

Numpy_01. 객체 Save & Load

CODE

```
1  import numpy as np
2
3  # 단일 객체 저장 및 불러오기
4  array = np.arange(0, 10)
5  print(array)
6
7  # .npy 파일에다가 저장하기
8  np.save("./22.12.05_d44_dataset/data/save.npy", array)
9
10 # 불러오기
11 result = np.load("./22.12.05_d44_dataset/data/save.npy")
12 print("result: ", result)
13 |
```

RESULT

```
[0 1 2 3 4 5 6 7 8 9]
result: [0 1 2 3 4 5 6 7 8 9]
PS C:\Users\user\Documents\Microsoft-AI-School> □
```

Numpy_02. 복수 객체 Save & Load

CODE

```
1  import numpy as np
2
3  # 복수 객체 저장을 위한 데이터 생성
4  array1 = np.arange(0, 10)
5  array2 = np.arange(0, 20)
6  print(array1, array2)
7
8  # 저장
9  np.savez("./22.12.05_d44_dataset/data/save.npz", array1=array1, array2=array2)
10
11 # 객체 불러오기
12 data = np.load("./22.12.05_d44_dataset/data/save.npz")
13 result1 = data["array1"]
14 result2 = data["array2"]
15
16 print("result1: ", result1)
17 print("result2: ", result2)
```

RESULT

```
[0 1 2 3 4 5 6 7 8 9] [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]
result1: [0 1 2 3 4 5 6 7 8 9]
result2: [ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]
PS C:\Users\user\Documents\Microsoft-AI-School> █
```

Numpy_03. 정렬 (오름차순 & 내림차순)

CODE

```
1  # Numpy
2  import numpy as np
3
4  # 1. 원소정렬
5  array = np.array([15,20,5,12,7])
6  np.save("./22.12.05_d44_dataset/data/array.npy", array) # .npy 파일에다가 저장
7
8  array_data = np.load("./22.12.05_d44_dataset/data/array.npy") # 불러오기
9  array_data.sort() # 오름차순
10 print("오름차순: ", array_data)
11 print("내림차순: ", array_data[::-1]) # 내림차순
```

RESULT

```
오름차순: [ 5  7 12 15 20]
내림차순: [20 15 12  7  5]
PS C:\Users\user\Documents\Microsoft-AI-School> 
```

Numpy_04. 정렬 (열 & 행)

CODE

```
1 import numpy as np
2 array = np.array([[5,2,7,6], [2,3,10,15]])
3 print("각 열을 기준으로 정렬 전\n", array)
4
5 array.sort(axis=0) # 0:열, 1:행
6 print("각 열을 기준으로 정렬 후\n", array)
```

RESULT

```
각 열을 기준으로 정렬 전
[[ 5  2  7  6]
 [ 2  3 10 15]]
각 열을 기준으로 정렬 후
[[ 2  2  7  6]
 [ 5  3 10 15]]
PS C:\Users\user\Documents\Microsoft-AI-School> 
```

Numpy_05. Linspace 활용

CODE

```
1 import numpy as np
2 # linspace[시작, 끝, 사이에 데이터 수]
3 array = np.linspace(0,10,5) # 0~10 의 5개의 데이터
4 print(array)
```

RESULT

```
[ 0.  2.5  5.  7.5 10.]
PS C:\Users\user\Documents\Microsoft-AI-School> 
```

Numpy_06. Random Seed 활용

CODE

```
1 import numpy as np
2 # 실행마다 결과 동일
3 # 랜덤값이 실행될때마다 변경되지않도록 seed 값 설정
4 np.random.seed(5)
5 print(np.random.randint(0,10,(2,3))) # (2,3) 행렬, 0~10 값 랜덤
```

RESULT

```
[[3 6 6]
 [0 9 8]]
PS C:\Users\user\Documents\Microsoft-AI-School> █
```

