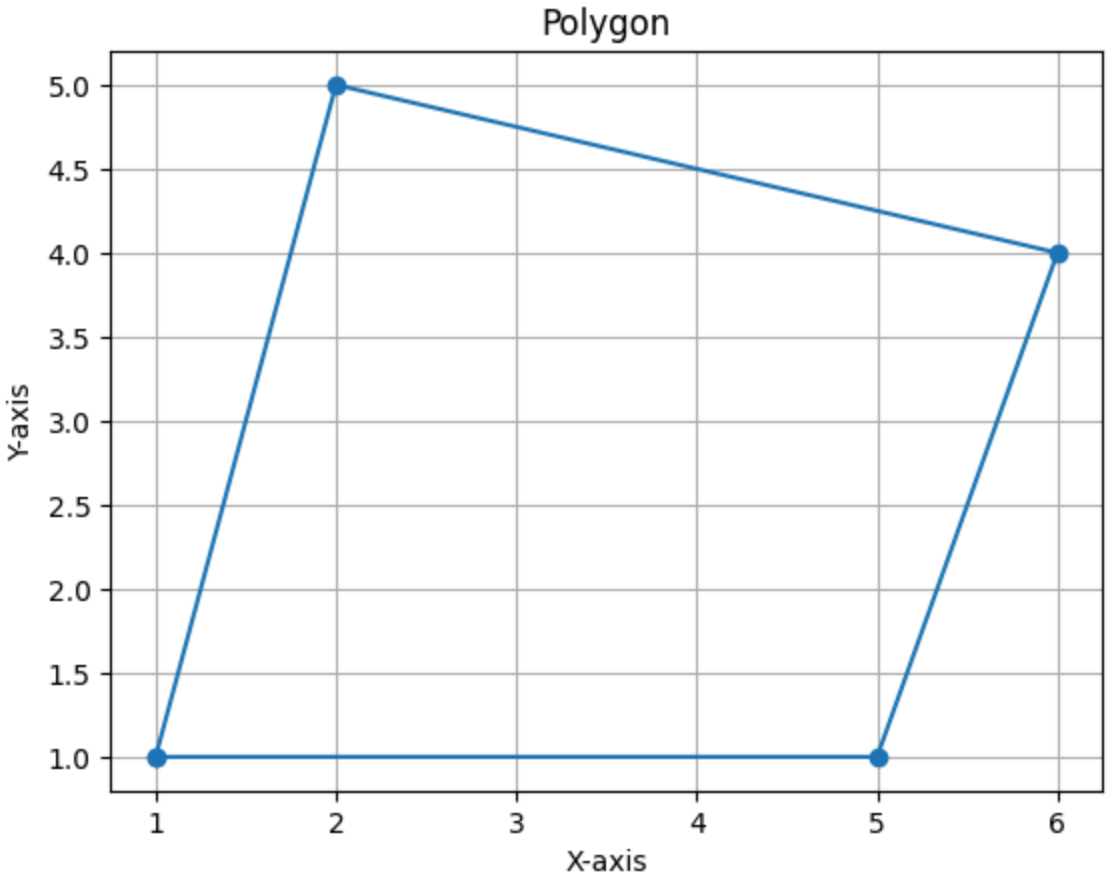
In [4]:
```
import matplotlib.pyplot as plt
from sympy import Point, Triangle
T = Triangle(Point(0, 0), Point(5, 0), Point(3, 4))
```

```
x = [p.x for p in T.vertices] + [T.vertices[0].x]
y = [p.y for p in T.vertices] + [T.vertices[0].y]
plt.plot(x, y, marker='o')
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Triangle")
plt.grid()
plt.show()
```



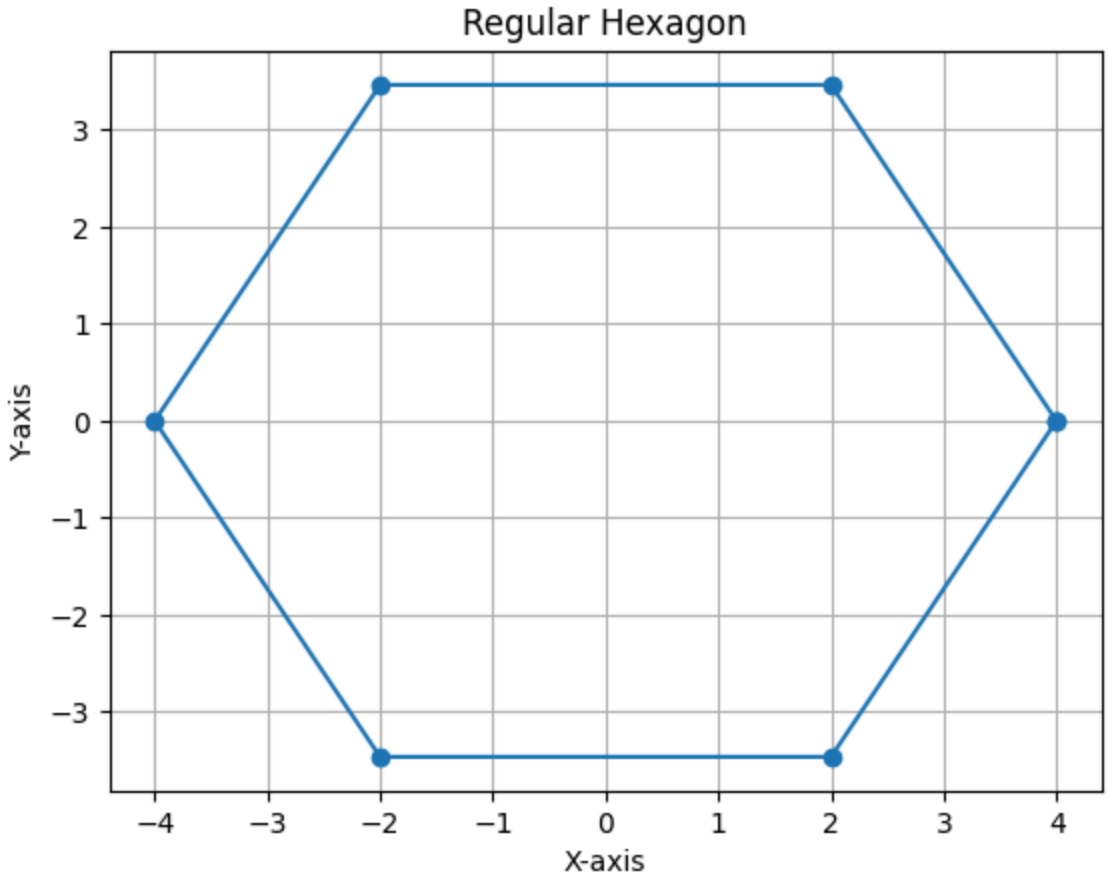

## Q4. Write a Python program to create and graphically represent a polygon using SymPy and Matplotlib with vertices (1,1), (5,1), (6,4), and (2,5)

In [5]:
```
import matplotlib.pyplot as plt
from sympy import Point, Polygon
P = Polygon(Point(1,1), Point(5,1), Point(6,4), Point(2,5))
x = [p.x for p in P.vertices] + [P.vertices[0].x]
y = [p.y for p in P.vertices] + [P.vertices[0].y]
plt.plot(x, y, marker='o')
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Polygon")
plt.grid()
plt.show()
```



