

In []:

```
import cv2
import numpy as np
```

In []:

```
src = np.zeros(shape=(512,512,3), dtype=np.uint8)
cv2.rectangle(src, (50, 100), (450, 400), (255, 255, 255), -1)
cv2.rectangle(src, (100, 150), (400, 350), (0, 0, 0), -1)
cv2.rectangle(src, (200, 200), (300, 300), (255, 255, 255), -1)
gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)

mode = cv2.RETR_EXTERNAL
#mode =cv2.RETR_LIST
method = cv2.CHAIN_APPROX_SIMPLE
#method =cv2.CHAIN_APPROX_NONE
contours, hierachy = cv2.findContours(gray, mode, method)
print('type(contours)=', type(contours))
print('type(contours[0])=', type(contours[0]))
print('len(contours)=', len(contours))
print('contours[0].shape=', contours[0].shape)
print('contours[0]=', contours[0])

cv2.drawContours(src, contours, -1, (255,0,0), 3) # 모든 윤곽선

for pt in contours[0][:]: # 윤곽선 좌표
    cv2.circle(src, (pt[0][0], pt[0][1]), 5, (0,0,255), -1)

cv2.imshow('src', src)
cv2.waitKey()
cv2.destroyAllWindows()
```

In []:

```
src = cv2.imread("./data/circles.jpg", cv2.IMREAD_COLOR)
gray = cv2.cvtColor(src, cv2.COLOR_RGB2GRAY)
ret, binary = cv2.threshold(gray, 127, 255, cv2.THRESH_BINARY_INV)

contours, hierachy = cv2.findContours(binary, cv2.RETR_CCOMP,
                                       cv2.CHAIN_APPROX_NONE)

for i in range(len(contours)):
    cv2.drawContours(src, [contours[i]], 0, (0, 0, 255), 2)
    cv2.putText(src, str(i), tuple(contours[i][0][0]), cv2.FONT_HERSHEY_COMPLEX,
                0.8, (0, 255, 0), 1)
    print(i, hierachy[0][i])
    cv2.imshow("src", src)
    cv2.waitKey()

cv2.destroyAllWindows()
```

In []:

