

기초데이터과학 (01분반)

Programming assignment 01

1. numpy를 사용하여 정수값을 갖는 4x4 배열에서 최대값의 위치(좌표)와 최소값의 위치(좌표)를 출력하는 코드를 작성하십시오.

```
In [35]: import numpy as np

# 4x4 배열 생성
arr = np.array([[10, 15, 30, 40],
                [50, 60, 70, 80],
                [90, 95, 85, 75],
                [65, 55, 45, 35]])

# 최대값, 최소값의 인덱스 - numpy의 argmax, argmin 함수 사용
i_max = np.argmax(arr)
i_min = np.argmin(arr)

# argmax, argmin 함수의 결과로 나온 인덱스를 4x4 배열에서의 위치로 변환 - numpy의
# unravel_index 함수 사용법은 검색하여 각자 찾아서 해볼 것
max_pos = np.unravel_index(i_max, arr.shape)
min_pos = np.unravel_index(i_min, arr.shape)

print("최대값 인덱스:", i_max)
print("4x4 배열에서의 최대값 위치:", max_pos)

print("최소값 인덱스:", i_min)
print("4x4 배열에서의 최소값 위치:", min_pos)

##### 아래의 출력처럼 결과가 나와야 함#####
```

```
최대값 인덱스: 9
4x4 배열에서의 최대값 위치: (2, 1)
최소값 인덱스: 0
4x4 배열에서의 최소값 위치: (0, 0)
```

2. 1부터 50까지의 숫자를 가진 numpy 배열을 생성하고, 그 배열에서 짝수만 추출하여 출력하는 코드를 작성하십시오.

```
In [34]: import numpy as np

# 1부터 50까지의 숫자를 가진 numpy 배열 생성
arr = np.arange(1, 51)

# 위 배열에서 짝수인 것은 True, 홀수인 것은 False 값을 갖도록 하여 출력
print(arr % 2 == 0)

# 위 배열에서 짝수인 것의 개수 출력
print((arr % 2 == 0).sum())
```

```
# 짝수만 추출 - 배열에 조건식을 사용하는 방법으로 짝수만 포함하는 배열 생성
even_numbers = arr[arr % 2 == 0]

# 짝수만 추출한 결과 출력
print(even_numbers)

##### 아래의 출력처럼 결과가 나와야 함#####
```

```
[False True False True False True False True False True False True
 False True False True False True False True False True False True
 False True False True False True False True False True False True
 False True]
25
[ 2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48
 50]
```

3. 1부터 10까지 정수 중 임의의 정수를 원소로 갖는 3x3 numpy 배열을 생성하고 배열 내에서 값이 1인 원소의 위치를 찾는 코드를 작성하시오.

```
In [33]: import numpy as np

# 1부터 10까지 임의의 정수값을 갖는 3x3 numpy 배열 생성 - random.randint 함수 사용
# numpy의 random.randint 함수 사용법은 검색하여 각자 찾아서 해볼 것
arr_2d = np.random.randint(1, 10, size=(3, 3))

# 위에서 생성한 배열 출력
print(arr_2d)

# 값이 1인 원소의 위치 찾기 - argwhere 함수 사용
# numpy의 argwhere 함수 사용법은 검색하여 각자 찾아서 해볼 것
indices_of_one = np.argwhere(arr_2d == 1)

# 결과 출력
print('값이 1인 원소의 위치:', indices_of_one)

##### 랜덤 값을 갖는 배열을 생성하기 때문에 아래의 출력과 동일하게 나오지 않을 수
```

```
[[4 8 6]
 [7 8 1]
 [5 3 4]]
값이 1인 원소의 위치: [[1 2]]
```

4. 평균 0, 표준편차 1인 정규분포를 따르는 10,000개의 숫자를 포함하는 배열을 생성하고, 그 배열의 평균과 표준편차를 계산하는 코드를 작성하시오.

```
In [32]: import numpy as np

# 평균 0, 표준편차 1인 정규분포로 이루어진 10,000개의 숫자를 포함하는 1차원 배열
# numpy의 random.randn 함수 또는 random.normal 함수 사용
# normal_dist_array = np.random.normal(0, 1, 10000)
normal_dist_array = np.random.randn(10000)

# 배열의 평균과 표준편차 계산
mean_value = np.mean(normal_dist_array)
std_dev_value = np.std(normal_dist_array)
```

```
# 결과 출력
print("배열의 평균:", mean_value)
print("배열의 표준편차:", std_dev_value)
```

랜덤 값을 갖는 배열을 생성하기 때문에 아래의 출력과 동일하게 나오지 않을 수

배열의 평균: -0.014092001687723774

배열의 표준편차: 1.0053163958999263

5. 1부터 10까지 임의의 정수를 원소로 갖는 4x4 배열을 랜덤하게 생성한 후, 배열 내에서 8보다 큰 원소의 개수와 3보다 작은 원소의 개수를 출력하는 코드를 작성하시오.

In [31]: `import numpy as np`

```
# 1부터 10까지 임의의 정수값을 갖는 4x4 numpy 배열 생성 - random.randint 함수 사용
arr_2d = np.random.randint(1, 10, size=(4, 4))
```

```
# 생성된 배열 출력
print(arr_2d)
```

```
# 8보다 큰 원소의 개수 출력
print("8보다 큰 원소의 개수:", (arr_2d > 8).sum())
```

```
# 3보다 작은 원소의 개수 출력
print("3보다 작은 원소의 개수:", (arr_2d < 3).sum())
```

랜덤 값을 갖는 배열을 생성하기 때문에 아래의 출력과 동일하게 나오지 않을 수

```
[[9 1 3 8]
```

```
 [9 2 5 2]
```

```
 [9 8 1 6]
```

```
 [1 5 3 6]]
```

8보다 큰 원소의 개수: 3

3보다 작은 원소의 개수: 5

In []: