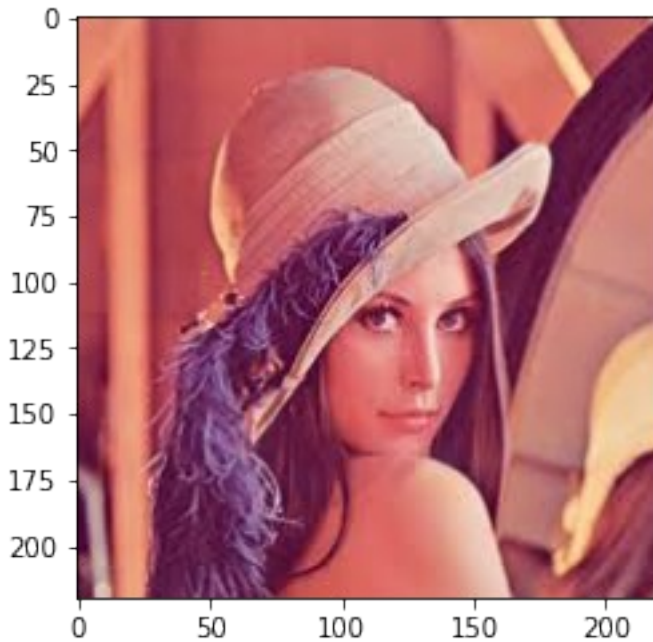


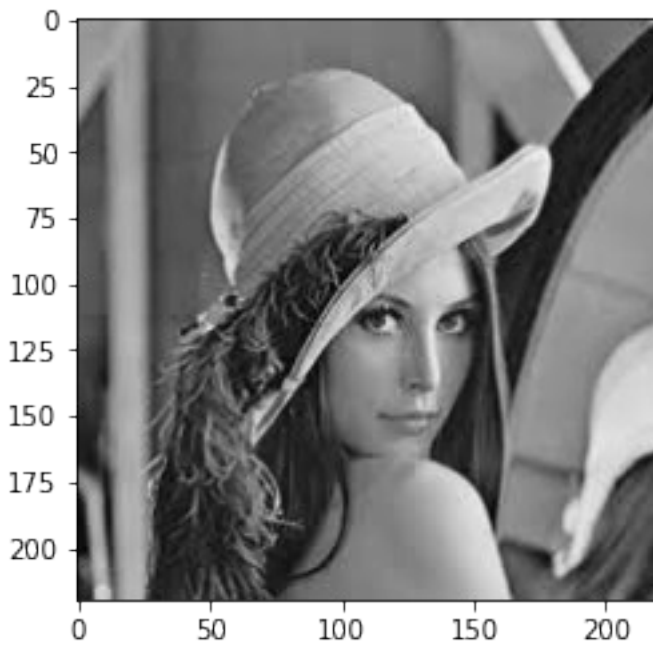
```
from skimage import data
import numpy as np
from skimage import io
from matplotlib import pyplot as plt

from skimage import io
data = io.imread('images/lena.jpg')
plt.imshow(data)

<matplotlib.image.AxesImage at 0x1345162e850>
```

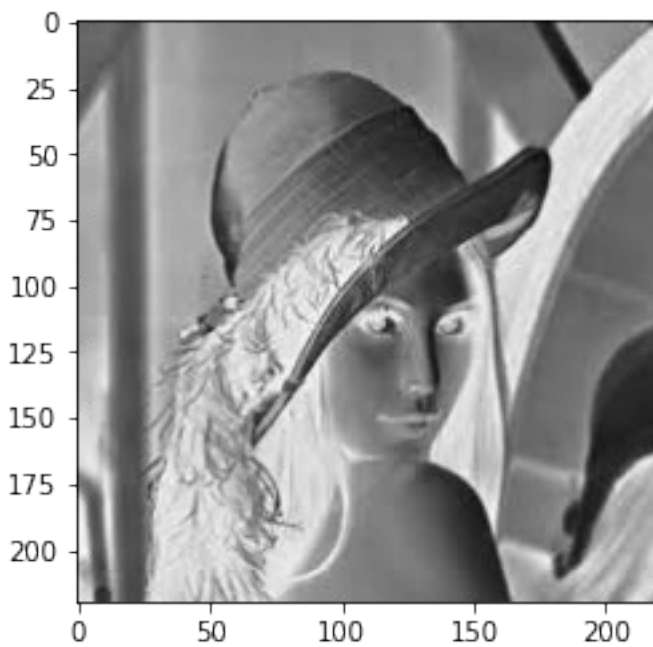


```
from skimage.color import rgb2gray
lena_gray=rgb2gray(data)
plt.imshow(lena_gray,cmap="gray")
plt.imsave('images/lena-gray.jpg', lena_gray)
```



Exercise 2

```
from skimage import util
inverted_img = util.invert(lena_gray)
plt.imshow(inverted_img, cmap="gray")
plt.savefig("images/Lena_Gray_Inv.jpg", inverted_img)
```

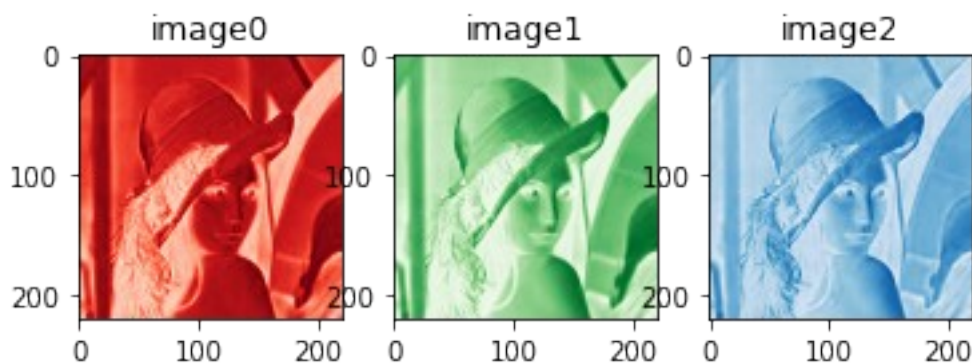


Exercise 3

```

# plt.imshow(data)
bande_Rouge=data[:, :,0]
# plt.imshow(bande_Rouge)
bande_Blue=data[:, :,2]
# plt.imshow(bande_Blue)
bande_Green=data[:, :,1]
# plt.imshow(bande_Green)
fig, axes = plt.subplots(1, 3)
axes[0].imshow(bande_Rouge,cmap="Reds")
axes[0].set_title('image0')
axes[1].imshow(bande_Green,cmap="Greens")
axes[1].set_title('image1')
axes[2].imshow(bande_Blue,cmap="Blues")
axes[2].set_title('image2')
plt.show()
plt.imsave("Images/Bande_rouge.jpg",bande_Rouge,cmap="Reds")
plt.imsave("Images/Bande_green.jpg",bande_Green,cmap="Greens")
plt.imsave("Images/Bande_bleu.jpg",bande_Blue,cmap="Blues")

```



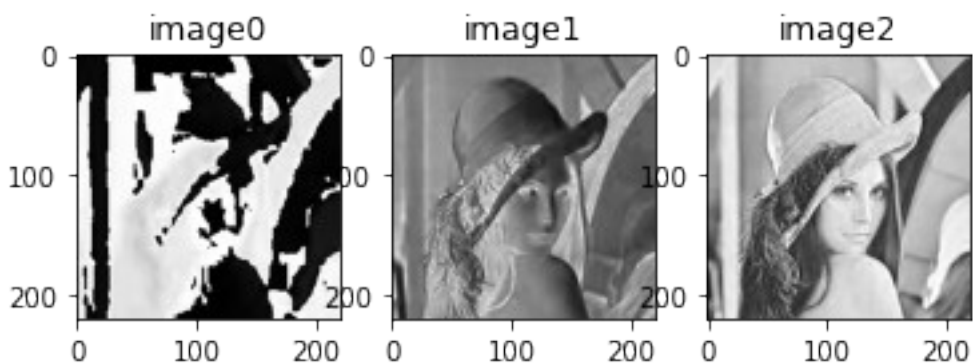
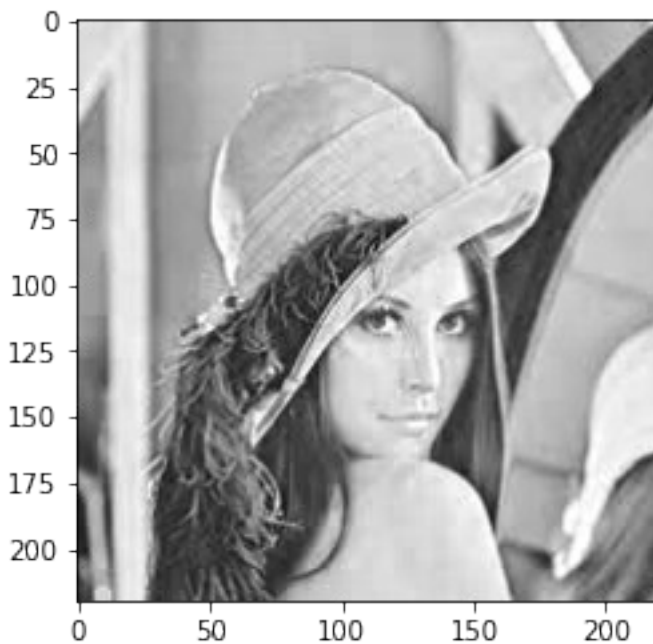
Exercice 4

```

from skimage.color import rgb2hsv
lena_HSV=rgb2hsv(data)
# plt.imshow(lena_HSV)
# plt.imshow(data)
bande1=lena_HSV[:, :,0]
# plt.imshow(bande1,cmap="gray")
bande2=lena_HSV[:, :,1]
# plt.imshow(bande2,cmap="gray")
bande3=lena_HSV[:, :,2]
plt.imshow(bande3,cmap="gray")
fig, axes = plt.subplots(1, 3)
axes[0].imshow(bande1,cmap="gray")
axes[0].set_title('image0')
axes[1].imshow(bande2,cmap="gray")
axes[1].set_title('image1')
axes[2].imshow(bande3,cmap="gray")
axes[2].set_title('image2')
plt.show()

```

```
plt.imsave("Images/Bande_Teinte.jpg",bande1,cmap="gray")
plt.imsave("Images/Bande_saturation.jpg",bande2,cmap="gray")
plt.imsave("Images/Bande_luminance.jpg",bande3,cmap="gray")
```

