(#%% => 구역을 나눠서 실행시킬수 있게함.)

Date. No.

<10주차> 2015137 안원영

• 행렬과 배열.

[[1 2 3] [4 5 6]] [[7] [8] [9]] 행렬곱(내적) => [[1×7+2×8+3×9] [4×7+5×8+6×9]]

(2×3) (3×1) → 일치해야 곱할수 있음. (2×1) → 2행 1열

만들어지는 결과에 영향.

→ ① numpy로 만들기

```
lta = [[1,2,3], [4,5,6]]  => 리스트는 연산하기 어려움.
ltb = [[7], [8], [9]]

npa = np.array[lta]  => 리스트 연산 어렵기 때문에 배열로 바꿔줌.
npb = np.array[ltb]

print(np.dot(npa, npb))
```

→ 내적 구해줌.

• 리스트안의 type 구분하기.

```
def isNumber(v):  => 숫자인지 아닌지 구분하는것
    if type(v) == int or type(v) == float:
        return true  => 숫자면 true 리턴.
    elif type(v) == list or type(v) == tuple:
        lt = [isNumber(k) for k in v]  → 타입이 리스트나 튜플인 경우
        return true if all(lt) else False  → 재귀호출
    else:
        return False  => 문자면 False
```

3항연산자 기법사용 ← return ...

all(lt) → 모든게 참인지 아닌지 물어보는 내장함수.

isNumber([1, 2]) => 리스트 형태

• 리스트에 들어있는 값이 모두 같은지 판별하는 함수

① Shift 해서 비교하기.

```
def ltall(lt):
    lt2 = lt.copy()  => 독립주소가지는 리스트 생성.
    lt.pop(0)  => 첫 원소 삭제 (1,2,1)
    lt2.pop()  => 마지막 원소 삭제 (1,1,2)
    return lt == lt2  => 모두 같다면 true.

print(ltall([1,1,2,1]))
```

② 좌우 반전해서 비교 => 단점: 가운데를 기점으로 반전되서 (홀수개의 경우) 다른값이 가운데에 있다면 결과가 이상하게 나옴.

```
def ltall2(lt):
    if len(lt)%2 != 0:  → 짝수로 만들기 위해.
        lt.insert(0, lt[0])  => 지정한 위치에 값 넣어주기
    lt2 = lt.copy()
    lt2.reverse()  => 좌우반전.
    return lt == lt2

print(ltall2([1,1,2,1,1]))
```

→ 아무값이나 넣어주기 배열에 있는값으로.

• 리스트의 모양(행과 열)을 출력하는 함수

```
def lt_shape(lt):  ※ lt = [[1,2,3],[4,5,6]]  → 2행 3열.
    row = len(lt)
    col_in_row = [len(lt[i]) for i in range(row)]
    if ltall(col_in_row):
        col = col_in_row[0]
    return row, col
```

→ 각 행별로 열길이 다를수 있기 때문에 모든 행의 열 길이를 가지는 리스트만듦