Date 01-07-2021

Opening Image files in a notebook

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import cv2
```

In [2]:
```
img=cv2.imread('00-puppy.jpg')
img
```

Out[2]:
```
array([[[78, 81, 95],
        [80, 83, 97],
        [81, 84, 98],
        ...,
        [22, 27, 25],
        [22, 27, 25],
        [22, 27, 25]],

       [[79, 81, 95],
        [79, 81, 95],
        [79, 82, 96],
        ...,
        [22, 27, 25],
        [22, 27, 25],
        [22, 27, 25]],

       [[78, 81, 95],
        [77, 80, 94],
        [77, 80, 94],
        ...,
        [22, 27, 25],
        [22, 27, 25],
        [22, 27, 25]],

       ...,

       [[29, 29, 19],
        [31, 30, 20],
        [31, 30, 20],
        ...,
        [22, 30, 23],
        [23, 31, 24],
        [23, 31, 24]],

       [[31, 30, 20],
        [31, 30, 20],
        [30, 29, 19],
        ...,
        [22, 30, 23],
        [23, 31, 24],
        [23, 31, 24]],

       [[31, 30, 20],
        [30, 29, 19],
        [30, 29, 19],
        ...,
        [22, 30, 23],
        [23, 31, 24],
        [23, 31, 24]]], dtype=uint8)
```

In [3]:
```
type(img)
```

Out[3]:
```
numpy.ndarray
```

In [4]:
```
img.shape
```

Out[4]:
```
(1300, 1950, 3)
```

In [5]:
```
wrong_path_image=cv2.imread('wrong/path/which/does/not/exists/on/my/computer.png')
type(wrong_path_image)
```

Out[5]:
```
NoneType
```

In [6]:
```
plt.imshow(img)
```

Out[6]:
```
<matplotlib.image.AxesImage at 0x1f32548754e0>
```

The above Output is a strange Output because -

MatPlotLib read image in format of RED-GREEN-BLUE

OpenCV read image in format of BLUE-GREEN-RED

In [7]:
```
img=cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

In [8]:
```
plt.imshow(img)
```

Out[8]:
```
<matplotlib.image.AxesImage at 0x1f32576af0>
```

In [9]:
```
img_gray=cv2.imread('00-puppy.jpg',cv2.IMREAD_GRAYSCALE)
img_gray
```

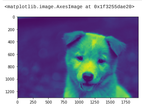
Out[9]:
```
array([[95, 97, 98, ..., 26, 26, 26],
       [95, 95, 96, ..., 26, 26, 26],
       [95, 94, 94, ..., 26, 26, 26],
       ...,
       [29, 28, 28, ..., 27, 28, 28],
       [30, 29, 29, ..., 27, 28, 28],
       [30, 29, 29, ..., 27, 28, 28]], dtype=uint8)
```

In [10]:
```
img_gray.shape
```

Out[10]:
```
(1300, 1950)
```

In [11]:
```
plt.imshow(img_gray)
```

Out[11]:
```
<matplotlib.image.AxesImage at 0x1f3255dae20>
```

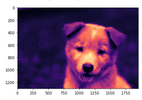


In [12]:
```
plt.imshow(img_gray,cmap='gray')
```

Out[12]:
```
<matplotlib.image.AxesImage at 0x1f325646760>
```

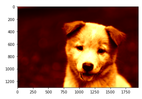
In [13]:
```
plt.imshow(img_gray,cmap='magma')
```

Out[13]:
```
<matplotlib.image.AxesImage at 0x1f32569afa0>
```



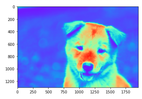
In [14]:
```
plt.imshow(img_gray,cmap='afmhot')
```

Out[14]:
```
<matplotlib.image.AxesImage at 0x1f325703820>
```



In [15]:
```
plt.imshow(img_gray,cmap='rainbow')
```

Out[15]:
```
<matplotlib.image.AxesImage at 0x1f32576b0d0>
```

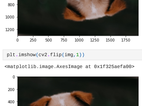


In [16]:
```
plt.imshow(cv2.resize(img,(2920,700)))
```

Out[16]:
```
<matplotlib.image.AxesImage at 0x1f3257be9d0>
```

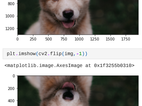
In [17]:
```
plt.imshow(cv2.flip(img,0))
```

