Name :T.sajeepan

INDEX NO:190539T

In [ ]:
```
for i in range(1,6):
  print(i,':',i**2)
```

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

In [ ]:
```
import sympy
for i in range(1,6):
  if not (sympy.isprime(i)):
    print(i,':',i**2)
```

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

In [ ]:
```
squares = [i**2 for i in range(1,6)]
for i, i2 in enumerate(squares):
  print(i,':',squares[i])
```

```
0 : 1
1 : 4
2 : 9
3 : 16
4 : 25
```

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

In [ ]:
```
A=np.array([[1,2],[3,4],[5,6]])
B=np.array([[7,8,9,1],[1,2,3,4]])
print(np.dot(A,B))
#A@B
```

```
[[ 9 12 15  9]
 [25 32 39 19]
 [41 52 63 29]]
```

In [ ]:
```
C=([[1,2],[3,4],[5,6]])
D=([[3,2],[5,4],[3,1]])
print(np.multiply(C,D))
```

```
[[ 3  4]
 [15 16]
 [15  6]]
```

In [ ]:
```
test_array=np.random.randint(10, size=(5, 7))
sub_array=test_array[2:4,3:4]
print(sub_array)
print(test_array)
```

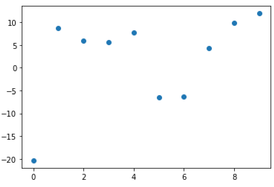
```
[[2]
 [9]]
[[0 0 1 7 3 3 1]
 [1 5 5 1 6 0 8]
 [0 8 8 2 0 1 3]
 [5 8 1 9 0 2 2]
 [7 2 6 5 6 0 0]]
```

In [ ]:
```
# one dimensional array addtion
A=np.array([1,2,3])
B=np.array([5,6,7])

print('one dimensional array addtion')
print(A+B)
# scalar and two-dimensional
S=2
C=np.array([[1,2,3],[7,6,5]])
print('scalar and two-dimensional')
print(C+S)

#one-dimensional and two-dimensional array addition
print('one-dimensional and two-dimensional array addition')
print(C+A)
```

```
one dimensional array addtion
[ 6  8 10]
scalar and two-dimensional
[[3 4 5]
 [9 8 7]]
one-dimensional and two-dimensional array addition
[[2 4 6]
 [8 8 8]]
```

In [ ]:
```
m,c =2, -4
N = 10
x = np.linspace (0,N-1,N).reshape (N, 1 )
sigma = 10
y = m*x + c + np.random.normal(0,sigma,(N, 1 ) )

plt.scatter(x,y)
```

Out[ ]:
```
<matplotlib.collections.PathCollection at 0x1590c1028e0>
```



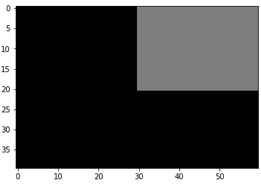
In [ ]:
```
X=np.append(np.ones((N,1)),x,axis =1)
ans=np.linalg.inv(X.T@X)@X.T@y
print(ans)
```

```
[[-4.44689879]
 [ 1.45877809]]
```

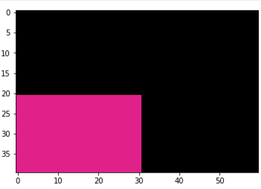
In [ ]:
```
import cv2 as cv

im = cv.imread(r'./gal_gaussian.png')

#view the original image
cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',im)
cv.waitKey(0)
cv.destroyAllWindows()

blur = cv.GaussianBlur(im,(5,5),0)
cv.imshow('Image',blur)
cv.waitKey(0)
cv.destroyAllWindows()
```

In [ ]:
```
im2 = cv.imread(r'./gal_sandp.png')

#view the original image
cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',im2)
cv.waitKey(0)
cv.destroyAllWindows()

medi = cv.medianBlur(im2,5)
cv.imshow('Image',medi)
cv.waitKey(0)
cv.destroyAllWindows()
```

In [ ]:
```
im3 = np.zeros((40,60),dtype=np.uint8)

im3[0:21,30:61]=125

fig, ax =plt.subplots()
ax.imshow(im3,cmap='gray',vmax=255,vmin=0)
plt.show()
```



In [ ]:
```
im4 = np.zeros((40,60,3),dtype=np.uint8)

im4[21:41,0:31]=[224,33,138]

fig, ax =plt.subplots()
ax.imshow(im4,vmax=255,vmin=0)
plt.show()
```



In [ ]:
```
im5 = cv.imread(r'./tom_dark.jpg')

alpha=2
beta=0
gamma=0

n_im= cv.addWeighted(im5,alpha,im5,beta,gamma)
cv.namedWindow('Image',cv.WINDOW_AUTOSIZE)
cv.imshow('Image',n_im)
cv.waitKey(0)
cv.destroyAllWindows()
```