

OpenCV/Python 설치 및 기본 사용법

OpenCV 개요

▶ OpenCV (Open source Computer Vision Library)

▶ 영상처리, 컴퓨터 비전, 기계학습 관련 라이브러리

- ▶ 2500개가 넘는 알고리즘으로 구성

- ▶ OpenCV 1.0 ~1.1 version은 C언어 개발

- ▶ OpenCV 2.0 ~이후 version은 C++ 언어로 갱신

- ▶ 2006년도 OpenCV 1.0, 2009년도 OpenCV 2.0, 2015년도 3.0버전 이후 현재 version 4.10.0까지 발표

- ▶ 여기서는 OpenCV 4.10.0 버전 사용

- ▶ 사용가능 Platform: Windows, Linux, Mac, Android, iOS

- ▶ 인터페이스 가능 언어: C, C++, Java, Python, Matlab, Javascript

▶ www.opencv.org

- ▶ Library/Release, Documentation(docs.opencv.org/4.10.0/), Tutorials, OpenCV-Python Tutorials, Main modules, Extra Modules

▶ 대한민국 OpenCV 사이트(<https://cafe.naver.com/opencv>)

OpenCV 주요 라이브러리 기능

- ▶ **opencv_core**
 - ▶ 기본자료구조와 기본 연산함수(행렬연산, DFT, 그리기 등)
- ▶ **opencv_imgproc**
 - ▶ 필터링, 히스토그램 처리, 컬러변환 등 기본 영상처리 알고리즘
- ▶ **opencv_highgui**
 - ▶ 윈도우 관련 GUI 기능, 영상 입출력
- ▶ **opencv_ml**
 - ▶ 기계학습 알고리즘(Bayes 분류기, SVM, 결정 트리, EM 등)
- ▶ **opencv_features2d**
 - ▶ 특징점 검출기(SIFT, FAST, ORB, GFTT, HARRIS 등) 및 특징점 매칭 함수
- ▶ **opencv_video**
 - ▶ 비디오 입출력, 움직임 검출, 특징 추적, 물체 추적, 전경 추출 함수 및 클래스



OpenCV 주요 라이브러리 기능

- ▶ `opencv_objdetect`
 - ▶ 영상에서 물체 검출 함수(얼굴 검출, 사람 검출)
- ▶ `opencv_calib3d`
 - ▶ 카메라 보정, 스테레오 영상 처리, 3D 영상 처리
- ▶ `opencv_flann`
 - ▶ 공간에서 이웃을 빨리 찾는 알고리즘(Fast Library for Approximate Nearest Neighbor)
- ▶ `opencv_stitching`
 - ▶ 여러 장의 영상을 이용해서 파노라마 영상을 생성
- ▶ `opencv_dnn`
 - ▶ 딥러닝 네트워크인 AlexNet, GoogLeNet, VGG, ResNet, MobileNet 등의 네트워크가 동작
 - ▶ 객체 검출과 관련해서는 VGG-SSD, MobileNet-SSD, Faster-RCNN, R-FCN, Mask-RCNN, EAST, YOLOv2, tiny YOLO, YOLOv3 등의 모델을 사용
- ▶ `opencv_contrib` ← 기본으로 설치되지 않음
 - ▶ 특허가 있는 알고리즘(xfeatures2d::SURF 등)
 - ▶ 라이브러리 전체에 통합되지 않은 새로운 기능 포함(ximgproc, text, face, tracker 등)
 - ▶ 주로 커뮤니티에서 기여한 기능들로 구성되어 있으며, 아직 안정적인 API가 없거나 충분히 테스트되지 않은 경우

OpenCV 주요 라이브러리 파일

- ▶ **opencv_core** 각 라이브러리마다 1, 2, 3 이 있음
 - 1 헤더파일: core.hpp
 - 2 LIB static 라이브러리
 - ▶ opencv_core460.lib, opencv_core460d.lib
 - 3 DLL dynamic 라이브러리
 - ▶ opencv_core460.dll, opencv_core460d.dll
 - ▶ d는 Debug 정보가 포함되었음을 의미(Debug 모드용)
- ▶ 다른 라이브러리도 동일한 형태임
- ▶ OpenCV 3.0버전부터는 통합라이브러리 사용
 - ▶ opencv_world460d.lib(Debug 모드)
 - ▶ opencv_world460.lib(Release 모드)
 - ▶ 통합헤더 파일 사용: opencv.hpp

OpenCV Documentation

ENHANCED BY Google



OpenCV

Open Source Computer Vision

4.8.0



[Main Page](#)

[Related Pages](#)

[Modules](#)

[Namespaces ▾](#)

[Classes ▾](#)

[Files ▾](#)

[Examples](#)

[Java documentation](#)

Search

OpenCV modules

- [Introduction](#)
- [OpenCV Tutorials](#)
- [OpenCV-Python Tutorials](#)
- [OpenCV.js Tutorials](#)
- [Tutorials for contrib modules](#)
- [Frequently Asked Questions](#)
- [Bibliography](#)
- Main modules:
 - core. [Core functionality](#)
 - imgproc. [Image Processing](#)
 - imgcodecs. [Image file reading and writing](#)
 - videoio. [Video I/O](#)
 - highgui. [High-level GUI](#)
 - video. [Video Analysis](#)
 - calib3d. [Camera Calibration and 3D Reconstruction](#)
 - features2d. [2D Features Framework](#)
 - objdetect. [Object Detection](#)
 - dnn. [Deep Neural Network module](#)
 - ml. [Machine Learning](#)
 - flann. [Clustering and Search in Multi-Dimensional Spaces](#)
 - photo. [Computational Photography](#)
 - stitching. [Images stitching](#)
 - gapi. [Graph API](#)

OpenCV 함수 선언

The screenshot shows the OpenCV documentation for the `Canny()` function. Handwritten blue annotations are present:

- A blue arrow points from the text "함수명" (function name) to the `Canny()` title.
- A blue arrow points from the text "C++ →" to the C++ signature: `void cv::Canny (InputArray image, OutputArray edges, double threshold1, double threshold2, int apertureSize = 3, bool L2gradient = false)`. A note next to it says "output을 매개변수 형태로 리턴" (return output as a parameter).
- A blue arrow points from the text "Python →" to the Python signature: `cv.Canny(image, threshold1, threshold2[, edges[, apertureSize[, L2gradient]]]) -> edges` and `cv.Canny(dx, dy, threshold1, threshold2[, edges[, L2gradient]]) -> edges`. A note next to it says "output을 object 형태로 리턴" (return output as an object).

