/* Most common used flex styles*/ /* Basic flexbox reverse styles */ /* Flexbox alignment */ /* Non-flexbox positioning helper styles */

```
def imshows(ImageData, LabelData, rows, cols, gridType = True):
  ImageArray = list(ImageData)
  LabelArray = list(LabelData)
  from matplotlib import pyplot as plt
  fig = plt.figure(figsize=(10,10))
  for i in range(1, cols*rows +1):
    fig.add_subplot(rows, cols, i)
    image = ImageArray[i - 1]
    if (len(image.shape)<3):
      plt.imshow(image, plt.cm.gray)
      plt.grid(gridType)
    else:
      plt.imshow(image)
      plt.grid(gridType)
    plt.title(LabelArray[i-1])
  plt.show()
```

```
from google.colab import drive
drive.mount('/content/gdrive')
linken = "gdirve/My Drive/TTH_Digital_Image_Processing/Dataset_07/"
```

```
import os
linken = "gdrive/My Drive/TTH_Digital_Image_Processing/Dataset_07/"
print(os.path.isdir(linken))
print(os.path.isfile(linken + "cat.jpg"))
print(os.path.isfile(linken + "Animal.jpeg"))
```

```
def segmentObjects1(pic):
  from pylab import imread
  from skimage.color import rgb2gray
  import cv2
  import numpy as np

  imglink= linken + pic
  img = imread(imglink)
  gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
  #imshows([img, gray],["Image","Gray"],1,2)

  gray_blur = cv2.GaussianBlur(gray,(5,5),0)
  #gray_blur = cv2.bilateralFilter(gray,9,50,50)
  threshing = cv2.adaptiveThreshold(gray_blur, 255,
                          cv2.ADAPTIVE_THRESH_GAUSSIAN_C,cv2.THRESH_BINARY_INV, 1
  #imshows([img, gray, gray_blur, threshing],['img','gray','gray_blur','threshin

  from skimage.morphology import erosion, dilation, closing, white_tophat
  from skimage.morphology import black_tophat, skeletonize, convex_hull_image
  from skimage.morphology import disk

  selem = disk(3)
  eroded = erosion(threshing, selem)
  image_removenoise = dilation(eroded, selem)

  kernel = np.ones((7,7),np.uint8)
  closing = cv2.morphologyEx(image_removenoise, cv2.MORPH_CLOSE, kernel, iterati

  selem = disk(5)
  closing = erosion(closing, selem)
  closing = dilation(closing, selem)

  #imshows([threshing, image_removenoise, closing],['threshing', 'image_removeno
  from scipy import ndimage

  cont_img = closing.copy()
  roi = img.copy()
  img_fill_holes = ndimage.binary_fill_holes(cont_img)
  inv2, contours, hierarchy = cv2.findContours(img_fill_holes.astype(np.uint8),cv

  cv2.drawContours(roi, contours, -1,(255,0,0),10)
  #imshows([closing, img_fill_holes, roi],['closing','img_fill_holes','roi'],1,3

  imshows([img, roi],['IMG', 'ROI'],1,2)
```

```
def segmentObjects2(pic):
 import numpy as np
 import cv2
 from matplotlib import pyplot as plt

 imglink= linken + pic
 img = cv2.imread(imglink)
 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

 gray_blur = cv2.GaussianBlur(gray,(15,15),0)
 gray_blur = cv2.bilateralFilter(gray,25,125,125)
 threshing = cv2.adaptiveThreshold(gray_blur, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_

 #imshows([img, gray, gray_blur, threshing],['img','gray','gray_blur','threshin

 from skimage.morphology import erosion, dilation, opening, closing, white_topha
 from skimage.morphology import black_tophat, skeletonize, convex_hull_image
 from skimage.morphology import disk

 selem = disk(3)
 eroded = erosion(threshing, selem)
 image_removenoise = dilation(eroded,selem)

 kernel = np.ones((3,3), np.uint8)
 closing = cv2.morphologyEx(image_removenoise, cv2.MORPH_CLOSE, kernel, iterati

 #imshows([threshing, image_removenoise, closing],['threshing','image_removenois

 from scipy import ndimage
 roi = img.copy()
 cont_img = closing.copy()
 img_fill_holes = ndimage.binary_fill_holes(cont_img)
 im2, contours, hierarchy = cv2.findContours(img_fill_holes.astype(np.uint8),
                                             cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPL
 cv2.drawContours(roi, contours, -1, (0, 255, 0), 5)

 #imshows([cont_img, img_fill_holes, roi],['closing','img_fill_holes','roi'],1,3
 imshows([img, roi],['IMG', 'ROI'],1,2)
```

```
segObj1 = "Animal.jpeg","cat.jpg","Flower.jpeg","Hand Sign.jpg","table.jpg","tra
for pic in segObj1:
  segmentObjects1(pic)
```

```
segObj2 = "coin-detection.jpg","gray_image.png"
for pic in segObj2:
  segmentObjects2(pic)
```