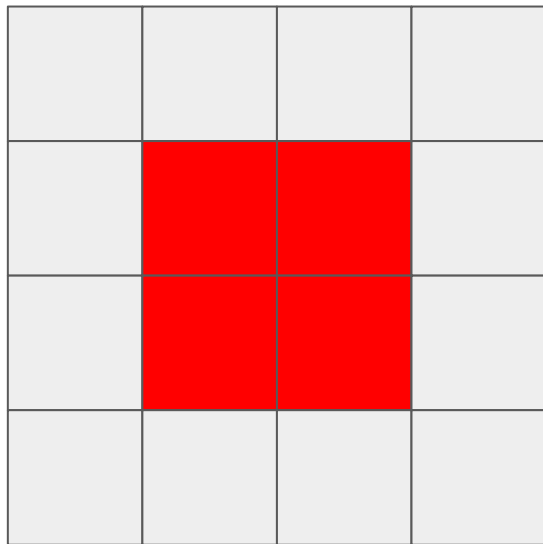


5강

이미지는 어떻게
표현할까?

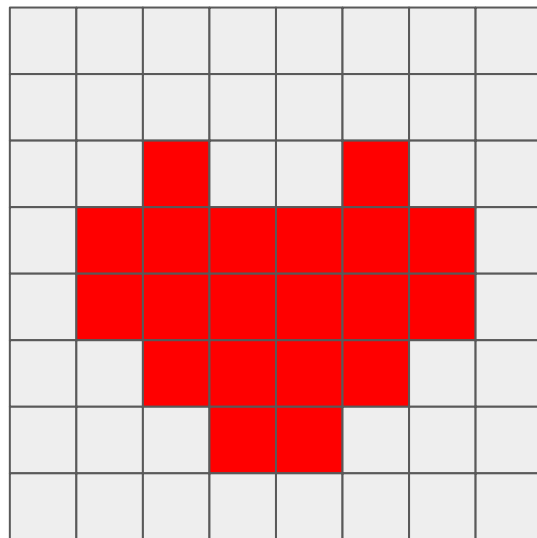
이미지는 어떻게 표현될까?

해상도 : 4 x 4



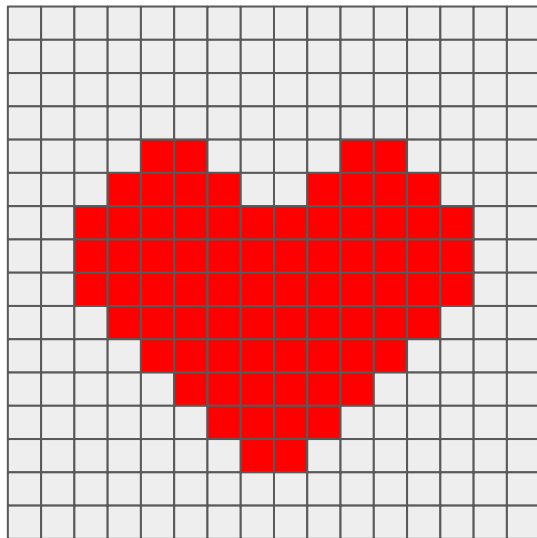
이미지는 어떻게 표현될까?

해상도 : 8 x 8



이미지는 어떻게 표현될까?

해상도 : 16 x 16



이미지는 어떻게 표현될까?

픽셀의 표현 방식



이미지는 어떻게 표현될까?

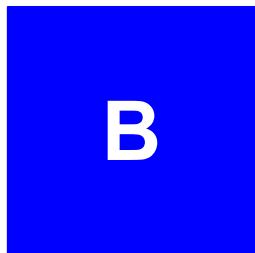
R G B



(255, 0, 0)



(0, 255, 0)



(0, 0, 255)

이미지는 어떻게 표현될까?

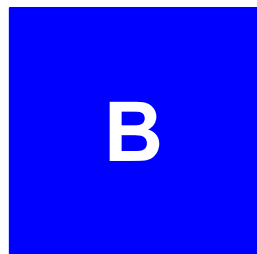


R G B

(255, 0, 0)



(0, 255, 0)

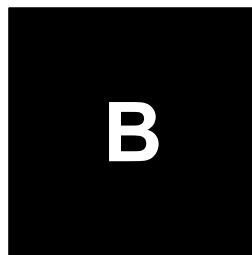


(0, 0, 255)



R G B

(?, ?, ?)



(?, ?, ?)

이미지는 어떻게 표현될까?

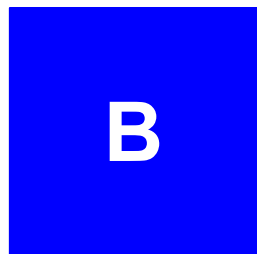


R G B

(255, 0, 0)



(0, 255, 0)

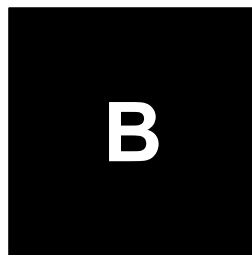


(0, 0, 255)



R G B

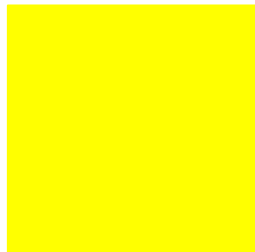
(255, 255, 255)



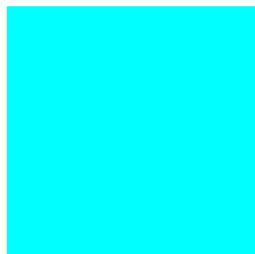
(0, 0, 0)

이미지는 어떻게 표현될까?

R G B



(255, 255, 0)



(0, 255, 255)

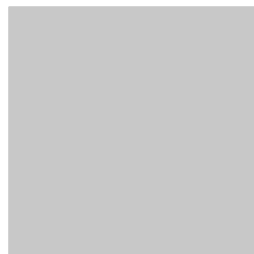


(255, 0, 255)

R G B



(100, 100, 100)



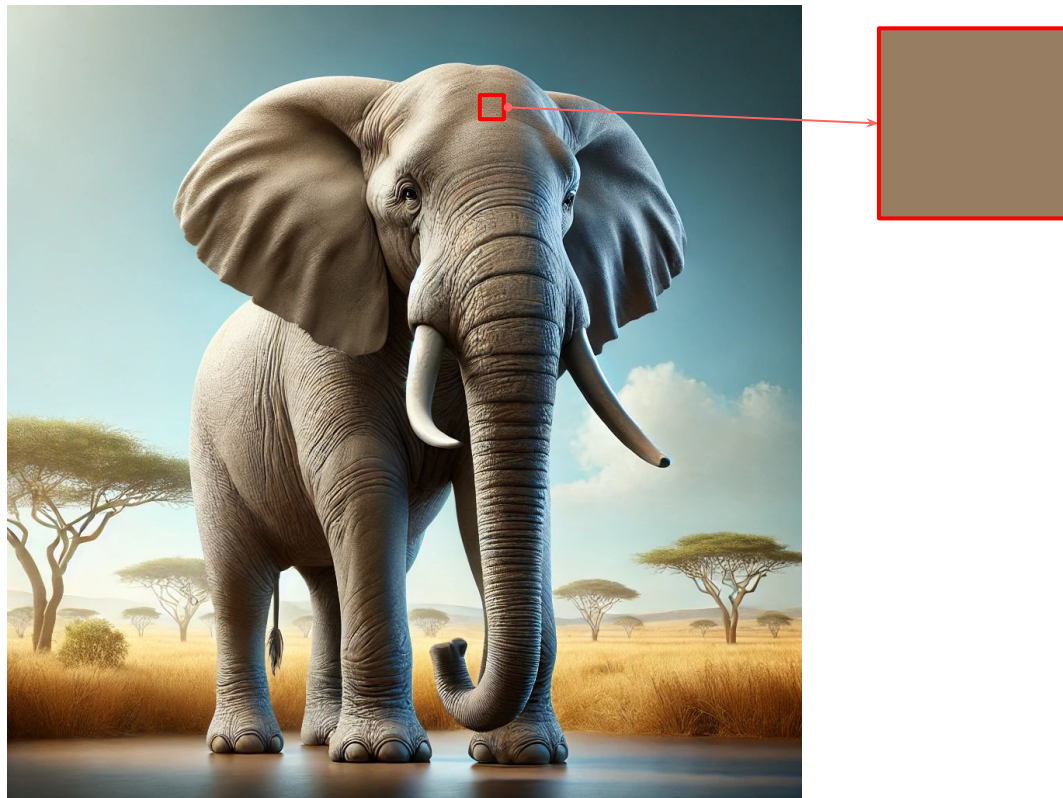
(200, 200, 200)



(255, 100, 100)

이미지는 어떻게 표현될까?

해상도 : 1024 x 1024



Python에서는 이미지를 어떻게
처리할까?

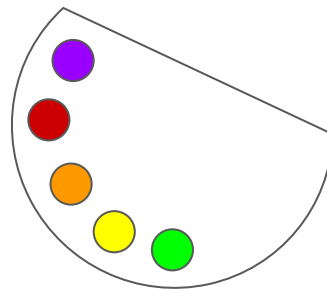
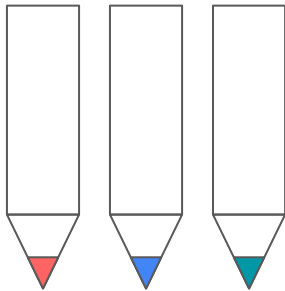
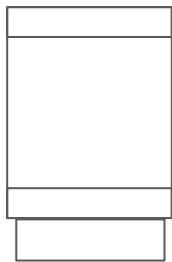
라이브러리란?

라이브러리란 ?

필요한 도구가 잘 정리된 도구
상자

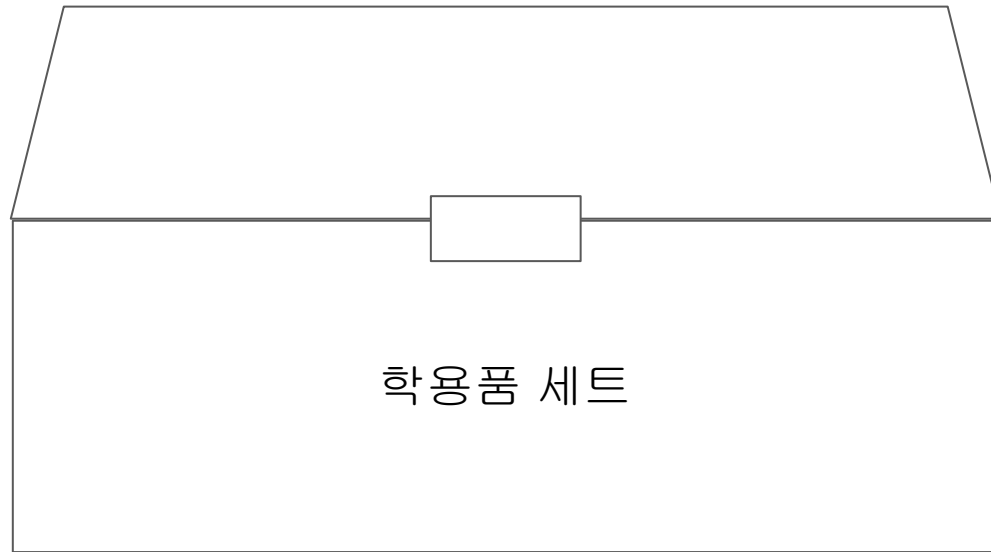
라이브러리란 ?

