연습문제 4장

1. 콜백 함수란 무엇인가?

|  |
| --- |
| 마우스, 트랙바, 키보드 등 특정한 이벤트를 처리하는 함수 |

2. 윈도우를 지정하는 cv2.namedWindow() 함수의 두 번째 인수(flags)에 대한 옵션은 여러 가지가 있다. 그 중에서 cv2.WINDOW\_NORMAL와 cv2.WINDOW\_AUTOSIZE 간의 차이를 설명하시오.

|  |
| --- |
| cv2.WINDOW\_NORMAL : 윈도우의 크기 변경을 자유롭게 할 수 있다. 이미지의 크기도 같이 변경된다.  Cv2.WINDOW\_AUTOSIZE : 이미지의 크기 변경 없이 윈도우의 크기만 변경된다. |

3. 타원을 그리는 cv2.ellipse() 함수의 인수를 자세히 설명하시오.

|  |
| --- |
| 그릴 대상, 타원의 중심, x축 반지름, y축 반지름, 각도, 호의 시작각도, 호의 종료 각도, 선 색상, 선 두께, 비트이동 |

4. OpenCV이 제공하는 마우스 이벤트와 트랙바 이벤트를 제어할 콜백 함수를 시스템에 등록하는 함수는 각각 무엇이며, 인수가 어떻게 구성되었는지 자세히 설명하시오.

|  |
| --- |
| Cv2.createTrackbar()   * trackbarname : 트랙바 이름 * winname : 트랙바의 부모 윈도우 이름 * value : 트랙바 슬라이더의 위치 값 * count : 트랙바 슬라이더의 최댓값 * onChange : 콜백 함수   Cv2.setMouseCallback()   * winname : 이벤트 발생을 확인할 윈도우 이름 * onMouse : 콜백 함수 * param : 이벤트 처리 함수로 전달할 추가적인 사용자 정의 인수 |

5. 다음 예시 코드의 실행 결과를 설명하시오.

|  |
| --- |
| import numpy as np import cv2  image = np.zeros((300, 400), np.uint8) image[:] = 100  title = 'Window' cv2.namedWindow(title, cv2.WINDOW\_NORMAL) cv2.moveWindow(title, 100, 200) cv2.imshow(title, image) cv2.waitKey(0) cv2.destroyAllWindows() |

|  |
| --- |
| 4행: 300 x 400 크기의 행렬 생성.  5행: 각 화소 값을 100으로 초기화.  8행: “Window”라는 이름의 창을 생성. 창의 크기를 자유롭게 변경 가능  9행: 창을 100, 200 위치로 이동.  10행: “Window”라는 이름의 창에 image을 띄움.  11행: 키가 입력될 때까지 대기  12행: 모든 창을 끔 |

6. 300행, 400열의 행렬을 회색 바탕색(100)으로 생성해서 500행, 600열의 윈도우에 표시하시오.

|  |
| --- |
| import numpy as np import cv2  image = np.zeros((300, 400), np.uint8) image[:] = 100  title = 'Window' cv2.namedWindow(title, cv2.WINDOW\_NORMAL) cv2.imshow(title, image) cv2.resizeWindow(title, 500, 600)  cv2.waitKey(0) cv2.destroyAllWindows() |

7. 다음 예시 코드는 컴파일 혹은 런타임 에러가 발생한다. 에러가 발생하는 부분을 수정하고 실행 결과를 적으시오.

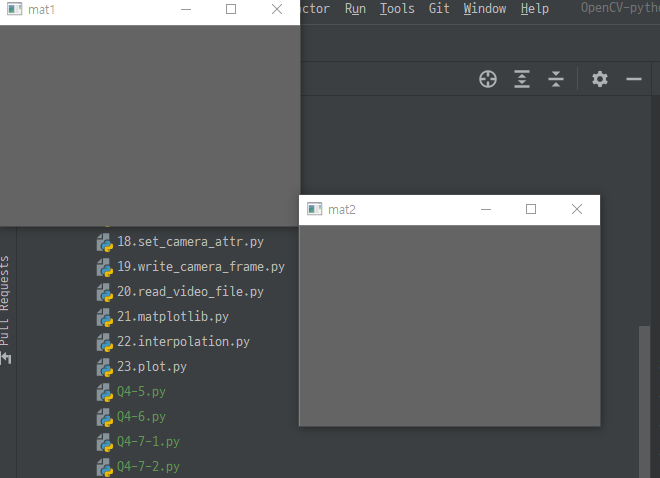
(1)

|  |
| --- |
| import numpy as np import cv2  image = np.zeros((300, 400, 3), np.uint8) image[:] = (255, 255, 255)  blue, green, red = (255, 0, 0), (0, 255, 0), (0, 0, 255) pt1, pt2 = (50, 130), (200, 300)  cv2.line(image, pt1, (100, 200), red) cv2.line(image, pt2, (100, 100), blue) cv2.rectangle(image, pt1, pt2, (255, 0, 255)) cv2.rectangle(image, pt1, pt2, (0, 0, 255))  title = "Line & Rectangle" cv2.namedWindow(title) cv2.imshow(title, image) cv2.waitKey(0) cv2.destroyAllWindows() |
| (1) line color에 대한 정보가 필요  (2) pt2의 좌표는 2차원에 행렬에 존재하므로 line의 양끝점은 길이가 2인 튜플 또는 리스트여야 한다. |

(2)

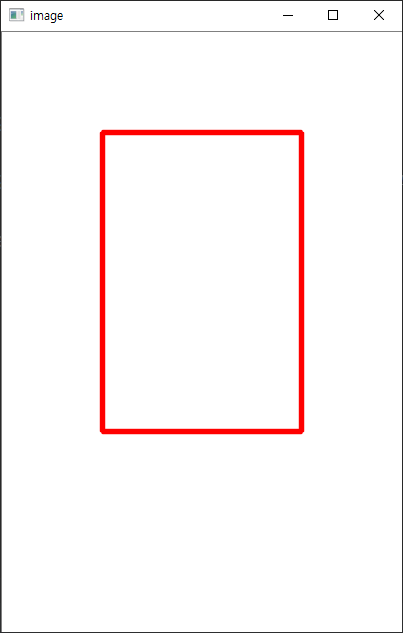
|  |
| --- |
| import numpy as np import cv2  def onMouse(event, x, y, flags, param):  global title  pt = (x, y)  if event == cv2.EVENT\_LBUTTONDOWN:  cv2.circle(image, pt, 5, 100, 1)  cv2.imshow(title, image)  elif event == cv2.EVENT\_RBUTTONDOWN:  **# pt + (30, 30) 과 pt, (30, 30)은 다름..!**  cv2.rectangle(image, pt + (30, 30), 100, 2)  cv2.imshow(title, image)    image = np.ones((300, 300), np.uint8) \* 255 title = "Draw Event" cv2.namedWindow(title) cv2.imshow(title, image) cv2.setMouseCallback(title, onMouse) cv2.waitKey(0) cv2.destroyAllWindows() |
| pt의 좌표가 초기화 되어있지 않았다.  cv2.rectangle 함수에서 시작, 종료 좌표에 대한 설정이 제대로 되어있지 않았다. |

8. 200행, 300열의 행렬 2개를 만들어서 다음과 같이 배치하시오.



|  |
| --- |
| import numpy as np import cv2  mat1 = np.full((200, 300), 100, np.uint8) mat2 = np.full((200, 300), 100, np.uint8)    cv2.imshow("mat1", mat1) cv2.imshow("mat2", mat2) cv2.moveWindow("mat1", 0, 0) cv2.moveWindow("mat2", 300, 200)  cv2.waitKey(0) cv2.destroyAllWindows() |

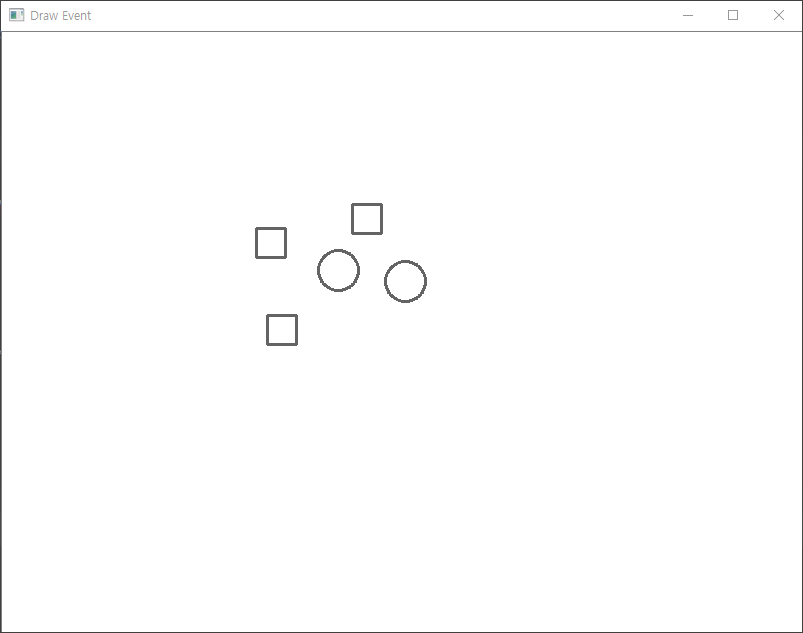
9. 600행, 400열의 윈도우를 만들고, 영상 안의 (100, 100) 좌표에 200x300 크기의 빨간 색 사각형을 그리시오.



|  |
| --- |
| import numpy as np import cv2  red = (0, 0, 255) image = np.ones((600, 400, 3), np.uint8) \* 255  cv2.rectangle(image, (100, 100) + (200, 300), red, 3, cv2.LINE\_AA)  cv2.imshow("image", image) cv2.waitKey(0) cv2.destroyAllWindows() |

10. 다음의 마우스 이벤트 제어 프로그램을 작성하시오.

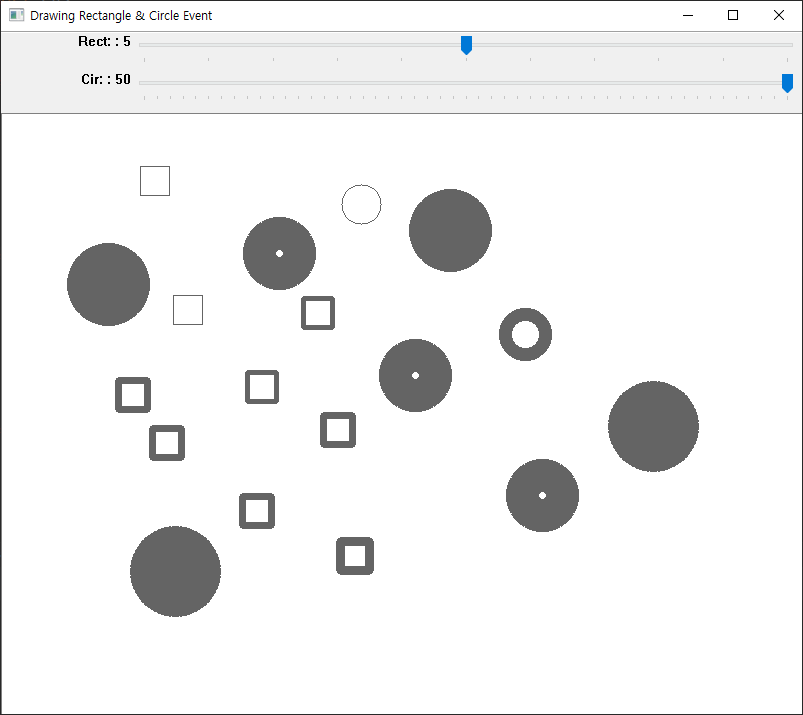
(1) 마우스 오른쪽 버튼 클릭 시 원(클릭 좌표에서 반지름 20화소)을 그린다.  
(2) 마우스 왼쪽 버튼 클릭 시 사각형(크기 30x30)을 그린다.



|  |
| --- |
| import numpy as np import cv2   def onMouse(event, x, y, flags, param):  global title  pt = (x, y)  if event == cv2.EVENT\_LBUTTONDOWN:  cv2.rectangle(image, pt + (30, 30), 100, 2)  cv2.imshow(title, image)  elif event == cv2.EVENT\_RBUTTONDOWN:  cv2.circle(image, pt, 20, 100, 2)  cv2.imshow(title, image)   image = np.ones((600, 800), np.uint8) \* 255 title = "Draw Event" cv2.namedWindow(title) cv2.imshow(title, image) cv2.setMouseCallback(title, onMouse) cv2.waitKey(0) cv2.destroyAllWindows() |

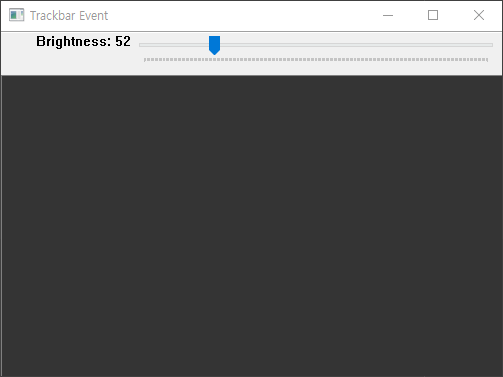
11. 10번 연습문제에서 다음을 추가하여 프로그램을 작성하시오.

(1) 트랙바를 추가해서 선의 굵기를 1~10픽셀로 조절한다.  
(2) 트랙바를 추가해서 원의 반지름을 1~50픽셀로 조절한다.



|  |
| --- |
| import numpy as np import cv2   def onChangeRec(value):  global image, title, rectangle\_thickness  print("Rectangle 굵기: {}".format(value))  rectangle\_thickness = value   def onChangeCir(value):  global image, title, circle\_thickness  print("Circle 굵기: {}".format(value))  circle\_thickness = value   def onMouse(event, x, y, flags, param):  global title, rectangle\_thickness, circle\_thickness  pt = (x, y)  if event == cv2.EVENT\_LBUTTONDOWN:  cv2.rectangle(image, pt + (30, 30), 100, rectangle\_thickness)  cv2.imshow(title, image)  elif event == cv2.EVENT\_RBUTTONDOWN:  cv2.circle(image, pt, 20, 100, circle\_thickness)  cv2.imshow(title, image)   rectangle\_thickness, circle\_thickness = 1, 1 image = np.ones((600, 800), np.uint8) \* 255 title = "Drawing Rectangle & Circle Event"  cv2.namedWindow(title) cv2.imshow(title, image)  cv2.setMouseCallback(title, onMouse) cv2.createTrackbar("Rect: ", title, 1, 10, onChangeRec) cv2.createTrackbar("Cir: ", title, 1, 50, onChangeCir)  cv2.waitKey(0) cv2.destroyAllWindows() |

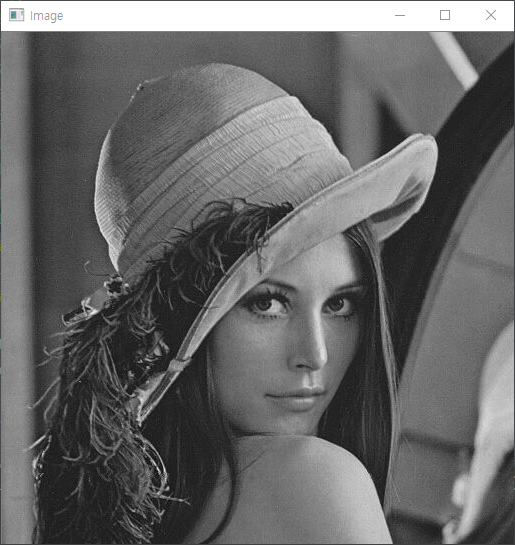
12. 예제\_4.2.3인 05.event\_trackbar.py에서 화살표 키로 트랙바를 이동하는 코드를 추가하시오.



|  |
| --- |
| import numpy as np import cv2  switch\_case ={  2424832: -1,  2555904: 1 }  # 트랙바 콜백 함수 def onChange(value):  global image, title   add\_value = value - int(image[0][0])  image = image + add\_value  cv2.setTrackbarPos("Brightness", title, image[0][0])  cv2.imshow(title, image)   image = np.zeros((300, 500), np.uint8)  title = "Trackbar Event" cv2.imshow(title, image)  cv2.createTrackbar("Brightness", title, image[0][0], 255, onChange)  while True:  key = cv2.waitKeyEx(100)  if key == 27:  break  try:  x = switch\_case[key]  if 0 <= image[0][0] + x <= 255:  onChange(image[0][0] + x)  except KeyError:  continue  cv2.waitKey(0) cv2.destroyAllWindows() |

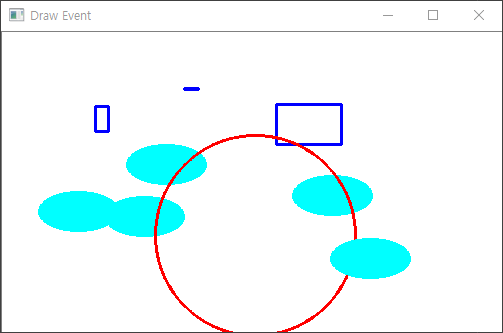
# 왼쪽으로 가면 수치는 밝기 값은 정상적으로 변하는데 띄워지는 이미지가 제대로 안 나옴..

13. 컬러 영상파일을 적재한 후의 행렬을 윈도우에 명암도 영상으로 표시하고, “test.jpg”와 “test.png” 파일로 각각 저장하시오. 이때 영상파일을 가장 좋은 화질로 압축하시오.



|  |
| --- |
| import cv2  image = cv2.imread("images/read\_color.jpg") if image is None:  raise Exception("File not found.")  gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)  cv2.imshow("Image", gray) cv2.imwrite("images/test.jpg", image, (cv2.IMWRITE\_JPEG\_QUALITY, 100)) cv2.imwrite("images/test.png", image, (cv2.IMWRITE\_PNG\_COMPRESSION, 9)) cv2.waitKey(0) |

14. 심화예제\_4.3.5인 11.event\_draw.py를 수정해서 마우스 중간버튼을 클릭하여 타원을 그리세요.