Out[17]:
```
<matplotlib.image.AxesImage at 0x1f32581f1f0>
```



In [18]:
```
plt.imshow(cv2.flip(img,1))
```

Out[18]:
```
<matplotlib.image.AxesImage at 0x1f325aefa00>
```



In [19]:
```
plt.imshow(cv2.flip(img,-1))
```

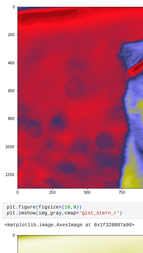
Out[19]:
```
<matplotlib.image.AxesImage at 0x1f32550310>
```

In [20]:
```
cv2.imwrite('Flipped Image.png',cv2.flip(img,-1))
```

Out[20]:
```
True
```

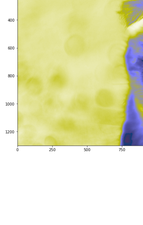
In [21]:
```
plt.figure(figsize=(16,9))
plt.imshow(img_gray,cmap='gist_stern')
```

Out[21]:
```
<matplotlib.image.AxesImage at 0x1f3253b90d0>
```



In [22]:
```
plt.figure(figsize=(16,9))
plt.imshow(img_gray,cmap='gist_stern_r')
```

Out[22]:
```
<matplotlib.image.AxesImage at 0x1f32809fa90>
```



Date 01-07-2021

Drawing on Images - Part One - Basic Shapes

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import cv2
```

In [2]:
```
def blank_image_generator(tuple_containing_size_of_blank_image=(900,1600,3),tuple_of_figsize=(16,9)):
    blank_img=np.zeros(tuple_containing_size_of_blank_image,dtype=int)
    plt.figure(figsize=tuple_of_figsize)
    return blank_img
```

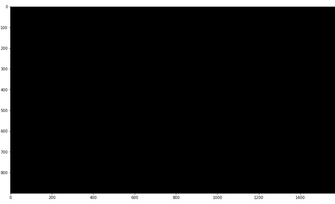
In [3]:
```
blank_image_generator().shape
```

Out[3]:
```
(900, 1600, 3)
```

<Figure size 1152x648 with 0 Axes>

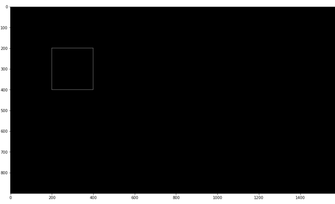
In [4]:
```
plt.imshow(blank_image_generator())
```

Out[4]:
```
<matplotlib.image.AxesImage at 0x248276a7fd0>
```



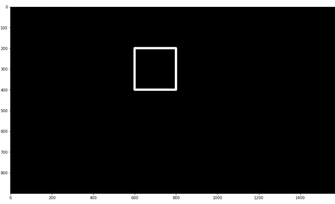
In [5]:
```
plt.imshow(cv2.rectangle(blank_image_generator(),pt1=(200,200),pt2=(400,400),color=(255,255,255)))
```

Out[5]:
```
<matplotlib.image.AxesImage at 0x24821bc8310>
```



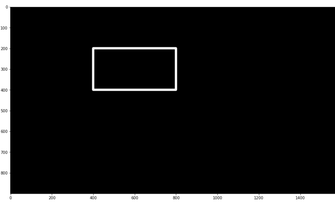
In [6]:
```
plt.imshow(cv2.rectangle(blank_image_generator(),pt1=(600,200),pt2=(800,400),color=(255,255,255),thickness=10))
```

Out[6]:
```
<matplotlib.image.AxesImage at 0x2482193bca0>
```



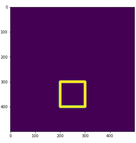
In [7]:
```
plt.imshow(cv2.rectangle(blank_image_generator(),pt1=(400,200),pt2=(800,400),color=(255,255,255),thickness=10))
```

Out[7]:
```
<matplotlib.image.AxesImage at 0x248219706a0>
```



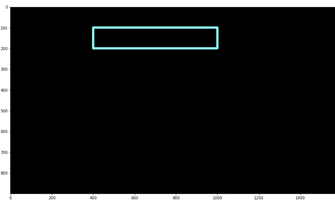
In [8]:
```
plt.imshow(cv2.rectangle(blank_image_generator((500,500),tuple_of_figsize=(8,6)),pt1=(200,300),pt2=(300,400),color=(255,255,255),thickness=10))
```

Out[8]:
```
<matplotlib.image.AxesImage at 0x24821d8dcc10>
```



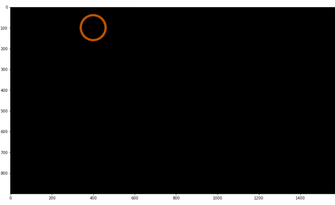
In [9]:
```
plt.imshow(cv2.rectangle(blank_image_generator(),pt1=(400,200),pt2=(1000,100),color=(143, 255, 251),thickness=10))
```

Out[9]:
```
<matplotlib.image.AxesImage at 0x24821e3b820>
```



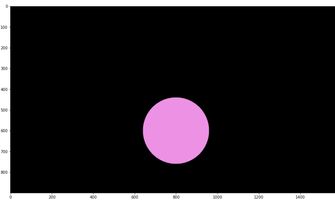
In [10]:
```
plt.imshow(cv2.circle(blank_image_generator(),center=(400,100),radius=60,color=(191,  84,  1),thickness=10))
```

Out[10]:
```
<matplotlib.image.AxesImage at 0x24821e90af0>
```



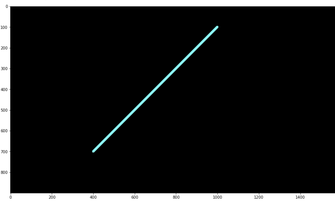
In [11]:
```
plt.imshow(cv2.circle(blank_image_generator(),center=(800,600),radius=160,color=(237, 145, 228),thickness=-1))
```

Out[11]:
```
<matplotlib.image.AxesImage at 0x24821ef9e20>
```



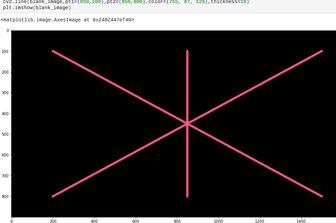
In [12]:
```
plt.imshow(cv2.line(blank_image_generator(),pt1=(400,700),pt2=(1000,100),color=(143, 255, 251),thickness=10))
```

Out[12]:
```
<matplotlib.image.AxesImage at 0x24822e61d00>
```



In [13]:
```
blank_image = blank_image_generator()
cv2.line(blank_image,pt1=(200,800),pt2=(1500,100),color=(255,  87,  129),thickness=10)
cv2.line(blank_image,pt1=(200,100),pt2=(1500,800),color=(255,  87,  129),thickness=10)
cv2.line(blank_image,pt1=(850,100),pt2=(850,800),color=(255,  87,  129),thickness=10)
plt.imshow(blank_image)
```

Out[13]:
```
<matplotlib.image.AxesImage at 0x2482447ef40>
```



Date 01-07-2021

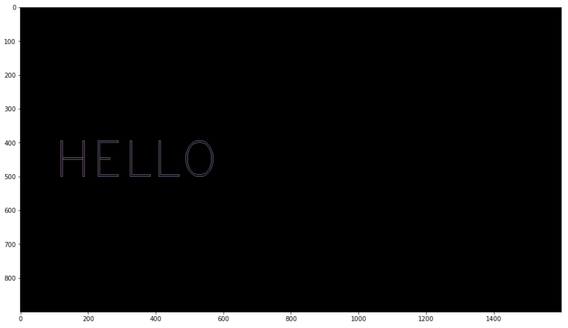
Drawing on Images Part Two - Text and Polygons

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import cv2
```