미술 시간



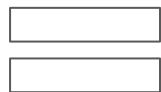
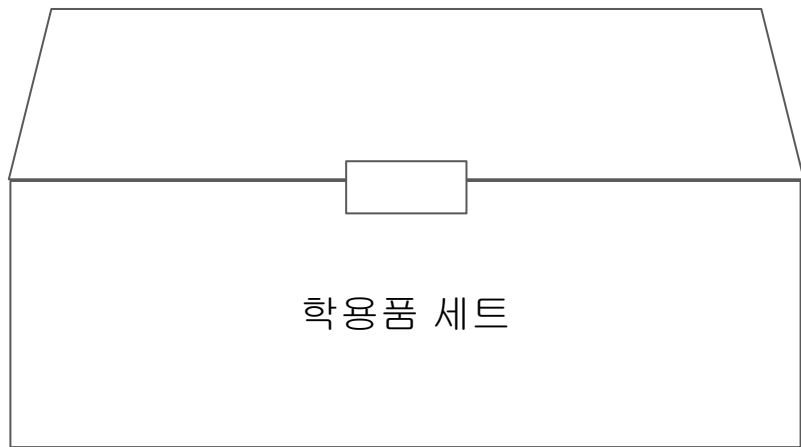
라이브러리란 ?

미술 시간



라이브러리란 ?

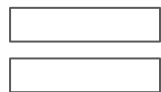
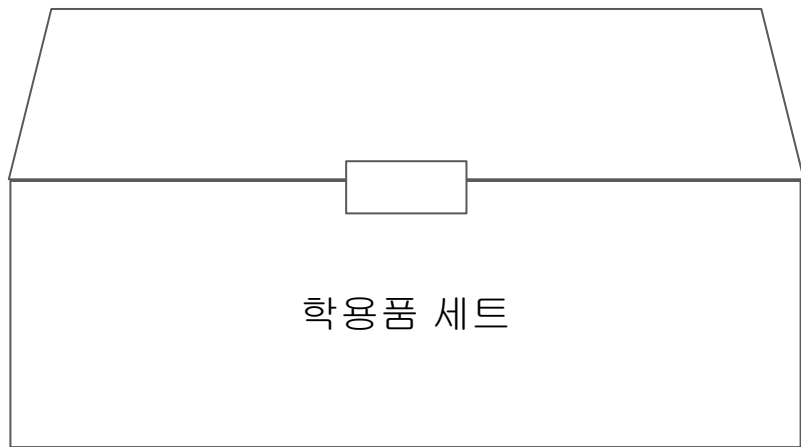
미술 시간



필요한 도구가 잘 정리된 도구
상자

라이브러리란 ?

미술 시간



라이브러리

Python에서는 이미지를 어떻게 처리할까?

Pillow

OpenCV

Matplotlib



Python에서는 이미지를 어떻게 처리할까?

Pillow

OpenCV

Matplotlib

Pillow 라이브러리는 이미지를 다루는 데 유용한
‘도구(코드)’가 많이 들어있어서,

- 이미지 파일 열기/저장하기
- 이미지 크기 조절/돌리기/색 바꾸기 등을
간단한 함수로 손쉽게 할 수 있습니다.

Python에서는 이미지를 어떻게 처리할까?

Pillow

OpenCV

Matplotlib

```
1 # Pillow 방식으로 업로드된 파일 읽기
2 from PIL import Image
3 image_path = 'img1.webp'
4 img = Image.open(image_path)
5 img
```

결과



Python에서는 이미지를 어떻게 처리할까?

Pillow

OpenCV

Matplotlib

```
1 # 이미지 파일명
2 print("이미지 파일명:", img.filename)
3 # → 우리가 불러온 파일 경로가 표시됨
4
5 # 이미지 형식(format)
6 print("이미지 형식:", img.format)
7 # → JPEG, PNG, WEBP 등 실제 이미지가 어떤 포맷인지 알려줌
8
9 # 이미지 크기(size) : (너비, 높이) 형태의 튜플
10 print("이미지 크기:", img.size)
11 # → 예: (640, 480)
12 #   => width=640, height=480
13
14 # 이미지 너비(width)
15 print("이미지 너비:", img.width)
16 # → 예: 640
17
18 # 이미지 높이(height)
19 print("이미지 높이:", img.height)
20 # → 예: 480
21
22 # 이미지의 색상 모드(mode)
23 print("이미지 모드:", img.mode)
24 # → 예: 'RGB', 'L'(흑백), 'RGBA'(투명 채널 포함), 'CMYK' 등
```

결과

이미지 파일명: /content/img1.png

이미지 형식: WEBP

이미지 크기: (1792, 1024)

이미지 너비: 1792

이미지 높이: 1024

이미지 모드: RGB

Python에서는 이미지를 어떻게 처리할까?

Pillow

OpenCV

Matplotlib

OpenCV 라이브러리는 **컴퓨터 비전(영상 분석)** 분야의
‘도구(코드)’가 많이 들어있어서,

- 이미지/영상 속 사물 찾기(객체 인식)
- 엣지(윤곽) 검출
- 얼굴 인식 등을 해보려 할 때, 이미 만들어진 함수를 불러와 쓰기만 하면 됩니다.

Python에서는 이미지를 어떻게 처리할까?

Pillow

OpenCV

Matplotlib

```
1 import cv2
2
3 # OpenCV 방식으로 업로드된 파일 읽기
4 img_cv = cv2.imread(image_path) # OpenCV 함수 imread로 이미지를 불러옵니다.
5
6 # img_cv에는 사진이 '넘파이(NumPy) 배열' 형태로 저장돼요.
7 img_cv
```

결과

```
array([[202, 226, 208],
       [199, 223, 206],
       [200, 225, 207],
       ...,
       [ 65,  55,  59],
       [ 73,  63,  67],
       [ 62,  51,  55]],

      [[206, 230, 213],
       [195, 219, 201],
       [200, 225, 207],
       ...,
```


Python에서는 이미지를 어떻게 처리할까?

Pillow
OpenCV
Matplotlib

```
1 # shape를 출력해보면 (세로길이, 가로길이, 색상채널 수)가 나옵니다.  
2  
3 # 이미지 크기(높이, 너비, 채널 수) 확인  
4 print("img_cv.shape:", img_cv.shape)  
5  
6 # 이미지 높이(세로길이)  
7 print("Height (img_cv.shape[0]):", img_cv.shape[0])  
8  
9 # 이미지 너비(가로길이)  
10 print("Width (img_cv.shape[1]):", img_cv.shape[1])  
11  
12 # 색상 채널 개수 (흑백이면 1, 컬러면 3 등)  
13 print("Color Channels (img_cv.shape[2]):", img_cv.shape[2])  
14
```

결과

```
img_cv.shape: (1024, 1792, 3)  
Height (img_cv.shape[0]): 1024  
Width (img_cv.shape[1]): 1792  
Color Channels (img_cv.shape[2]): 3
```

Python에서는 이미지를 어떻게 처리할까?

Pillow

OpenCV

Matplotlib

Matplotlib 라이브러리는 데이터/이미지 시각화와 관련된
‘도구(코드)’를 담고 있어서,

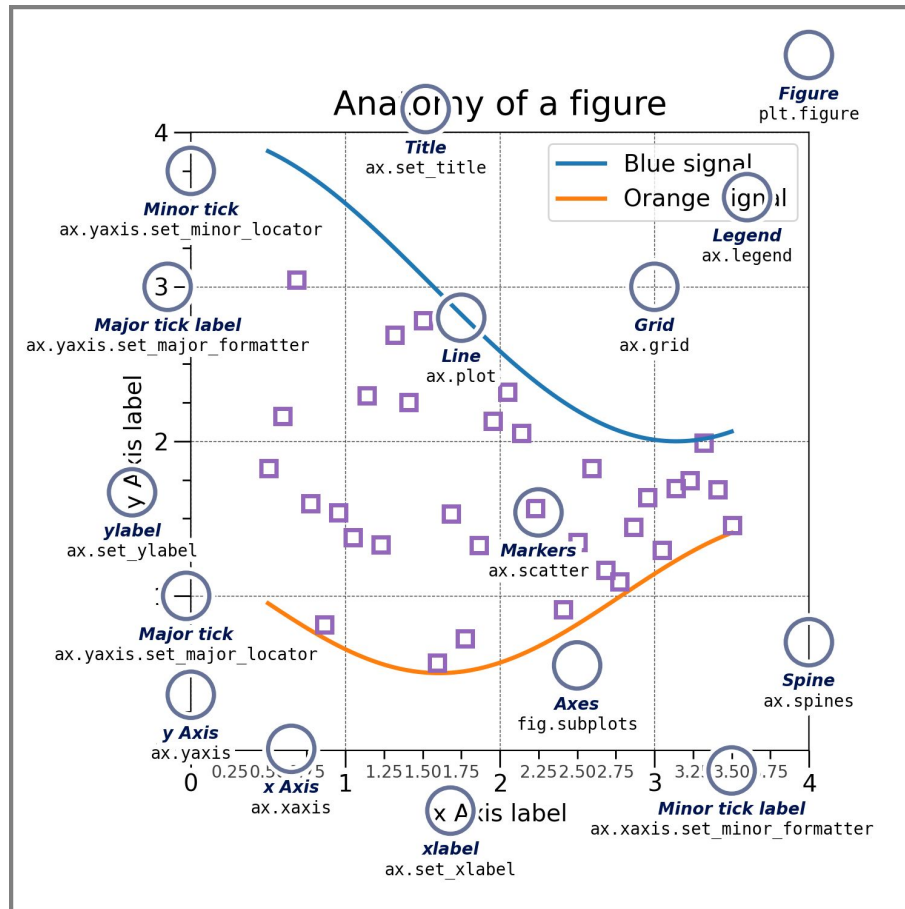
- 그래프(막대그래프, 꺾은선그래프 등) 그리기
- 이미지를 화면에 표시하기 등을 간단한 명령으로 그릴 수 있습니다.

Python에서는 이미지를 어떻게 처리할까?

Pillow

OpenCV

Matplotlib



Python에서는 이미지를 어떻게 처리할까?

Pillow
OpenCV
Matplotlib

```
1 import matplotlib.pyplot as plt
2
3 # 첫 번째로 업로드된 파일 읽기
4 for filename in uploaded.keys():
5     # 이미지를 열기
6     img = Image.open(io.BytesIO(uploaded[filename]))
7     # img = Image.open(image_path)
8     break
9
10 # 원본 이미지 시각화
11 plt.figure(figsize=(30, 20))
12 plt.subplot(1, 4, 1)
13 plt.title(f"Original {img.size}")
14 plt.imshow(img)
15 plt.axis('off')
16
```

결과

Original (1792, 1024)



Python에서는 이미지를 어떻게 처리할까?

이미지 Transformation

Grayscale



일반적으로 **ITU-R BT.601** 규격에 따른 **가중치 합**을 이용해 **그레이스케일 (흑백)** 이미지를 만듭니다.

$$\text{Gray} = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

공식에 따라 각 픽셀에 대해 **가중평균**을 구한 뒤,
그 결과를 새로운 픽셀 값(**0~255** 사이)로 삼아 **흑백 모드**로 변환합니다.

Python에서는 이미지를 어떻게 처리할까?

이미지 Transformation

Rotated 45°



`img`라는 이름의 **이미지**가 있다고 할 때, `img.rotate(45)`라는 명령어를 사용하면,

그 이미지를 **반시계 방향으로 45도** 회전시킬 수 있습니다.

