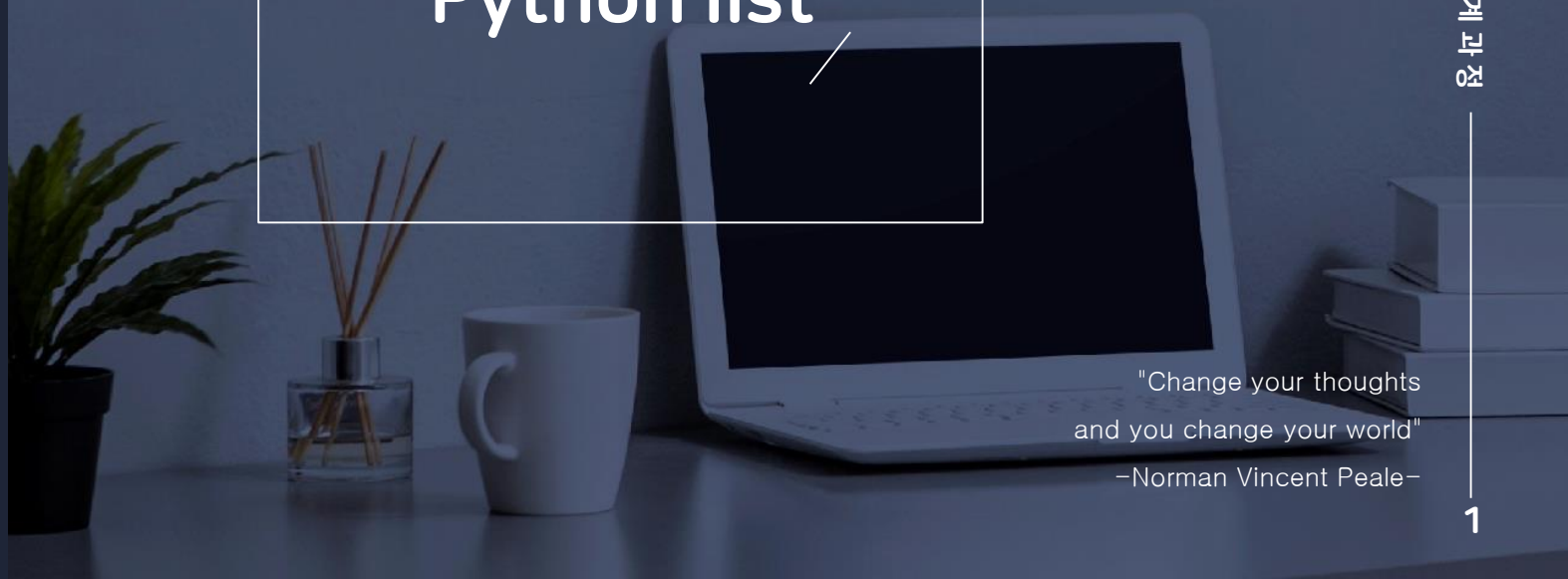
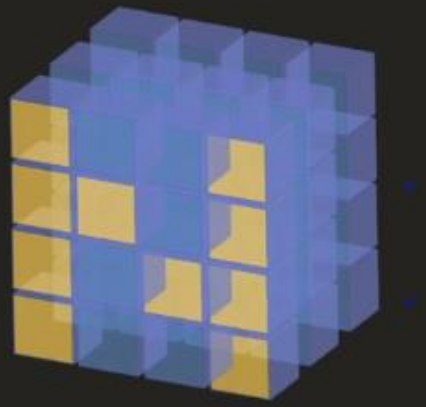
Numpy array
vs
Python list



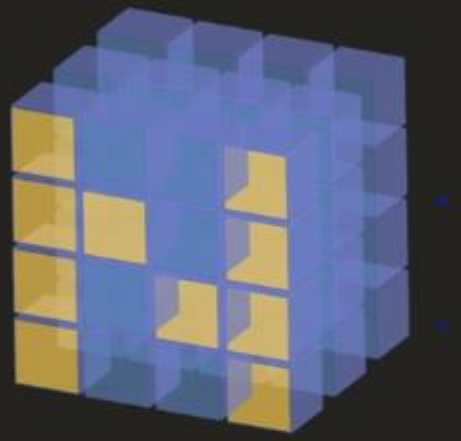
"Change your thoughts
and you change your world"
-Norman Vincent Peale-



numpy array

python list

여러 값을 보관한다는 공통점



numpy array

굳이 numpy array를 쓰는 이유는?

문법 차이

[10, 5, 3, 7, 1, 5]
+ [10, 5, 3, 7, 1, 5]

[20, 10, 6, 14, 2, 10]

[10, 5, 3, 7, 1, 5]
+ [10, 5, 3, 7, 1, 5]

[10, 5, 3, 7, 1, 5, 10, 5, 3, 7, 1, 5]

문법 차이

[10, 5, 3, 7, 1, 5]

+ 5

[15, 10, 8, 12, 6, 10]

[10, 5, 3, 7, 1, 5]

+ 5

~~빨셈~~

~~나눗셈~~

문법 차이

[10, 5, 3, 7, 1, 5]

* 3

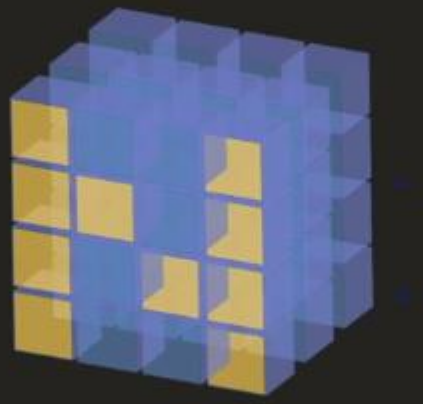
[30, 15, 9, 21, 3, 15]

[10, 5, 3, 7, 1, 5]

* 3

[10, 5, 3, 7, 1, 5, 10, 5, 3, 7, 1, 5, 10, 5, 3, 7, 1, 5]

성능 차이

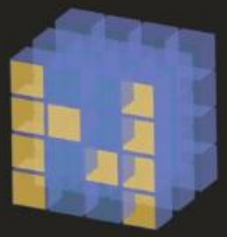


문법이 간단 + 뛰어난 성능

왜 이런 성능 차이가 있죠?

```
[17, 9, True, 'Hello', 1, 5]
```

속도 개선

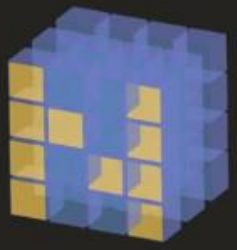


```
[17, 9, 10, 4, 1, 5]

['Hello', 'Cat', ‘Banana', ‘Pig']
```

언제 어떤 걸 써야 하나요?

값을 추가하고 제거하는 일



수치 계산이 많고 복잡할 때

행렬같은 다차원 배열의 경우

Python으로 Numpy 행렬 만들기

"Change your thoughts
and you change your world"
-Norman Vincent Peale-

실습과제

1) $A = \begin{pmatrix} 1 & 2 & 3 \\ -1 & 0 & 2 \\ 3 & 4 & 5 \\ 2 & -2 & -3 \end{pmatrix}$ 4X3 행렬 A를 numpy array로 정의해보기

2) $B = \begin{pmatrix} 0 & 2 \\ 1 & -1 \\ 3 & -4 \end{pmatrix}$ 3x2 행렬 B를 numpy array로 정의해보기

3) 정의한 행렬 A 의 2 행 2 열 원소 찾기

4) 정의한 행렬 B 의 3 행 1열 원소 찾기

(프로그래밍할 때는 인덱스를 0부터 세는 걸 잊지마세요!!)

질문 1

$$\begin{bmatrix} 1 & 2 & -1 \\ 1 & 1 & -3 \end{bmatrix} + \begin{bmatrix} 3 & 2 & -2 \\ -1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$ 다음 식에서 a_{13} 에 들어갈 값을 쓰세요.

질문 2

$$\begin{bmatrix} 1 & 2 & -1 \\ 1 & 1 & -3 \end{bmatrix} \times \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix}$$ 다음 식에서 a_{11} 에 들어갈 값을 고르세요.

질문 3

$$\begin{bmatrix} 1 & 2 & -1 \\ 1 & 1 & -3 \end{bmatrix} \times \begin{bmatrix} 3 & 2 \\ 1 & 1 \\ 2 & -1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$ 다음 식에서 a_{12} 에 들어갈 값을 쓰세요.

최댓값, 최솟값

max 메소드와 min 메소드를 사용하면 numpy array의 최댓값과 최솟값을 구할 수 있습니다.

```
import numpy as np

array1 = np.array([14, 6, 13, 21, 23, 31, 9, 5])

print(array1.max()) # 최댓값
print(array1.min()) # 최솟값
```

```
31
5
```

평균값

mean 메소드를 사용하면 numpy array의 평균값을 구할 수 있습니다.

```
import numpy as np

array1 = np.array([14, 6, 13, 21, 23, 31, 9, 5])

print(array1.mean()) # 평균값
```

```
15.25
```

위 예시에서, 총합$(14 + 6 + 13 + 21 + 23 + 31 + 9 + 5)$을 총 개수$(8)$로 나누면 15.25입니다.