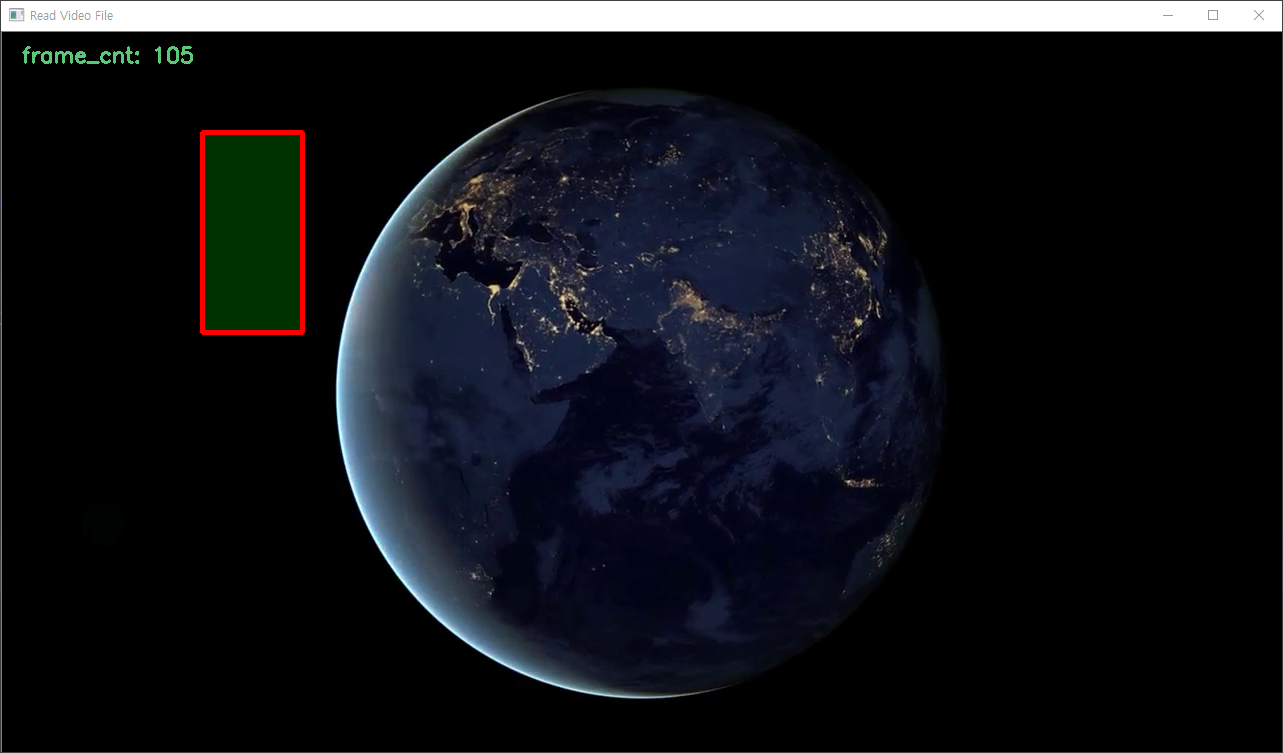


|  |
| --- |
| import numpy as np import cv2   def onMouse(event, x, y, flags, param):  global title, pt   if event == cv2.EVENT\_LBUTTONDOWN:  # 값이 음수일 때는 첫 번째 마우스 클릭을 의미  if pt[0] < 0:  pt = (x, y)  else:  # 파란색 사각형  cv2.rectangle(image, pt, (x, y), (255, 0, 0), 2)  cv2.imshow(title, image)  pt = (-1, -1) # 시작 좌표 초기화   elif event == cv2.EVENT\_RBUTTONDOWN:  if pt[0] < 0:  pt = (x, y)  else:  dx, dy = pt[0] - x, pt[1] - y  radius = int(np.sqrt(dx\*dx + dy\*dy)) # 반지름 계산  cv2.circle(image, pt, radius, (0, 0, 255), 2)  cv2.imshow(title, image)  pt = (-1, -1) # 시작 좌표 초기화   elif event == cv2.EVENT\_MBUTTONDOWN:  pt = (x, y)  cv2.ellipse(image, pt, (40, 20), 0, 0, 360, (255, 255, 0), -1)  cv2.imshow(title, image)  pt = (-1, -1)  image = np.full((300, 500, 3), (255, 255, 255), np.uint8) pt = (-1, -1) title = "Draw Event" cv2.imshow(title, image) cv2.setMouseCallback(title, onMouse) cv2.waitKey(0) |

15. 심화예제\_4.5.2인 18.set\_camera\_attr.py를 수정해서 트랙바로 카메라 영상의 밝기와 대비 변경할 수 있도록 수정하시오.

|  |
| --- |
| import cv2 from Common.utils import put\_string   # 줌 조절 콜백 함수 def zoom\_bar(value):  global capture  capture.set(cv2.CAP\_PROP\_ZOOM, value) # 줌 설정   # 초점 조철 콜백 함수 def focus\_bar(value):  global capture  capture.set(cv2.CAP\_PROP\_FOCUS, value)  def bright\_bar(value):  global capture  capture.set(cv2.CAP\_PROP\_BRIGHTNESS, value)  capture = cv2.VideoCapture(0) if not capture.isOpened():  raise Exception("카메라 연결 안됨")  capture.set(cv2.CAP\_PROP\_FRAME\_WIDTH, 400) # 카메라 프레임 너비 capture.set(cv2.CAP\_PROP\_FRAME\_HEIGHT, 300) # 카메라 프레임 높이 capture.set(cv2.CAP\_PROP\_AUTOFOCUS, 0) # 자동 초점 중지 capture.set(cv2.CAP\_PROP\_BRIGHTNESS, 100) # 프레임 밝기 초기화  title = "Change Camera Properties" cv2.namedWindow(title) cv2.createTrackbar('zoom', title, 0, 10, zoom\_bar) cv2.createTrackbar('focus', title, 0, 40, focus\_bar) cv2.createTrackbar('brightness', title, 0, 255, bright\_bar)  while True:  ret, frame = capture.read()  if not ret:  break  if cv2.waitKey(30) >= 0:  break   # 카메라 속성 가져오기  zoom = int(capture.get(cv2.CAP\_PROP\_ZOOM))  focus = int(capture.get(cv2.CAP\_PROP\_FOCUS))  bright = int(capture.get(cv2.CAP\_PROP\_BRIGHTNESS))   put\_string(frame, 'brightness : ', (10, 210), bright)  put\_string(frame, 'zoom : ', (10, 240), zoom) # 줌값 표시  put\_string(frame, 'focus : ', (10, 270), focus) # 초점 값 표시  cv2.imshow(title, frame)  capture.release() |

