190456K

Avishka Perera

```
!pip install sympy
```

```
Requirement already satisfied: sympy in /usr/local/lib/python3.7/dist-packages (1.7.1
Requirement already satisfied: mpmath>=0.19 in /usr/local/lib/python3.7/dist-packages
```

```
import cv2
from sympy import isprime

import  numpy as np
import matplotlib.pyplot as plt

from google.colab import drive
```

```
drive.mount('/content/drive')
% cd '/content/drive/MyDrive/Academics/UoM-Course-Work/Semester-04/Computer-Vision/Inclass
! ls
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
/content/drive/MyDrive/Academics/UoM-Course-Work/Semester-04/Computer-Vision/Inclass/
gal_gaussian.png  gal_sandp.png  tom_dark.jpg
```

1.

```
for i in range(1,6):
  print(f'{i} :  {i**2}')
```

```
1 :  1
2 :  4
3 :  9
4 :  16
5 :  25
```

2.

```
for i in range(1,6):
  if not isprime(i):
    print(f'{i} :  {i**2}')
```

```
1 :  1
4 :  16
```

3.

```
sqrs = [i**2 for i in range(1,6)]
for i in range(5):
  print(f'{i+1} :  {sqrs[i]}')
```

```
1 :  1
2 :  4
3 :  9
4 :  16
5 :  25
```

4.

```
sqrs = [i**2 for i in range(1,6) if not isprime(i)]
for i in range(len(sqrs)):
  print(f'{i+1} :  {sqrs[i]}')
```

```
1 :  1
2 :  16
```

5.

(a)

```
mat_1 = np.array([
        [1,2],
        [3,4],
        [5,6]
       ])
mat_2 = np.array([
        [7,8,9,1],
        [1,2,3,4]
       ])
mat_c = np.matmul(mat_1, mat_2)
mat_c
```

```
array([[ 9, 12, 15,  9],
       [25, 32, 39, 19],
       [41, 52, 63, 29]])
```

(b)

```
mat_a = np.array([
        [1,2],
        [3,4],
        [5,6]
])

mat_b = np.array([
                [3,2],
                [5,4],
                [3,1]
])
mat_a*mat_b
```

```
array([[ 3,  4],
       [15, 16],
       [15,  6]])
```

6.

```
rand_arr = np.random.randint(0, 10, (5, 7))
res_arr = rand_arr[2:5, 1:3]

print(rand_arr)
print(res_arr)
print(f'The size of the resulting array is {res_arr.size}')
```

```
[[6 3 0 8 4 0 4]
 [8 0 7 4 6 0 1]
 [9 5 3 2 0 9 7]
 [3 4 1 2 9 7 8]
 [8 8 4 7 1 3 9]]
[[5 3]
 [4 1]
 [8 4]]
The size of the resulting array is 6
```

7.

```
print('Original array:\n', mat_a)
print('Broadcasted element:\n', 3 + mat_a)
print('Broadcasted row:\n', [[1,2]] + mat_a)
print('Broadcasted column:\n', [[1],[2],[3]] + mat_a)
```

```
Original array:
 [[1 2]
 [3 4]
 [5 6]]
Broadcasted element:
 [[4 5]
 [6 7]
 [8 9]]
```

```
Broadcasted row:
 [[2 4]
 [4 6]
 [6 8]]
Broadcasted column:
 [[2 3]
 [5 6]
 [8 9]]
```

8.

(a), (b)

```
m, c = 2 , -4
N = 10
ones_column = np.ones((N,1))
x = np.linspace(0 , N-1, N).reshape(N, 1)
x = np.hstack((x, ones_column))
sigma = 10
y = m*x + c + np.random.normal(0 , sigma , (N, 1 ))

np.matmul(np.matmul(np.linalg.inv(np.matmul(x.T,x)), x.T), y)
```

```
array([[ 1.25688069, -0.74311931],
       [-2.38978123, -0.38978123]])
```

9.

(a)

```
def sqrt_hyp_est(s):
  e = 0
  running_mantissa = s

  if s>=10:
    while running_mantissa >= 10:
      running_mantissa /= 10
      e += 1

  else:
    while running_mantissa < 1:
      running_mantissa *= 10
      e -= 1

  a = running_mantissa
  n = e/2
  sqrt = (-190/(a+20)+10)*10**(n)

  return sqrt
```

(c)

```
print(sqrt_hyp_est(64))
print(sqrt_hyp_est(75))
print(sqrt_hyp_est(100))
print(sqrt_hyp_est(1600))
```

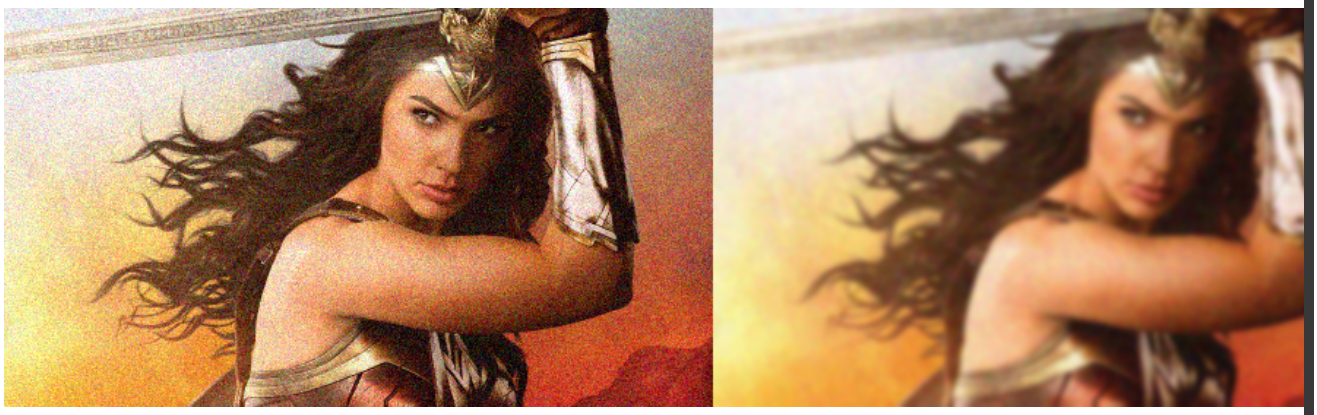
```
8.86396010804773
9.774312767793173
9.523809523809526
38.064453316841615
```