The documentation also includes the header `#include <opencv2/imgproc.hpp>`, a description of the function, and a list of parameters:

- image**: 8-bit input image.
- edges**: output edge map; single channels 8-bit image, which has the same size as image.
- threshold1**: first threshold for the hysteresis procedure.
- threshold2**: second threshold for the hysteresis procedure.
- apertureSize**: aperture size for the Sobel operator.
- L2gradient**: a flag, indicating whether a more accurate L_2 norm $= \sqrt{(dI/dx)^2 + (dI/dy)^2}$ should be used to calculate the image magnitude (`L2gradient=true`), or whether the default L_1 norm $= |dI/dx| + |dI/dy|$ is enough (`L2gradient=false`).

그림 2-7 OpenCV 함수 선언-Canny 함수 사례

파이썬 설치하기

■ 세 가지 소프트웨어 설치 필요함

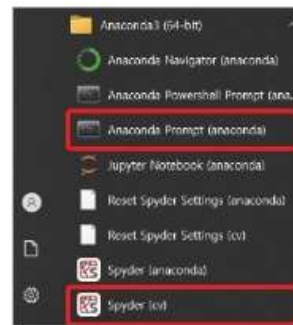
- 파이썬
- 통합 개발 환경(IDE)
- 라이브러리

■ 보통 아나콘다를 활용하여 설치

- 세 가지 소프트웨어를 한꺼번에 설치해줌
- 가상 환경 virtual environment 제공 (base)
 - 버전 충돌을 피하기 위해 새로운 프로젝트는 새로운 가상 환경에서 제작
- 이 책은 cv라는 가상 환경에서 프로그래밍 수행



(a) 아나콘다 다운로드 화면



(b) 윈도우 [시작] 버튼을 클릭해 아나콘다 설치 확인

그림 2-3 아나콘다와 스파이더 설치

아나콘다 설치

- ▶ 데이터 분석 및 기계학습을 위한 패키지들이 기본적으로 포함
- ▶ 개발도구 제공: 주피터 노트북, 스파이더
- ▶ 설치파일 다운로드
 - ▶ <https://www.anaconda.com/download>
 - ▶ PC 운영체제(설정->시스템->정보)에 맞게 다운로드(64-bit)
 - ▶ Anaconda3-2024.06-1-Windows-x86_64.exe // Python 3.12
 - ▶ 설치 폴더 적절하게 변경 후에 나머지는 디폴트로 진행

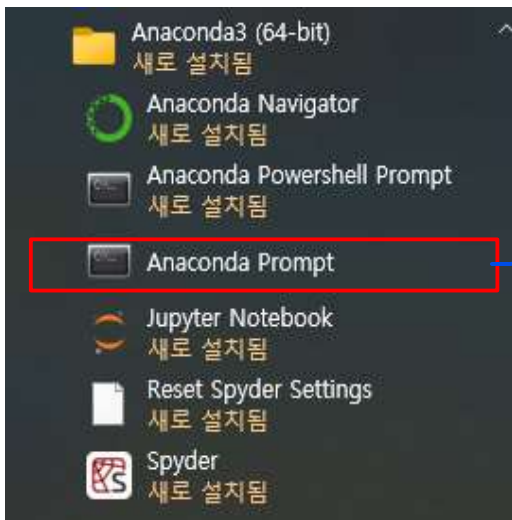


OpenCV와 Tensorflow 설치

▶ default 상태로 next 클릭(설치 폴더는 원하는 위치로 변경)

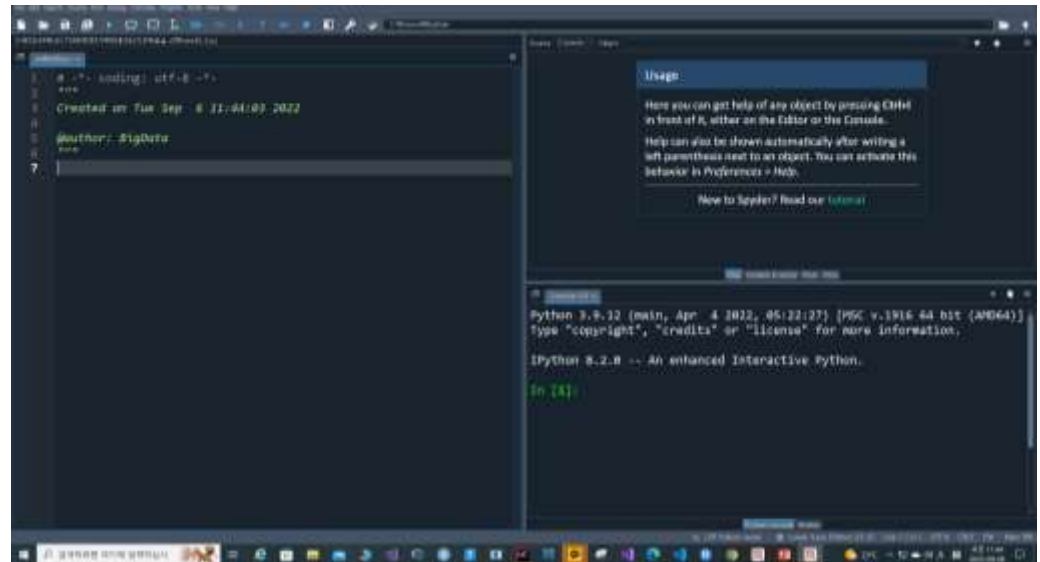
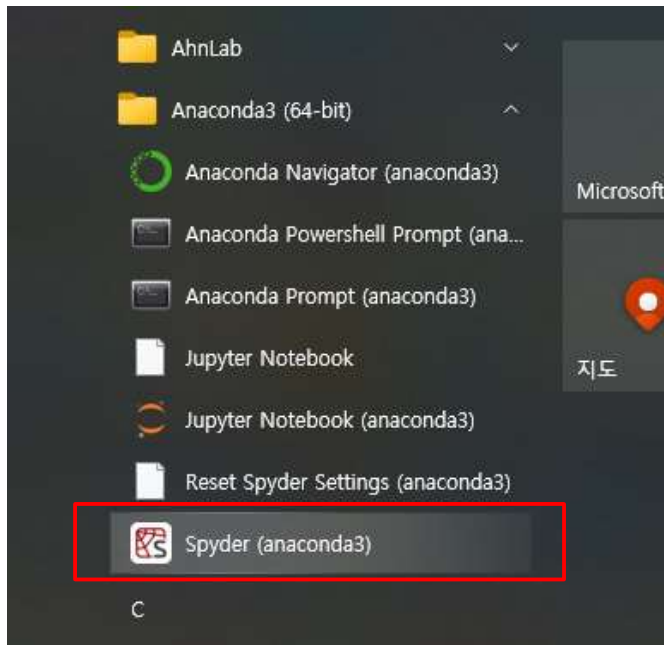
▶ 설치가 끝난 후 시작메뉴 창에서 Anaconda Prompt 실행

가상환경으로
접속



- (base) C:/users> python --version
 - #아나콘다에 포함되어 있는 파이썬의 버전 확인
- (base) C:/users> conda --version #아나콘다 버전 확인
- **OpenCV 설치**
- (base) C:/users> pip install opencv-python
 - (contrib 설치: pip install opencv-contrib-python)
- **Tensorflow 설치**
- (base) C:/users> pip install tensorflow
- (base) C:/users> python
- >>> import cv2
- >>> cv2.__version__ *cv2 에 있는 version 함수 실행*
- >>> import tensorflow as **tf**
- >>> **tf.**__version__
- >>> exit()

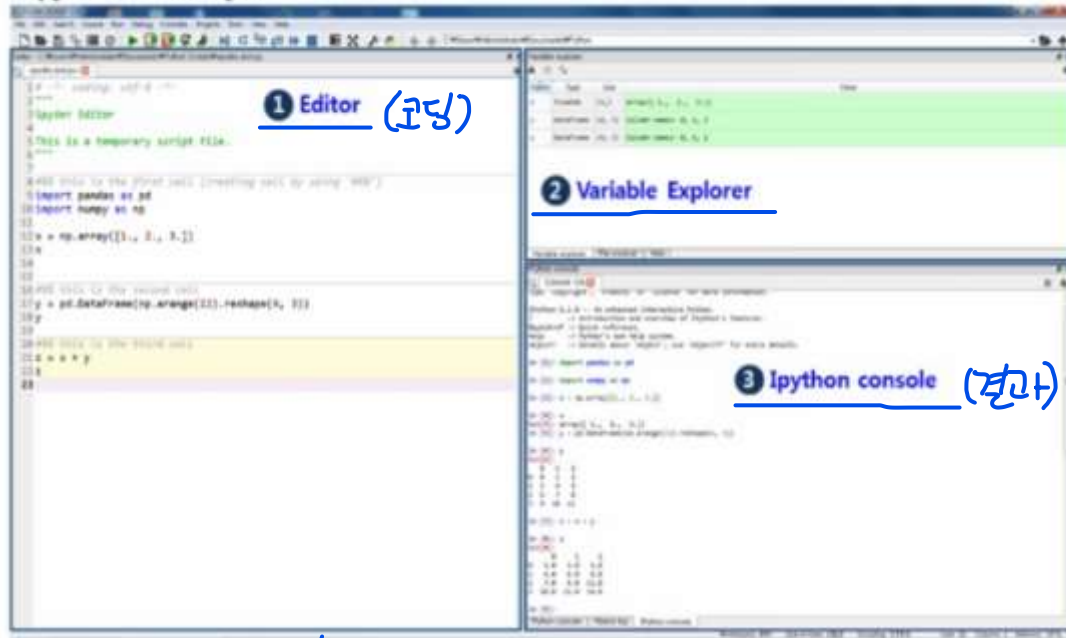
spyder(IDE) 실행



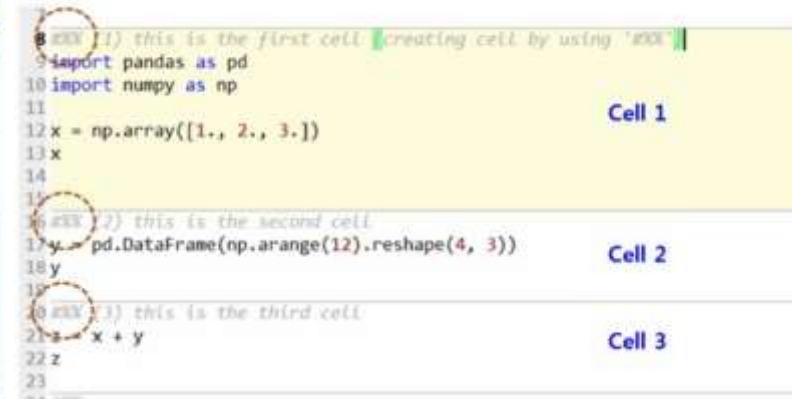
(base) C:/users> conda list #현재 base 가상환경에 설치된 프로그램과 라이브러리 목록을 볼 수 있다.

스파이더 화면

Spyder default layout

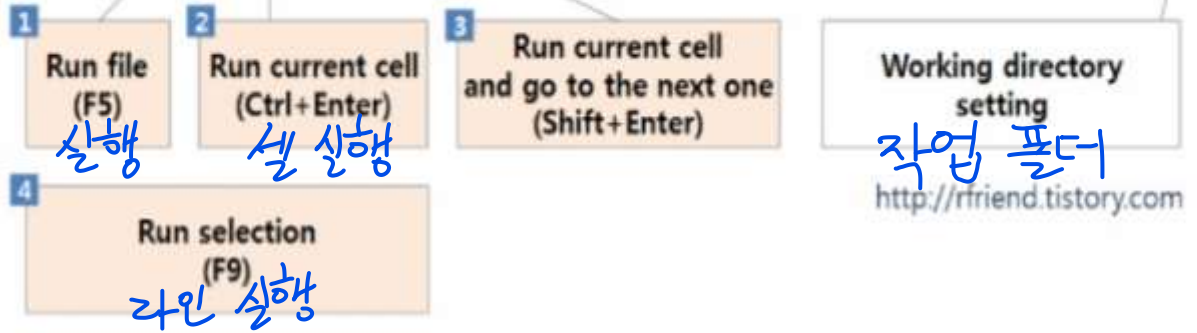
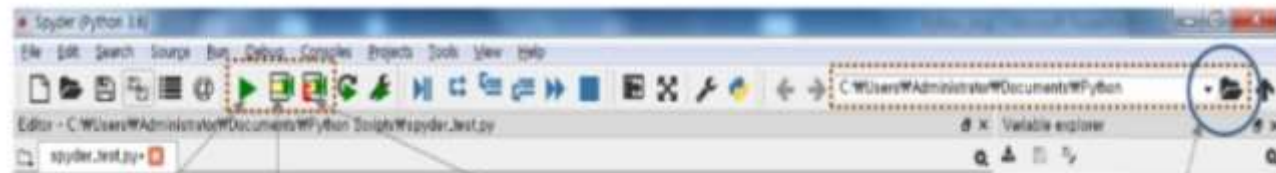


Spyder : Cell (###)



프로그램을 분리할 수 있다

여기서 cell 은 '#%%' 로 구분이 되며, 위/아래로 '선(line)'으로 구분이 된 덩어리를 말합니다.



<http://rfriend.tistory.com>

객체 다루기

■ 파이썬은 객체지향 언어

프로그램 2-1

numpy.ndarray 클래스 형의 객체를 만들고 멤버 함수 적용하기

```
01 import numpy as np
02
03 a=np.array([4,5,0,1,2,3,6,7,8,9,10,11])
04 print(a) ①
05 print(type(a)) ②
06 print(a.shape) ③
07 a.sort()
08 print(a) ④
09
10 b=np.array([-4.3,-2.3,12.9,8.99,10.1,-1.2])
11 b.sort()
12 print(b) ⑤
13
14 c=np.array(['one','two','three','four','five','six','seven'])
15 c.sort()
16 print(c) ⑥
```

```
[ 4  5  0  1  2  3  6  7  8  9 10 11] ①
<class 'numpy.ndarray'> ②
(12,) ③
[ 0  1  2  3  4  5  6  7  8  9 10 11] ④
[-4.3 -2.3 -1.2  8.99 10.1 12.9 ] ⑤
['five' 'four' 'one' 'seven' 'six' 'three' 'two'] ⑥
```

객체 다루기

■ 객체지향 특성과 강점

- 객체 내부에 있는 함수를 멤버함수 혹은 메소드, 변수를 멤버 변수라고 한다.
- 객체는 능동적: 자신이 소유한 멤버 함수를 능동적으로 호출
- 필요한 만큼 얼마든지 찍어낼 수 있음



(a) a 객체



(b) b 객체



(c) c 객체

그림 2-8 np.ndarray 클래스로 찍어낸 객체들

객체 확인하기 - **type**과 **dir** 내장 함수

- 객체의 클래스를 알려주는 **type**과 사용 가능한 멤버 함수를 알려주는 **dir**

```
In [1]: type(a)
numpy.ndarray
In [2]: dir(a)
['T', '__abs__', '__add__', '__and__', ..., 'all', 'any', ..., 'max', ..., 'shape', ..., 'sort', ...]
```

- 함수가 도움 말은 **help** 혹은 명령어에 커저 위치 한 후 **ctrl + i**

```
In [3]: help(a.sort)
Help on built-in function sort:

sort(...) method of numpy.ndarray instance
    a.sort(axis=-1, kind=None, order=None)

    Sort an array in-place. Refer to `numpy.sort` for full documentation.

    Parameters
    -----
    axis : int, optional
        Axis along which to sort. Default is -1, which means sort along the last
        axis.
```

2.4 [프로그래밍 예제 1] 영상을 읽고 표시하기

■ 처음 해보는 OpenCV 프로그래밍

프로그램 2-2

영상 파일을 읽고 윈도우에 디스플레이하기

```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('soccer.jpg')    # 영상 읽기
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 cv.imshow('Image Display',img) # 윈도우에 영상 표시
10
11 cv.waitKey()
12 cv.destroyAllWindows()
```

None: <class: NoneType>

값이 존재하지 않는 경우(값을 알 수 없는 경우)

값이 "없음"일 경우

값이 정의되지 않은 경우

값을 초기화하여 사용하는 경우

"None" 객체는 "is None"으로 체크합니다.

자체 "None"은 True 또는 False 체크 시, "False"로 판정!



OpenCV에서 영상은 `numpy.ndarray` 클래스 형의 객체

■ `numpy`는 다차원 배열을 위한 사실상 표준 모듈

- 이런 이유로 OpenCV는 영상을 `numpy.ndarray`로 표현
- OpenCV가 다루는 영상은 `numpy`가 제공하는 다양한 기능(함수)을 사용할 수 있음

```
In [1]: type(img)
        numpy.ndarray
In [2]: img.shape
        (948,1434,3)
```

■ 영상에서의 좌표: (y, x) 혹은 (r, c)

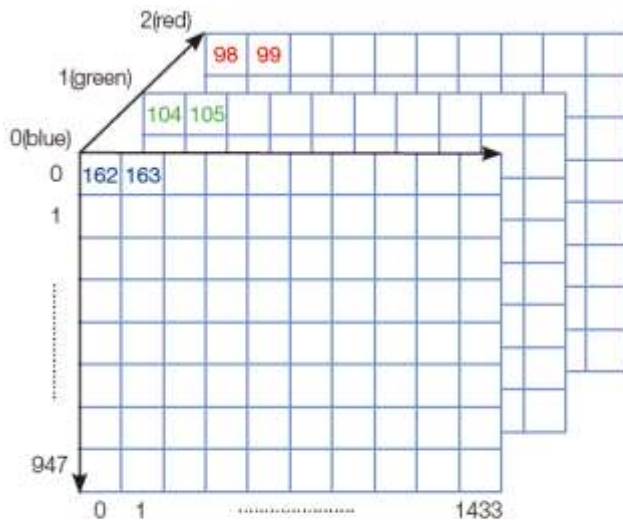


OpenCV에서 영상은 `numpy.ndarray` 클래스 형의 객체

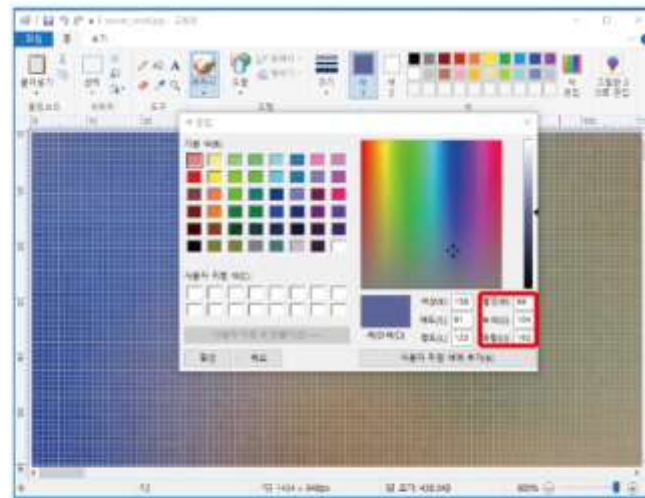
■ 영상의 표현

- 화소(pixel: picture element)의 위치 (r,c) 또는 (y,x)
- 화소값 조사

```
In [3]: print(img[0,0,0], img[0,0,1], img[0,0,2])    # (0,0) 화소 조사  
162 104 98  
In [4]: print(img[0,1,0], img[0,1,1], img[0,1,2])    # (0,1) 화소 조사  
163 105 99
```



(a) 프로그램으로 조사



(b) 그림판으로 조사



그림 2-9 img 객체가 표현하는 영상의 구조와 내용

2.5 [프로그래밍 예제2] 영상 형태 변환하고 크기 축소하기

프로그램 2-3

영상을 명암 영상으로 변환하고 반으로 축소하기

```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('soccer.jpg')
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 gray=cv.cvtColor(img,cv.COLOR_BGR2GRAY)          # BGR 컬러 영상을 명암 영상으로 변환
10 gray_small=cv.resize(gray,dsize=(0,0),fx=0.5,fy=0.5) # 반으로 축소
11
12 cv.imwrite('soccer_gray.jpg',gray)                # 영상을 파일에 저장
13 cv.imwrite('soccer_gray_small.jpg',gray_small)
14
15 cv.imshow('Color image',img)
16 cv.imshow('Gray image',gray)
17 cv.imshow('Gray image small',gray_small)
18
19 cv.waitKey()
20 cv.destroyAllWindows()
```

결과 크기 이미지 dsize를 (0, 0)으로 주면, 결과 이미지 크기를 설정하지 않은 경우이며, 이 경우는 **fx와 fy를 사용하여 이미지의 비율을 조절** 함.



2.5 [프로그래밍 예제2] 영상 형태 변환하고 크기 축소하기

■ cvtColor 함수가 컬러 영상을 명암 영상으로 바꾸는 방법

$$I = \text{round}(0.299 \times R + 0.587 \times G + 0.114 \times B) \quad (2.1)$$

(참조: https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html)

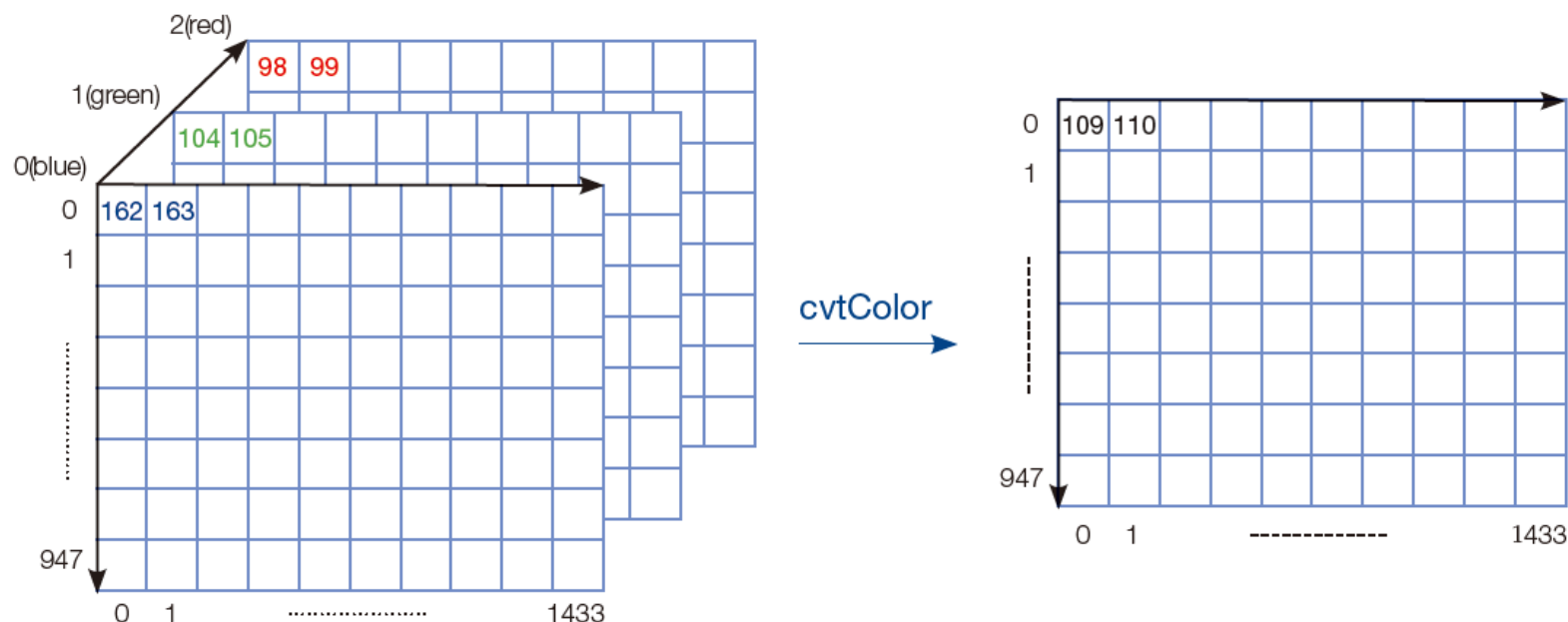


그림 2-10 BGR 컬러 영상을 명암 영상으로 변환

2.6 [프로그래밍 예제 3] 웹 캠에서 비디오 읽기

프로그램 2-4

웹 캠으로 비디오 획득하기

```
01 import cv2 as cv
02 import sys
03
04 cap=cv.VideoCapture(0,cv.CAP_DSHOW) # 카메라와 연결 시도
05
06 if not cap.isOpened():
07     sys.exit('카메라 연결 실패')
08
09 while True:
10     ret,frame=cap.read()           # 비디오를 구성하는 프레임 획득
11
12     if not ret:
13         print('프레임 획득에 실패하여 루프를 나갑니다.')
14         break
15
16     cv.imshow('Video display',frame)
17
18     key=cv.waitKey(1)              # 1밀리초 동안 키보드 입력 기다림
19     if key==ord('q'):              # 'q' 키가 들어오면 루프를 빠져나감
20         break
21
22 cap.release()                    # 카메라와 연결을 끊음
23 cv.destroyAllWindows()
```

`cap=cv.VideoCapture('tunnel.mpeg')`

`ord(문자)`: 하나의 문자를 인자로 받고 해당 문자에 해당하는 유니코드 정수를 반환

카메라와 동영상 열기

■ cv2.VideoCapture 클래스로 처리

- open()으로 카메라나 동영상을 열고 read()로 프레임을 가져옴

■ 카메라 열기

- cv2.VideoCapture(index, apiPreference=None) -> retval
- index : camera_id, 시스템 기본 카메라를 기본 방법으로 열려면 index에 0을 전달. 장치 관리자에 등록되어 있는 카메라 순서대로 인덱스가 설정되어 있음
- apiPreference : 선호하는 카메라 처리 방법을 지정합니다.
- retval : cv2.VideoCapture 객체를 반환합니다.

■ 동영상 열기

- cv2.VideoCapture(filename, apiPreference=None) -> retval
- filename : 비디오 파일 이름, 정지 영상 시퀀스, 비디오 스트림 URL 등,

■ 프레임 받아오기 - cv2.VideoCapture.read()

- cv2.VideoCapture.read(image=None) -> retval, image
- retval : 성공하면 True, 실패하면 False.
- image : 현재 프레임 (numpy.ndarray)



2.6 [프로그래밍 예제 3] 웹 캠에서 비디오 읽기

■ 비디오에서 영상 수집하기(numpy의 hstack 함수 사용)

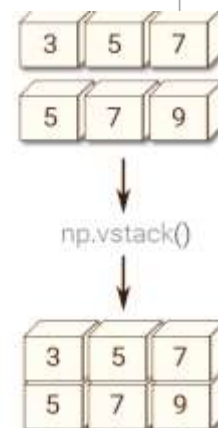
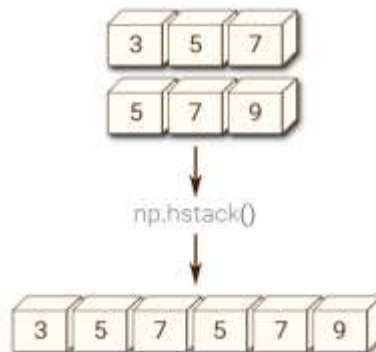
프로그램 2-5

비디오에서 수집한 영상을 이어 붙이기

```
01 import cv2 as cv
02 import numpy as np
03 import sys
04
05 cap=cv.VideoCapture(0,cv.CAP_DSHOW)      # 카메라와 연결 시도
06
07 if not cap.isOpened():
08     sys.exit('카메라 연결 실패')
09
10 frames=[]
11 while True:
12     ret,frame=cap.read()                  # 비디오를 구성하는 프레임 획득
13
14     if not ret:
15         print('프레임 획득에 실패하여 루프를 나갑니다.')
16         break
17
```

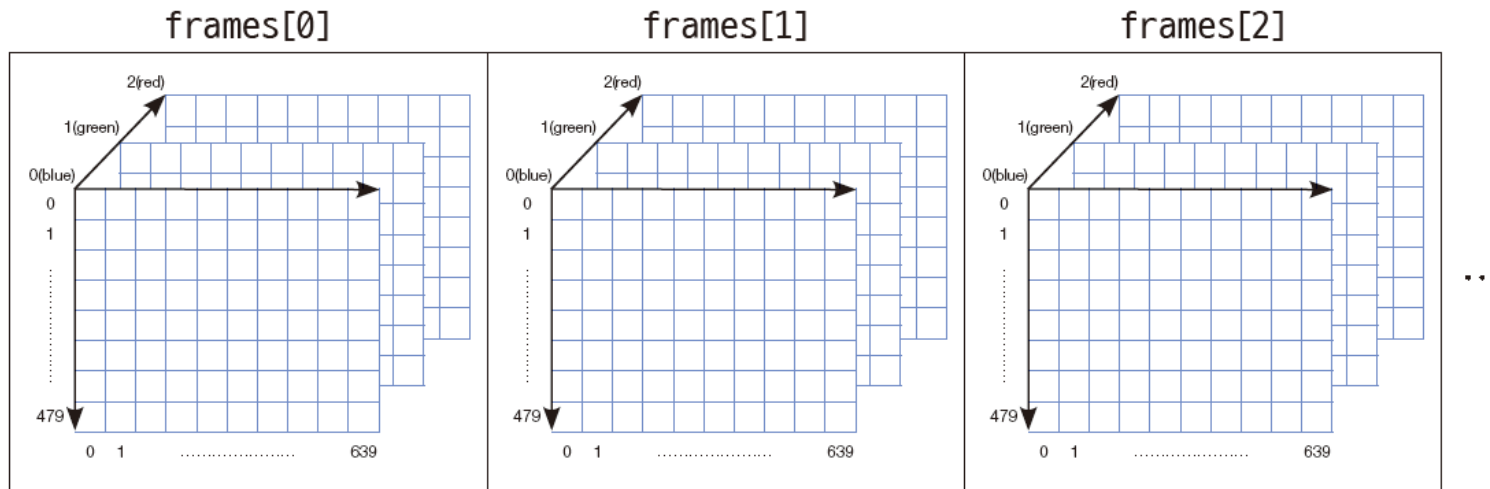
2.6 [프로그래밍 예제 3] 웹 캠에서 비디오 읽기

```
18 cv.imshow('Video display',frame)
19
20 key=cv.waitKey(1)           # 1밀리초 동안 키보드 입력 기다림
21 if key==ord('c'):          # 'c' 키가 들어오면 프레임을 리스트에 추가
22     frames.append(frame)
23 elif key==ord('q'):         # 'q' 키가 들어오면 루프를 빠져나감
24     break
25
26 cap.release()              # 카메라와 연결을 끊음
27 cv.destroyAllWindows()
28
29 if len(frames)>0:           # 수집된 영상이 있으면
30     imgs=frames[0]
31     for i in range(1,min(3,len(frames))): # 최대 3개까지 이어 붙임
32         imgs=np.hstack((imgs,frames[i]))
33
34 cv.imshow('collected images',imgs)
35
36 cv.waitKey()
37 cv.destroyAllWindows()
```

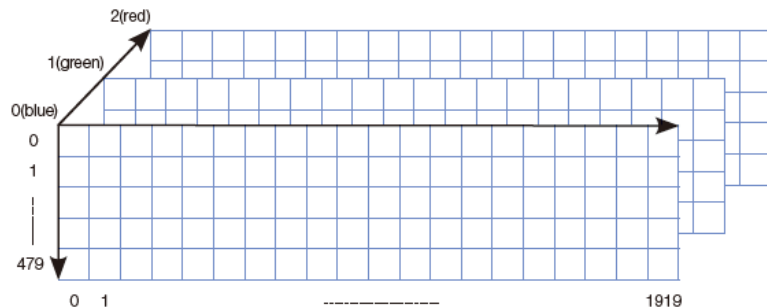


2.6 [프로그래밍 예제 3] 웹 캠에서 비디오 읽기

■ [프로그램 2-5]의 자료구조



(a) frames 리스트



(b) imgs 배열

```
In [1]: len(frames)
7
In [2]: frames[0].shape
(480,640,3)
In [3]: type(imgs)
numpy.ndarray
In [4]: imgs.shape
(480,1920,3)
```

2.7 [프로그래밍 예제 4] 그래픽 기능과 사용자 인터페이스 만들기

■ OpenCV의 그래픽 기능

- 영상에 글씨나 도형을 넣는데 유용([그림 2-6]의 Gui features in OpenCV 참조)
- line, rectangle, polylines, circle, ellipse, putText 함수



2.7 [프로그래밍 예제 4] 그래픽 기능과 사용자 인터페이스 만들기

■ [프로그램 2-6] 영상에 도형을 그리고 글씨 쓰기

프로그램 2-6

영상에 도형을 그리고 글자 쓰기

```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('girl_laughing.jpg')
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 cv.rectangle(img,(830,30),(1000,200),(0,0,255),2) # 직사각형 그리기
10 cv.putText(img,'laugh',(830,24),cv.FONT_HERSHEY_SIMPLEX,1,(255,0,0),2) # 글씨 쓰기
11
12 cv.imshow('Draw',img)
13
14 cv.waitKey()
15 cv.destroyAllWindows()
```



red

굵기

blue

2.7 [프로그래밍 예제 4] 그래픽 기능과 사용자 인터페이스 만들기



2.7 [프로그래밍 예제 4] 그래픽 기능과 사용자 인터페이스 만들기

■ 함수 선언에 대한 이해

◆ rectangle() [1/2]

```
void cv::rectangle ( InputOutputArray img,  
                    Point pt1,  
                    Point pt2,  
                    const Scalar & color,  
                    int thickness = 1 ,  
                    int lineType = LINE_8 ,  
                    int shift = 0  
                  )
```

Python:

```
cv.rectangle( img, pt1, pt2, color[, thickness[, lineType[, shift]]] ) -> img  
cv.rectangle( img, rec, color[, thickness[, lineType[, shift]]] ) -> img
```

```
#include <opencv2/imgproc.hpp>
```

Draws a simple, thick, or filled up-right rectangle.

The function `cv::rectangle` draws a rectangle outline or a filled rectangle whose two opposite corners are `pt1` and `pt2`.

Parameters

img	Image.
pt1	Vertex of the rectangle.
pt2	Vertex of the rectangle opposite to <code>pt1</code> .
color	Rectangle color or brightness (grayscale image).
thickness	Thickness of lines that make up the rectangle. Negative values, like <code>FILLED</code> , mean that the function has to draw a filled rectangle.
lineType	Type of the line. See <code>LineTypes</code> .
shift	Number of fractional bits in the point coordinates.

그림 2-12 OpenCV 공식 사이트가 제공하는 `rectangle` 함수의 선언(https://docs.opencv.org/4.x/d6/d6e/group__imgproc__draw.html#ga07d2f74cadcf8e305e810ce8eed13bc9)

2.7 [프로그래밍 예제 4] 그래픽 기능과 사용자 인터페이스 만들기

■ [프로그램 2-7] 마우스를 통한 상호작용(콜백 함수에 대한 이해 필요)

프로그램 2-7 마우스로 클릭한 곳에 직사각형 그리기

```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('girl_laughing.jpg')
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 def draw(event,x,y,flags,param):      # 콜백 함수
10     if event==cv.EVENT_LBUTTONDOWN:   # 마우스 왼쪽 버튼 클릭했
11         cv.rectangle(img,(x,y),(x+200,y+200),(0,0,255),2)
12     elif event==cv.EVENT_RBUTTONDOWN: # 마우스 오른쪽 버튼 클릭
13         cv.rectangle(img,(x,y),(x+100,y+100),(255,0,0),2)
14
15     cv.imshow('Drawing',img)
16
17 cv.namedWindow('Drawing')
18 cv.imshow('Drawing',img)
19
20 cv.setMouseCallback('Drawing',draw)   # Drawing 윈도우에 draw 콜백 함수 지정
21
22 while(True):                          # 마우스 이벤트가 언제 발생할지 모르므로 무한 반복
23     if cv.waitKey(1)==ord('q'):
24         cv.destroyAllWindows()
25     break
```



마우스이벤트 처리

- 마우스 이벤트를 처리하는 MouseCallback 함수를 만든다
 - cv::setMouseCallback()를 통하여 시스템에 등록
 - 시스템이 마우스 이벤트를 감지했을 때 callback 함수 호출
 - void setMouseCallback(const string& winname, MouseCallback onMouse, void* userdata=0)
 - typedef void(*MouseCallback)(int event, int x, int y, int flags, void* userdata)
-
- void setMouseCallback()
 - string& winname: 이벤트 발생을 검사할 윈도우 이름
 - MouseCallback onMouse: 마우스 이벤트를 처리할 콜백함수 이름(함수포인터)
 - void* userdata : 추가적으로 콜백함수로 전달할 사용자 정의 인수
 - void(*MouseCallback)()
 - event: 발생한 마우스 이벤트 종류
 - int x, int y : 이벤트 발생시 x,y좌표
 - int flags: 마우스 이벤트와 동시에 특수키(shift, alt, ctrl)이 눌러졌는지 여부

마우스이벤트

MouseEventType 열거형 상수	값	설명
EVENT_MOUSEMOVE	0	마우스가 창 위에서 움직인 경우
EVENT_LBUTTONDOWN	1	마우스 왼쪽 버튼을 누른 경우
EVENT_RBUTTONDOWN	2	마우스 오른쪽 버튼을 누른 경우
EVENT_MBUTTONDOWN	3	마우스 가운데 버튼을 누른 경우
EVENT_LBUTTONUP	4	마우스 왼쪽 버튼을 떼는 경우
EVENT_RBUTTONUP	5	마우스 오른쪽 버튼을 떼는 경우
EVENT_MBUTTONUP	6	마우스 가운데 버튼을 떼는 경우
EVENT_LBUTTONDBLCLK	7	마우스 왼쪽 버튼을 더블클릭하는 경우
EVENT_RBUTTONDBLCLK	8	마우스 오른쪽 버튼을 더블클릭하는 경우
EVENT_MBUTTONDBLCLK	9	마우스 중간 버튼을 더블클릭하는 경우
EVENT_MOUSEWHEEL	10	마우스 휠을 돌리는 경우
EVENT_MOUSEHWHEEL	11	마우스 휠을 좌우로 움직이는 경우

flag 종류는 다음과 같다.

cv2.EVENT_FLAG_LBUTTON	왼쪽 버튼 누름
cv2.EVENT_FLAG_RBUTTON	오른쪽 버튼 누름
cv2.EVENT_FLAG_MBUTTON	가운데 버튼 누름
cv2.EVENT_FLAG_ALTKEY	Alt 누름
cv2.EVENT_FLAG_CTRLKEY	Ctrl 누름
cv2.EVENT_FLAG_SHIFTKEY	Shift 누름

2.7 [프로그래밍 예제 4] 그래픽 기능과 사용자 인터페이스 만들기

■ [프로그램 2-8] 마우스 드래그로 도형 크기 조절하기

프로그램 2-8

마우스 드래그로 직사각형 그리기

```
01 import cv2 as cv
02 import sys
03
04 img=cv.imread('girl_laughing.jpg')
05
06 if img is None:
07     sys.exit('파일을 찾을 수 없습니다.')
08
09 def draw(event,x,y,flags,param):
10     global ix,iy
11
12     if event==cv.EVENT_LBUTTONDOWN: # 마우스 왼쪽 버튼 클릭했을 때 초기 위치 저장
13         ix,iy=x,y
14     elif event==cv.EVENT_LBUTTONUP: # 마우스 왼쪽 버튼 클릭했을
15         cv.rectangle(img,(ix,iy),(x,y),(0,0,255),2)
16
17     cv.imshow('Drawing',img)
18
19 cv.namedWindow('Drawing')
20 cv.imshow('Drawing',img)
21
22 cv.setMouseCallback('Drawing',draw)
23
24 while(True):
25     if cv.waitKey(1)==ord('q'):
26         cv.destroyAllWindows()
27     break
```



2.8 [프로그래밍 예제 5] 페인팅

■ [프로그램 2-9]: 페인팅 기능

프로그램 2-9

빨간색 붓과 파란색 붓으로 페인팅하기

```
01  import cv2 as cv
02  import sys
03
04  img=cv.imread('soccer.jpg')
05
06  if img is None:
07      sys.exit('파일을 찾을 수 없습니다.')
08
09  BrushSiz=5                                # 붓의 크기
10  LColor,RColor=(255,0,0),(0,0,255)         # 파란색과 빨간색
```



2.8 [프로그래밍 예제 5] 페인팅

```
11
12 def painting(event,x,y,flags,param):
13     if event==cv.EVENT_LBUTTONDOWN:
14         cv.circle(img,(x,y),BrushSiz,LColor,-1)    # 마우스 왼쪽 버튼 클릭하면 파란색
15     elif event==cv.EVENT_RBUTTONDOWN:
16         cv.circle(img,(x,y),BrushSiz,RColor,-1)    # 마우스 오른쪽 버튼 클릭하면 빨간색
17     elif event==cv.EVENT_MOUSEMOVE and flags==cv.EVENT_FLAG_LBUTTON:
18         cv.circle(img,(x,y),BrushSiz,LColor,-1)    # 왼쪽 버튼 클릭하고 이동하면 파란색
19     elif event==cv.EVENT_MOUSEMOVE and flags==cv.EVENT_FLAG_RBUTTON:
20         cv.circle(img,(x,y),BrushSiz,RColor,-1)    # 오른쪽 버튼 클릭하고 이동하면 빨간색
21
22     cv.imshow('Painting',img)                        # 수정된 영상을 다시 그림
23
24 cv.namedWindow('Painting')
25 cv.imshow('Painting',img)
26
27 cv.setMouseCallback('Painting',painting)
28
29 while(True):
30     if cv.waitKey(1)==ord('q'):
31         cv.destroyAllWindows()
32     break
```

`cv2.circle(img, center, radius, color[, thickness[, lineType[, shift]])` → img

parameter	내용
img	이미지 파일
center	원의 중심 좌표(x, y)
radius	원의 반지름
color	색상 (blue, green, red) 0 ~ 255
thickness	선 두께 (default 1)
lineType	선 종류 (default cv.Line_8) - LINE_8 : 8-connected line - LINE_4 : 4-connected line - LINE_AA : antialiased line
shift	fractional bit (default 0)

두께값을 -1로 하면 내부가 채워진 원



2.8 [프로그래밍 예제 5] 페인팅

