

**Linear algebra codes**

**Pythonic Code**

**최성철 교수**  
**Director of TEAMLAB**

01100  
00110



# Vector representation of python

- Vector를 파이썬으로 표시하는 다양한 방법 존재

```
vector_a = [1, 2, 10] # List로 표현했을 경우  
vector_b = (1, 2, 10) # Tuple로 표현했을 경우  
vector_c = {'x': 1, 'y': 1, 'z': 10} # dict 표현했을 경우  
  
print(vector_a, vector_b, vector_c)
```

- 최선의 방법은 없음
- 값의 변경 유무, 속성값 유무에 따라 선택할 수 있음
- 본 수업에서는 기본적으로 list로 vector 연산 실시

# Vector의 계산

```
u = [2, 2]
```

```
v = [2, 3]
```

```
z = [3, 5]
```

```
result = []
```

```
for i in range(len(u)):
```

```
    result.append(u[i] + v[i] + z[i])
```

```
print(result)
```

**[2, 2]+[2, 3]+[3, 5]=[7, 10]**

**이런 코드는 쓰면 안됨  
파이썬 답지 못하고  
안 아름다움**

# Vector의 계산

u = [2, 2]

v = [2, 3]

z = [3, 5]

$$[2, 2] + [2, 3] + [3, 5] = [7, 10]$$

```
result = [sum(t) for t in zip(u,v,z)]
```

```
print (result)
```

# Vector의 계산: Scalar-Vector product

```
u = [1, 2, 3]  2([1, 2, 3] + [4, 4, 4]) = 2[5, 6, 7] = [10, 12, 14 ]  
v = [4, 4, 4]  
alpha = 2
```

```
result = [alpha*sum(t) for t in zip(u,v)]  
print(result)
```

# Matrix representation of python

- Matrix 역시 Python으로 표시하는 다양한 방법이 존재

```
matrix_a = [[3, 6], [4, 5]] # List로 표현했을 경우  
matrix_b = [(3, 6), (4, 5)] # Tuple로 표현했을 경우  
matrix_c = {(0, 0): 3, (0, 1): 6, (1, 0): 4, (1, 1): 5} # dict 표현했을 경우
```

$\begin{bmatrix} 3 & 6 \\ 4 & 5 \end{bmatrix}$

- 특히 dict로 표현할 때는 무궁무진한 방법이 있음
- 본 수업에서는 기본적으로 two-dimensional list 형태로 표현함
- [[1번째 row], [2번째 row], [3번째 row]]

# Matrix의 계산: Matrix addition

$$C = A + B = \begin{bmatrix} 3 & 6 \\ 4 & 5 \end{bmatrix} + \begin{bmatrix} 5 & 8 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 8 & 14 \\ 10 & 12 \end{bmatrix}$$

```
matrix_a = [[3, 6], [4, 5]]  
matrix_b = [[5, 8], [6, 7]]  
result = [[sum(row) for row in zip(*t)] for t in zip(matrix_a, matrix_b)]  
print(result)
```

tuple이 ( [3,6], [5,8] )  
각각의 배열  
편

[3, 6]

[5, 8]

# Matrix의 계산: Scalar-Matrix Product

$$\alpha \times A = 4 \times \begin{bmatrix} 3 & 6 \\ 4 & 5 \end{bmatrix} = \begin{bmatrix} 12 & 24 \\ 16 & 20 \end{bmatrix}$$

```
matrix_a = [[3, 6], [4, 5]]  
alpha = 4  
result = [[alpha * element for element in t] for t in matrix_a]  
  
print(result)
```



# Matrix의 계산: Matrix Transpose

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

```
matrix_a = [[1, 2, 3], [4, 5, 6]]  
result = [ [element for element in t] for t in zip(*matrix_a) ]  
print (result)
```

# Matrix의 계산: Matrix Product

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, B = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix} \text{ 이면 } C = A \times B = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 5 & 8 \\ 5 & 6 \end{bmatrix}$$

```
matrix_a = [[1, 1, 2], [2, 1, 1]]  
matrix_b = [[1, 1], [2, 1], [1, 3]]  
result = [[sum(a * b for a, b in zip(row_a, column_b)) W  
           for column_b in zip(*matrix_b)] for row_a in matrix_a]  
  
print(result)
```



**Human knowledge belongs to the world.**