```
from google.colab import drive
drive.mount('/content/drive')
```

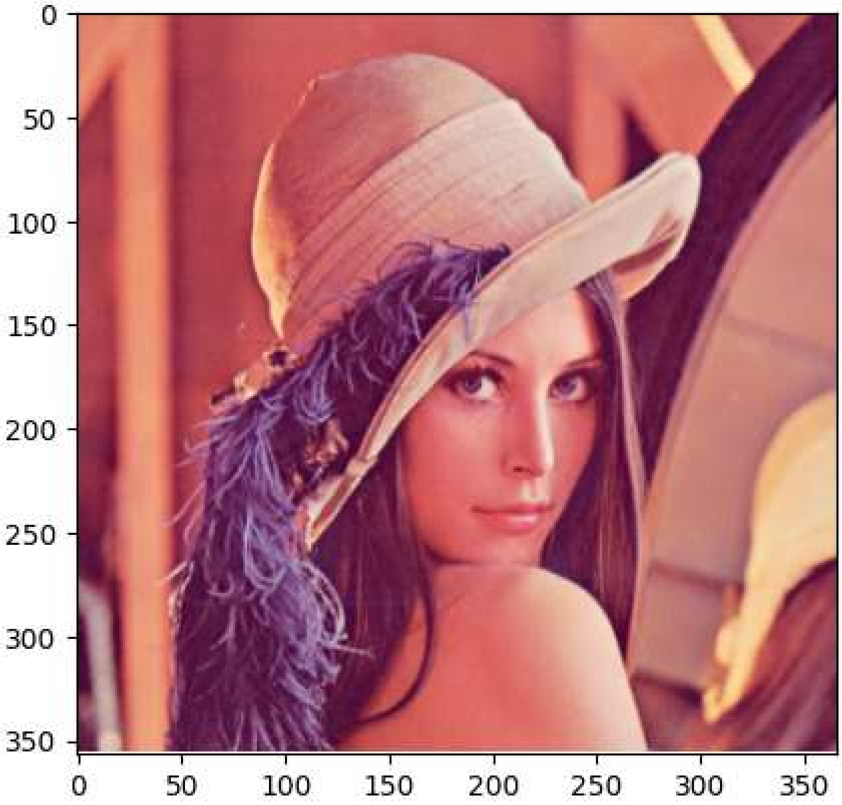
```
Mounted at /content/drive
```

1. Read and display the image

```
# Read the image using the Pillow library and display it.

!pip install pillow
from PIL import Image
image_colored = Image.open("/content/drive/MyDrive/Final-Year AI/lenna_image.png")
display(image_colored)
```

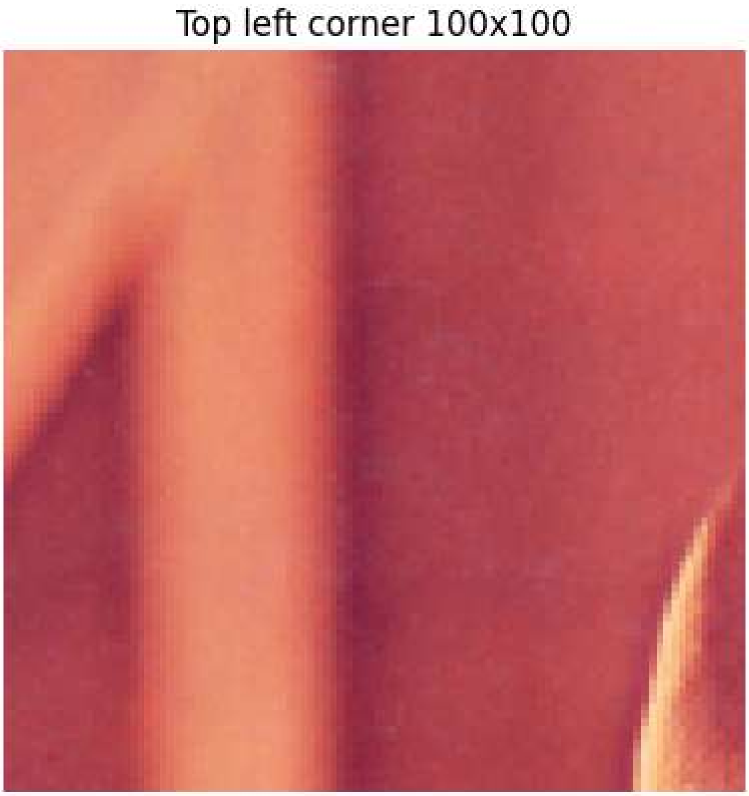
```
Requirement already satisfied: pillow in /usr/local/lib/python3.11/dist-packages (11.1.0
```

```
# You can also use matplotlib to display the image.
import matplotlib.pyplot as plt
image_colored = Image.open("/content/drive/MyDrive/Final-Year AI/lenna_image.png")
plt.imshow(image_colored)
plt.show()
```



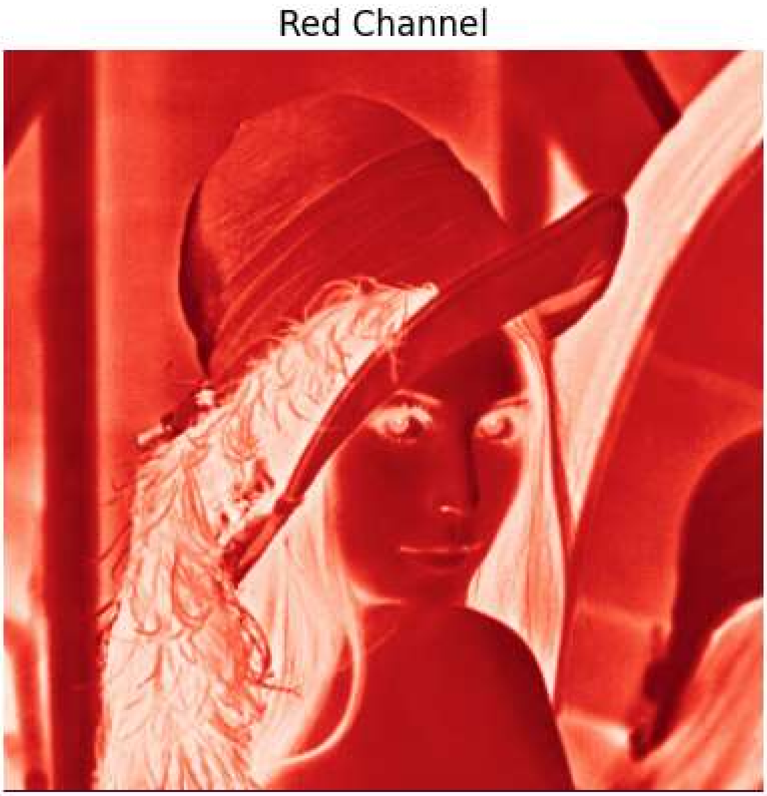
2. Display only the top left corner of 100x100 pixels.

```
import numpy as np
image = Image.open("/content/drive/MyDrive/Final-Year AI/lenna_image.png")
image_array = np.array(image)
top_left = image_array[:100, :100]  # Extract top-left 100x100 pixels
plt.imshow(top_left)
plt.title("Top left corner 100x100")
plt.axis("off")
plt.show()
```



3. Show the three color channels (R, G, B).

```
r = image_array[:, :, 0]
plt.imshow(r, cmap='Reds')
plt.title("Red Channel")
plt.axis("off")
plt.show()
```

