



UAEM

Universidad Autónoma
del Estado de México

Universidad Autónoma del Estado de México

Centro Universitario UAEM Zumpango

Ingeniería en Computación

Graficacion Computacional

Alumno: Diego Argel Navarrete Godines

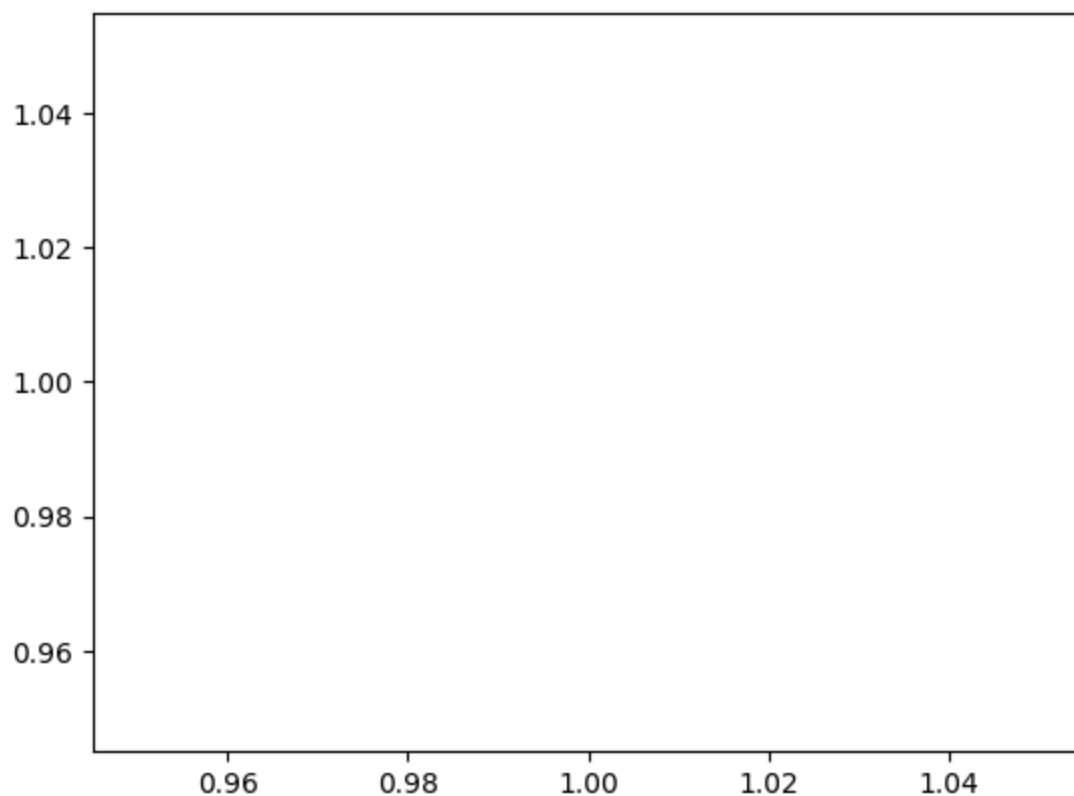
Profesora: Hazem Álvarez Rodríguez

Fecha: 09 de octubre de 2024

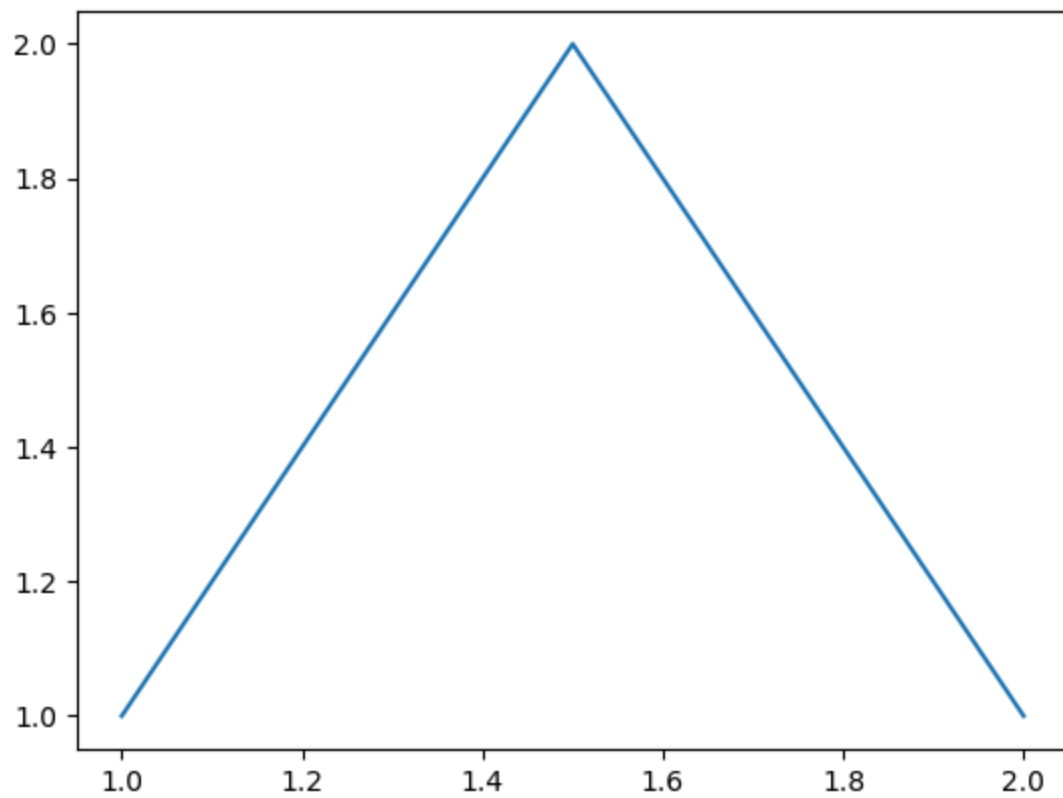
Descripcion: Traslación, Rotación y Escalado de Figuras

```
In [2]: import matplotlib.pyplot as plt  
import numpy as np
```

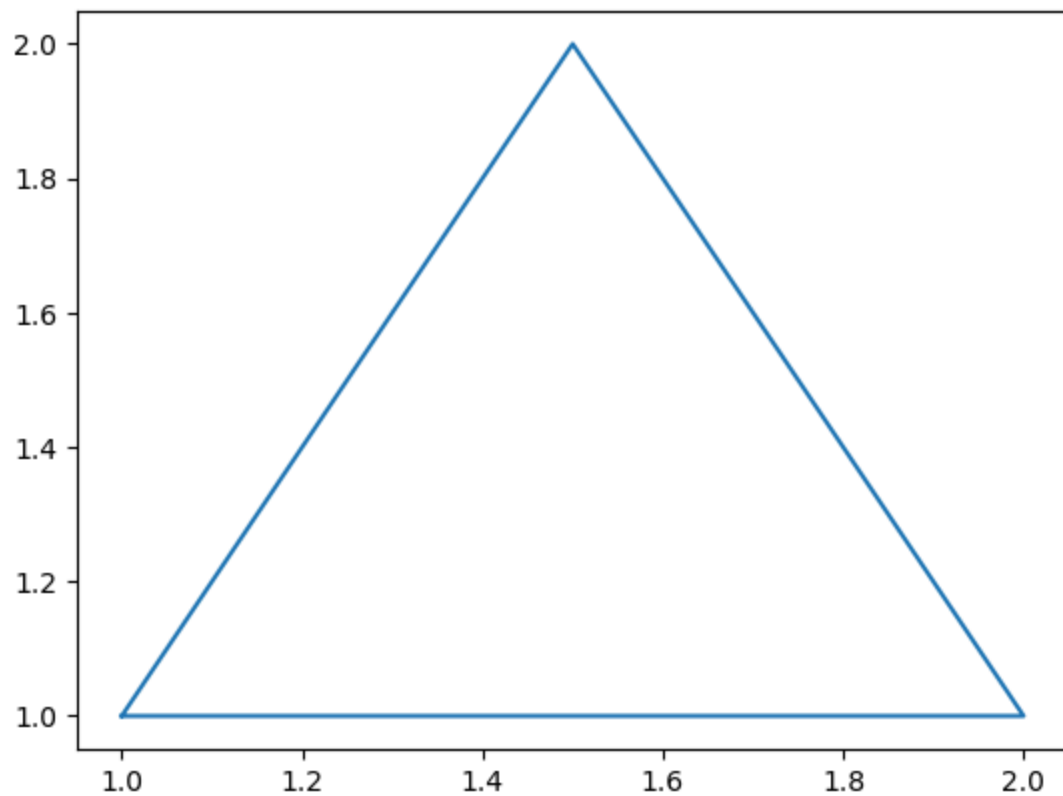
```
In [3]: # Une los puntos [x][y], [1-1, 1.5-2, 2-1, 1-1]  
plt.plot([1],[1])  
plt.show()
```