Python에서는 이미지를 어떻게 처리할까?

이미지 Transformation

Resized (1/2)



원본 크기와 다른 해상도로 키우거나 줄일 때, 픽셀들이 합쳐지거나, 인접 픽셀 값을 바탕으로 보정(interpolation)되면서 새로운 이미지가 만들어집니다.

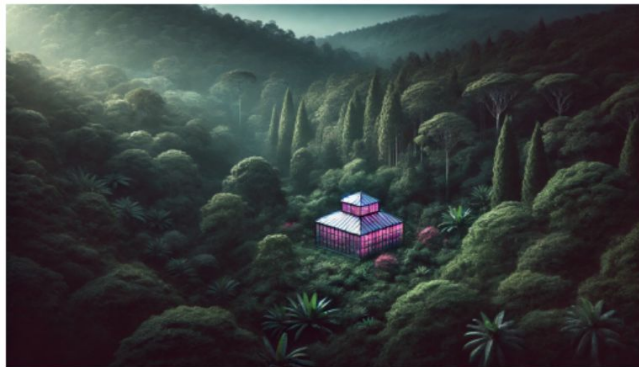
예를 들어, 원본이 1000×800 픽셀이라면, 절반 크기는 500×400 픽셀이 됩니다.

이 과정에서 어떤 보간(Interpolation) 방식을 쓰느냐에 따라 선명도나 화질이 조금씩 달라질 수 있습니다

Python에서는 이미지를 어떻게 처리할까?

이미지 해상도

Original (1792, 1024)



50% Resolution (896, 512)



25% Resolution (448, 256)



10% Resolution (179, 102)



Python에서는 이미지를 어떻게 처리할까?

이미지 Transformation

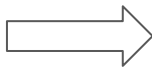
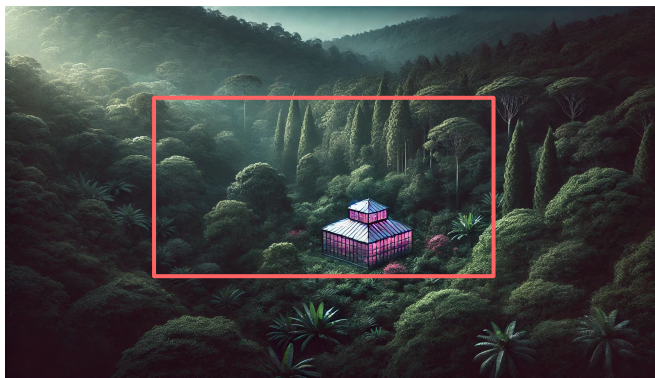
Flipped (Horizontal)



이미지를 **왼쪽↔오른쪽**으로 뒤집는 모드(좌우 반전)입니다.
마치 세로로 서 있는 거울을 중앙에 놓고, 그 거울에 비친
모습을 보는 것처럼 뒤집힙니다.

Python에서는 이미지를 어떻게 처리할까?

이미지 Transformation



Cropped



잘라낼 영역(직사각형)의 **왼쪽, 위쪽, 오른쪽, 아래쪽** 좌표를 지정하고 원본 이미지에서 해당 영역의 데이터만 잘라냅니다.

Project1. 이미지를 연속적으로 crop해서 비디오 만들기



Project1. 이미지를 연속적으로 crop해서 비디오 만들기



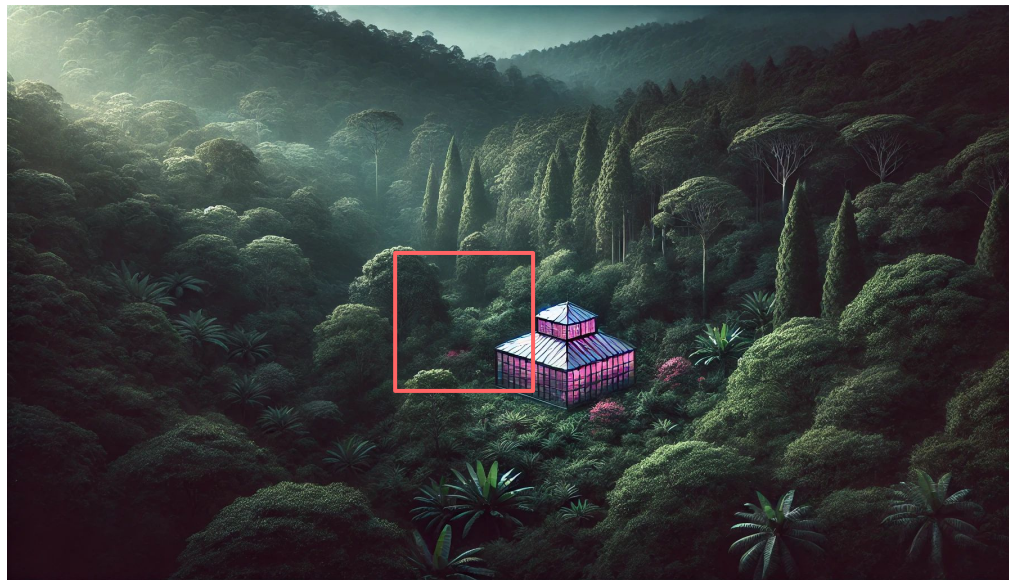
Project1. 이미지를 연속적으로 crop해서 비디오 만들기



Project1. 이미지를 연속적으로 crop해서 비디오 만들기



Project1. 이미지를 연속적으로 crop해서 비디오 만들기



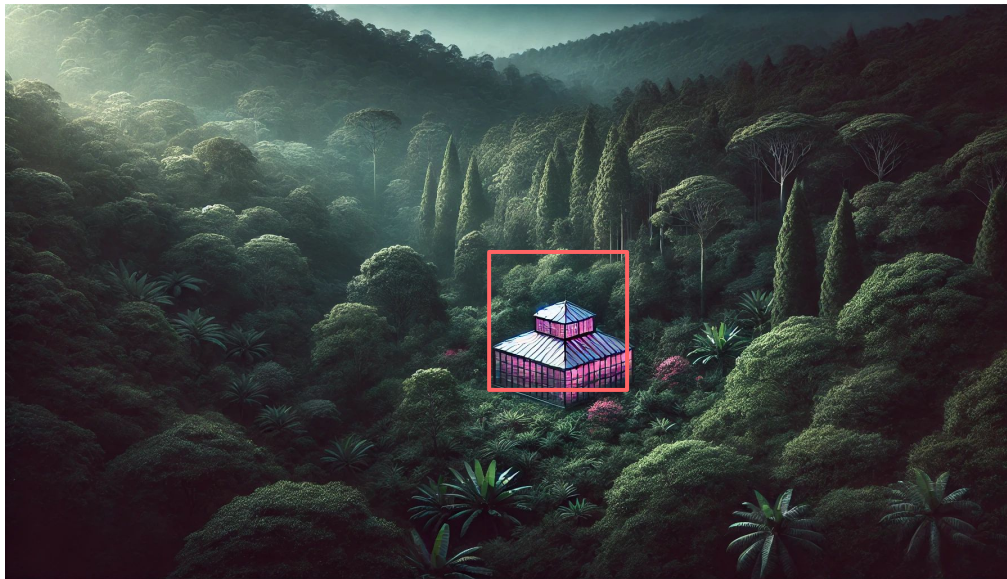
Project1. 이미지를 연속적으로 crop해서 비디오 만들기



Project1. 이미지를 연속적으로 crop해서 비디오 만들기



Project1. 이미지를 연속적으로 crop해서 비디오 만들기



Project1. 이미지를 연속적으로 crop해서 비디오 만들기



실습