Mask

It is called thresholding.

grayscale img

```
ret, mask = cv2.threshold(img, threshold_val, maxVal, cv2.TYPE)
```

ret: advanced usage
mask: our mask
threshold_val: 0 <= val <= 255
cv2.TYPE: cv2.THRESH_BINARY, cv2.THRESH_BINARY_INV, ...

Symbols: page 14

This mask only uses one value as a limit to ~~make~~ give white/black outputs.

in future pages, we will see:

1) Range of ~~to~~ mask (color ranges of thresholding) (page 12)
2) Adaptive thresholding

→ Here, we will see how to use plots ~~to~~ to show results, and how plots work.

Plots

- once you want to plot/show an image, you define the size of the grid & the position of the image in the grid.

```
plt.subplot(2, 10, 1); plt.imshow(img, 'gray')
```

2: rows; 10: cols; 1: position of the img; 'gray': grayscale

so, now we have the grid, we can use it:

```
plt.subplot(2, 10, 3); plt.imshow(img); plt.title('titles')
```

'titles': img title

once we position our images, we do:

```
plt.show() # and our images will appear.
```

if we use a different grid ~~we will~~ the program will create another plot slide, ~~and we will~~ and we will need to call plt.show() twice to see both of the plot slides.

```
plt.subplot(1, 5, 2); ....
plt.show()                  # if we didn't use this here, we
plt.subplot(2, 4, 8); .... # would get only the second plot
plt.show()                  # slide (the latest)
```

```
plt.xticks(), plt.yticks()
```