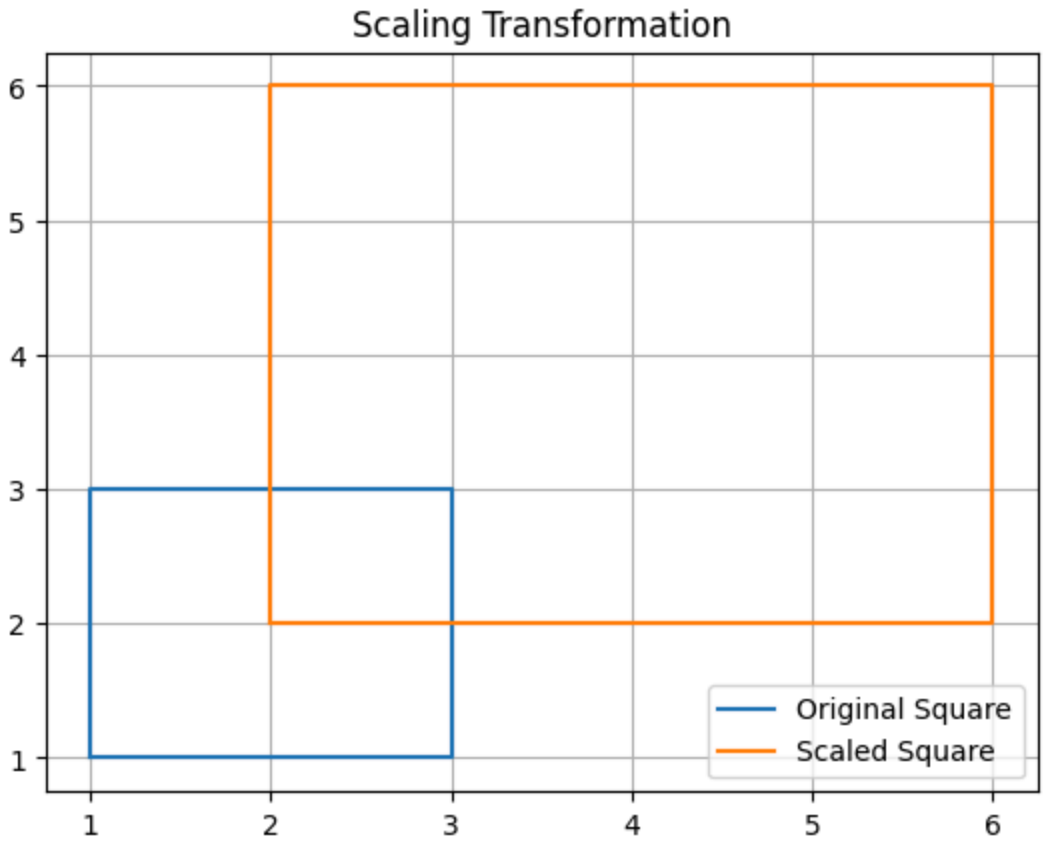

## Q5. Write a Python program to generate and plot a regular hexagon using the RegularPolygon class in SymPy.

In [6]:
```
import matplotlib.pyplot as plt
from sympy import RegularPolygon, Point
R = RegularPolygon(Point(0,0), 4, 6)
x = [v.x for v in R.vertices] + [R.vertices[0].x]
y = [v.y for v in R.vertices] + [R.vertices[0].y]
plt.plot(x, y, marker='o')
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.title("Regular Hexagon")
plt.grid()
plt.show()
```




## Q6. Plot a square and its scaled image.

In [7]:
```
import matplotlib.pyplot as plt
from sympy import Polygon, Point
S = Polygon(Point(1,1), Point(3,1), Point(3,3), Point(1,3))
S_s = S.scale(2, 2)
def draw(P, label):
    x = [p.x for p in P.vertices] + [P.vertices[0].x]
    y = [p.y for p in P.vertices] + [P.vertices[0].y]
    plt.plot(x, y, label=label)
draw(S, "Original Square")
draw(S_s, "Scaled Square")
plt.legend()
plt.grid()
plt.title("Scaling Transformation")
plt.show()
```


Scaling Transformation
Original Square
Scaled Square
1
2
3
4
5
6