

▼ 2021-1 영상정보처리 3주차 과제 템플릿

마감: 2021년 3월 25일 오후 11시 59분 이름: 박민혁

학번: 32151671

구글 드라이브를 연결하고 자신의 노트북이 저장되어 있는 폴더로 이동하시오

```
from google.colab import drive
drive.mount('/gdrive')
%cd /gdrive/My Drive/Classroom/[영상정보처리] 2000004793-2021-1/Report Week 3
```

```
Drive already mounted at /gdrive; to attempt to forcibly remount, call drive.mount("/gdrive", fo
/gdrive/My Drive/Classroom/[영상정보처리] 2000004793-2021-1/Report Week 3
```



다음에 이미지 경로 '[../Dongkeun-OpenCV-ImgData/logo.png](#)' 를 변경하지 말고 이미지를 읽고, 해당 이미지에 대한 type, size, dtype 등의 속성을 체크하시오.

```
import matplotlib.pyplot as pyplot
import matplotlib.cm as cm
import cv2
```

```
img = cv2.imread('../Dongkeun-OpenCV-ImgData/logo.png')
print('type = ', type(img))
print('size = ', img.size)
print('dtype = ', img.dtype)
```

```
↳ type = <class 'numpy.ndarray'>
size = 876600
dtype = uint8
```

다음의 셀에 읽은 이미지에서 좌표 $y = 100$, $x = 50$ 에 있는 화소의 각 색요소 값을 한번에 읽어 출력하시오. 또한 각 색요소의 값을 별도로 가져오는 예를 작성하시오.

```
pyplot.imshow(img)
pyplot.show()
```

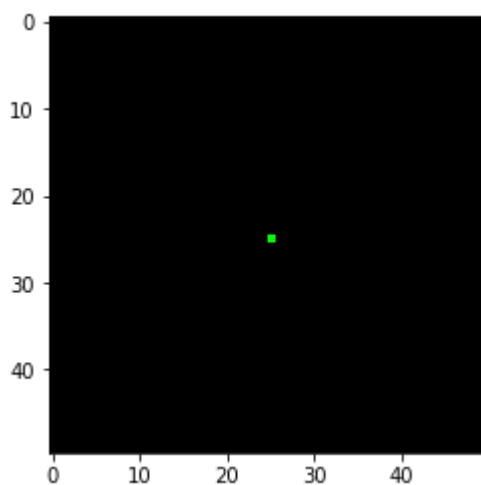


```
(b, g, r) = img[100, 50]
print(b, g, r)
b = img[4, 250, 2]
print(b)
g = img[300, 50, 1]
print(g)
r = img[400, 350, 0]
print(r)
```

```
0 0 0
255
255
255
```

다음의 셀에 읽은 이미지에서 좌표 $y = 100, x = 50$ 의 화소를 순수 녹색(green) 으로 세팅하는 코드를 작성하고 해당 분야를 ROI 를 이용해 확대해서 출력하시오.

```
img[100, 50] = (0, 255, 0)
format_image = img[75:125, 25:75]
pyplot.imshow(format_image)
pyplot.show()
```

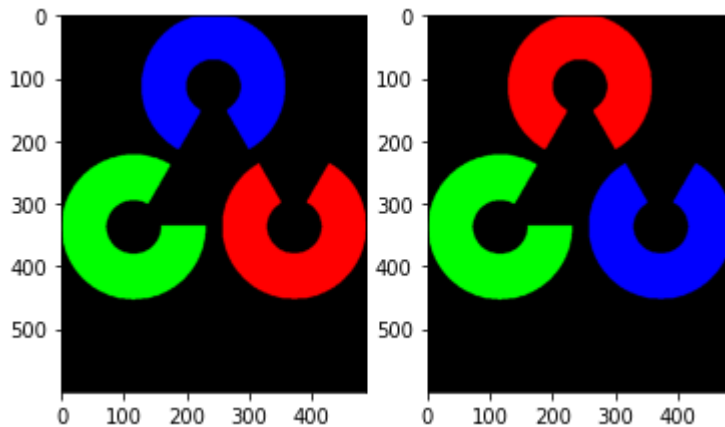


▼ Accessing and Manipulating pixels in opencv with grayscale images

위에서 사용한 동일 입력 이미지를 읽어서 matplotlib.pyplot에 맞는 채널 순서로 변환하고 본래 이미

```
opencv_img = cv2.imread('../Dongkeun-OpenCV-ImgData/logo.png')
b, g, r = cv2.split(opencv_img)
matplot_img = cv2.merge((r, g, b))

pyplot.subplot(121)
pyplot.imshow(opencv_img)
pyplot.subplot(122)
pyplot.imshow(matplot_img)
pyplot.show()
```



▼ In Depth Example

강의노트 'In Depth' 부분에 있는 예제를 임의의 값을 이용하여 예시하고 간단한 설명을 붙여 자기 노트를 완성하시오.

▼ Python Slice Indexing with Numpy

```
import numpy as np

np.array(8)

array(8)

np.array([2.3, 0.1, -9.1,])

array([ 2.3,  0.1, -9.1])

np.array([[93, 95], [84, 100], [99, 87]])

array([[ 93,  95],
       [ 84, 100],
```

```

        [ 99, 87]])

np.array([[[0, 1], [2, 3]], [[4, 5], [6, 7]]])

array([[[0, 1],
        [2, 3]],

       [[4, 5],
        [6, 7]]])

x = np.array([[[0, 1], [2, 3]], [[4, 5], [6, 7]]])
print(x)

```

```

[[[0 1]
  [2 3]]

 [[4 5]
  [6 7]]]

```

```

print(x[0, :, ::-1])
# 첫 번째는 0행, 두 번째는 전체 열, 세 번째는 순서 뒤집기이다.

```

```

[[1 0]
 [3 2]]

```

▼ One - Dimensional Array In Numpy

```

simple_array = np.array([2.3, 0.1, -9.1])
print(simple_array)
print(simple_array[0])
print(simple_array[-2])
print(simple_array[1:3])
# print(simple_array[3]) -> 에러 마지막 인덱스는 2이다. 인덱스 범위 초과.

```

```

[ 2.3  0.1 -9.1]
2.3
0.1
[ 0.1 -9.1]

```

▼ Two - Dimensional Arrays

```

grades = np.array([93, 95, 84, 100, 99, 87])
print(grades)
grades = grades.reshape(3,2)
# 3 x 2 행렬로 만들어 달라는 의미이다.
print(grades)

```

```

print(grades[1, 0])

```

```
print(grades[1, 0])  
# 1번째 행 0번째 열 출력.  
print(grades[-2, 0])  
# -2는 뒤에서 2번째 즉 1번째 행 0번째 열 출력.
```

```
[ 93  95  84 100  99  87]  
[[ 93  95]  
 [ 84 100]  
 [ 99  87]]  
84  
84
```