<Numpy 복습 및 과제>

1. 모듈 선언하기

import numpy as np

2. 배열 만드는 여러 가지 방법 (

1) List를 넘파이 배열로 변환 (array 함수 사용)

-> **np.array([1,2,3])**

-> **np.array([[1,2,3],[4,5,6],[7,8,9]])** -- 2차원 list

2) Tuple을 넘파이 배열로 변환 (array 함수 사용)

-> **np.array((1,2,3))**

3) Numpy 자체 함수 사용해서 배열 만들기

① **zeros()** : 0으로 채워진 배열을 반환

-> x = np.zeros((3,2))

② **ones()** : 1로 채워진 배열을 반환

-> x = np.ones((3,2))

③ **arange(start, end, step)** : range() 함수와 유사한 함수

start: 배열의 시작점(생략가능)

end: 생성된 배열의 끝점

step: 배열이 증가하는 크기)

-> x = np.arange(1, 10, 2) : [1 3 5 7 9]

*** 다차원 배열 만들기!

2차원: np.arange(1,10).reshape(3,3)

3차원: np.arange(1, 13).reshape(3, 2, 2)

(단, reshape 뒤 숫자 곱햇을 때 개수가 맞아야 변형가능)

3. 행렬 연산하기

1) 각 원소들끼리 계산 (덧셈, 뺄셈, 곱셈, 나눗셈)

ex) data = np.arange(1, 10).reshape(3,3) -- 2차원 배열 생성

① 덧셈: **data + data** (행렬의 원소끼리 덧셈)

② 뺄셈: **data - 1 (모든 원소에 1을 뺌)**

③ 곱셈: **data * 10 (모든 원소에 10을 곱함)**

④ 나눗셈: **data / data** (행렬의 원소끼리 나누기)

심화) 행렬의 곱: np.dot(data, data) 또는 data@data

과제1) 행렬 원소의 곱셈을 파이썬 리스트로 코딩해보자.
1부터 10000까지 원소를 만든 뒤 원소에 2를 곱한 결과를 list 로 구현해보자.

(for문 사용)

```
arr = np.arange(10000)
list = arr.tolist()  //넘파이 배열을 리스트로 바꿔주는 함수

print(arr * 2)
```