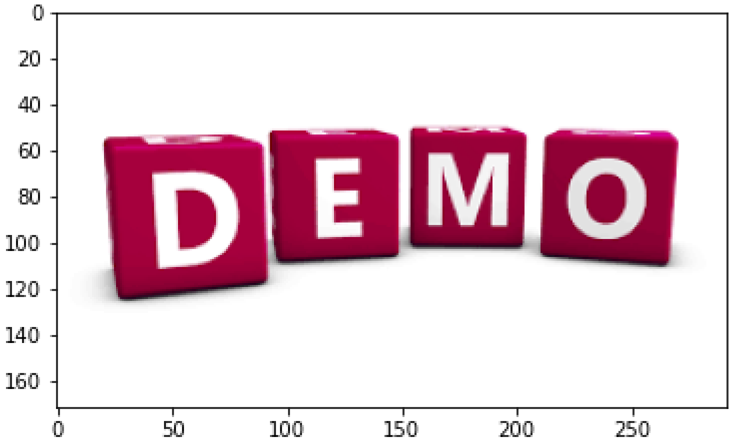
1806554 Ganesh Bhandarkar Python Assignment 4

1

In [157…

```
# 1 method
import matplotlib.pyplot as plt
import cv2
i = cv2.imread('demo.png')
plt.imshow(i)
# 2 method
out = cv2.imshow('Image Output',i)
cv2.waitKey(0)
cv2.destroyAllWindows()
# 3 method
from PIL import Image
im = Image.open(r"demo.png")
im.show()
```



2

In [6]:

```
i.shape
```

Out[6]:

```
(172, 292, 3)
```

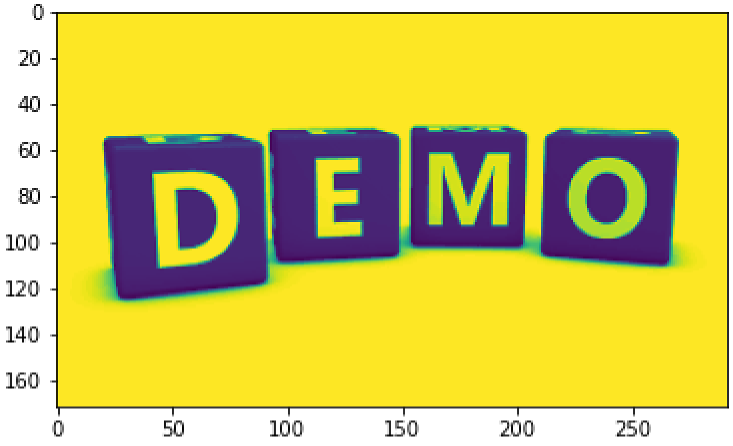
3

In [7]:

```
gi = cv2.cvtColor(i, cv2.COLOR_BGR2GRAY)
plt.imshow(gi)
```

Out[7]:

```
<matplotlib.image.AxesImage at 0x1abd0295310>
```



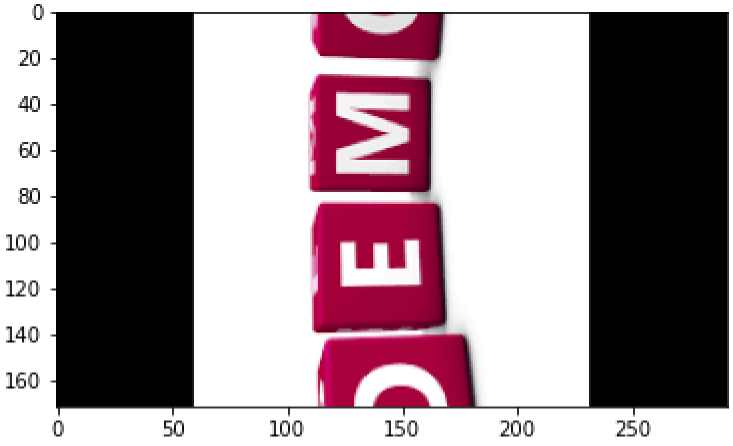
4

In [15]:

```
import imutils as im
ih = im.rotate(i,90)
plt.imshow(ih)
```

Out[15]:

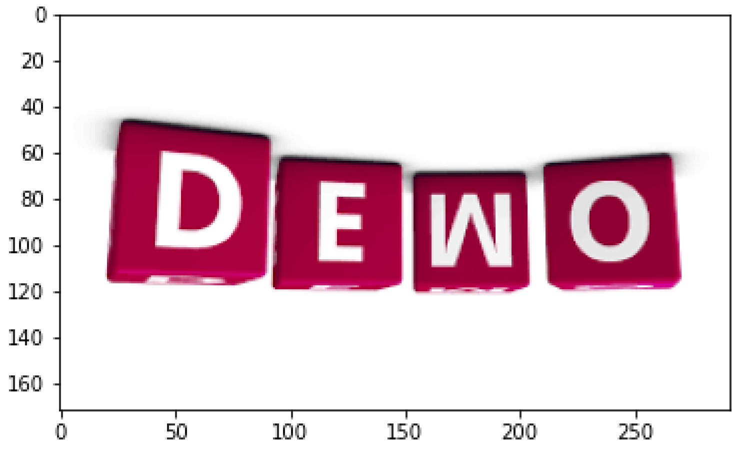
```
<matplotlib.image.AxesImage at 0x1abd0815e50>
```



In [20]:

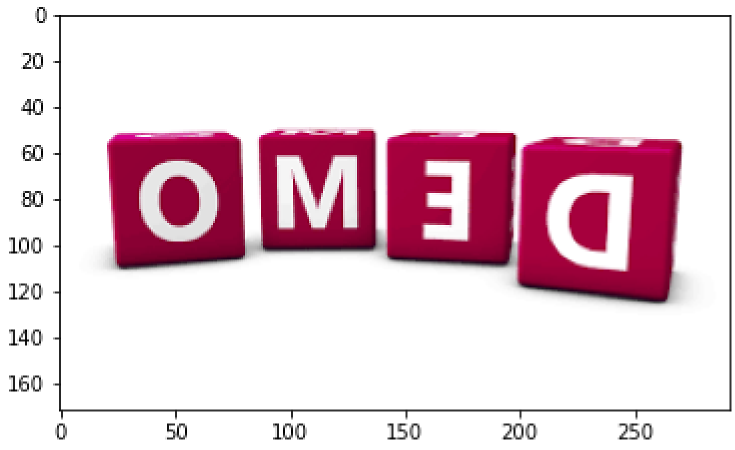
```
# vertical flip
iv = cv2.flip(i,0)
plt.imshow(iv)
```

Out[20]:
```
<matplotlib.image.AxesImage at 0x1abd072a5e0>
```



In [21]:
```
# horizontal flip
iv = cv2.flip(i,1)
plt.imshow(iv)
```

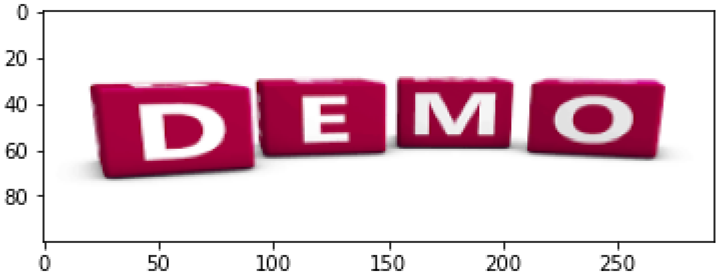
Out[21]:
```
<matplotlib.image.AxesImage at 0x1abd1306790>
```



5

In [23]:
```
# resize
new_size = 100
ds = (i.shape[1],new_size)
output = cv2.resize(i, ds, interpolation = cv2.INTER_AREA)
plt.imshow(output)
```

Out[23]:
```
<matplotlib.image.AxesImage at 0x1abd11b2160>
```



6

In [24]:

```
# new window
out = cv2.imshow('Image Output',i)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

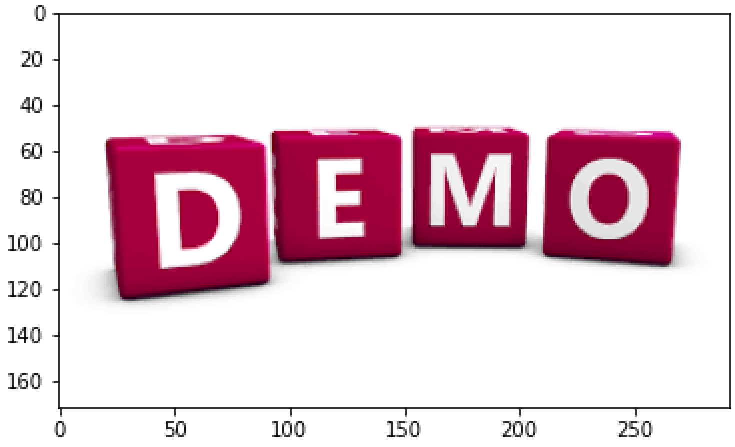
7

In [25]:

```
# show in jupyter
plt.imshow(i)
```

Out[25]:

```
<matplotlib.image.AxesImage at 0x1abd0e44ee0>
```



8

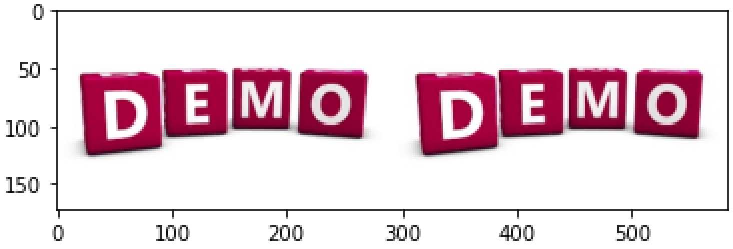
In [37]:

```
# concat side by side horizontal
im_h = cv2.hconcat([i,i])
cv2.imwrite('opencv_hconcat.jpg', im_h)
```

```
j = cv2.imread('opencv_hconcat.jpg')
plt.imshow(j)
```

Out[37]:

```
<matplotlib.image.AxesImage at 0x1abd1094550>
```



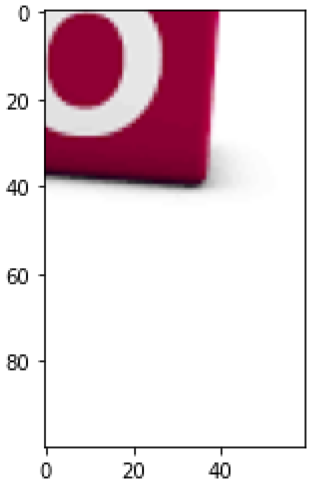
9

In [46]:

```
# crop image
#i.shape
val = input()
(hs,he,ws,we) = map(int,val.strip().split(' '))
cropped = i[hs:he, ws:we]
plt.imshow(cropped)
```

Out[46]:

```
<matplotlib.image.AxesImage at 0x1abd07b6190>
```



10

In [53]:

```
# Binarized Image
import numpy as np
img = cv2.imread('demo.png')
```

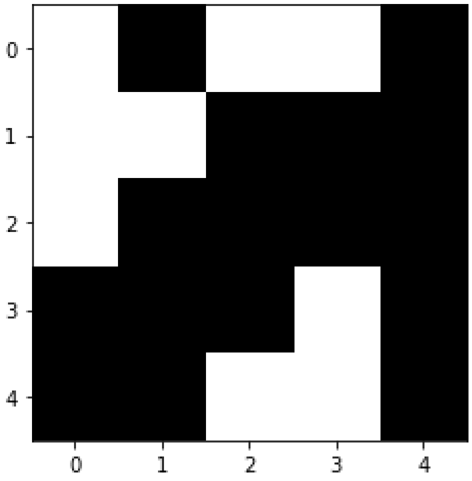
```
height,width,channels = img.shape
img_binary = np.zeros((height,width,1))
img_grayscale = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
(thresh, img_binary) = cv2.threshold(img_grayscale, 128, 255,
cv2.THRESH_BINARY)
cv2.imwrite('image_binary.jpg',img_binary)
cv2.imshow('image',img_binary)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

11

In [154…

```
# binary image
matz = np.random.randint(2,size = (5,5))
print(matz)
plt.imshow(matz, cmap="gray")
plt.show()
```

```
[[1 0 1 1 0]
 [1 1 0 0 0]
 [1 0 0 0 0]
 [0 0 0 1 0]
 [0 0 1 1 0]]
```