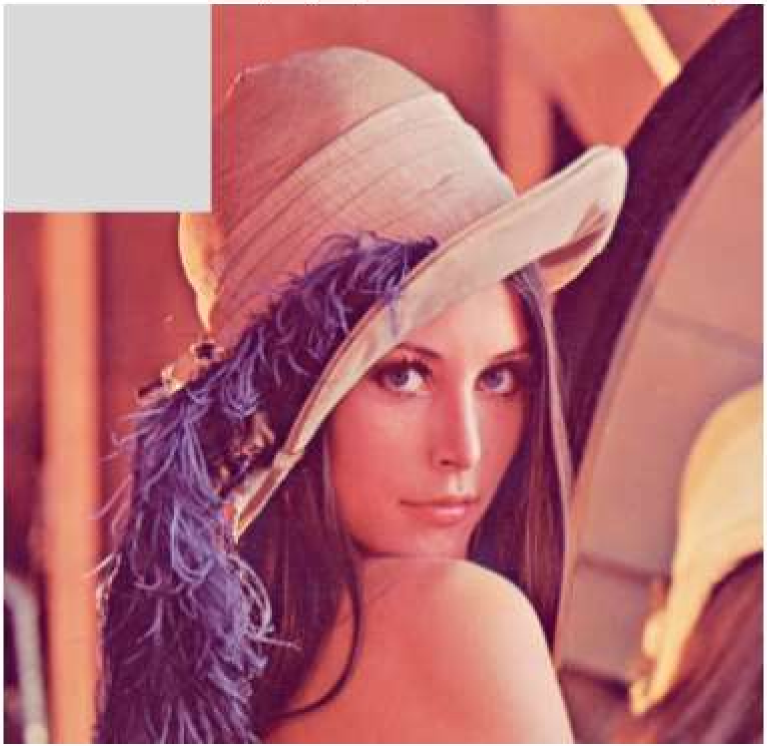


```
g = image_array[:, :, 1]
plt.imshow(g, cmap='Greens')
plt.title("Green Channel")
plt.axis("off")
plt.show()
```

```
b = image_array[:, :, 2]
plt.imshow(b, cmap='Blues')
plt.title("Blue Channel")
plt.axis("off")
plt.show()
```

4. Modify the top 100 × 100 pixels to a value of 210 and display the resulting image

```
image_array[:100, :100] = 210
plt.imshow(image_array)
plt.title("Modified Image (Top-left 100x100 to 210)")
plt.axis("off")
plt.show()
```



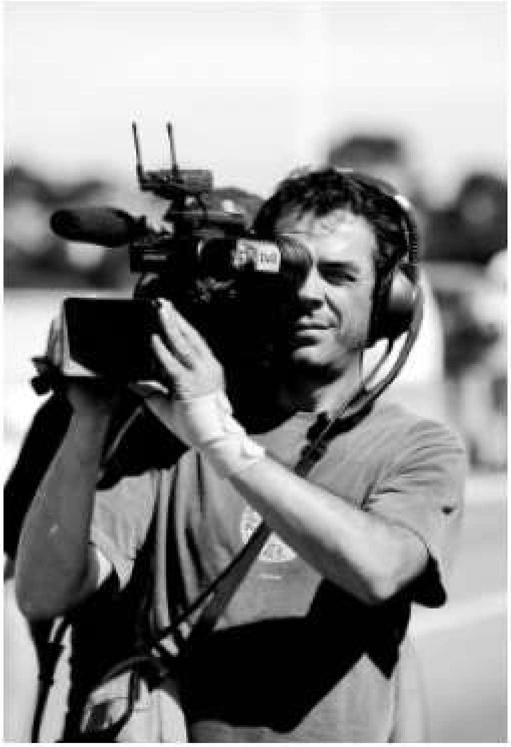
Modified Image (Top-left 100x100 to 210)

Exercise - 2

1. Load and display a grayscale image.

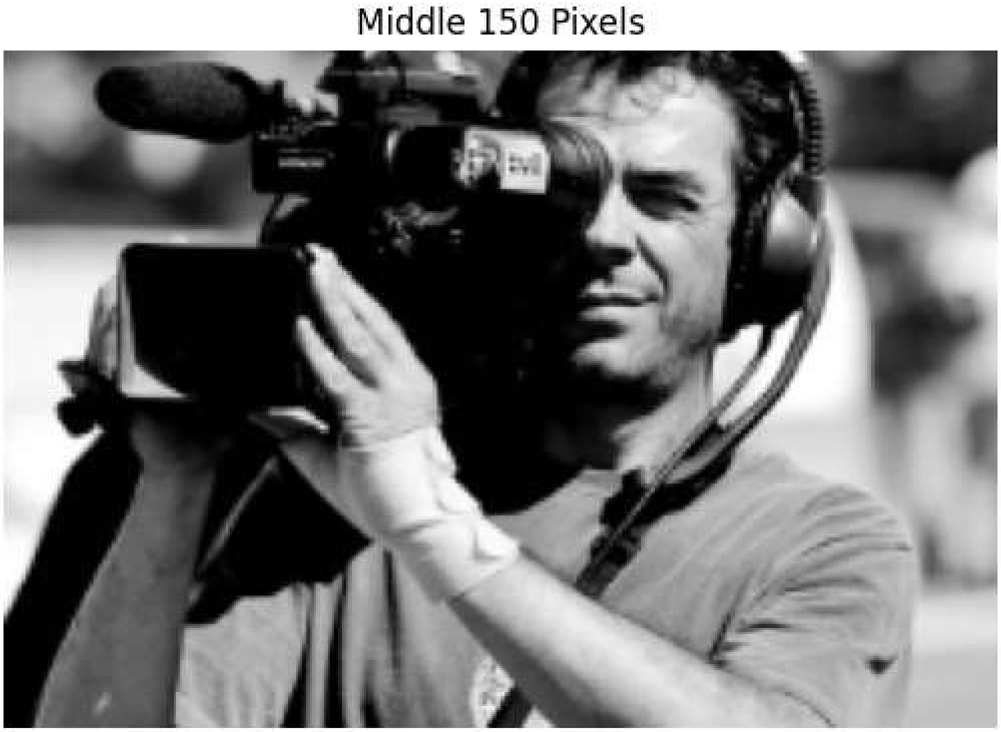
```
#Load a grayscale image using the Pillow library.
image_colored = Image.open("/content/drive/MyDrive/Final-Year AI/camera_man.jpg")
image_grayscale = image_colored.convert("L")
display(image_grayscale)
```

```
#Display the grayscale image using matplotlib.
image = Image.open("/content/drive/MyDrive/Final-Year AI/camera_man.jpg")
gray_image = image.convert("L")
plt.imshow(gray_image, cmap="gray")
plt.axis("off")
plt.show()
```



2. Extract and display the middle section of the image (150 pixels).

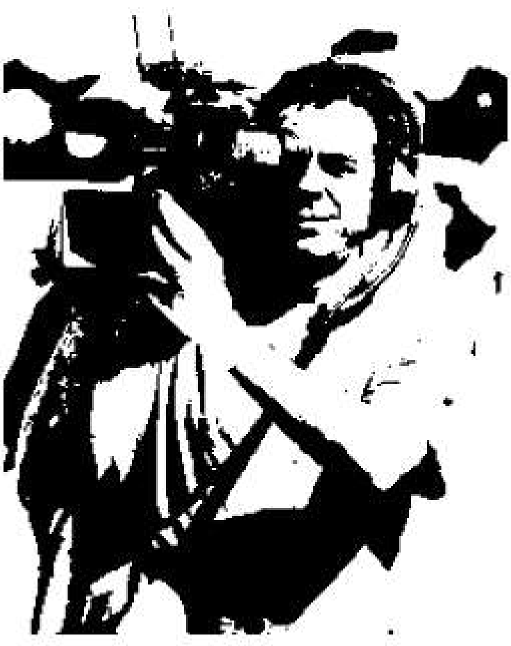
```
gray_array = np.array(gray_image)
h, w = gray_array.shape

# Crop middle 150 pixels in height
cropped = gray_array[h//2 - 75 : h//2 + 75, :]
plt.imshow(cropped, cmap="gray")
plt.title("Middle 150 Pixels")
plt.axis("off")
plt.show()
```



3. Apply a simple threshold to the image (e.g., set all pixel values below 100 to 0).

```
binary_image = np.where(gray_array < 100, 0, 255)
plt.imshow(binary_image, cmap="gray")
plt.title("Thresholding Image")
plt.axis("off")
plt.show()
```

Thresholding Image



4. Rotate the image 90 degrees clockwise and display the result.

```
rotated = np.rot90(gray_array, k=-1)
plt.imshow(rotated, cmap="gray")
plt.title("Rotated Image (90° Clockwise)")
plt.axis("off")
plt.show()
```

Rotated Image (90° Clockwise)

5. Convert the grayscale image to an RGB image.

```
rgb_image = np.stack([gray_array]*3, axis=-1)  # Copy grayscale values to R, G, B channels
plt.imshow(rgb_image)
plt.title("Converted RGB")
plt.axis("off")
plt.show()
```

Converted RGB

Exercise 3:Image Compression and Decompression using PCA.

1. Load and preprocess image

```
from PIL import Image
import numpy as np
```