In [2]:
```
def blank_image_generator(tuple_containing_size_of_blank_image=(900,1600,3),tuple_of_figsize=(16,9)):
    blank_img=np.zeros(tuple_containing_size_of_blank_image,dtype=int)
    plt.figure(figsize=tuple_of_figsize)
    return blank_img
```

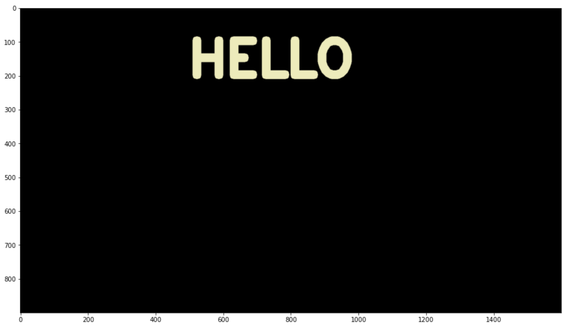
In [3]:
```
plt.imshow(cv2.putText(blank_image_generator(),
                       text="HELLO",
                       org=(100,500),
                       fontFace=cv2.FONT_HERSHEY_DUPLEX,
                       fontScale=5,
                       color=(216, 187, 237)))
```

Out[3]:
```
<matplotlib.image.AxesImage at 0x1e48912d1f0>
```



In [4]:
```
plt.imshow(cv2.putText(blank_image_generator(),
                       text="HELLO",
                       org=(500,200),
                       fontFace=cv2.FONT_HERSHEY_DUPLEX,
                       fontScale=5,
                       color=(237, 235, 187),
                       lineType=cv2.LINE_8,
                       thickness=20))
```

Out[4]:
```
<matplotlib.image.AxesImage at 0x1e4880e0760>
```



Custom Polygons

In [5]:
```
custom_polygon_vertices_1=np.array([ [100,300], [200,200], [400,300], [200,400] ],
                                   dtype=int)
custom_polygon_vertices_1
```

Out[5]:
```
array([[100, 300],
       [200, 200],
       [400, 300],
       [200, 400]])
```

In [6]:
```
custom_polygon_vertices_1.shape
```

Out[6]:
```
(4, 2)
```

In [7]:
```
custom_polygon_vertices_reshaped_1=custom_polygon_vertices_1.reshape((4,1,2))
custom_polygon_vertices_reshaped_1
```

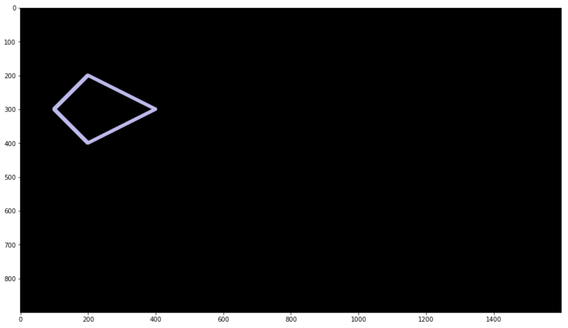
Out[7]:
```
array([[[100, 300]],

       [[200, 200]],

       [[400, 300]],

       [[200, 400]]])
```

In [8]:
```
custom_polygon_vertices_reshaped_1.shape
```

Out[8]:
```
(4, 1, 2)
```

In [9]:
```
plt.imshow(cv2.polylines(blank_image_generator(),
           [custom_polygon_vertices_reshaped_1],
           isClosed=True,
           color=(190, 187, 237),
           thickness=10))
```

Out[9]:
```
<matplotlib.image.AxesImage at 0x1e488328af0>
```



In [10]:
```
custom_polygon_vertices_2=np.array([ [200,150],[300,400],[800,300],[600,200] ],
                                   dtype=int)
custom_polygon_vertices_2
```

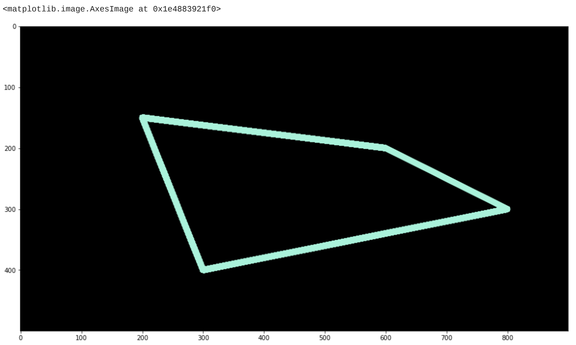
Out[10]:
```
array([[200, 150],
       [300, 400],
       [800, 300],
       [600, 200]])
```

In [11]:
```
custom_polygon_vertices_reshaped_2=custom_polygon_vertices_2.reshape((4,1,2))
custom_polygon_vertices_reshaped_2
```

Out[11]:
```
array([[[200, 150]],

       [[300, 400]],

       [[800, 300]],

       [[600, 200]]])
```

In [12]:
```
plt.imshow(cv2.polylines(blank_image_generator(tuple_containing_size_of_blank_image=(500,900,3)),
           [custom_polygon_vertices_reshaped_2],
           isClosed=True,
           color=(170, 242, 219),
           thickness=10))
```

Out[12]:
```
<matplotlib.image.AxesImage at 0x1e4883921f0>
```



Date 02-07-2021

Image-Basics-Assessment

Image Basics Assessment

Complete the Tasks in bold below. Keep in mind, you may need to run some of these tasks as Python scripts.

TASK: Open the *dog_backpack.jpg* image from the DATA folder and display it in the notebook. Make sure to correct for the RGB order.