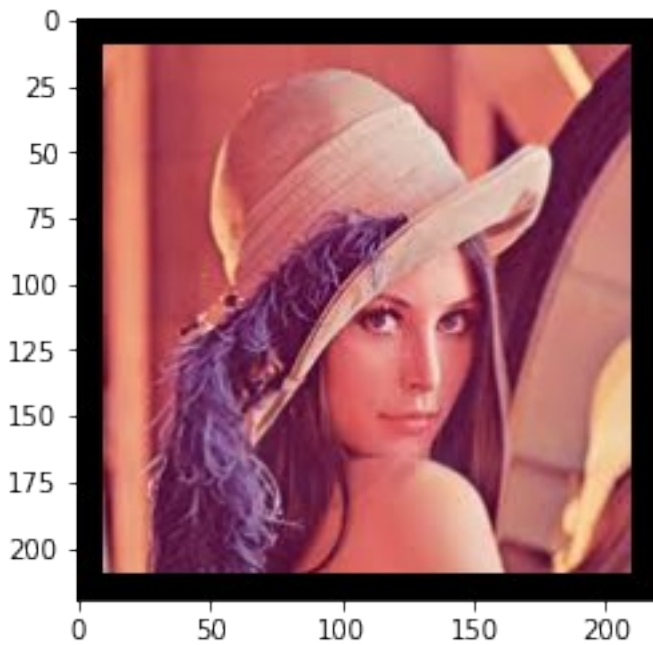


Exercice 5

```
def cadreNoire(image,ep):
    image1=np.array(image)
    for i in range(ep):
        image1[i]=np.array([0,0,0])
        image1[image.shape[0]-i-1]=np.array([0,0,0])
        image1[:,i,:]=np.array([0,0,0])
        image1[:,image.shape[0]-i-1,:]=np.array([0,0,0])
    return image1

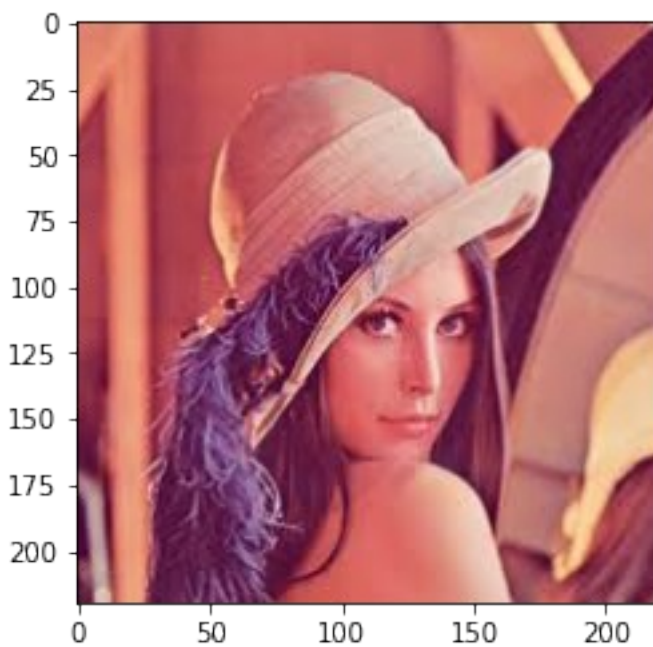
plt.imshow(cadreNoire(data,10))

<matplotlib.image.AxesImage at 0x13451edc580>
```



```
plt.imshow(data)
```

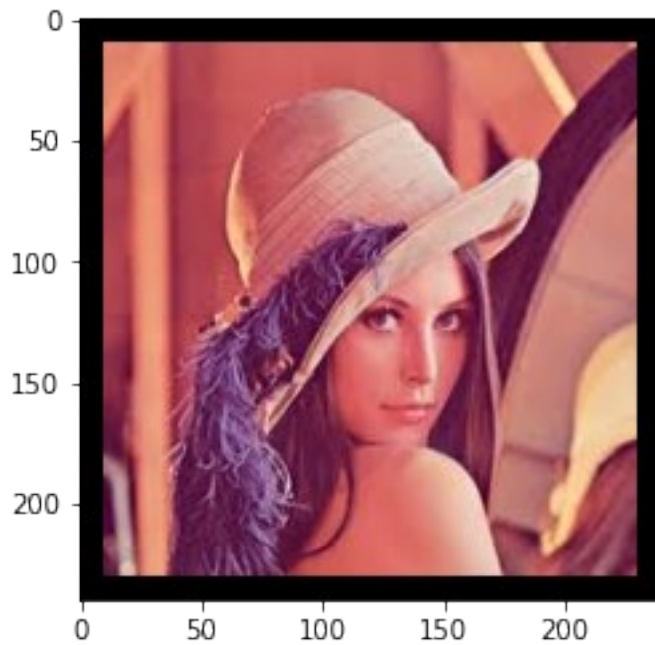
```
<matplotlib.image.AxesImage at 0x134549ba850>
```



Exercise 6

```
from PIL import Image, ImageOps
img = Image.open('images/lena.jpg')
img_with_border = ImageOps.expand(img, border=10, fill='black')
# img_with_border.save('images/lena-with-border.jpg')
```

```
plt.imshow(img_with_border)
img_with_border.size
(240, 240)
```



Methode 2

```
def cadre1(image,ep):
    lines=image.shape[0]
    columns=image.shape[1]
    im1=np.zeros((lines+2*ep,columns+2*ep,3),dtype=np.uint8)
    print(im1.shape)
    for i in range(image.shape[0]):
        for j in range(image.shape[1]):
            for k in range(3):
                im1[i+ep,j+ep,k]=image[i,j,k]
    return im1
```