중앙값

median 메소드를 사용하면 중간값을 구할 수 있는데요. 특이하게 median은 numpy array의 메소드가 아니라 numpy의 메소드입니다.

```
import numpy as np

array1 = np.array([8, 12, 9, 15, 16])
array2 = np.array([14, 6, 13, 21, 23, 31, 9, 5])

print(np.median(array1)) # 중앙값
print(np.median(array2)) # 중앙값
```

```
12.0
13.5
```

array1을 정렬하면 중앙값이 12입니다.

array2에는 짝수개의 요소가 있기 때문에 중앙값이 13과 14 두 개입니다. 둘의 평균값을 내면 13.5입니다.

특수 행렬들

특수 행렬들

전치 행렬

단위 행렬

역행렬

전치 행렬 (transposed matrix)

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 3 & 2 & 2 \end{bmatrix} \qquad A^T = \begin{bmatrix} 1 & 3 \\ 2 & 2 \\ 1 & 2 \end{bmatrix}$$

단위 행렬 (identity matrix)

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3 x 3 단위 행렬

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

단위 행렬 (identity matrix)

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$n \times 1 = n$$

선형 대수학에서는 단위 행렬이 같은 역할

$$A = \begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix}$$

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$AI = \begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix}$$

역행렬

$$10 \times \frac{1}{10} = 1 \qquad 5 \times \frac{1}{5} = 1$$

역수

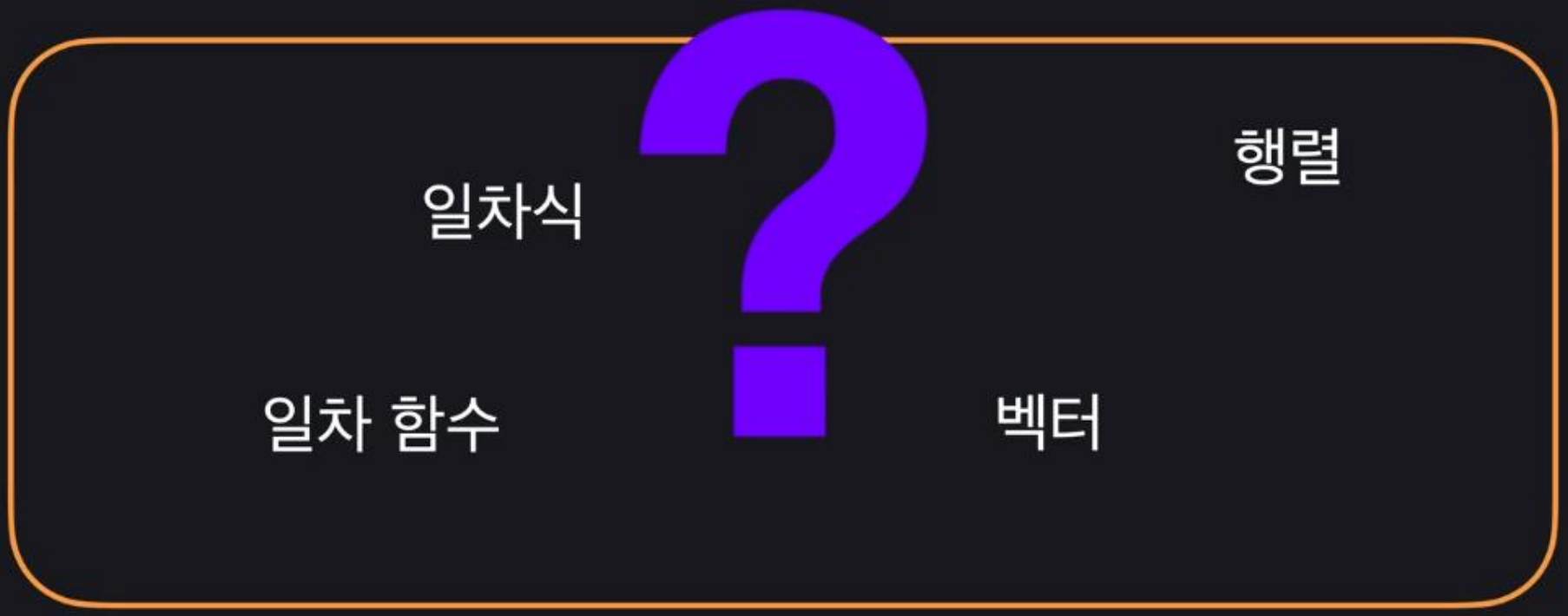
선형 대수학에서는 역행렬이 같은 역할

역행렬

$$A = \begin{bmatrix} 3 & 4 \\ 1 & 2 \end{bmatrix} \qquad A^{-1} = \begin{bmatrix} 1 & -2 \\ -\frac{1}{2} & \frac{3}{2} \end{bmatrix}$$

모든 행렬에 역행렬이 있는 건 아니다!

선형 대수학



선형 시스템

$$2x_0 - 4x_1 + x_2 = 3$$

$$3x_0 + x_1 - 6x_2 = 10$$

$$x_0 + x_1 + x_2 = 5$$

$$Ax = y$$

$$\begin{bmatrix} 2 & -4 & 1 \\ 3 & 1 & -6 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 10 \\ 5 \end{bmatrix}$$

아무리 복잡한 선형 시스템도
행렬과 벡터로 쉽게 표현할 수 있음!

$$\begin{bmatrix} 2x_0 - 4x_1 + x_2 \\ 3x_0 + x_1 - 6x_2 \\ x_0 + x_1 + x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 10 \\ 5 \end{bmatrix}$$

선형 대수학  머신 러닝

어떻게 사용?

아파트 가격 예측

$$\text{집 값} = \text{크기} \times a_1 + \text{지하철 역 거리} \times a_2 + \text{층수} \times a_3$$

일차식

첫 번째 아파트 가격: $110 \times a_1 + 400 \times a_2 + 20 \times a_3$

두 번째 아파트 가격: $100 \times a_1 + 1000 \times a_2 + 5 \times a_3$

세 번째 아파트 가격: $180 \times a_1 + 10 \times a_2 + 30 \times a_3$

네 번째 아파트 가격: $50 \times a_1 + 300 \times a_2 + 5 \times a_3$

아파트 가격 예측

$$X = \begin{bmatrix} 110 & 400 & 20 \\ 100 & 1000 & 5 \\ 180 & 10 & 30 \\ 50 & 300 & 5 \\ & \vdots & \end{bmatrix} \qquad a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

깔끔하게 요약

모든 집 값 $= Xa$

요약

머신 러닝을 할 때는 데이터를 일차식에 사용하는 경우가 많다

행렬을 이용하면 정돈된 형태로 효율적이게 계산을 할 수 있다

선형 대수학은 일차식, 일차 함수, 행렬, 벡터를 다루는 학문이기 때문에 필수

실습

"Change your thoughts
and you change your world"
-Norman Vincent Peale-

실습과제

일본에는 한식 열풍이 불고 있습니다. 기회를 엿본 영훈이는 대기업을 퇴사하고 신주쿠에 프랜차이즈 '흥부부대찌개' 가맹점을 냈습니다.

그러나 보수적인 아버지께서는 번듯한 직장을 박차고 나온 영훈이가 못마땅합니다. 아버지를 안심시켜 드리기 위해 매달 매출을 보고하려고 하는데요. 엔화(¥)로 저장한 매출 데이터를 원화(₩)로 변환하는 작업이 필요합니다.

마침 numpy를 배운 우리가 도와줄 수 있겠네요. 엔화 매출이 담겨 있는 파이썬 리스트가 주어졌습니다. 1엔에 10.08원이라고 가정하고, 원화 매출이 담긴 numpy array를 만들어 출력해 주세요.

반복문은 사용하면 안 됩니다!

참고자료

"Change your thoughts
and you change your world"
-Norman Vincent Peale-

- (파이어폭스, 크롬) 사이버 실습실 : http:/matrix.skku.ac.kr/LA-Lab/
- http:/matrix.skku.ac.kr/LA-Lab/Solution/
- 무료 선형대수 계산 도구 : http:/matrix.skku.ac.kr/2014-Album/MC.html
- LA-interact(이상구) : http:/matrix.skku.ac.kr/2012-LAwithSage/interact/
- Full course 강의 동영상 : http:/matrix.skku.ac.kr/LinearAlgebra.htm
- 강좌 기록 http:/matrix.skku.ac.kr/2015-LA-FL/Linear-Algebra-Flipped-Class-SKKU.htm
- 선형대수학 강좌 포트폴리오: http:/matrix.skku.ac.kr/2017-LA-portfolio-sglee/
- http:/matrix.skku.ac.kr/2017-Album/LA-syllabus.htm
- http:/matrix.skku.ac.kr/2017-album/2017-Spring-Lectures.htm
- http:/matrix.skku.ac.kr/2015-Album/[Big-Book]LinearAlgebra-F6.pdf
- http:/ibook.skku.edu/Viewer/LA-Texbook -전자책 (무료 다운로드)
- 샘플 중간 및 기말 고사 **http://matrix.skku.ac.kr/LA-K/ (강의록)**
  http:/matrix.skku.ac.kr/2015-Album/2015-LA-S-Exam-All-Sol.pdf
  http:/matrix.skku.ac.kr/2015-Album/CLA-Final-Sample-Exam.pdf
  http:/matrix.skku.ac.kr/LA/2016-S-LA-Midterm-Final-Solution.pdf
- http:/matrix.skku.ac.kr/Lab-Book/Sage-Lab-Manual-2.htm 실습실