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import cv2
```

In [2]:
```
img=cv2.imread('dog_backpack.jpg')
img
```

Out[2]:
```
array([[[249, 248, 250],
        [249, 248, 250],
        [249, 248, 250],
        ...,
        [249, 248, 250],
        [249, 248, 250],
        [249, 248, 250]],

       [[249, 248, 250],
        [249, 248, 250],
        [249, 248, 250],
        ...,
        [249, 248, 250],
        [249, 248, 250],
        [249, 248, 250]],

       [[249, 248, 250],
        [249, 248, 250],
        [249, 248, 250],
        ...,
        [249, 248, 250],
        [249, 248, 250],
        [249, 248, 250]],

       ...,

       [[ 39,  40,  50],
        [ 39,  39,  51],
        [ 38,  38,  52],
        ...,
        [ 86,  80,  73],
        [ 81,  77,  72],
        [ 79,  75,  70]],

       [[ 40,  41,  51],
        [ 40,  40,  52],
        [ 39,  38,  52],
        ...,
        [ 88,  82,  75],
        [ 84,  78,  73],
        [ 79,  75,  70]],

       [[ 40,  42,  52],
        [ 38,  40,  51],
        [ 39,  39,  53],
        ...,
        [ 89,  83,  78],
        [ 84,  78,  71],
        [ 78,  75,  67]]], dtype=uint8)
```

In [3]:
```
plt.imshow(img)
```

Out[3]:
```
<matplotlib.image.AxesImage at 0x244dbcc0b80>
```

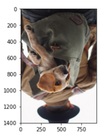
In [4]:
```
img=cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img)
```

Out[4]:
```
<matplotlib.image.AxesImage at 0x244dbdac700>
```

TASK: Flip the image upside down and display it in the notebook.

In [5]:
```
plt.imshow(cv2.flip(img,0))
```

Out[5]:
```
<matplotlib.image.AxesImage at 0x244dbe08ac0>
```



TASK: Draw an empty RED rectangle around the dogs face and display the image in the notebook.

In [6]:
```
img_empty_rect=img.copy()
img_empty_rect = cv2.rectangle(img_empty_rect,pt1=(250,400), pt2=(600,700), color=(255,0,0), thickness=10)
plt.imshow(img_empty_rect)
```

Out[6]:
```
<matplotlib.image.AxesImage at 0x244dbe61c10>
```

In [7]:
```
plt.imshow(img)
```

Out[7]:
```
<matplotlib.image.AxesImage at 0x244dbec2310>
```

TASK: Draw a BLUE TRIANGLE in the middle of the image. The size and angle is up to you, but it should be a triangle (three sides) in any orientation.

In [8]:
```
custom_triangle_vertices=np.array([ [500, 300], [150, 700], [650 ,700]],
                                  dtype=int)

custom_triangle_vertices_reshaped_1=custom_triangle_vertices.reshape((-1,1,2))

img_empty_triangle=img.copy()
img_empty_triangle =cv2.polylines(img_empty_triangle,
            [custom_triangle_vertices_reshaped_1],
            isClosed=True,
            color=(0, 0, 255),
            thickness=10)
plt.imshow(img_empty_triangle)
```

Out[8]:
```
<matplotlib.image.AxesImage at 0x244dbf1a340>
```

BONUS TASK. Can you figure our how to fill in this triangle? It requires a different function that we didn't show in the lecture! See if you can use google search to find it.

CLICK ME FOR A DIRECT LINK TO THE HINT

In [9]:
```
custom_triangle_vertices=np.array([ [500, 300], [150, 700], [650 ,700]],
                                  dtype=int)

custom_triangle_vertices_reshaped_1=custom_triangle_vertices.reshape((-1,1,2))

img_filled_triangle=img.copy()
img_filled_triangle =cv2.polylines(img_filled_triangle,
            [custom_triangle_vertices_reshaped_1],
            isClosed=True,
            color=(0, 0, 255),
            thickness=10)
plt.imshow(img_filled_triangle)
```

Out[9]:
```
<matplotlib.image.AxesImage at 0x244dbf69c10>
```

In [10]:
```
custom_triangle_vertices=np.array([ [500, 300], [150, 700], [650 ,700]],
                                  dtype=int)

custom_triangle_vertices_reshaped_1=custom_triangle_vertices.reshape((-1,1,2))

img_filled_triangle=img.copy()


img_filled_triangle=cv2.fillPoly(img_filled_triangle, [custom_triangle_vertices_reshaped_1], color=(0, 0, 255))

plt.imshow(img_filled_triangle)
```

Out[10]:
```
<matplotlib.image.AxesImage at 0x244dbfc66a0>
```

TASK: (NOTE: YOU WILL NEED TO RUN THIS AS A SCRIPT). Create a script that opens the picture and allows you to draw empty red circles whever you click the RIGHT MOUSE BUTTON DOWN.