16. PC 카메라를 통해서 받아온 프레임에 대해 다음의 영상처리를 수행하고, 결과 영상을 윈도우에 표시하는 프로그램을 작성하시오.  
# camera가 없어서 임시로 샘플 영상 파일을 활용

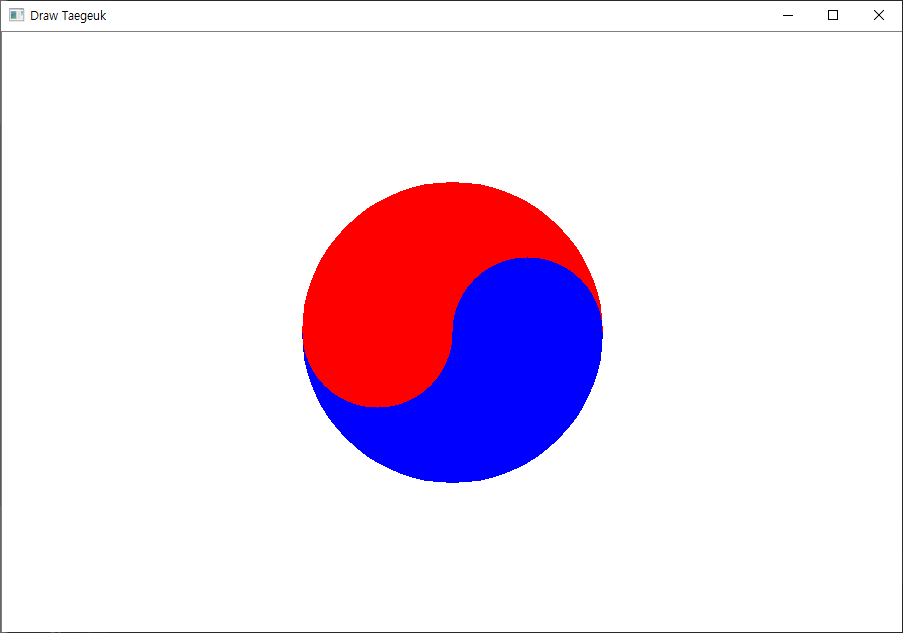


|  |
| --- |
| import cv2 from Common.utils import put\_string  capture = cv2.VideoCapture("images/video\_file.avi") if not capture.isOpened():  raise Exception("동영상 파일 개방 안됨")  frame\_rate = capture.get(cv2.CAP\_PROP\_FPS) delay = int(1000 / frame\_rate) frame\_cnt = 0 # 현재 프레임 번호  while True:  ret, frame = capture.read()  if not ret or cv2.waitKey(delay) >= 0:  break   # 컬러 영상 채널 분리 (BGR)  blue, green, red = cv2.split(frame)  frame\_cnt += 1   cv2.add(green[100:300, 200:300], 50, green[100:300, 200:300])   # 단일 채널 영상 합성  frame = cv2.merge([blue, green, red])  put\_string(frame, 'frame\_cnt: ', (20, 30), frame\_cnt)  cv2.rectangle(frame, (200, 100), (300, 300), (0, 0, 255), 3)  cv2.imshow("Read Video File", frame)  capture.release() |

17. PC 카메라를 통해서 받아온 프레임을 좌우로 뒤집어서 “flip\_test.avi” 이름의 동영상 파일로 저장하는 프로그램을 작성하시오.

|  |
| --- |
| import cv2  capture = cv2.VideoCapture(0) if not capture.isOpened():  raise Exception("카메라 연결 안됨")  fps = 15 # 초당 프레임 수 delay = round(1000/fps) # 프레임 간 지연 시간 size = (640, 480) # 동영상 파일 해상도 fourcc = cv2.VideoWriter\_fourcc(\*'DIVX') # 압축 코덱 설정  # 카메라 속성 실행창에 출력 print("width x height: ", size) print("VideoWriterfourcc: %s" % fourcc) print("delay: %2d ms" % delay) print("fps: %.2f" % fps)  # 카메라 속성 지정 capture.set(cv2.CAP\_PROP\_ZOOM, 1) capture.set(cv2.CAP\_PROP\_FOCUS, 0) capture.set(cv2.CAP\_PROP\_FRAME\_WIDTH, size[0]) # 해상도 설정 capture.set(cv2.CAP\_PROP\_FRAME\_HEIGHT, size[1])   # 동영상 파일 개방 및 코덱, 해상도 설정 writer = cv2.VideoWriter("images/flip\_test.avi", fourcc, fps, size) if not writer.isOpened():  raise Exception("동영상 파일 개방 안됨")  while True:  ret, frame = capture.read()  if not ret:  break  if cv2.waitKey(delay) >= 0:  break  # frame = frame[:, ::-1]  frame = cv2.flip(frame, 1)  # frame을 동영상으로 저장  writer.write(frame)  cv2.imshow("View Frame from Camera", frame)  writer.release() capture.release() |

18. 다음과 같이 태극 문양을 그리는 프로그램을 작성하시오.



|  |
| --- |
| import numpy as np import cv2  red, blue, white = (0, 0, 255), (255, 0, 0), (255 ,255, 255) image = np.full((600, 900, 3), white, np.uint8)  center = (450, 300)  # 타원의 중점 표시 cv2.ellipse(image, center, (150, 150), 0, 0, 180, blue, cv2.FILLED) cv2.ellipse(image, center, (150, 150), 0, 180, 360, red, cv2.FILLED) cv2.circle(image, (375, 300), 75, (0, 0, 255), cv2.FILLED) cv2.circle(image, (525, 300), 75, (255, 0, 0), cv2.FILLED)  cv2.imshow("Draw Taegeuk", image) cv2.waitKey() |