SCALING

- You can resize a ~~the~~ image, using:

INTERPOLATION METHODS.

```
result = cv2. resize( img , None , fx = x , fy = y, interpolations cv2.INTER_CUBIC)
                                                                         INTER_AREA
                                                                         INTER_LINEAR
```

img to be resized

size of the new image. for example I can write (400, 500) instead of None, and it will look like:

```
result = cv2 resize(img, (400,500))
```

You can add an interpolation method here, `cv2.resize(img,(x,y), cv2.INTER_....)`

width

height

x and y define the percentage of the new resized img. if we want the result to be the same as the input img we give x = 1 and y = 1. if we want half the original, then x = y = 1/2.

- Note that we can skip the interpolation method, and use just:

```
result = cv2. resize( img, None, fx = x fy = y)
```

It's an cool trick to take only the rows/cols of the original img and use them for the new size:

```
rows, cols = img.shape[:2]
```

```
result = cv2. resize( img, (2*cols, 2*rows), interpolation= ---)
```

shrinking an img, looks good with: INTER_AREA
while enlargening it, with INTER_CUBIC or INTER_LINEAR