Numpy_07. Array 복사 & 수정

CODE

```
1  import numpy as np
2  arr1 = np.arange(0,10)
3  arr2 = arr1
4  arr3 = arr1.copy()
5  arr2[0] = 99
6  print("array1: ", arr1) # arr1도 변경
7  print("array3: ", arr3) # arr3은 그대로
8
9  # numpy 중복된 원소 제거
10 array = np.array([1, 2, 1, 2, 3, 4, 3, 4, 5])
11 print("중복 처리 전: ", array)
12 # 중복 처리 전 >> [1 2 1 2 3 4 3 4 5]
13 print("중복 처리 후: ", np.unique(array))
14
15
16 # np.isin() -> 내가 찾는게 있는지 여부 각 index 위치에 true false
17 array = np.array([1, 2, 3, 4, 5, 6, 7])
18
19 iwantit = np.array([1, 2, 3, 10])
20
21 print(np.isin(array, iwantit))
22 # [ True  True  True False False False False]
```

RESULT

```
array1: [99  1  2  3  4  5  6  7  8  9]
array3: [0  1  2  3  4  5  6  7  8  9]
중복 처리 전: [1  2  1  2  3  4  3  4  5]
중복 처리 후: [1  2  3  4  5]
[ True  True  True False False False False]
PS C:\Users\user\Documents\Microsoft-AI-School> 
```

Image_Numpy_01. Image Save & Load & Compress

CODE

```
1 import numpy as np
2 import cv2
3
4 # 이미지 경로
5 x = cv2.imread("./22.12.05_d44_dataset/data/cat.jpg", 0) # 흑백
6 y = cv2.imread("./22.12.05_d44_dataset/data/cat.jpg", 1) # 컬러
7
8 # 여러개 파일 저장 .npz
9 np.savez("./22.12.05_d44_dataset/data/image.npz", array1= x, array2= y)
10
11 # 압축 방법
12 np.savez_compressed("./22.12.05_d44_dataset/data/image_compressed.npz", array1= x, array2= y)
13
14 # npz 데이터 불러오기
15 data = np.load("./22.12.05_d44_dataset/data/image_compressed.npz")
16
17 result1 = data["array1"]
18 result2 = data["array2"]
19
20 cv2.imshow("result1", result1)
21 cv2.waitKey(0)
```

RESULT



Image_01. Image의 타입 & 사이즈 정보 확인

CODE

```
1  import cv2
2
3  img_path = "./22.12.05_d44_dataset/data/cat.jpg"
4  img = cv2.imread(img_path)
5
6  h, w , _ = img.shape
7
8  print("이미지 타입: ", type(img))
9  print(f"이미지 높이는: {h}, 이미지 넓이는: {w}")
```

RESULT

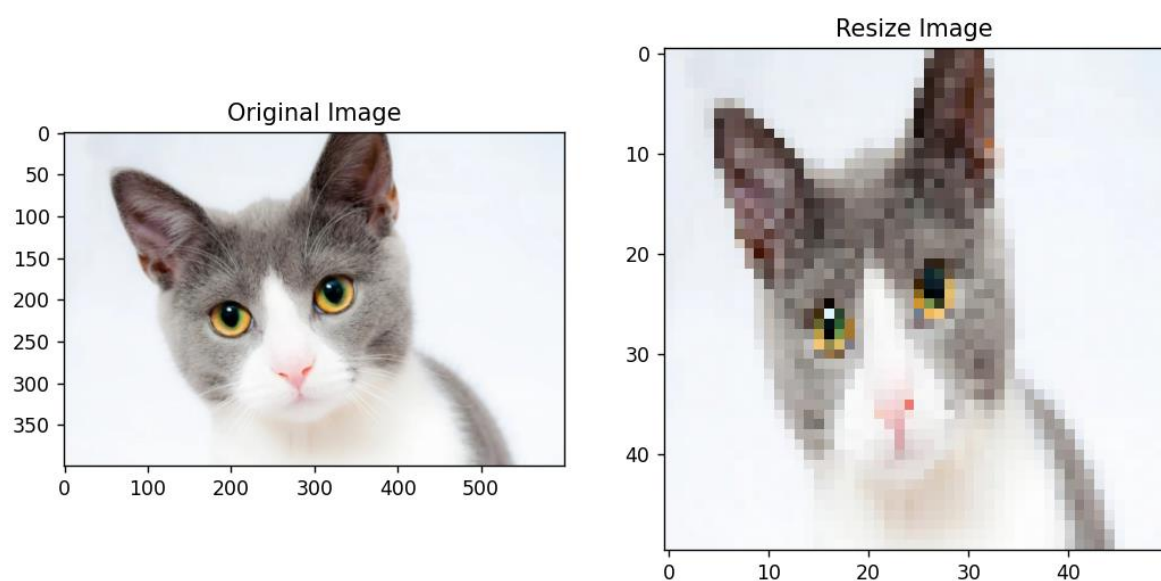
```
이미지 타입: <class 'numpy.ndarray'>
이미지 높이는: 399, 이미지 넓이는: 600
```

Image_02. RGB 변환, 사이즈 변환

CODE

```
1  import cv2
2  import matplotlib.pyplot as plt
3  image_path = "./22.12.05_d44_dataset/data/cat.jpg" # 경로
4
5  # 이미지 읽기
6  image = cv2.imread(image_path)
7
8  # RGB 변환
9  image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
10
11 # 사이즈 변환
12 image_50_by_50 = cv2.resize(image, (50,50))
13
14 # 이미지 저장
15 cv2.imwrite("./22.12.05_d44_dataset/data/cat_resized.jpg", image_50_by_50)
16 image_50_by_50 = cv2.imread("./22.12.05_d44_dataset/data/cat_resized.jpg")
17
18 flg, ax = plt.subplots(1, 2, figsize= (10, 5))
19 ax[0].imshow(image_rgb)
20 ax[0].set_title("Original Image")
21 ax[1].imshow(image_50_by_50)
22 ax[1].set_title("Resize Image")
23 plt.show()
```

RESULT

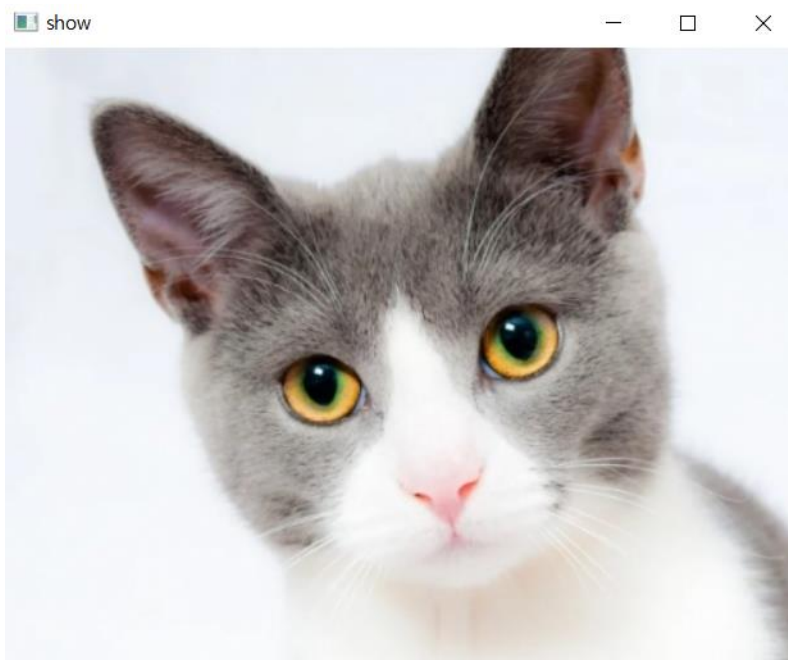


Image_03. Image 크롭

CODE

```
1  import cv2
2
3  def image_show(image):
4      cv2.imshow("show", image)
5      cv2.waitKey(0)
6
7  image_path = "./22.12.05_d44_dataset/data/cat.jpg"
8
9  # 이미지 읽기
10 image = cv2.imread(image_path)
11
12 # 이미지 크롭 [시작 : 끝 : 단계]
13 image_crop = image[10: , :500]
14
15 # 저장코드 추가 png 파일 저장
16 image_show(image_crop)
17 cv2.imwrite("./22.12.05_d44_dataset/data/cat_crop.png", image_crop)
18
```

RESULT



Image_04. Blur 처리

CODE

```
1  import cv2
2  # from image_03 import image_show
3
4  def imshow(image):
5      cv2.imshow("show", image)
6      cv2.waitKey(0)
7
8  image_path = "./22.12.05_d44_dataset/data/cat.jpg"
9
10 # 이미지 읽기
11 image = cv2.imread(image_path)
12
13 # 이미지 Blur
14 image_blury = cv2.blur(image, (5,5))
15 imshow(image_blury)
16
```

RESULT

