1. line, rectangle drawing

In [1]:

```
import cv2
import numpy as np

# White 배경 생성
img = np.zeros(shape=(512,512,3), dtype=np.uint8) + 255
#img = np.ones((512,512,3), np.uint8) * 255
#img = np.full((512,512,3), (255, 255, 255), dtype= np.uint8)
#img = np.zeros((512,512, 3), np.uint8) # Black 배경
pt1 = 100, 100
pt2 = 400, 400
cv2.rectangle(img, pt1, pt2, (0, 255, 0), 2)

cv2.line(img, (0, 0), (500, 0), (255, 0, 0), 5)
cv2.line(img, (0, 0), (0, 500), (0,0,255), 5)

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

2. cross point

In [9]:

```
img = np.zeros(shape=(512,512,3), dtype=np.uint8) + 255

x1, x2 = 100, 400
y1, y2 = 100, 400
cv2.rectangle(img, (x1, y1), (x2, y2), (0, 0, 255))

pt1 = 120, 50
pt2 = 300, 500
cv2.line(img, pt1, pt2, (255,0,0), 2)

imgRect = (x1, y1, x2-x1, y2-y1)
retval, rpt1, rpt2 = cv2.clipLine(imgRect, pt1, pt2)
if retval:
    cv2.circle(img, rpt1, radius=5, color=(0, 255, 0), thickness=-1)
    cv2.circle(img, rpt2, radius=5, color=(0, 255, 0), thickness=-1)

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

3. circle drawing

In [3]:

```
img = np.zeros(shape=(512,512,3), dtype=np.uint8) + 255
cx = img.shape[0]//2
cy = img.shape[1]//2

for r in range(200, 0, -100):
    cv2.circle(img, (cx, cy), r, color=(255, 0, 0))

cv2.circle(img, (cx, cy), radius=50, color=(0, 0, 255), thickness=-1)

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

4. ellipse drawing

In [4]:

```
img = np.zeros(shape=(512,512,3), dtype=np.uint8) + 255
ptCenter = img.shape[0]//2, img.shape[1]//2
size = 200,100

cv2.ellipse(img, ptCenter, size, 0, 0, 360, (255, 0, 0))
cv2.ellipse(img, ptCenter, size, 45, 0, 360, (0, 0, 255))

box = (ptCenter, size, 0)
cv2.ellipse(img, box,  (255, 0, 0), 5)

box = (ptCenter, size, 45)
cv2.ellipse(img, box,  (0, 0, 255), 5)

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

5. polyline drawing - 1

In [5]:

```
img = np.zeros(shape=(512,512,3), dtype=np.uint8) + 255

pts1 = np.array([[100, 100], [200, 100], [200, 200], [100, 200]])
pts2 = np.array([[300, 200], [400, 100], [400, 200]])

cv2.polylines(img, [pts1, pts2], isClosed=True, color=(255, 0, 0))

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

6. polyline drawing - 2

In [8]:

```
img = np.zeros(shape=(512,512,3), dtype=np.uint8) + 255

ptCenter = img.shape[0]//2, img.shape[1]//2
size = 200,100

cv2.ellipse(img, ptCenter, size, 0, 0, 360, (255, 0, 0))
pts1 = cv2.ellipse2Poly(ptCenter, size,  0, 0, 360, delta=30)

cv2.ellipse(img, ptCenter, size, 45, 0, 360, (255, 0, 0))
pts2 = cv2.ellipse2Poly(ptCenter, size, 45, 0, 360, delta=30)

cv2.polylines(img, [pts1, pts2], isClosed=True, color=(0, 0, 255))

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

7. rotated rectangle

In [9]:

```
img = np.zeros(shape=(512,512,3), dtype=np.uint8) + 255

x, y = 256, 256
size = 200

for angle in range(0, 90, 10):
    rect = ((256, 256), (size, size), angle)
    box = cv2.boxPoints(rect).astype(np.int32)
    r = np.random.randint(256)
    g = np.random.randint(256)
    b = np.random.randint(256)
    cv2.polylines(img, [box], True, (r, g, b), 2)

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

8. polygon drawing - 1

In [6]:

```
img = np.zeros(shape=(512,512,3), dtype=np.uint8) + 255

pts1 = np.array([[100, 100], [200, 100], [200, 200], [100, 200]])
pts2 = np.array([[300, 200], [400, 100], [400, 200]])

cv2.fillConvexPoly(img, pts1, color=(255, 0, 0))
cv2.fillPoly(img, [pts2], color=(0, 0, 255))
#cv2.fillPoly(img, [pts1, pts2], color=(0, 0, 255))

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

9. polygon drawing - 2

In [ ]:

```
b = (255, 0, 0)
g = (0, 255, 0)
r = (0, 0, 255)
w = (255, 255, 255)

img1 = np.zeros((400, 400, 3), np.uint8)
img2 = np.zeros((400, 400, 3), np.uint8)
pt1 = np.array([[100, 100], [270, 110], [300, 330], [170, 170], [150, 250]], np.int32)

cv2.fillConvexPoly(img1, pt1, g)
cv2.fillPoly(img2, [pt1], r)

cv2.imshow('convex', img1)
cv2.imshow('poly', img2)

cv2.waitKey()
cv2.destroyAllWindows()
```

10. textout

In [13]:

```
img = np.zeros(shape=(512,512,3), dtype=np.uint8) + 255
text = 'OpenCV Programming'
org = (50,100)
font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(img,text, org, font, 1, (255,0,0), 2)

size, baseLine = cv2.getTextSize(text, font, 1, 2)
#print('size=', size)
#print('baseLine=', baseLine)

cv2.rectangle(img, org, (org[0]+size[0], org[1]-size[1]), (0, 0, 255))
cv2.circle(img, org, 3, (0, 255,0), 2)

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

In [ ]:

```

```