In [10]:
```
import numpy as np
```

In [3]:
```
np.zeros((3,4))
```

Out[3]:
```
array([[0., 0., 0., 0.],
       [0., 0., 0., 0.],
       [0., 0., 0., 0.]])
```

In [4]:
```
np.ones((2, 3, 4), dtype = np.int16)
```

Out[4]:
```
array([[[1, 1, 1, 1],
        [1, 1, 1, 1],
        [1, 1, 1, 1]],

       [[1, 1, 1, 1],
        [1, 1, 1, 1],
        [1, 1, 1, 1]]], dtype=int16)
```

In [6]:
```
np.ones((2, 3, 4), dtype = np.float64)
```

Out[6]:
```
array([[[1., 1., 1., 1.],
        [1., 1., 1., 1.],
        [1., 1., 1., 1.]],

       [[1., 1., 1., 1.],
        [1., 1., 1., 1.],
        [1., 1., 1., 1.]]])
```

In [9]:
```
x = np.ones((2, 3, 4), dtype = np.float64)

x.itemsize
```

Out[9]:
```
8
```

In [10]:
```
y = np.ones((2, 3, 4), dtype = np.int16)

y.itemsize
```

Out[10]:
```
2
```

In [15]:
```
np.arange(15).reshape(5,3)
```

Out[15]:
```
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8],
       [ 9, 10, 11],
       [12, 13, 14]])
```

In [16]:
```
np.linspace(0, 2, 9) # starting, ending, element number
```

Out[16]:
```
array([0.  , 0.25, 0.5 , 0.75, 1.  , 1.25, 1.5 , 1.75, 2.  ])
```

In [17]:
```
A = np.array([[1,1],[0,1]])
A
```

Out[17]:
```
array([[1, 1],
       [0, 1]])
```

In [18]:
```
B = np.array([[2,0],[3,4]])
B
```

Out[18]:
```
array([[2, 0],
       [3, 4]])
```

In [19]:
```
# element wise multiplication
A*B
```

Out[19]:
```
array([[2, 0],
       [0, 4]])
```

In [21]:
```
# matrix multiplication
A@B
A.dot(B)
```

Out[21]:
```
array([[5, 4],
       [3, 4]])
```

In [22]:
```
# matrix addition
C = A + B
C
```

Out[22]:
```
array([[3, 1],
       [3, 5]])
```

In [25]:
```
# random array with size 3x4
D = np.random.rand(3,4)
D
```

Out[25]:
```
array([[0.29754584, 0.80957221, 0.91056837, 0.27611579],
       [0.38452282, 0.10479543, 0.95465046, 0.70818055],
       [0.75584115, 0.4458542 , 0.96624044, 0.44058877]])
```

In [26]:
```
D.shape
```

Out[26]:
```
(3, 4)
```

In [14]:
```
import cv2 as cv
```

In [4]:
```
image = cv.imread('../Homework1/images/image1.jpg', 1)
image.shape
```

Out[4]:
```
(675, 1200, 3)
```

In [7]:
```
image_gray = cv.imread('../Homework1/images/image1.jpg', 0)
image_gray.shape
```

Out[7]:
```
(675, 1200)
```

In [5]:
```
cv.imshow('image gray', image)

cv.waitKey(0)
cv.destroyAllWindows()
cv.waitKey(1)
```

Out[5]:
```
-1
```

In [8]:
```
cv.imshow('image gray', image_gray)

cv.waitKey(0)
cv.destroyAllWindows()
cv.waitKey(1)
```

Out[8]:
```
-1
```

In [59]:
```
A1 = np.zeros((300, 300,3), dtype = np.int16)
```

In [60]:
```
A1[1:100,:,0]= 255
```

In [61]:
```
cv.imshow('image', A1)

cv.waitKey(0)
cv.destroyAllWindows()
cv.waitKey(1)
```

Out[61]:
```
-1
```

In [ ]:
```

```

In [27]:
```
A_image = [0*A1, 127*A1]
```

In [48]:
```
gray_level = 127
```

In [50]:
```
gray_image = 0* np.ones((400,500,3), dtype = np.uint8)
```

In [51]:
```
gray_image1 = 0* np.ones((400,500,3), dtype = np.uint8)
```

In [53]:
```
for i im range(0,410,200):
    gray_image[:,i:i+50,0] = 255
    gray_image[:,i+50:i+100,1] = 255
    gray_image[:,i+150:i+200,2] = 255
```

In [54]:
```
cv.imshow('image gray', gray_image)

cv.waitKey(0)
cv.destroyAllWindows()
cv.waitKey(1)
```

Out[54]:
```
-1
```

In [55]:
```
n_im = gray_image[:,:,0]

cv.imshow('image gray', n_im)

cv.waitKey(0)
cv.destroyAllWindows()
cv.waitKey(1)
```

Out[55]:
```
-1
```

In [58]:
```
n_im1 = image[:,:,0]

cv.imshow('image gray', n_im1)

cv.waitKey(0)
cv.destroyAllWindows()
cv.waitKey(1)
```

Out[58]:
```
-1
```

In [67]:
```
im_second = cv.imread('../Homework1/images/image2.jpg')
cv.imshow('Second Image', im_second)

cv.waitKey(0)
cv.destroyAllWindows()
cv.waitKey(1)
```

Out[67]:
```
-1
```

In [68]:
```
# drowing line
cv.line(im_second, (0,0), (100,251), (255,0,0), 30) # starting point, ending point, color, thickness
cv.line(im_second, (0,512), (251,251), (0,0,255), 30)
cv.imshow('second image', im_second)

cv.waitKey(0)
cv.destroyAllWindows()
cv.waitKey(1)
```

Out[68]:
```
-1
```

In [72]:
```
# write to the local storage

is_written = cv.imwrite('new_image_second.jpg',im_second)

if is_written:
    print('successfully written')
else:
    print('failed!')
```

```
successfully written
```

In [80]:
```
height = int(im_second.shape[0] * 50 / 100)
width = int(im_second.shape[1] * 50 / 100)

dimension = (width, height)

resized_image = cv.resize(im_second, dimension, interpolation = cv.INTER_AREA)

resized_image.shape

cv.imshow('Scaled Image', resized_image)

cv.waitKey(0)
cv.destroyAllWindows()
cv.waitKey(1)
```

Out[80]:
```
-1
```

In [ ]:
```

```