**OpenCV 그리기 함수**

**2016301035 박상민**

* **교재에 있는 그리기 함수 예제들을 모아서 한 개의 이미지(blank\_500.jpg)에 그리시오.**
* **프로그램 설명**

**import numpy as np**

**import cv2**

**> numpy와 cv2 모듈을 포함하는 import 코드. 짧은 키워드로 참조하기 위해 as np 라고 사용한다.**

**img = np.full((500,500,3), 255, dtype=np.uint8)**

**cv2.imwrite('img/blank\_500.jpg', img)**

**> opneCV를 이용해서 그림을 그리기 위해서는, 그림판 역할을 하는 빈 NumPy 배열 이미지가 필요하다.**

**> 따라서 500\*500 크기의 컬러(3픽셀)를 저장할 빈 numpy 배열을 흰색(255)으로 채워서 저장한다.**

**> 해당 경로에 배열을 이미지로 저장한다. 앞으로 해당 이미지를 불러와서 그림 그릴 판으로 사용한다.**

**img = cv2.imread('./img/blank\_500.jpg')**

**cv2.putText(img, "Lines", (30,20), cv2.FONT\_ITALIC, 0.5, (0,0,0)) # 텍스트**

**> 이미 만들어진 흰색 blank 이미지 파일을 해당 경로에서 읽어온다.**

**> 해당 이미지 즉, 흰색 numpy 배열에 cv2 모듈의 putText 함수를 이용해 text를 출력한다.**

**# lines**

**cv2.line(img, (50,30), (150,30), (255,0,0))**

**cv2.line(img, (200,30), (300,30), (0,255,0))**

**cv2.line(img, (350,30), (450,30), (0,0,255))**

**> 시작과 끝 (x, y) 좌표, 색상값을 매개변수로 주어 line 함수로 좌표상에 표시한다.**

**> openCV에서는 색상의 순서가 RGB와 반대 순서이기 때문에, (Blue, Green, Red)를 표시한다.**

**> 따라서 위 코드는 각각 가로로 100 길이의 파랑, 초록, 빨간색 직선을 그리는 코드이다.**

**# rectanglues**

**cv2.putText(img, "Rectanglue", (30,60), cv2.FONT\_HERSHEY\_PLAIN, 1, (255,0,255)) # 텍스트**

**cv2.rectangle(img, (80,70), (120,110), (255,0,0)) # 채우기 없음**

**cv2.rectangle(img, (230,70), (270,110), (0,255,0), 10) # 선 두께 10**

**cv2.rectangle(img, (380,70), (420,110), (0,0,255), -1) # -1은 채워 그리기**

**> putText 함수로 출력할 위치, 폰트 종류, 색깔 등을 매개변수로 주어 text를 출력한다.**

**> rectangle 함수를 이용해 대각선 양쪽 끝 꼭지점 좌표와 색깔을 매개변수로 주어 사각형을 출력한다.**

**> 마지막 매개변수는 -1일 경우 채우기, 양수 값을 줄 경우 두께를 설정하는 속성이다.**

**# Polylines**

**cv2.putText(img, "Polyline", (30,150), cv2.FONT\_HERSHEY\_COMPLEX\_SMALL, 1, (0,210,210)) # 텍스트**

**pts1 = np.array([[25,160],[70,200],[50,200],[95,240]], dtype=np.int32) # 번개모양**

**pts2 = np.array([[165,160],[115,240],[215,240]], dtype=np.int32) # 삼각형**

**pts3 = np.array([[290,160],[240,240],[340,240]], dtype=np.int32) # 삼각형**

**pts4 = np.array([[405,150],[455,200],[430,250],[380,250],[355,200]], dtype=np.int32) # 오각형**

**> 다각형을 만들기 위해 numpy 배열로 그리려는 도형의 각 좌표를 만들어 저장한다.**

**> 위의 경우, pts1부터 번개모양, 삼각형, 삼각형, 오각형의 좌표이다.**

**cv2.polylines(img, [pts1], False, (255,0,0)) # 열린 파란색 번개모양 다각형 그리기**

**cv2.polylines(img, [pts2], False, (0,0,0), 10) # 열린 검정색 두께 10의 삼각형 그리기**

**cv2.polylines(img, [pts3], True, (0,0,255), 10) # 닫힌 빨간색 두께 10의 삼각형 그리기**

**cv2.polylines(img, [pts4], True, (0,0,0)) # 닫힌 오각형 그리기**

**> polylines 함수를 이용해서 빈 NumPy(img)에 그리려는 좌표를 담은 배열(pts)을 매개변수로 준다.**

**> 세 번째 매개변수는, True이면 첫 번째 좌표와 마지막 좌표를 이어 도형이 닫힌 상태가 되도록 그린다.**

**> 그 다음은 마찬가지 BGR 순서로 색상 값을 주고, 선의 두께 값을 주어 도형을 그린다.**

**# Circle**

**cv2.putText(img, "Circle", (30,280), cv2.FONT\_HERSHEY\_SCRIPT\_COMPLEX, 1, (255,0,0)) # 텍스트**

**cv2.circle(img, (50,340), 50, (255,0,0)) # 반지름이 50인 파란색 원 그리기**

**cv2.circle(img, (50,410), 40, (0,255,0), 5) # 두께 5, 반지름이 40인 초록색 원 그리기**

**cv2.circle(img, (50,450), 30, (0,0,255), -1) # 반지름이 30인 빨간색 채운 원 그리기**

**> 원을 그리기 위해서 circle 이라는 함수를 사용한다.**

**> 그려질 판 역할의 배열 이미지, 중심 좌표, 반지름의 길이, BGR 순서의 색상 값, 채우기 또는 선의 두께 값을 매개변수로 주어 원을 그린다.**

**cv2.ellipse(img, (150,340), (50,50), 0, 0, 180, (255,0,0)) # 아래 반원**

**cv2.ellipse(img, (200,340), (50,50), 0, 181, 360, (0,0,255)) # 위 반원**

**cv2.ellipse(img, (325,340), (75,50), 0, 0, 360, (0,255,0)) # 납작 타원**

**cv2.ellipse(img, (450,340), (50,75), 0, 0, 360, (255,0,255)) # 홀쭉 타원**

**cv2.ellipse(img, (190,445), (40,60), 50, 0, 360, (0,0,0)) # 회전 타원**

**cv2.ellipse(img, (310,445), (40,60), 45, 0, 180, (0,0,255)) # 회전 반원**

**cv2.ellipse(img, (350,445), (40,60), 45, 181, 360, (255,0,0)) # 회전 타원**

**> 타원이나 호, 반원 등을 그리기 위해서는 elipse 함수를 사용한다.**

**> 마찬가지로 이미지, 두 중점의 좌표를 튜플 형태로 주고, 그려질 도형이 회전한 각도, 그리기 시작할 좌표부터 마지막 좌표까지, BGR 순서의 색상 값을 주어 타원을 그린다.**

**> 그려지는 시작값은 우측 중앙부터 시계방향으로 그려진다.**

**cv2.imshow('draw', img)**

**cv2.waitKey(0)**

**cv2.destroyAllWindows()**

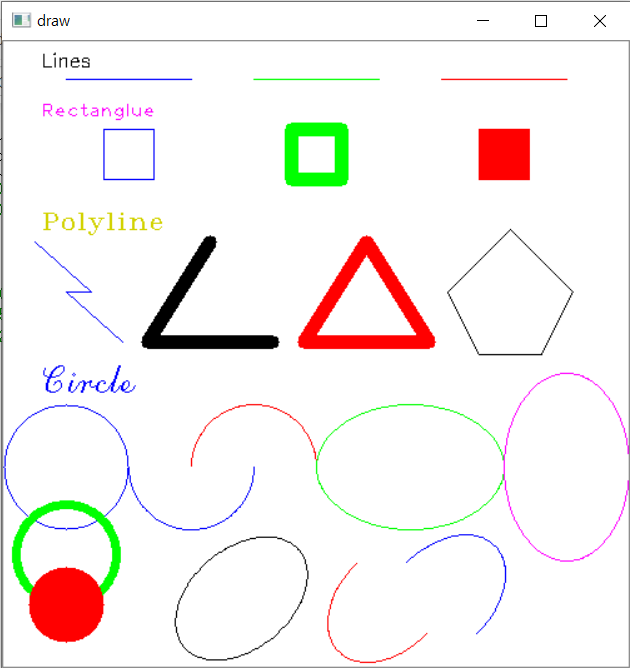
**> imshow 함수를 호출해서 해당 이미지를 그려주고, waitKey 함수를 호출해 키보드 입력을 기다린다.**

**> waitKey의 매개변수는 ms 단위로 입력을 기다릴 시간을 주고, 0일 경우 무한대로 기다린다.**

**> destroyAllWindows는, 떠있는 모든 창을 닫도록 하는 함수이다.**

**> 특정 창을 닫으려면 destroyWindow 함수로 닫으려는 창 이름을 값으로 주어 닫으면 된다.**

* **실행 결과**

****