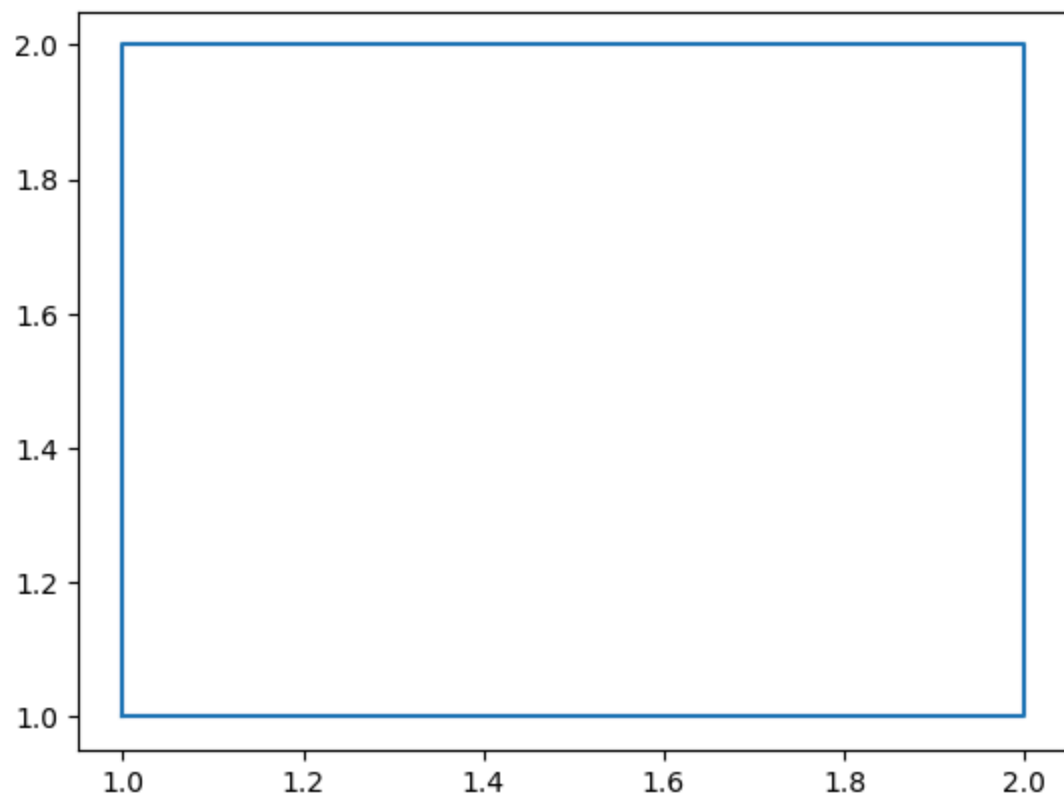
```
In [5]: plt.plot([1,1.5,2],[1,2,1])  
plt.show()
```



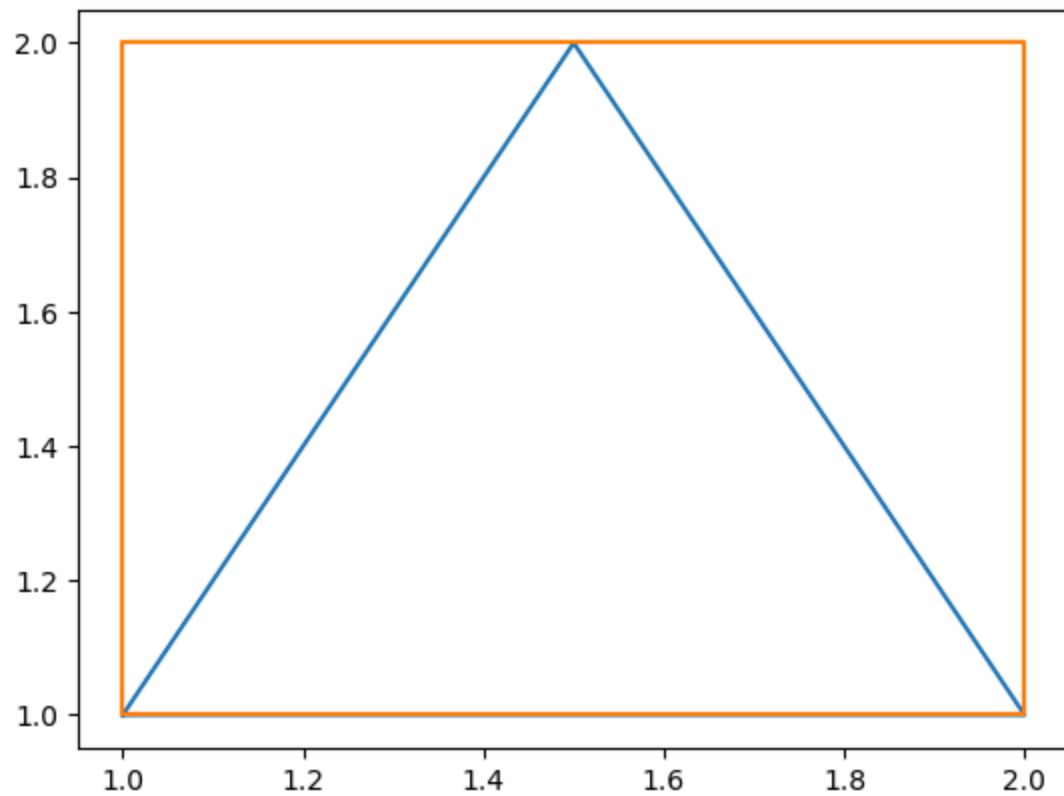
```
In [20]: plt.plot([1,1.5,2,1],[1,2,1,1])  
plt.show()
```



```
In [32]: plt.plot([1,1,2,2,1],[1,2,2,1,1])  
plt.show()
```



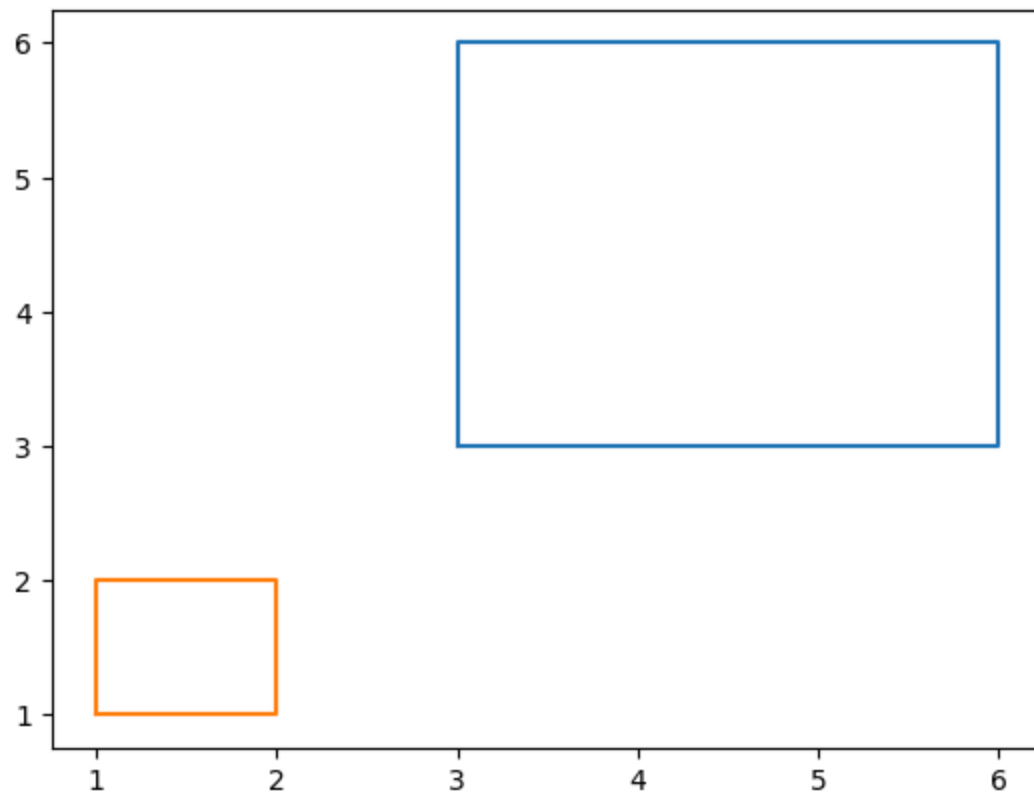
```
In [33]: plt.plot([1,1.5,2,1],[1,2,1,1])
plt.plot([1,1,2,2,1],[1,2,2,1,1])
plt.show()
```



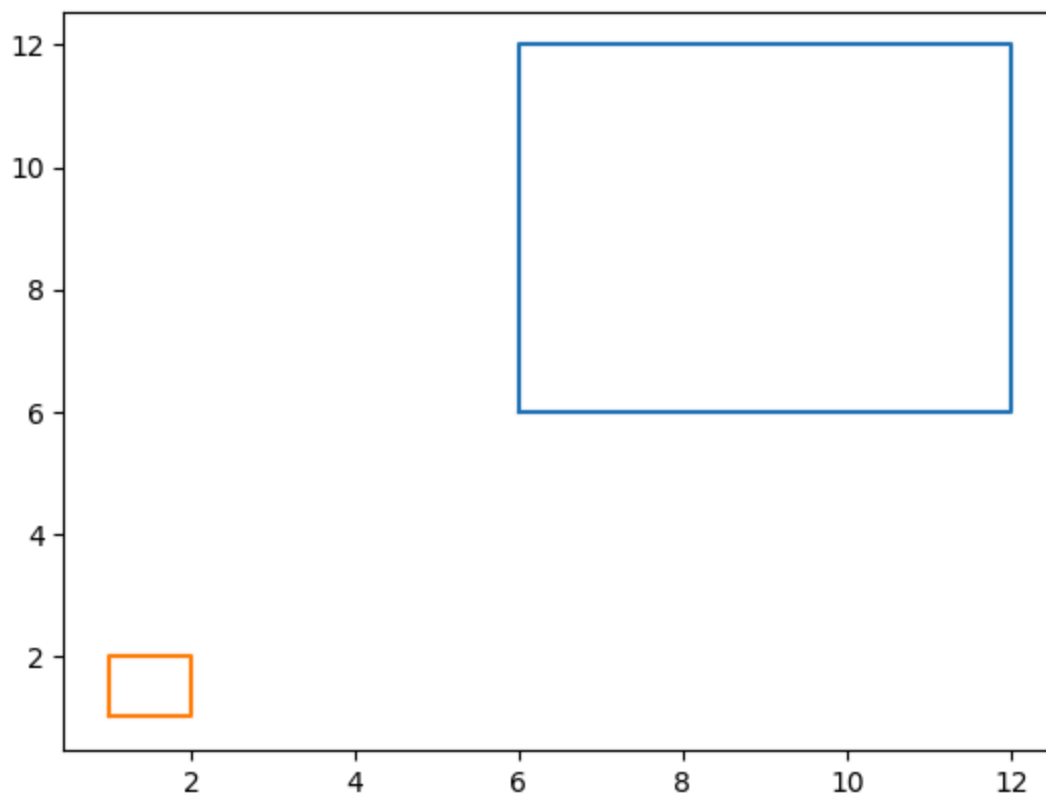
ESCALAR

```
In [109... esca = 3
plt.plot([1*esca, 2*esca, 2*esca, 1*esca, 1*esca],[1*esca, 1*esca, 2*esca, 2*esca, 1*esca])
```

```
plt.plot([1,1,2,2,1],[1,2,2,1,1])  
plt.show()
```



```
In [6]: esc = 6  
        esc2 = 6  
  
        cuadradoX = np.array([1, 1, 2, 2, 1])  
        cuadradoY = np.array([1, 2, 2, 1, 1])  
  
        cX = esc * cuadradoX  
        cY = esc2 * cuadradoY  
  
        # escalado  
        plt.plot(cX, cY)  
        # original  
        plt.plot(cuadradoX, cuadradoY)  
        plt.show()
```

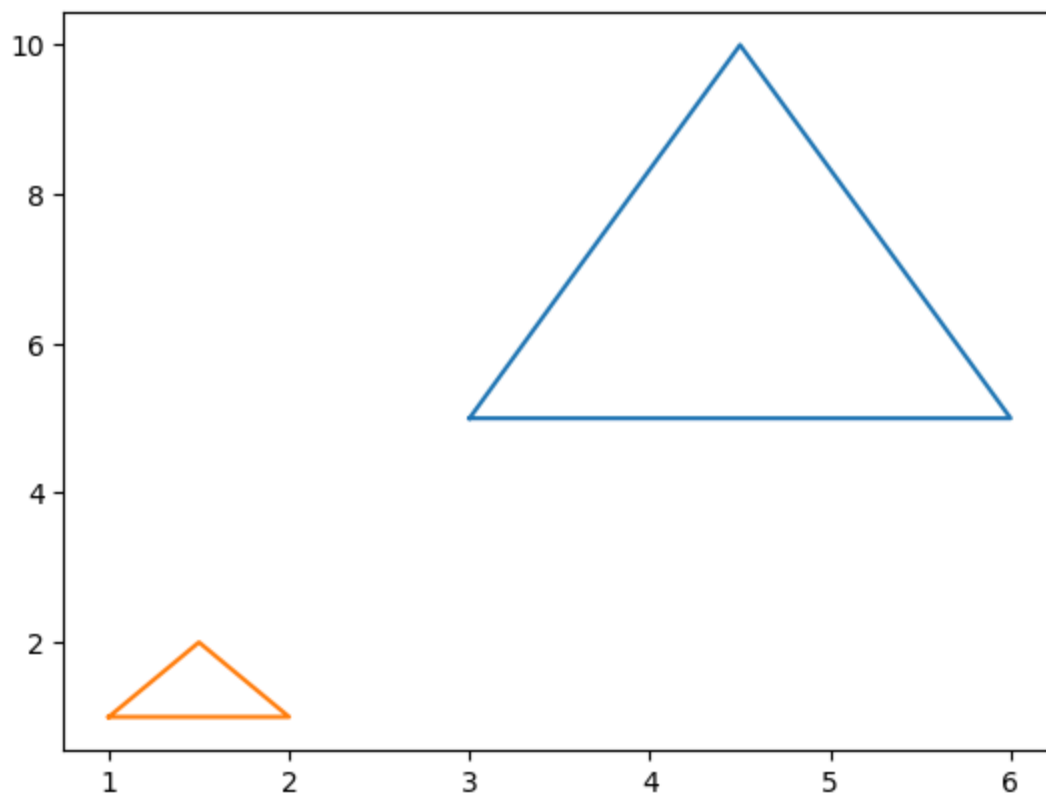


```
In [7]: esc = 3
esc2 = 5

trianguloX = np.array([1, 1.5, 2, 1])
trianguloY = np.array([1, 2, 1, 1])

tX = esc * trianguloX
tY = esc2 * trianguloY

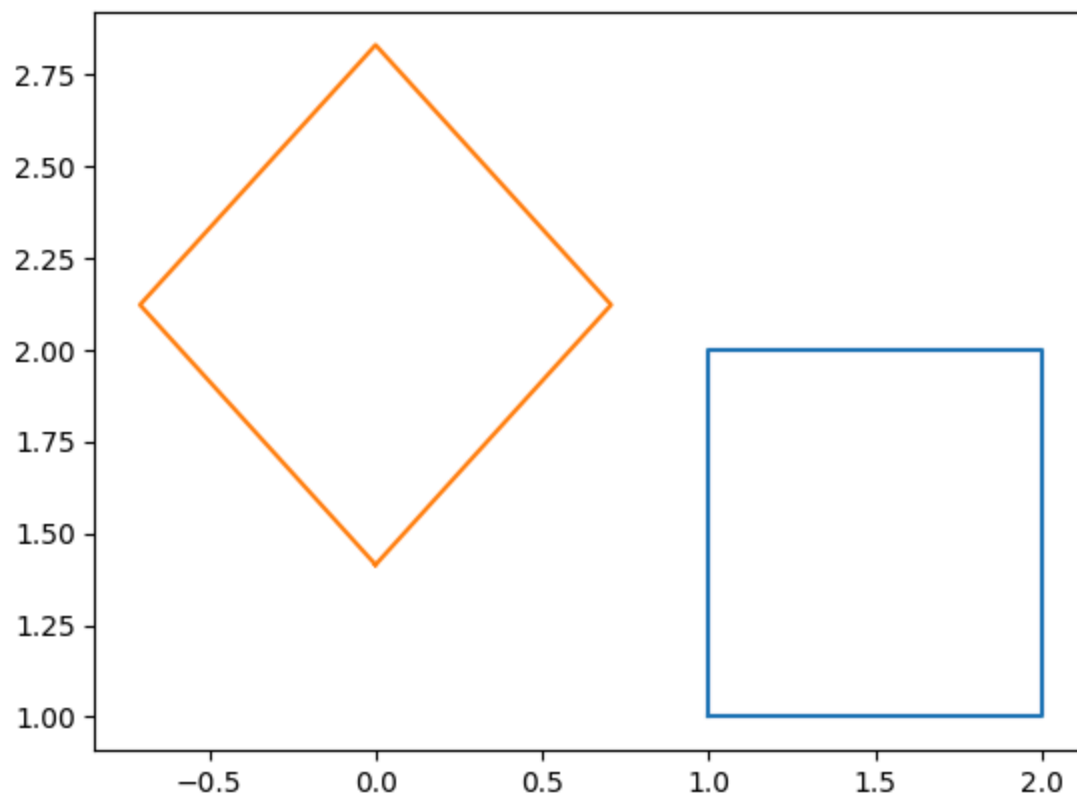
# escalado
plt.plot(tX, tY)
# original
plt.plot(trianguloX, trianguloY)
plt.show()
```



ROTACIÓN

In [104...

```
import numpy as np
x = [1, 2, 2, 1, 1]
y = [1, 1, 2, 2, 1]
plt.plot(x,y)
tetha = np.radians(45)
rotacionX = np.cos(tetha) * np.array(x) - np.sin(tetha) * np.array(y)
rotacionY = np.sin(tetha) * np.array(x) + np.cos(tetha) * np.array(y)
plt.plot(rotacionX, rotacionY)
plt.show()
```

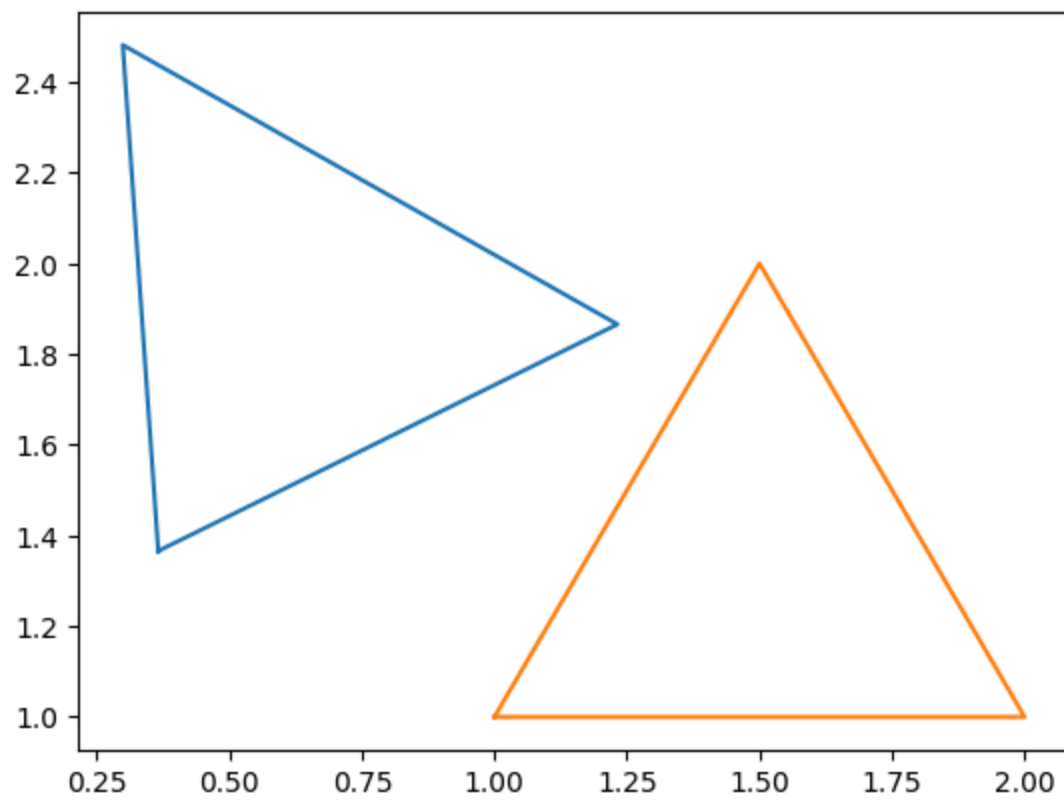


```
In [13]: theta = np.radians(30)

trianguloX = np.array([1, 1.5, 2, 1])
trianguloY = np.array([1, 2, 1, 1])

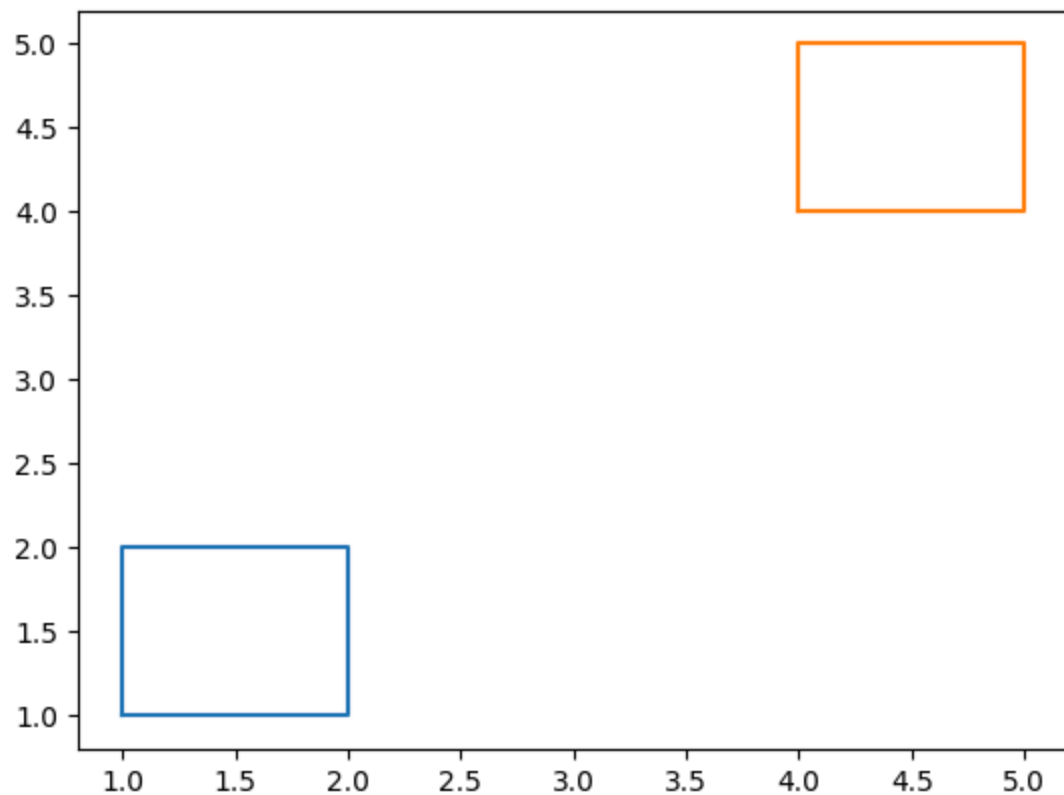
tX = np.cos(theta) * trianguloX - np.sin(theta) * trianguloY
tY = np.sin(theta) * trianguloX + np.cos(theta) * trianguloY

# rotacion
plt.plot(tX, tY)
# original
plt.plot(trianguloX, trianguloY)
plt.show()
```



TRANSLACIÓN

```
In [75]: plt.plot([1, 2, 2, 1, 1],[1, 1, 2, 2, 1])
plt.plot([4, 5, 5, 4, 4],[4, 4, 5, 5, 4])
plt.show()
```



```
In [9]: tras = 3
tras2 = 5
```



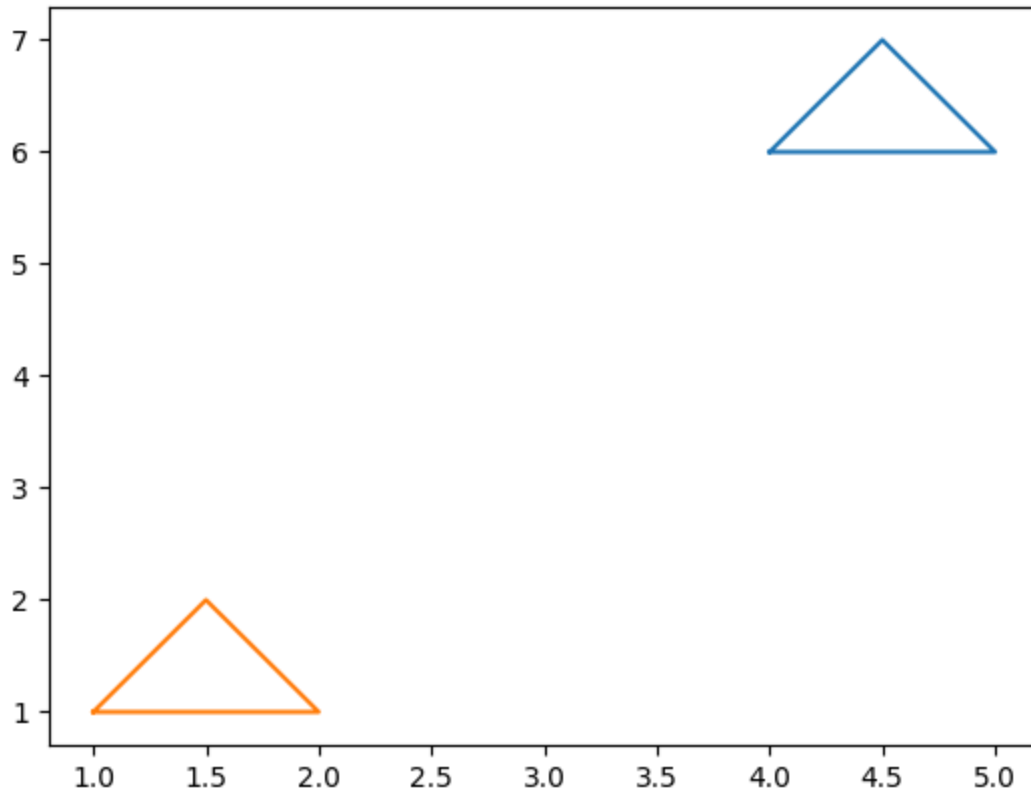
```

trianguloX = np.array([1, 1.5, 2, 1])
trianguloY = np.array([1, 2, 1, 1])

tX = tras + trianguloX
tY = tras2 + trianguloY

# traslación
plt.plot(tX, tY)
# original
plt.plot(trianguloX, trianguloY)
plt.show()

```



```

In [11]: tras = -2
tras2 = 2

cuadradoX = np.array([1, 1, 2, 2, 1])
cuadradoY = np.array([1, 2, 2, 1, 1])

cX = tras + cuadradoX
cY = tras2 + cuadradoY

# traslación
plt.plot(cX, cY)
# original
plt.plot(cuadradoX, cuadradoY)
plt.show()

```

