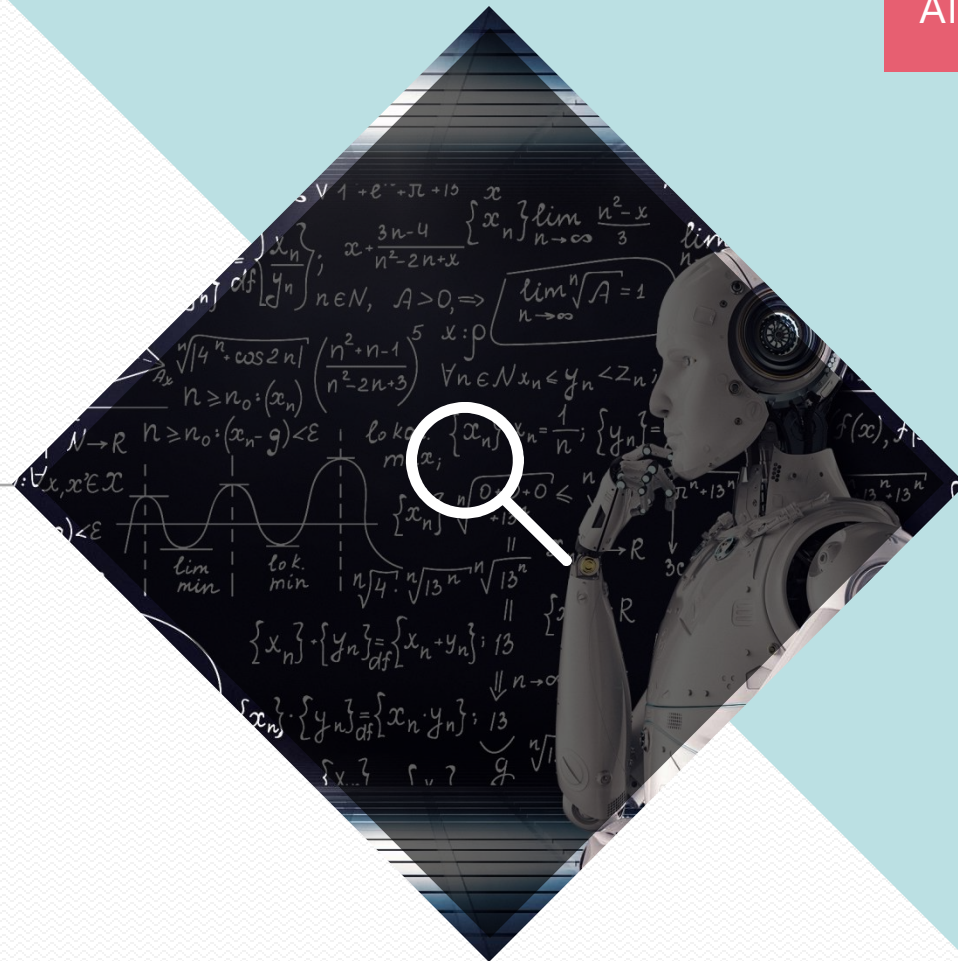


OpenCV



Overview

<https://opencv.org/>



- **Open**-source **C**omputer **V**ision
- 컴퓨터 비전 관련 작업을 위한 라이브러리 집합
- C++, Python, Java script 등의 언어 지원

Contents

1. OpenCV 란?

2. 개발 환경 설정

3. 이미지 - Basic Operation

4. 동영상 - Basic Operation

5. 카메라 - Basic Operation

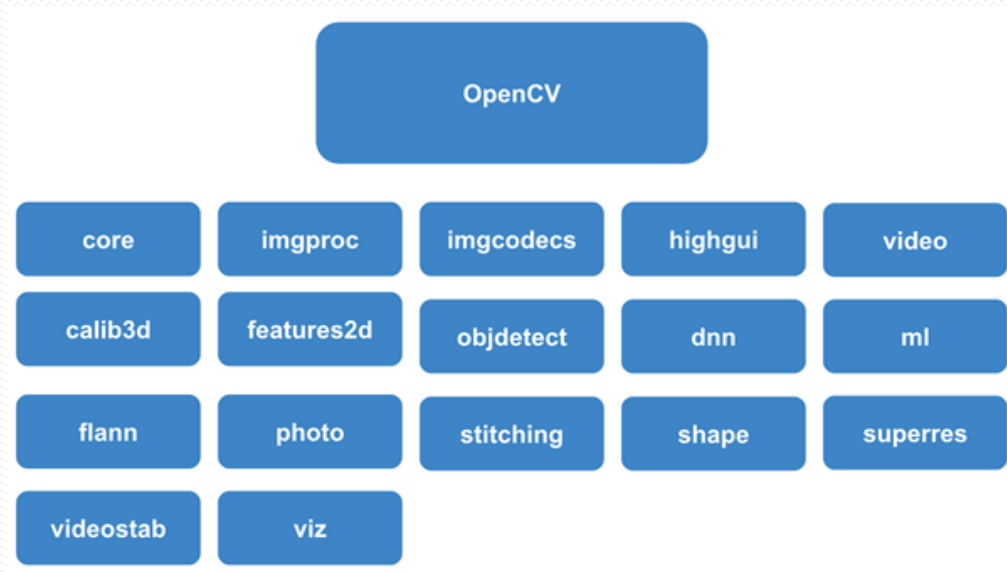
6. OpenCV 에서 제공하는 GUI Input 활용하기

7. Readme.md 작성법

1. OpenCV 란?

<https://docs.opencv.org/4.x/>

▪ OpenCV Modules



•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](#)

•Extra modules:

- alphamat. [Alpha Matting](#)
- aruco. [Aruco markers, module functionality was moved to objdetect module](#)
- bgsegm. [Improved Background-Foreground Segmentation Methods](#)
- bioinspired. [Biologically inspired vision models and derived tools](#)
- ccalib. [Custom Calibration Pattern for 3D reconstruction](#)
- cudaarithm. [Operations on Matrices](#)
- cudabgsegm. [Background Segmentation](#)
- cudacodec. [Video Encoding/Decoding](#)
- cudafeatures2d. [Feature Detection and Description](#)

⋮
skip...

Contents

1. OpenCV 란?

2. 개발 환경 설정 (Python)

3. 이미지 - Basic Operation

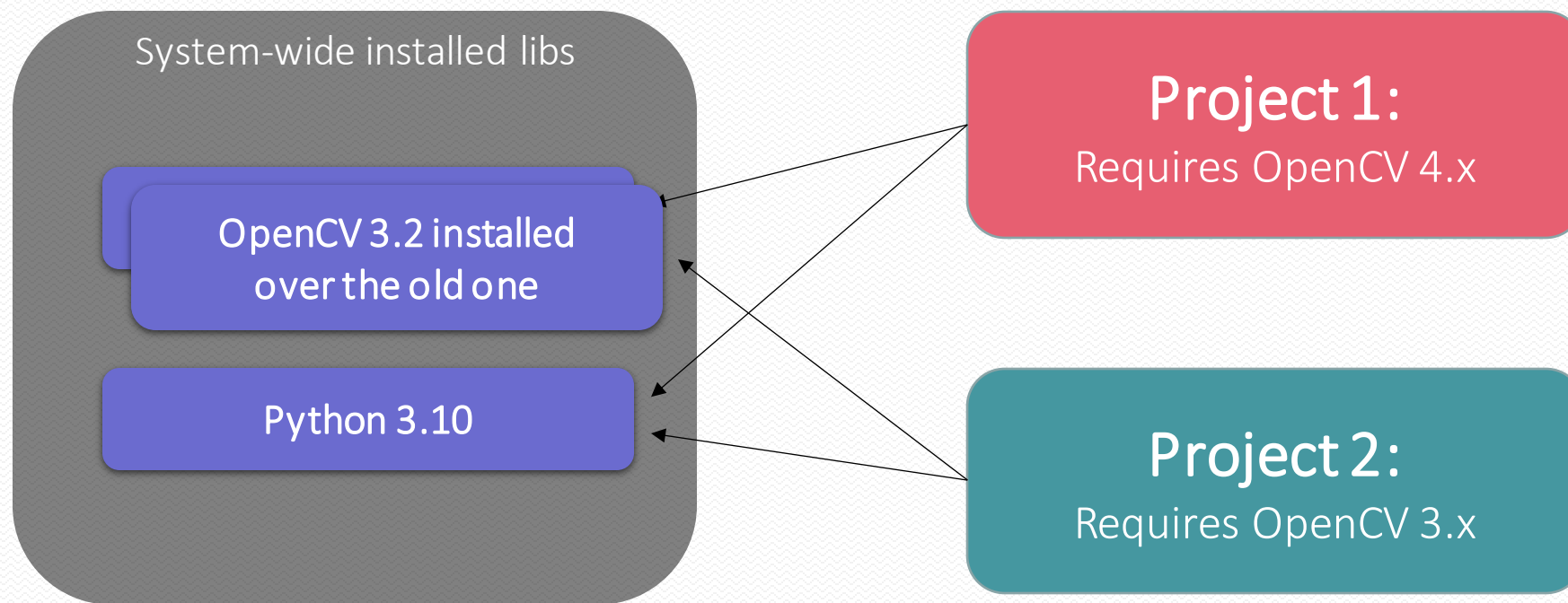
4. 동영상 - Basic Operation

5. 카메라 - Basic Operation

6. OpenCV 에서 제공하는 GUI Input 활용하기

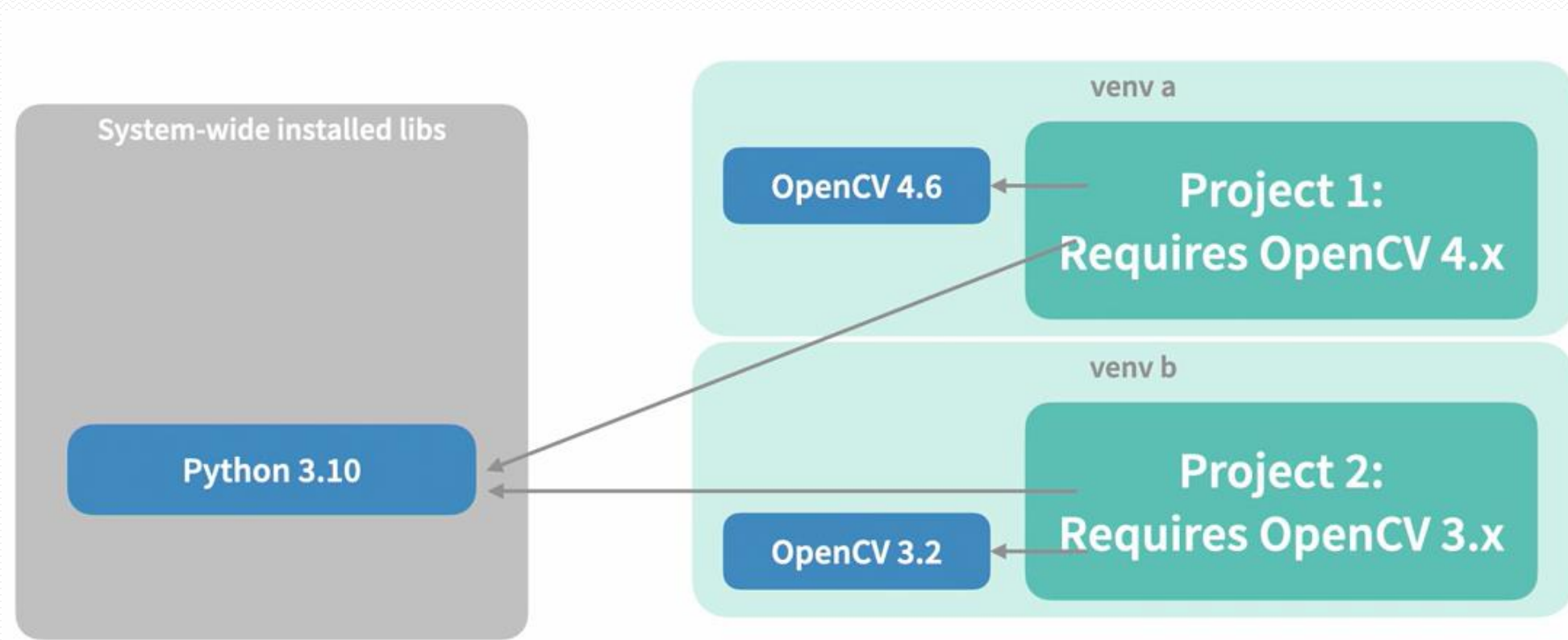
2. 개발 환경 설정

- 왜 가상환경을 사용해야 하나? **Problem: Dependency Conflicts**



2. 개발 환경 설정

- 왜 가상환경을 사용해야 하나?



2. 개발 환경 설정 - Windows

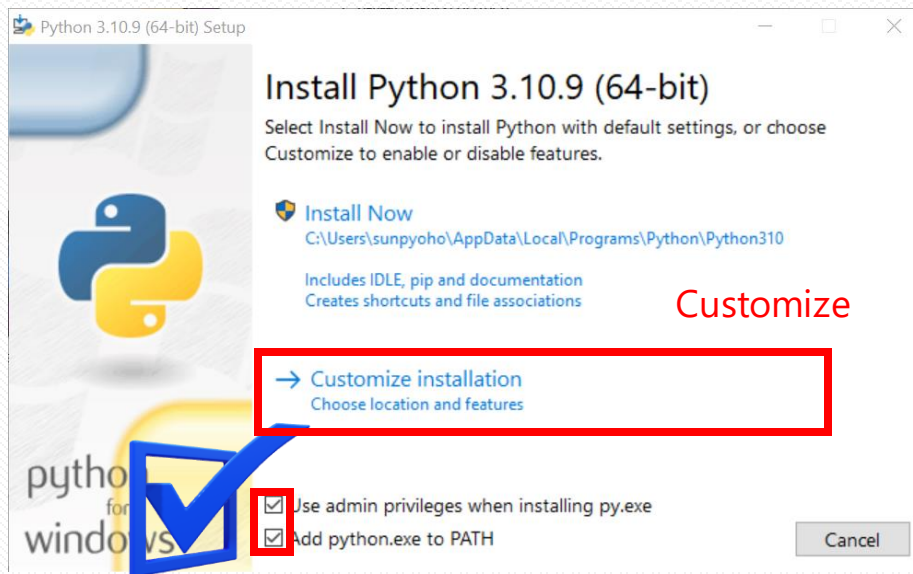
1. Python 설치

만약 본인 Windows PC 에 Python 이 설치되지 않았다면 공식 사이트에서 Python 3.7, 3.8, 3.9 또는 3.10 중 하나의 설치 파일을 다운로드 받아 설치한다. 이미 설치되어 있다면 이 단계는 생략 가능하다.

Python 3.10 의 경우 다음 링크에서 설치 파일을 다운로드/설치한다. <https://www.python.org/ftp/python/3.10.9/python-3.10.9-amd64.exe>

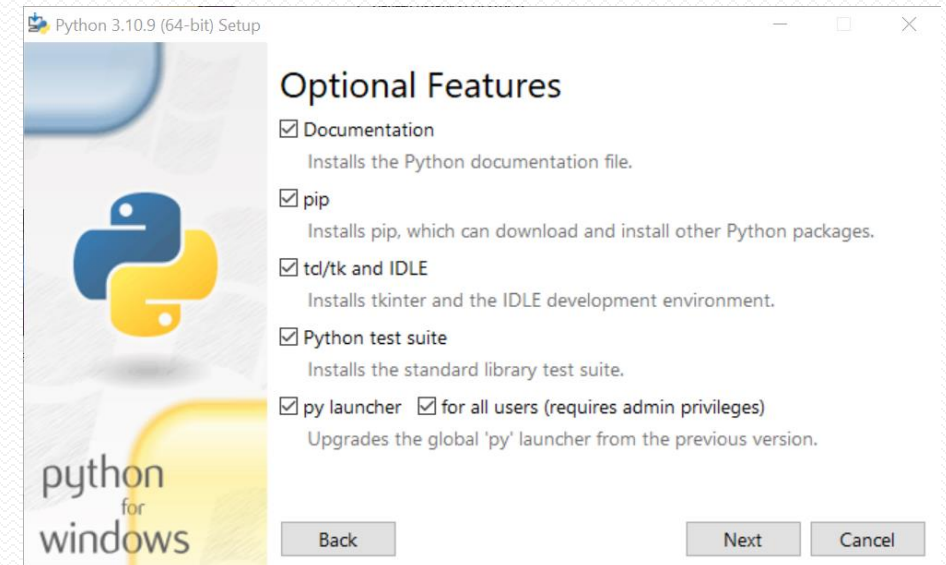


설치파일
더블클릭

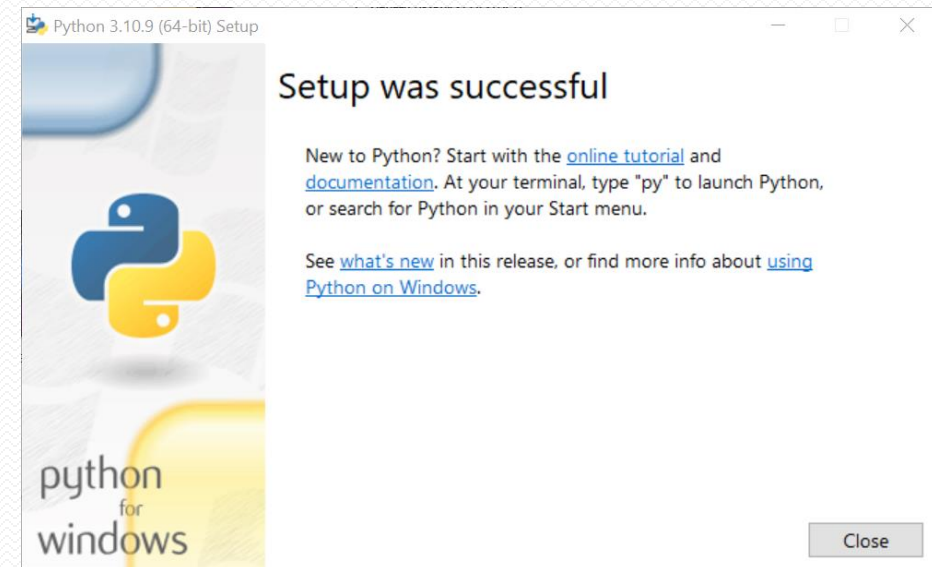
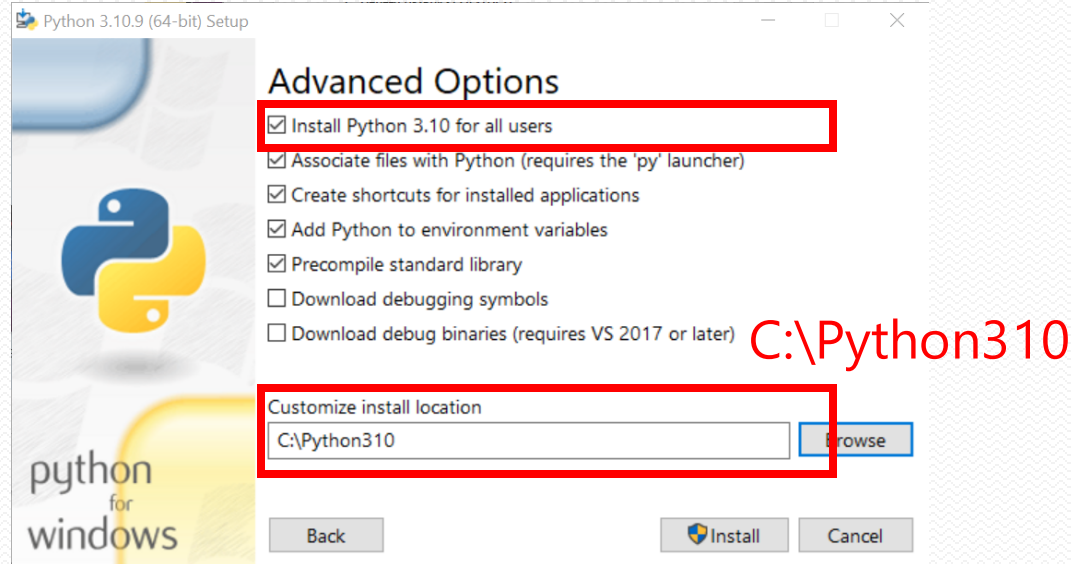


Customize

✓ 체크박스 2개 모두 선택



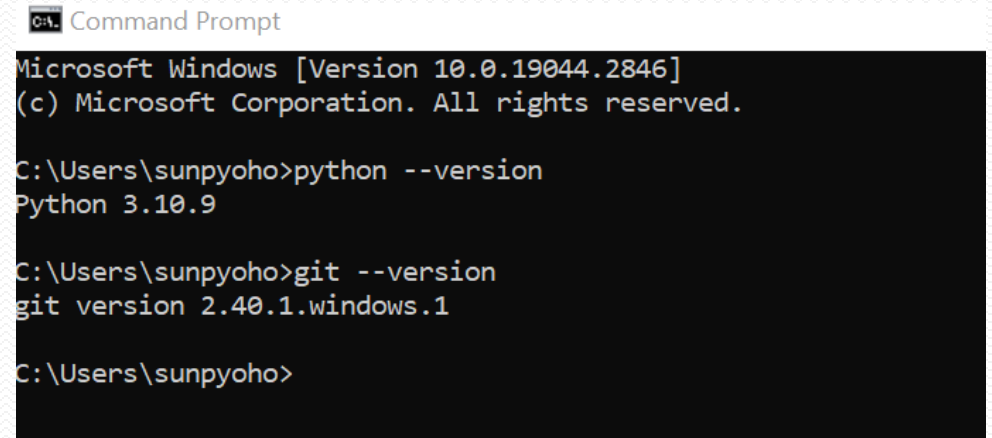
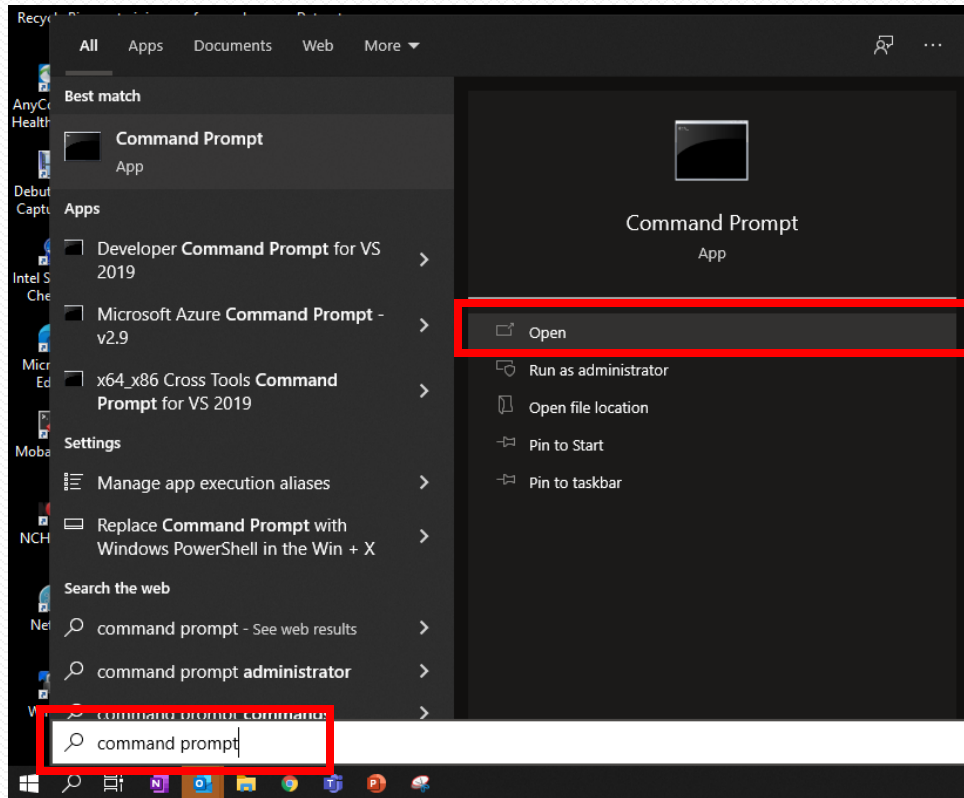
2. 개발 환경 설정 - Windows



2. 개발 환경 설정 - Windows

2. Python 가상환경 설정

- Windows Command 창을 연다. → Windows Key + "command prompt"



2. 개발 환경 설정 - Windows

2. Python 가상환경 설정

- Command Prompt 창에서 아래 명령어를 입력하여 Python 가상 환경을 생성한다. opencv_env 는 가상환경 이름으로 본인 취향따라
명명 가능

```
python -m venv opencv_env
```

- 아래 명령어를 입력하여 가상 환경을 활성화 한다.

```
opencv_env\Scripts\activate
```

- Command Prompt 창에 아래와 같이 가상환경 이름이 출력되면 성공적으로 가상환경이 활성화 된 것이다.

```
C:\Users\maxkim\opencv>opencv_env\Scripts\activate
```

```
(opencv_env) C:\Users\maxkim\opencv>
```

2. 개발 환경 설정 - Windows

3. 필요한 Python Package 들을 설치

- Python OpenCV 설치

```
(opencv_env) C:\[project path]>pip install opencv-python
```

```
Collecting opencv-python
  Using cached opencv_python-4.7.0.72-cp37-abi3-win_amd64.whl (38.2 MB)
Collecting numpy>=1.17.0
  Downloading numpy-1.25.0-cp39-cp39-win_amd64.whl (15.1 MB)
  | 15.1 MB 6.4 MB/s
Installing collected packages: numpy, opencv-python
Successfully installed numpy-1.25.0 opencv-python-4.7.0.72
```

- 설치 검증 - np.__version__ / cv2.__version__ 의 버전명이 출력되는지 확인한다.

```
(opencv_env) C:\Users\maxkim\opencv>python
Python 3.9.10 (tags/v3.9.10:f2f3f53, Jan 17 2022, 15:14:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> np.__version__
'1.25.0'
>>> import cv2
>>> cv2.__version__
'4.7.0'
```

2. 개발 환경 설정 - Ubuntu

1. Ubuntu 22.04 사용 시 default 로 Python 3.10 이 기 설치 되어 있어 별도 Python 설치 필요없음.
2. 첫째 날 수업의 Ubuntu 설치 과정에서 이미 python 가상환경 설정을 완료했으므로 해당 환경 사용.
 - Python3 가상환경 activation
 - Python OpenCV 설치 (아래 2가지 중 택일)

```
(.env) $ pip install opencv-python # only contains main modules
```

```
(.env) $ pip install opencv-contrib-python # contains both main and contrib modules
```

- 설치 검증 - np.__version__ / cv2.__version__ 의 버전명이 출력되는지 확인한다.

```
(.env) $ python3
>>> import cv2
>>> import numpy as np
>>> print("cv2 ver:", cv2.__version__, np ver:", np.__version__)
cv2 ver: 4.8.1 , np ver: 1.26.1 # your version might be different
```

Contents

1. OpenCV 란?
2. 개발 환경 설정
- 3. 이미지 - Basic Operation**
4. 동영상 - Basic Operation
5. 카메라 - Basic Operation
6. OpenCV 에서 제공하는 GUI Input 활용하기

https://www.w3schools.com/colors/colors_picker.asp

Colors HSL and HSLA

Colors HSL and HSLA

< Previous

Next >

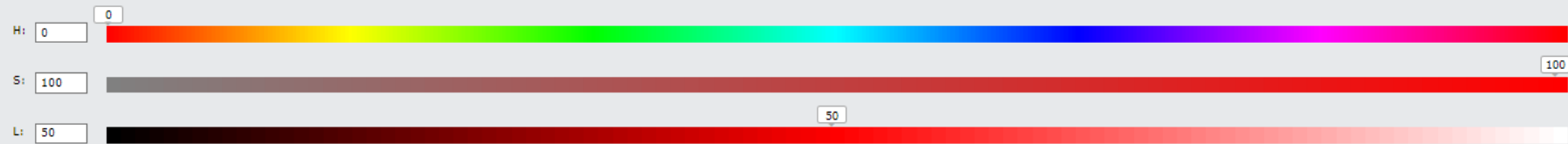
HSL Calculator



hsl(0, 100%, 50%)

rgb(255, 0, 0)

#ff0000



[Use this color in our Color Picker](#)

3. 이미지 - Basic Operation

■ 이미지를 Read / Write / Display

```
import numpy as np
import cv2

# 이미지 파일을 Read
img = cv2.imread("my_input.jpg")

# Image 란 이름의 Display 창 생성
cv2.namedWindow("image", cv2.WINDOW_NORMAL)

# Numpy ndarray H/W/C order
print(img.shape)

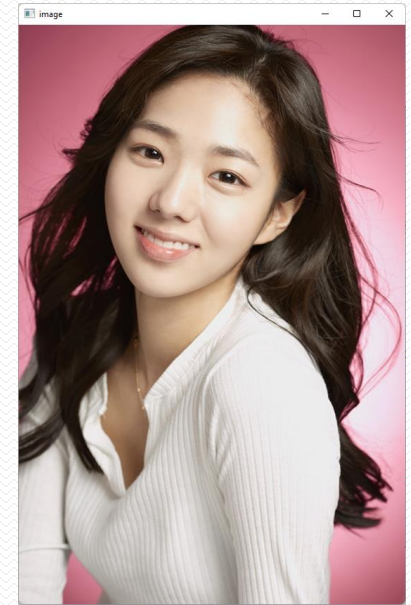
# Read 한 이미지 파일을 Display
cv2.imshow("image", img)

# 별도 키 입력이 있을때 까지 대기
cv2.waitKey(0)

# output.png 로 읽은 이미지 파일을 저장
cv2.imwrite("output.png", img)

# Destroy all windows
cv2.destroyAllWindows()
```

Quiz



1. `print(img.shape)` 의 출력 결과는 무슨 의미일까?
2. 본인이 좋아하는 사진을 web 에서 다운받아서 OpenCV API 를 사용해서 Display 및 파일로 저장해 보자.
3. 현재는 별도의 키 입력이 있을때 까지 `cv2.waitKey(0)` 함수에서 대기하게 된다. 코드를 추가해서 소문자 "s" 키를 입력받을때만 이미지 파일을 저장하고 다른 키가 입력되면 이미지 파일을 저장하지 않게 수정해 보자.

3. 이미지 - Basic Operation

RGB/HSV Color Space (색 공간)

```
# 이미지 파일을 Read 하고 Color space 정보 출력
color = cv2.imread("strawberry.jpg", cv2.IMREAD_COLOR)
#color = cv2.imread("strawberry_dark.jpg", cv2.IMREAD_COLOR)
print(color.shape)
```

```
height,width,channels = color.shape
cv2.imshow("Original Image",color)
```

```
# Color channel 을 B,G,R 로 분할하여 출력
b,g,r = cv2.split(color)
rgb_split = np.concatenate((b,g,r),axis=1)
cv2.imshow("BGR Channels",rgb_split)
```

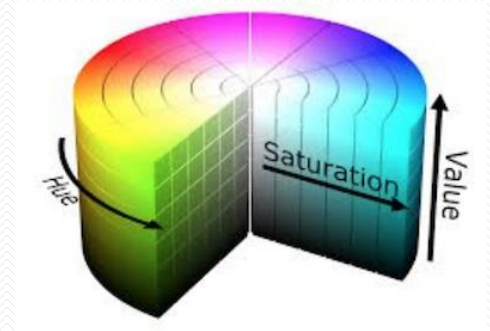
```
# 색공간을 BGR 에서 HSV 로 변환
hsv = cv2.cvtColor(color, cv2.COLOR_BGR2HSV)
```

```
# Channel 을 H,S,V 로 분할하여 출력
h,s,v = cv2.split(hsv)
hsv_split = np.concatenate((h,s,v),axis=1)
cv2.imshow("Split HSV",hsv_split)
```

```
enum cv::ImreadModes {
    cv::IMREAD_UNCHANGED = -1,
    cv::IMREAD_GRAYSCALE = 0,
    cv::IMREAD_COLOR = 1,
    cv::IMREAD_ANYDEPTH = 2,
    cv::IMREAD_ANYCOLOR = 4,
    cv::IMREAD_LOAD_GDAL = 8,
    cv::IMREAD_REDUCED_GRAYSCALE_2 = 16,
    cv::IMREAD_REDUCED_COLOR_2 = 17,
    cv::IMREAD_REDUCED_GRAYSCALE_4 = 32,
    cv::IMREAD_REDUCED_COLOR_4 = 33,
    cv::IMREAD_REDUCED_GRAYSCALE_8 = 64,
    cv::IMREAD_REDUCED_COLOR_8 = 65,
    cv::IMREAD_IGNORE_ORIENTATION = 128
}
```



RGB color space



HSV color space

Quiz

1. 위 색공간 이미지의 링크로 이동해서 각 색 공간의 표현 방법을 이해해 보자.
2. HSV color space 가 어떤 경우에 효과적으로 사용될까?
3. HSV 로 변환된 이미지를 BGR 이 아닌 RGB 로 다시 변환해서 출력해 보자.
4. COLOR_RGB2GRAY 를 사용해서 흑백으로 변환해 출력해 보자.

3. 이미지 - Basic Operation

▪ Crop / Resize (자르기 / 크기 조정)

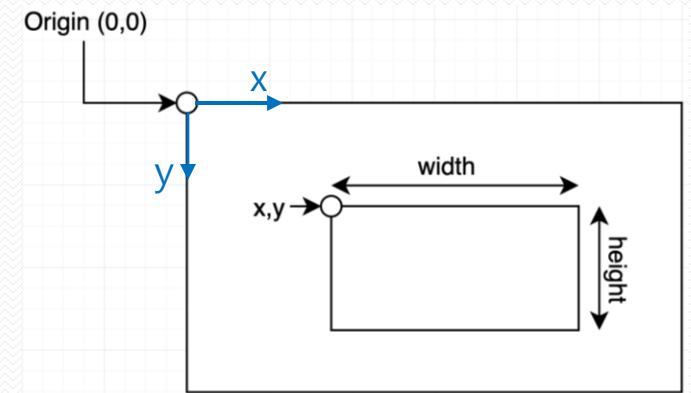
```
# 이미지 파일을 Read
img = cv2.imread("my_input.jpg")

# Crop 300x400 from original image from (100, 50)=(x,y)
cropped = img[50:450, 100:400]

# Resize cropped image from 300x400 to 400x200
resized = cv2.resize(cropped, (400,200))

# Display all
cv2.imshow("Original", img)
cv2.imshow("Cropped image", cropped)
cv2.imshow("Resized image", resized)

cv2.waitKey(0)
cv2.destroyAllWindows()
```



< OpenCV 에서 2D 이미지 좌표 개념 >

Quiz

1. Input image 를 본인이 좋아하는 인물 사진으로 변경해서 적용하자. 그리고 본인이 사용한 input image 의 size 를 확인해 보자.
2. 본인이 사용한 이미지의 얼굴 영역만 crop 해서 display 해 보자.
3. 원본 이미지의 정확히 1.5배만큼 이미지를 확대해서 파일로 저장해 보자.
4. openCV 의 rotate API 를 사용해서 우측으로 90도만큼 회전된 이미지를 출력해 보자.

3. 이미지 - Basic Operation

- 역상 (Reverse Image)

```
import cv2

src = cv2.imread("my_input.jpg",
cv2.IMREAD_COLOR)
dst = cv2.bitwise_not(src)

cv2.imshow("src", src)
cv2.imshow("dst", dst)
cv2.waitKey()
cv2.destroyAllWindows()
```

Quiz

1. AND, OR, XOR 연산에 대해서 확인해 보자.

3. 이미지 - Basic Operation

- 이진화 (Binary)

```
import cv2

src = cv2.imread("my_input.jpg",
cv2.IMREAD_COLOR)

gray = cv2.cvtColor(src,
cv2.COLOR_BGR2GRAY)
ret, dst = cv2.threshold(gray, 100, 255,
cv2.THRESH_BINARY)

cv2.imshow("dst", dst)
cv2.waitKey()
cv2.destroyAllWindows()
```

Quiz

- 임계값을 변화시켜 보자.

3. 이미지 - Basic Operation

- 흐림효과 (Blur)

```
import cv2

src = cv2.imread("my_input.jpg",
cv2.IMREAD_COLOR)
dst = cv2.blur(src, (9, 9), anchor=(-1, -
1), borderType=cv2.BORDER_DEFAULT)

cv2.imshow("dst", dst)
cv2.waitKey()
cv2.destroyAllWindows()
```

Quiz

1. Kernel Size를 변경하여 보자.
2. borderType을 변경하여 보자.
(cv2.BORDER_REFLECT)

3. 이미지 - Basic Operation

- 가장자리 검출 (Edge)

```
import cv2

src = cv2.imread("Image/wheat.jpg", cv2.IMREAD_COLOR)
gray = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY)

sobel = cv2.Sobel(gray, cv2.CV_8U, 1, 0, 3)

cv2.imshow("sobel", sobel)

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

Quiz

1. Laplacian 변환을 적용해 보자.
2. Canny Edge Detection을 적용해 보자

3. 이미지 - Basic Operation

- 배열 병합 (add Weighted)

```
import cv2

src = cv2.imread("tomato.jpg", cv2.IMREAD_COLOR)
hsv = cv2.cvtColor(src, cv2.COLOR_BGR2HSV)
h, s, v = cv2.split(hsv)

lower_red = cv2.inRange(hsv, (0, 100, 100), (5, 255, 255))
upper_red = cv2.inRange(hsv, (170, 100, 100), (180, 255, 255))
added_red = cv2.addWeighted(lower_red, 1.0, upper_red, 1.0, 0.0)

red = cv2.bitwise_and(hsv, hsv, mask = added_red)
red = cv2.cvtColor(red, cv2.COLOR_HSV2BGR)

cv2.imshow("red", red)
cv2.waitKey()
cv2.destroyAllWindows()
```



Quiz

1. lower_red 값의 범위를 변경해 보자.
2. upper_red 값의 범위를 변경해 보자.
3. addWeighted의 gamma 값을 변경해 보자

3. 이미지 - Basic Operation

- 채널 분리 및 병합

```
import cv2

src = cv2.imread("tomato.jpg",
cv2.IMREAD_COLOR)
b, g, r = cv2.split(src)
inverse = cv2.merge((r, g, b))

cv2.imshow("b", b)
cv2.imshow("g", g)
cv2.imshow("r", r)
cv2.imshow("inverse", inverse)
cv2.waitKey()
cv2.destroyAllWindows()
```



Quiz

1. Numpy 형태의 채널 분리를 적용해 보자.

```
b = src[:, :, 0]
g = src[:, :, 1]
r = src[:, :, 2]
```

2. 빈 이미지를 적용해 보자.

```
height, width, channel = src.shape
zero = np.zeros((height, width, 1),
dtype=np.uint8)
bgz = cv2.merge((b, g, zero))
```

Contents

1. OpenCV 란?
2. 개발 환경 설정
3. 이미지 - Basic Operation
- 4. 동영상 - Basic Operation**
5. 카메라 - Basic Operation
6. OpenCV 에서 제공하는 GUI Input 활용하기

4. 동영상 - Basic Operation

■ 동영상 파일을 읽고 보여주기

```
import numpy as np
import cv2

# Read from the recorded video file
cap = cv2.VideoCapture("ronaldinho.mp4")

# 동영상 파일이 성공적으로 열렸으면 while 문 반복
while(cap.isOpened()):
    # 한 프레임을 읽어옴
    ret, frame = cap.read()

    if ret is False:
        print("Can't receive frame (stream end?). Exiting ...")
        break

    # Display
    cv2.imshow("Frame", frame)

    # 1 ms 동안 대기하며 키 입력을 받고 'q' 입력 시 종료
    key = cv2.waitKey(1)
    if key & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Length	00:01:08
Frame width	1280
Frame height	720
Data rate	2562kbps
Total bitrate	2565kbps
Frame rate	24.00 frames/second



Quiz

1. 동영상이 너무 빠르게 재생된다. 이유를 찾아보고 정상적인 속도로 재생될 수 있도록 수정해 보자.
2. 동영상이 끝까지 재생되면 더이상 frame 을 읽어오지 못해 종료된다. 동영상이 끝까지 재생되면 다시 처음부터 반복 될 수 있도록 수정해 보자.
3. 동영상 크기를 반으로 resize 해서 출력해 보자.
4. 동영상 재생 중 'c' 키 입력을 받으면 해당 프레임을 이미지 파일로 저장하게 코드를 수정해 보자. 파일 이름은 001.jpg, 002.jpg 등으로 overwrite 되지 않게 하자.

4. 동영상 - Basic Operation

▪ 캡처 및 녹화 (Capture & Record)

```
import datetime
import cv2

capture = cv2.VideoCapture("ronaldinho.mp4")
fourcc = cv2.VideoWriter_fourcc(*'XVID')
record = False

while True:
    if(capture.get(cv2.CAP_PROP_POS_FRAMES) ==
       capture.get(cv2.CAP_PROP_FRAME_COUNT)):
        capture.open("/Image/Star.mp4")

    ret, frame = capture.read()
    cv2.imshow("VideoFrame", frame)

    now = datetime.datetime.now().strftime("%d_%H-%M-%S")
    key = cv2.waitKey(33)

    if key == 27:
        break
    elif key == 26:
        print("캡처")
        cv2.imwrite("D:" + str(now) + ".png", frame)
```

```
elif key == 24:
    print("녹화 시작")
    record = True
    video = cv2.VideoWriter("D:" + str(now) + ".avi", fourcc,
20.0, (frame.shape[1], frame.shape[0]))
elif key == 3:
    print("녹화 중지")
    record = False
    video.release()

if record == True:
    print("녹화 중..")
    video.write(frame)

capture.release()
cv2.destroyAllWindows()
```

Contents

1. OpenCV 란?
2. 개발 환경 설정
3. 이미지 - Basic Operation
4. 동영상 - Basic Operation
- 5. 카메라 - Basic Operation**
6. OpenCV 에서 제공하는 GUI Input 활용하기

5. 카메라 - Basic Operation

- 카메라로부터 input 을 받아 보여주고 동영상 파일로 저장하기

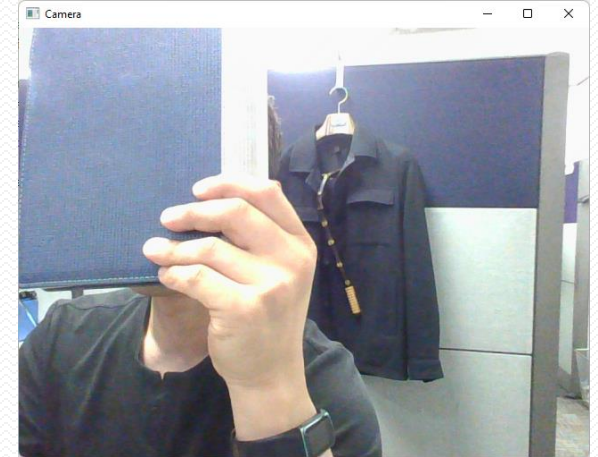
```
# Read from the first camera device
cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)

w = 640#1280#1920
h = 480#720#1080
cap.set(cv2.CAP_PROP_FRAME_WIDTH, w)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, h)

# 성공적으로 video device 가 열렸으면 while 문 반복
while(cap.isOpened()):
    # 한 프레임을 읽어옴
    ret, frame = cap.read()
    if ret is False:
        print("Can't receive frame (stream end?). Exiting ...")
        break

    # Display
    cv2.imshow("Camera", frame)

    # 1 ms 동안 대기하며 키 입력을 받고 'q' 입력 시 종료
    key = cv2.waitKey(1)
    if key & 0xFF == ord('q'):
        break
```



Quiz

- 가지고 있는 카메라의 지원 가능한 해상도를 확인 후 카메라 해상도를 변경해 보자.
- 카메라 Input 을 "output.mp4" 동영상 파일로 저장하도록 코드를 추가해 보자.

Contents

1. OpenCV 란?
2. 개발 환경 설정
3. 이미지 - Basic Operation
4. 동영상 - Basic Operation
5. 카메라 - Basic Operation
6. OpenCV 에서 제공하는 GUI Input 활용

6. OpenCV 에서 제공하는 GUI Input 활용하기

▪ Text / Line / Rectangle

```
# Read from the first camera device
cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)

topLeft = (50, 50)
bottomRight = (300, 300)

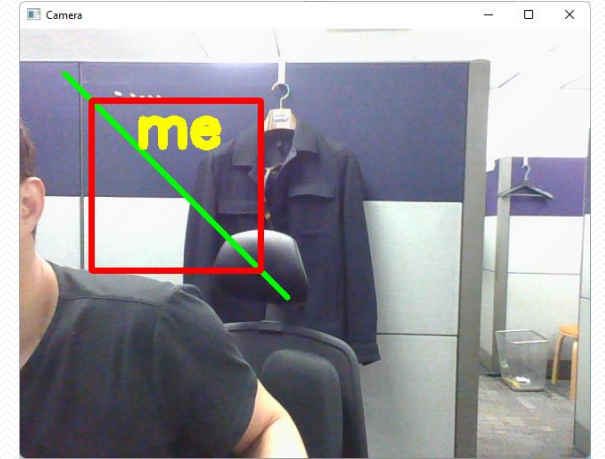
# 성공적으로 video device 가 열렸으면 while 문 반복
while(cap.isOpened()):
    # 한 프레임을 읽어옴
    ret, frame = cap.read()

    # Line
    cv2.line(frame, topLeft, bottomRight, (0, 255, 0), 5)

    # Rectangle
    cv2.rectangle(frame,
                  [pt+30 for pt in topLeft], [pt-30 for pt in bottomRight], (0, 0, 255), 5)

    # Text
    font = cv2.FONT_HERSHEY_SIMPLEX
    cv2.putText(frame, 'me',
                [pt+80 for pt in topLeft], font, 2, (0, 255, 255), 10)

    # Display
    cv2.imshow("Camera", frame)
```



Quiz

1. Text 문구 / Font / 색상 / 크기 / 굵기 / 출력위치 등 모든 값을 변경해 보자.
2. 동그라미를 그리는 함수를 찾아서 적용해 보자.
3. 마우스 왼쪽 버튼을 click 하면 해당 위치에 동그라미가 그려지도록 코드를 추가해 보자.
(Reference : cv2.EVENT_LBUTTONDOWN)

6. OpenCV 에서 제공하는 GUI Input 활용하기

▪ Trackbar

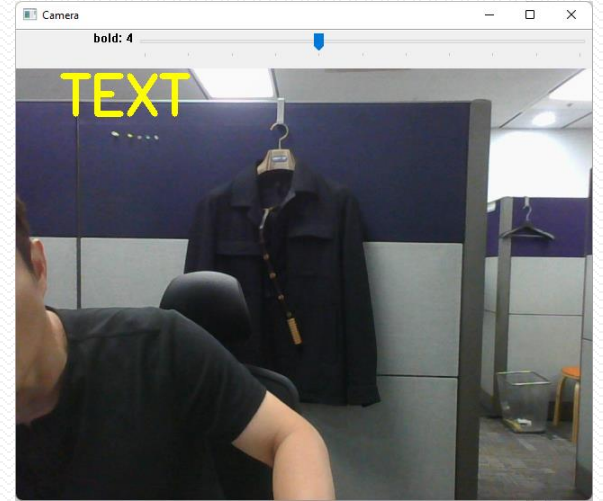
```
topLeft = (50, 50)
bold = 0
# Callback function for the trackbar
def on_bold_trackbar(value):
    #print("Trackbar value:", value)
    global bold
    bold = value

cv2.namedWindow("Camera")
cv2.createTrackbar("bold", "Camera", bold, 10, on_bold_trackbar)

# 성공적으로 video device 가 열렸으면 while 문 반복
while(cap.isOpened()):
    # 한 프레임을 읽어옴
    ret, frame = cap.read()
    if ret is False:
        print("Can't receive frame (stream end?). Exiting ...")
        break

    # Text
    cv2.putText(frame, "TEXT",
                topLeft, cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 255, 255), 1 + bold)

    # Display
    cv2.imshow("Camera", frame)
```



Quiz

1. Trackbar 를 control 해서 TEXT 의 굵기가 변하는 것을 확인해 보자.
2. Trackbar 를 추가해서 font size 를 변경 / 적용해 보자.
3. R/G/B Trackbar 를 각각 추가해서 글자의 font color 를 변경해 보자.

Contents

1. OpenCV 란?
2. 개발 환경 설정
3. 이미지 - Basic Operation
4. 동영상 - Basic Operation
5. 카메라 - Basic Operation
6. OpenCV 에서 제공하는 GUI Input 활용
7. Readme.md 작성법

7. Readme.md 작성법

```
1 # Project ABC
2
3 * (간략히 프로젝트 중 목표가 무엇인지에 대해 기술)
4
5 ## High Level Design
6
7 * (프로젝트 아키텍처 기술, 전반적인 diagram 으로 설명을 권장)
8
9 ## Clone code
10
11 ```shell
12 git clone https://github.com/xxx/yyy/zzz
13 ```
14
15 ## Prerequisite
16
17 * (프로젝트를 실행하기 위해 필요한 dependencies 및 configuration들이 있다면, 설치 및 설정 방법에 대해 기술)
18
19 ```shell
20 python -m venv .venv
21 source .venv/bin/activate
22 pip install -r requirements.txt
23 ```
```

```
Preview Code Blame 59 lines (37 loc) · 1.14 KB

Project ABC

• (간략히 프로젝트를 설명하고, 최종 목표가 무엇인지에 대해 기술)

High Level Design

• (프로젝트 아키텍처 기술, 전반적인 diagram 으로 설명을 권장)

Clone code

• (각 팀에서 프로젝트를 위해 생성한 repository에 대한 code clone 방법에 대해서 기술)



```
git clone https://github.com/xxx/yyy/zzz
```



Prerequisite

• (프로젝트를 실행하기 위해 필요한 dependencies 및 configuration들이 있다면, 설치 및 설정 방법에 대해 기술)



```
python -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
```


```

* 프로젝트개요를 readme.md에 기입

* 개요에는 간략한 프로젝트 설명/High Level Design/repository에 대한 code clone 방법/dependencies 및 configuration/실행및 빌드절차/실행결과 혹은 산출물에 대한 내용이 포함되야함.



Python3 Cheat Sheet



Python 3 Beginner's Reference Cheat Sheet

Alvaro Sebastian

<http://www.sixthresearcher.com>

Main data types

`boolean` = `True` / `False`
`integer` = 10
`float` = 10.01
`string` = "123abc"
`list` = [value1, value2, ...]
`dictionary` = { key1:value1, key2:value2, ... }

Numeric operators

`+` addition
`-` subtraction
`*` multiplication
`/` division
`**` exponent
`%` modulus
`//` floor division

Comparison operators

`==` equal
`!=` different
`>` higher
`<` lower
`>=` higher or equal
`<=` lower or equal

Boolean operators

`and` logical AND
`or` logical OR
`not` logical NOT

Special characters

`#` coment
`\n` new line
`\<char>` scape char

String operations

`string[i]` retrieves character at position `i`
`string[-1]` retrieves last character
`string[i:j]` retrieves characters in range `i` to `j`

List operations

`list = []` defines an empty list
`list[i] = x` stores `x` with index `i`
`list[i]` retrieves the item with index `i`
`list[-1]` retrieves last item
`list[i:j]` retrieves items in the range `i` to `j`
`del list[i]` removes the item with index `i`

Dictionary operations

`dict = {}` defines an empty dictionary
`dict[k] = x` stores `x` associated to key `k`
`dict[k]` retrieves the item with key `k`
`del dict[k]` removes the item with key `k`

String methods

`string.upper()` converts to uppercase
`string.lower()` converts to lowercase
`string.count(x)` counts how many times `x` appears
`string.find(x)` position of the `x` first occurrence
`string.replace(x,y)` replaces `x` for `y`
`string.strip(x)` returns a list of values delimited by `x`
`string.join(L)` returns a string with `L` values joined by string
`string.format(x)` returns a string that includes formatted `x`

List methods

`list.append(x)` adds `x` to the end of the list
`list.extend(L)` appends `L` to the end of the list
`list.insert(i,x)` inserts `x` at `i` position
`list.remove(x)` removes the first list item whose value is `x`
`list.pop(i)` removes the item at position `i` and returns its value
`list.clear()` removes all items from the list
`list.index(x)` returns a list of values delimited by `x`
`list.count(x)` returns a string with list values joined by `S`
`list.sort()` sorts list items
`list.reverse()` reverses list elements
`list.copy()` returns a copy of the list

Dictionary methods

`dict.keys()` returns a list of keys
`dict.values()` returns a list of values
`dict.items()` returns a list of pairs (key,value)
`dict.get(k)` returns the value associated to the key `k`
`dict.pop()` removes the item associated to the key and returns its value
`dict.update(D)` adds keys-values (`D`) to dictionary
`dict.clear()` removes all keys-values from the dictionary
`dict.copy()` returns a copy of the dictionary

Legend: `x,y` stand for any kind of data values, `s` for a string, `n` for a number, `L` for a list where `i,j` are list indexes, `D` stands for a dictionary and `k` is a dictionary key.