10.

```
from google.colab.patches import cv2_imshow
```

```
img = cv2.imread('gal_gaussian.png')
gaussian = cv2.GaussianBlur(img, (45, 45), 2)
compare = np.concatenate((img, gaussian), axis=1) #side by side comparison

cv2_imshow(compare)
# cv2.waitKey(0)
# cv2.destroyAllWindows()
```



11.

```
img = cv2.imread('gal_sandp.png')
median = cv2.medianBlur(img, 5)
compare = np.concatenate((img, median), axis=1) #side by side comparison

cv2_imshow(compare)
```

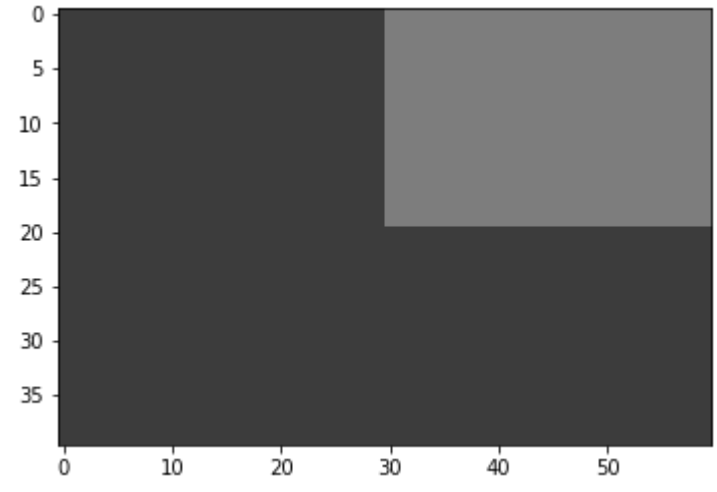
```
# cv2.imshow('img', compare)
# cv2.waitKey(0)
# cv2.destroyAllWindows()
```

12.

```
img = np.ones((40, 60), dtype=np.uint8)*60
img[:20, 30:] = 125

fig, ax = plt.subplots()

ax.imshow(img, cmap='gray', vmin=0, vmax=255)
plt.show()
```



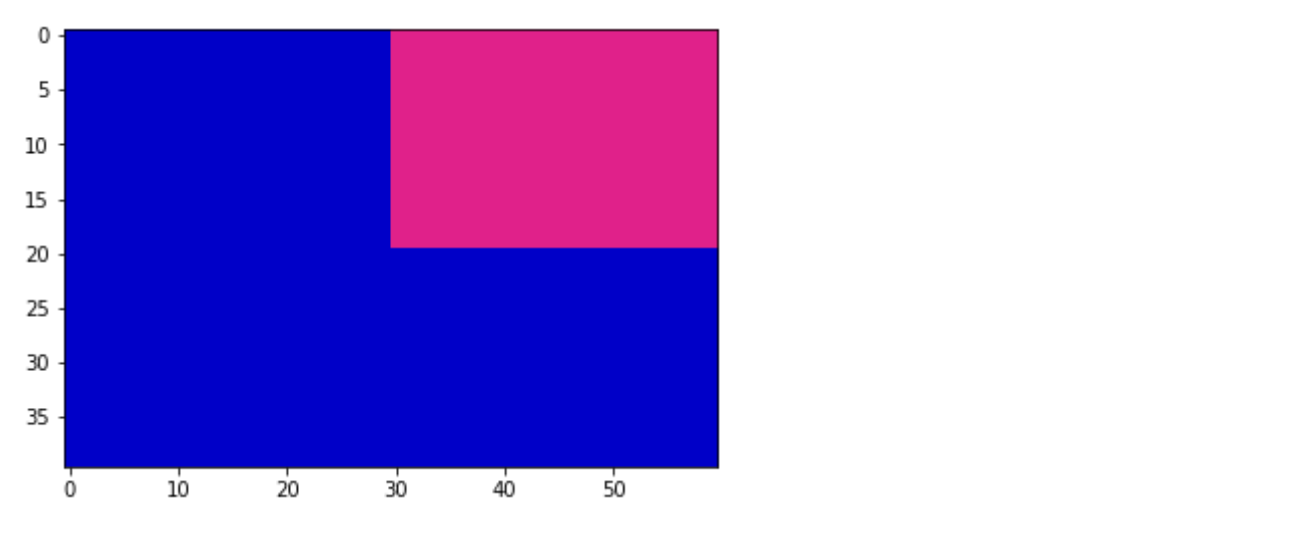
13.

```
img = np.zeros((40, 60, 3), dtype=np.uint8)
img[:,:,2] = 200
```

```
# Barbie pink's hex = #e0218a
img[:20, 30:, 0] = int('e0', 16)
img[:20, 30:, 1] = int('21', 16)
img[:20, 30:, 2] = int('8a', 16)

fig, ax = plt.subplots()

ax.imshow(img, vmin=0, vmax=255)
plt.show()
```



14.

```
img = cv2.imread('tom_dark.jpg')
bright = img + 30
compare = np.concatenate((img, bright), axis=1) #side by side comparison

cv2_imshow(compare)
```

✓ 2s completed at 7:21 PM