

# Lab12. NumPy and Pandas

*CSED101 LAB*

# NumPy

# NumPy

---

- 계산 과학 분야에 이용되는 핵심 라이브러리
- 다차원 배열 객체를 연산하는 도구 제공

```
import numpy as np
```

# NumPy 생성 및 사용

```
# 1차원 배열 생성  
# [0 1 2 3 4 5]  
a = np.array([0, 1, 2, 3, 4, 5])  
print(a)  
  
print(type(a)) # numpy.ndarray
```

```
# 2차원 배열 생성  
a = np.array([[0, 1, 2], [3, 4, 5]])  
print(a)
```

# 넘파이 함수 실습  
np.zeros()  
np.ones()  
np.arange()  
np.transpose()

# 인덱싱/슬라이싱 실습  
# 아래와 같은 구조의 배열 생성 후 실습  
# [[ 1 2 3 4 5]  
# [ 6 7 8 9 10]  
# [11 12 13 14 15]]  
  
# Lab12\_numpy.ipynb 에 있는 문제로 실습

# 실습 1

- 3이상의 정수 n을 입력 받아 아래와 같이 경계값은 1이고, 나머지는 0을 원소로 가지는  $n \times n$  행렬을 생성하시오.

실행 예시)

3 이상의 정수 입력: 5

```
[[1. 1. 1. 1. 1.]  
 [1. 1. 1. 1. 1.]  
 [1. 1. 1. 1. 1.]  
 [1. 1. 1. 1. 1.]  
 [1. 1. 1. 1. 1.]]
```

```
[[1. 1. 1. 1. 1.]  
 [1. 0. 0. 0. 1.]  
 [1. 0. 0. 0. 1.]  
 [1. 0. 0. 0. 1.]  
 [1. 1. 1. 1. 1.]]
```

```
import numpy as np  
  
n = int(input("3 이상의 정수 입력: "))  
  
# n행 n열의 모든 값이 1인 배열 생성  
a = np.ones((n, n))  
  
print(a)  
print()  
# 경계값 제외한 내부의 값 0으로 설정  
a[1:-1, 1:-1] = 0  
  
print(a)
```

# Array 연산

```
A = np.array([[1, 2],  
             [3, 4]])  
  
B = np.array([[5, 6],  
             [7, 8]])  
  
# Elementwise sum  
# [[ 6  8]  
# [10 12]]  
print(A + B)  
print(np.add(A, B))  
print()  
  
# Elementwise difference  
# [[-4 -4]  
# [-4 -4]]  
print(A - B)  
print(np.subtract(A, B))  
print()
```

```
# Elementwise division  
# [[0.2 0.33333333]  
# [0.42857143 0.5 ]]  
print(A / B)  
print(np.divide(A, B))  
  
# Elementwise product  
# [[ 5 12]  
# [21 32]]  
print(A * B)  
print(np.multiply(A, B))  
print()  
  
# Dot product  
# 행렬곱은 @연산자 또는 np.matmul()을 사용할 것을 권장  
# [[19 22]  
# [43 50]]  
print(np.matmul(A, B)) # A @ B
```

- **numpy.linalg**
  - 역행렬, 행렬곱, 행렬식 등 선형대수 관련 함수
  - <https://numpy.org/doc/stable/reference/routines.linalg.html>
- **numpy.random**
  - 난수 생성 관련 함수
  - <https://numpy.org/doc/stable/reference/random/index.html>

## 실습 2

■ 연립 일차방정식 풀기 풀기 (`np.matmul()`) 또는 `@` 사용할 것!)

■ `def linear_system_solver(ndarr, v):`

- `ndarr`: 연립방정식의 왼쪽을 표현하는 정사각행렬(크기  $2 \times 2$  이상)

- `v`: 연립방정식의 오른쪽을 표현하는 1차원 `ndarray`

- `return` 값: 각 미지수의 값

- $\begin{cases} a + b = 15 \\ 2a + b = 25 \end{cases}$  는 다음과 같이 표현 가능

- $$\begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 15 \\ 25 \end{bmatrix}$$

- $$\therefore \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 15 \\ 25 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 15 \\ 25 \end{bmatrix} = \begin{bmatrix} 10 \\ 5 \end{bmatrix}$$

- 해가 없거나 무한한 경우에 대해서는 고려하지 않음

# Pandas

Lab12\_pandas.ipynb 자료를 이용하여 실습을 진행합니다.