12

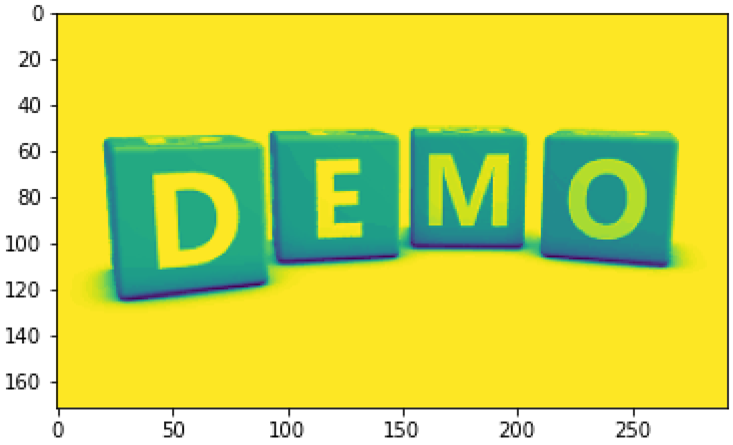
In [80]:

```
# RGB Channels
rgbImage = cv2.imread('demo.png')
```

```
# redChannel = rgbImage[:,:,1]
# greenChannel = rgbImage[:,:,2]
# blueChannel = rgbImage[:,:,2]
def channelShift(i):
    return rgbImage[:,:,i]
x = (0,1,2)
r,g,b = map(channelShift,x)
plt.imshow(r)
```

Out[80]:

```
<matplotlib.image.AxesImage at 0x1abd47e9880>
```

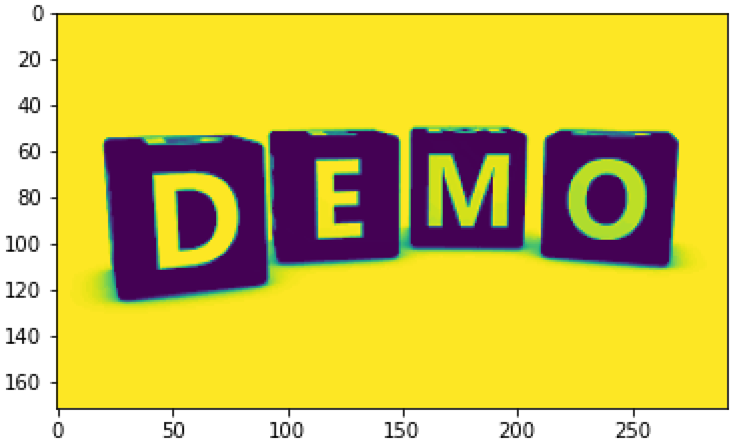


In [81]:

```
plt.imshow(g)
```

Out[81]:

```
<matplotlib.image.AxesImage at 0x1abd483eee0>
```

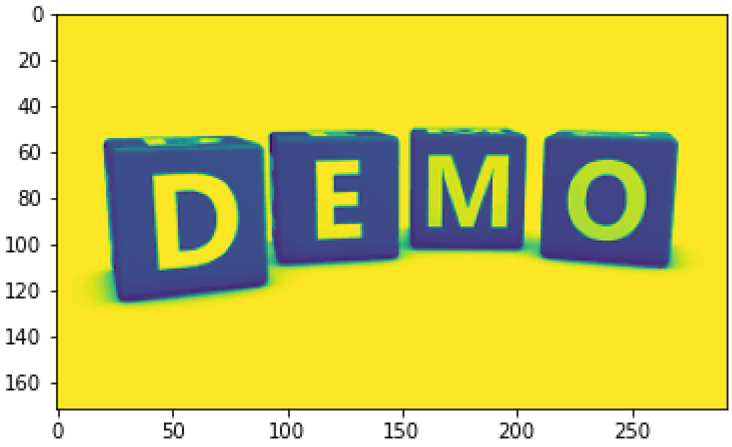


In [82]:

```
plt.imshow(b)
```

Out[82]:

```
<matplotlib.image.AxesImage at 0x1abd489bb20>
```



13

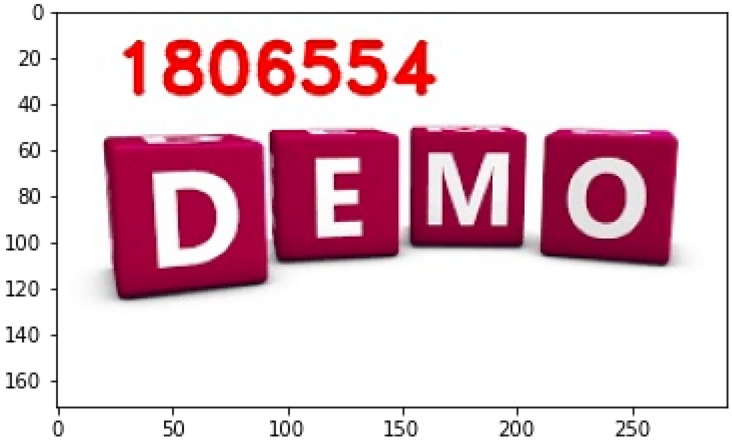
In [155…

```
# Text on Image
img = cv2.imread('demo.png')
font = cv2.FONT_HERSHEY_DUPLEX
org = (25,35)
fontScale = 1
color = (255,0,0)
thickness = 2
sentence = input()
image =
cv2.putText(img,sentence,org,font,fontScale,color,thickness,cv2.L

cv2.imwrite('opencv_T.jpg',image)
j = cv2.imread('opencv_T.jpg')
plt.imshow(j)
```

Out[155…

```
<matplotlib.image.AxesImage at 0x1abd0afd040>
```



14

In [109…

```
# images in folder
import cv2
import os

def printImageNames(folder):
    images = []
    for filename in os.listdir(folder):
        img = cv2.imread(os.path.join(folder,filename))
        if img is not None:
            images.append(filename)
    images.sort()
    return images
folder="C:/Users/KIIT/Documents/College-Stuff/T&T/python scripts/"
printImgNames(folder)
```

Out[109…

```
['demo.png', 'image_binary.jpg', 'opencv_T.jpg', 'opencv_hconcat.jpg']
```

15

In [136…

```
# count Images
def cntImages(path):
    cnt = 0
    for filename in os.listdir(path):
        img = cv2.imread(os.path.join(path,filename))
        if img is not None:
            cnt+=1
    return cnt
path="C:/Users/KIIT/Documents/College-Stuff/T&T/python scripts/"
cntImages(path)
```

Out[136… 4

16

In [152…

```
# copy and Name change
import shutil
import os
src_dir = "C:/Users/KIIT/Documents/College-Stuff/T&T/python scripts/"
dst_dir = "C:/Users/KIIT/Documents/College-Stuff/T&T/python scripts/work/"
for filename in glob.glob(os.path.join(src_dir, '*.jpg')):
    shutil.copy(filename, dst_dir)
for cnt,filename in enumerate(os.listdir("work")):
    new_name = str(filename)
    dst = new_name[:-4] + "_" + str(cnt) + ".jpg"
    src = 'work/' + filename
    dst = 'work/' + dst
    os.rename(src,dst)
```

17

In [127…

```
# Specifications
def Info(folder):
    for filename in os.listdir(folder):
        img = cv2.imread(os.path.join(folder,filename))
        if img is not None:
            with open("images_info.txt","a") as f:
                f.write("filename : "+str(filename))
                f.write(" Info -> \n")
                f.write("height : "+str(img.shape[0])+" "
                        "width : "+str(img.shape[1])+" "
                        "channels : "+str(img.shape[2]))
                f.write("\n")
    with open("images_info.txt") as f:
        for line in f:
            print(line,end = "")
folder="C:/Users/KIIT/Documents/College-Stuff/T&T/python scripts/"
Info(folder)
```

```
filename : demo.png Info ->
height : 172 width : 292 channels : 3
filename : image_binary.jpg Info ->
height : 172 width : 292 channels : 3
filename : opencv_hconcat.jpg Info ->
height : 172 width : 584 channels : 3
filename : opencv_T.jpg Info ->
height : 172 width : 292 channels : 3
```

In [ ]: