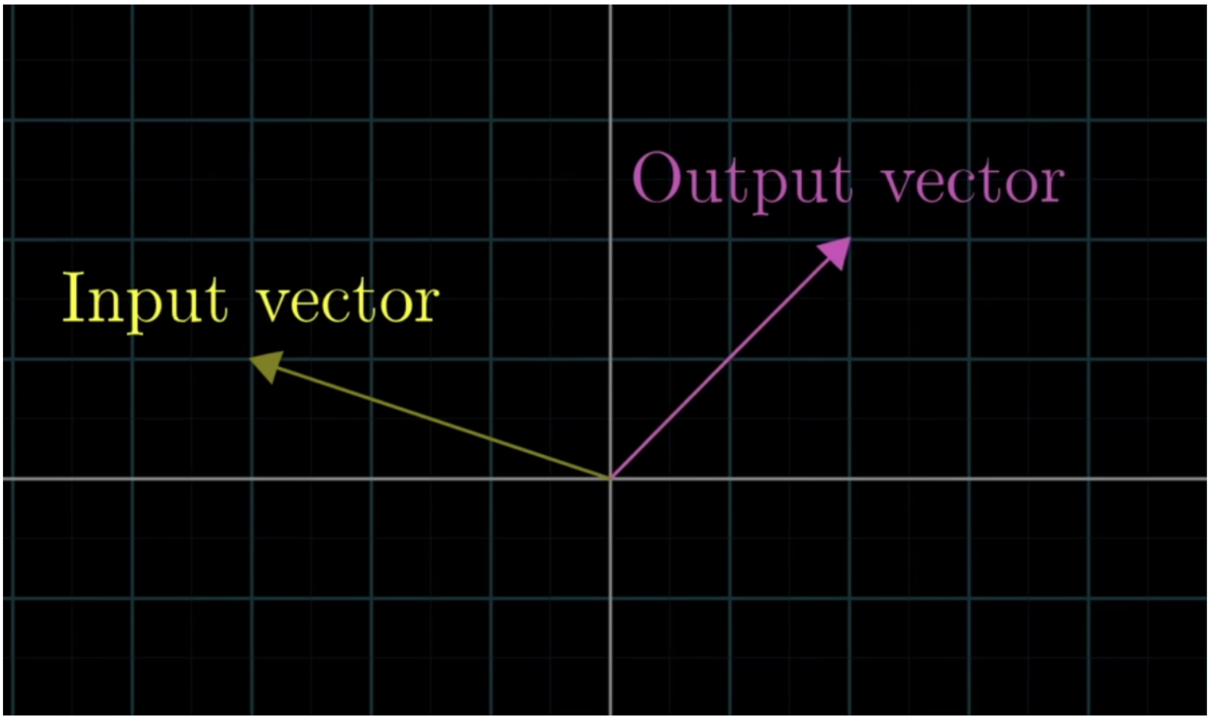
Matrices

선형 변환(Linear Transformation)

변환 = 함수 : 인풋을 아웃풋으로 변환한다.

선형대수에서 '변환'은 특정 벡터를 다른 벡터로 변경하는 것과 같은 것이다.

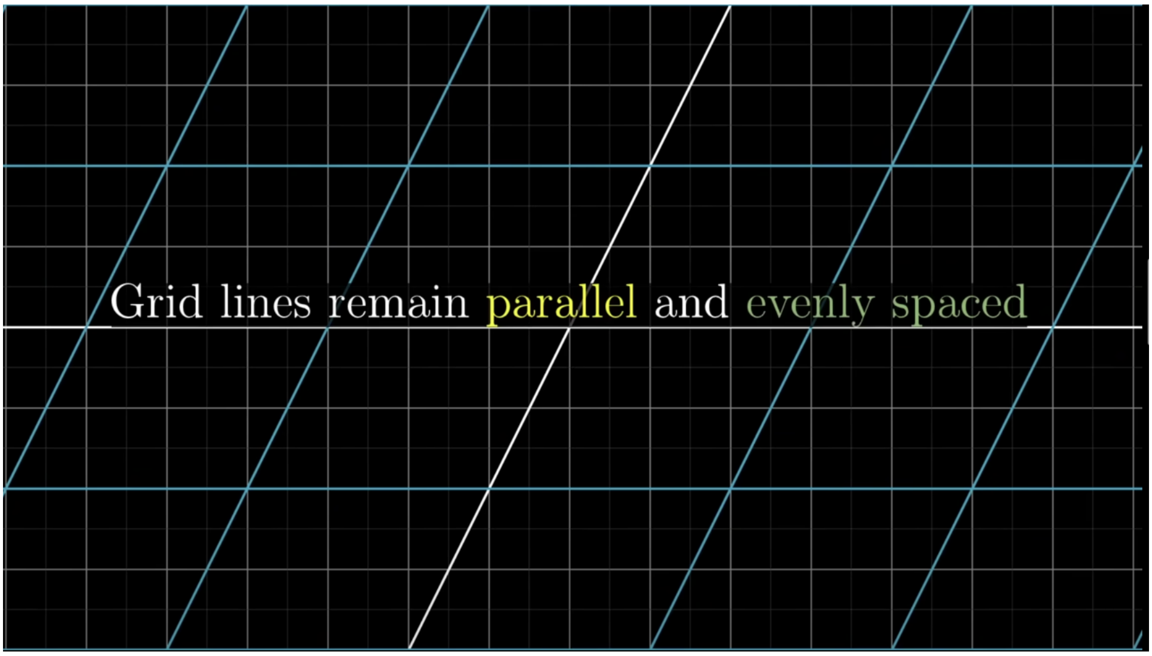


'변환'은 움직임으로 이해하면 좋다. 즉 입력 벡터를 출력 벡터로 변환하는 것은 입력 벡터를 움직여서 출력 벡터로 바꾸는 것으로 생각 할 수 있다.

선형대수에서 변환은 몇 가지로 제한 된다. 그 중 하나가 '선형 변환' 이다.

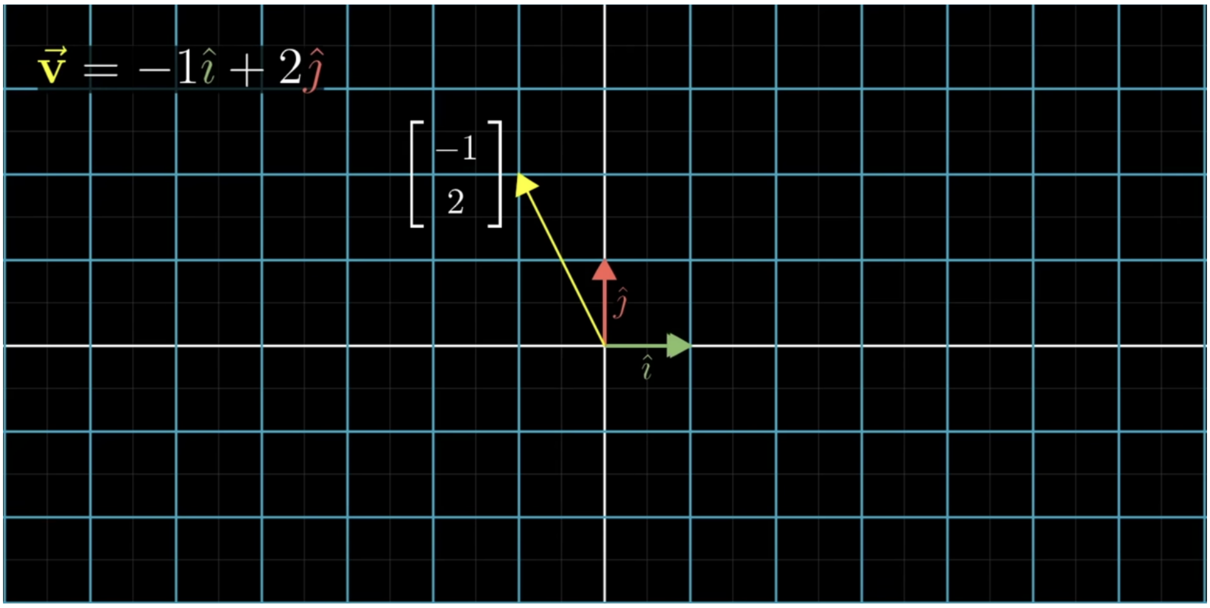
선형적이다는 것은 두가지 속성을 가진다.

1. 모든 선 들은 휘지 않은 직선이어야 한다.
2. 원점은 변환 이후에도 여전히 원점이어야 한다.

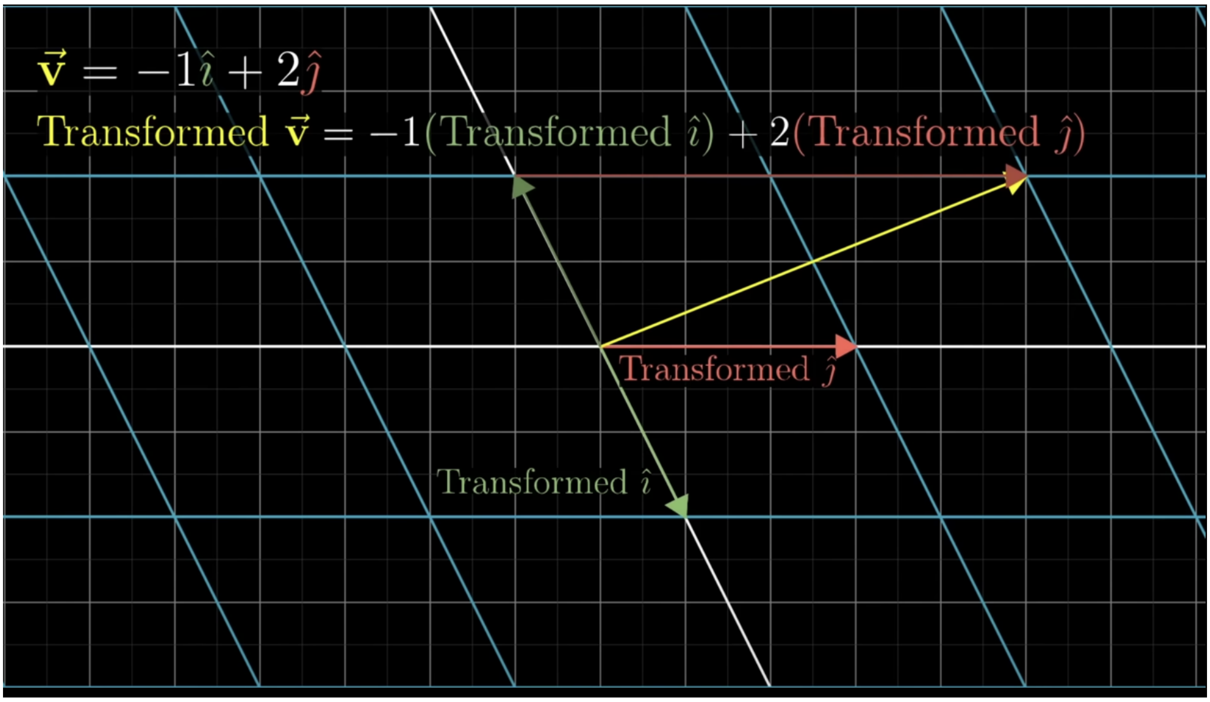


따라서 선형 변환에서는 격자 라인들이 변형 이후에도 여전히 ‘평행’하고 ‘동일한 간격’으로 있어야 한다. (그렇지 않으면 대각선을 그으면 구부러지기 때문)

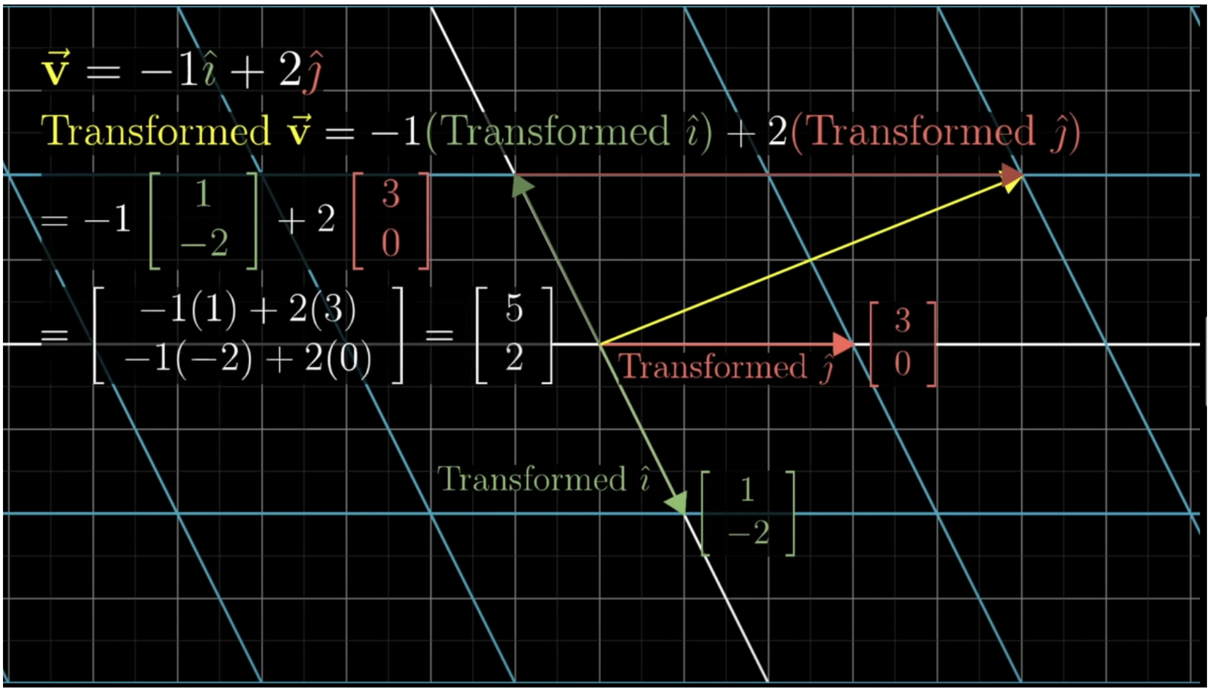
그렇다면 ‘선형 변환’ 은 어떻게 수치적으로 표현할 수 있을까?



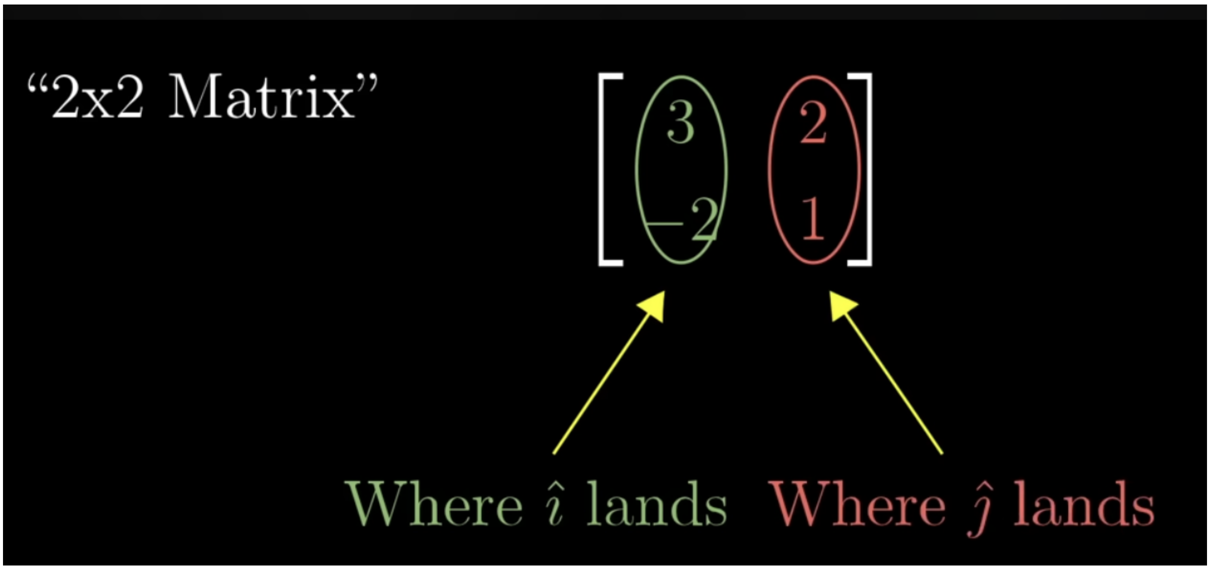
두 개의 기저 벡터가 어떻게 변하는지만 알면 된다. 위 그림에서 벡터 v[-1,2]는 i-hat, j-hat 두 기저 벡터가 있을 때, i-hat 벡터의 -1배, j-hat 벡터의 2배를 의미한다.



변환 후에도 i의 -1배, j의 2배로 같은 선형 결합을 유지하는 것을 확인 할 수 있다. 이 격자선들이 계속 평행하고 균등하게 분포하는 것이 중요하다.

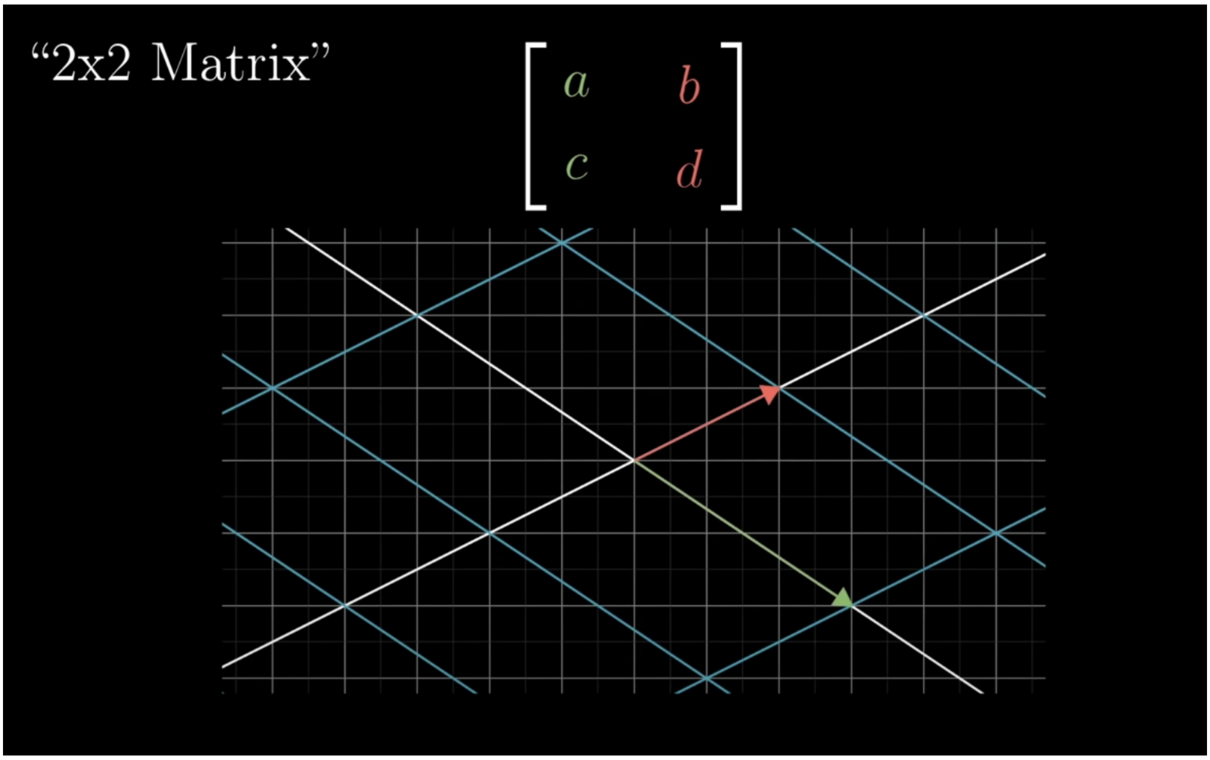


즉 i-hat과 j-hat의 변형 위치만 알면 변환 후의 벡터 v-1[5,-2]을 추론할 수 있는 것이다. (변환을 볼 필요가 없다.)



다시 말해 2차원 선형 변환은 4개의 숫자 = i-hat의 x,y 좌표와 j-hat의 x,y좌표 로 설명이 가능 한데, 이 숫자들은 2*2 숫자 형태로 나타내는 것이 일반적이고 이것이 2*2행렬이다.

즉 **선형 변환은 매트릭스**로 나타낼 수 있다.



매트릭스는 선형변환에 필요한 정보를 나타낼 뿐이다.
즉 첫째열 [a,c]과 둘째열[b,d]는 각각 기저벡터의 도착점이다.

"2x2 Matrix"

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad \begin{bmatrix} x \\ y \end{bmatrix}$$

$$x\begin{bmatrix} a \\ c \end{bmatrix} + y\begin{bmatrix} b \\ d \end{bmatrix} = \begin{bmatrix} ax+by \\ cx+dy \end{bmatrix}$$

이 변환(매트릭스)를 어떤 벡터에 적용하면 위와 같다. 즉 벡터가 매트릭스를 통해 선형변화가 이루어진 것이다.

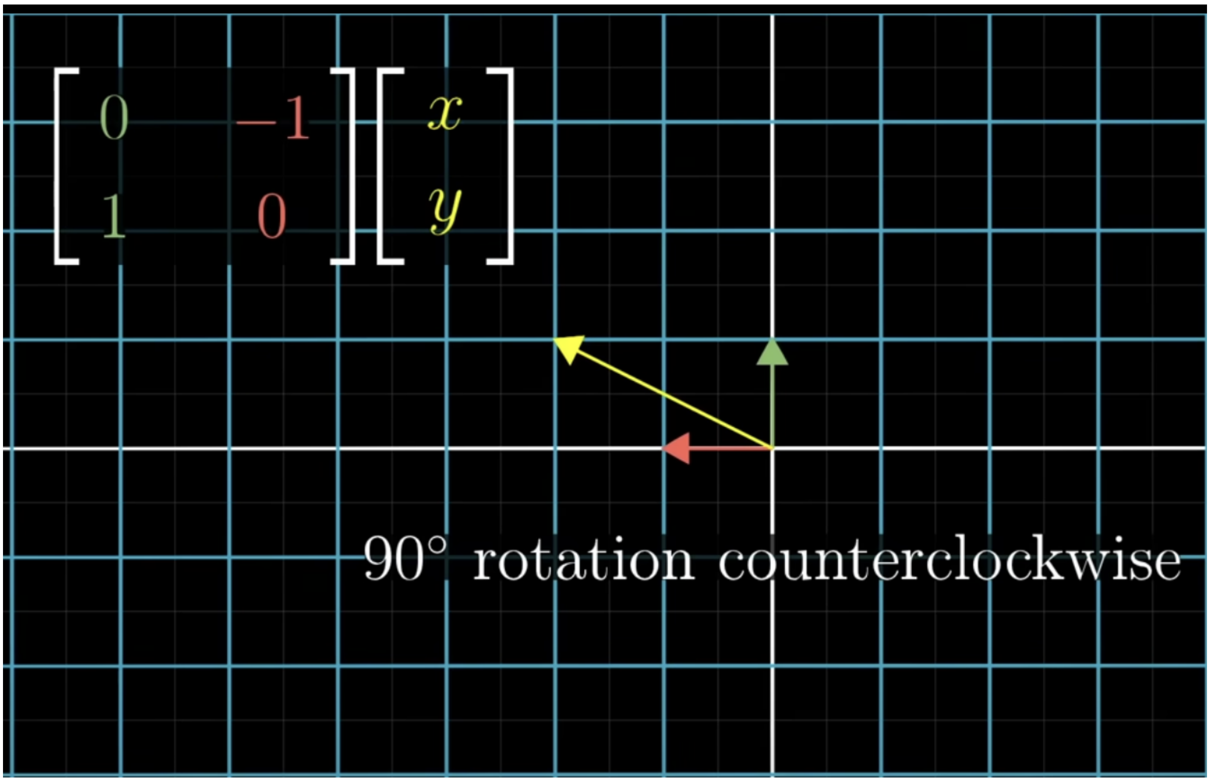
"2x2 Matrix"

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = x\begin{bmatrix} a \\ c \end{bmatrix} + y\begin{bmatrix} b \\ d \end{bmatrix} = \begin{bmatrix} ax+by \\ cx+dy \end{bmatrix}$$

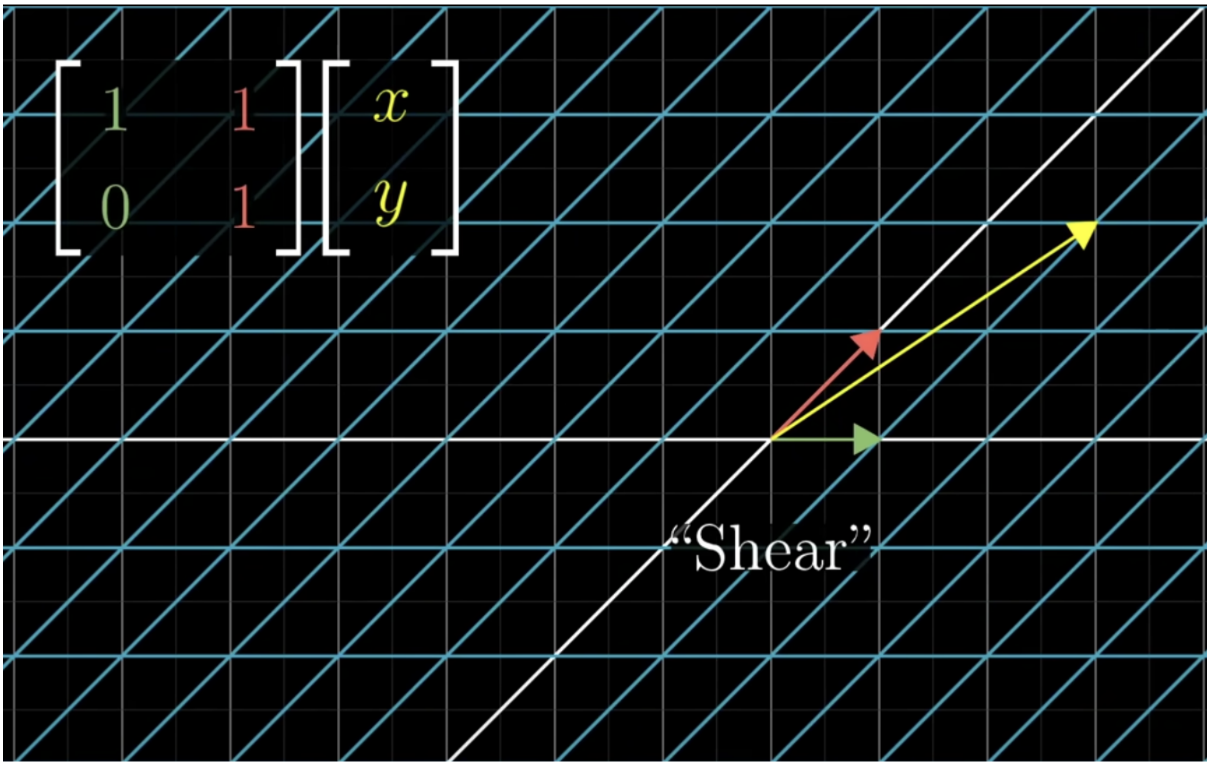
고등학교때 배운 행렬의 곱셈으로 표현이 가능하다.

몇 가지 변환의 예

몇 가지 변환을 알아보면 아래와 같다.

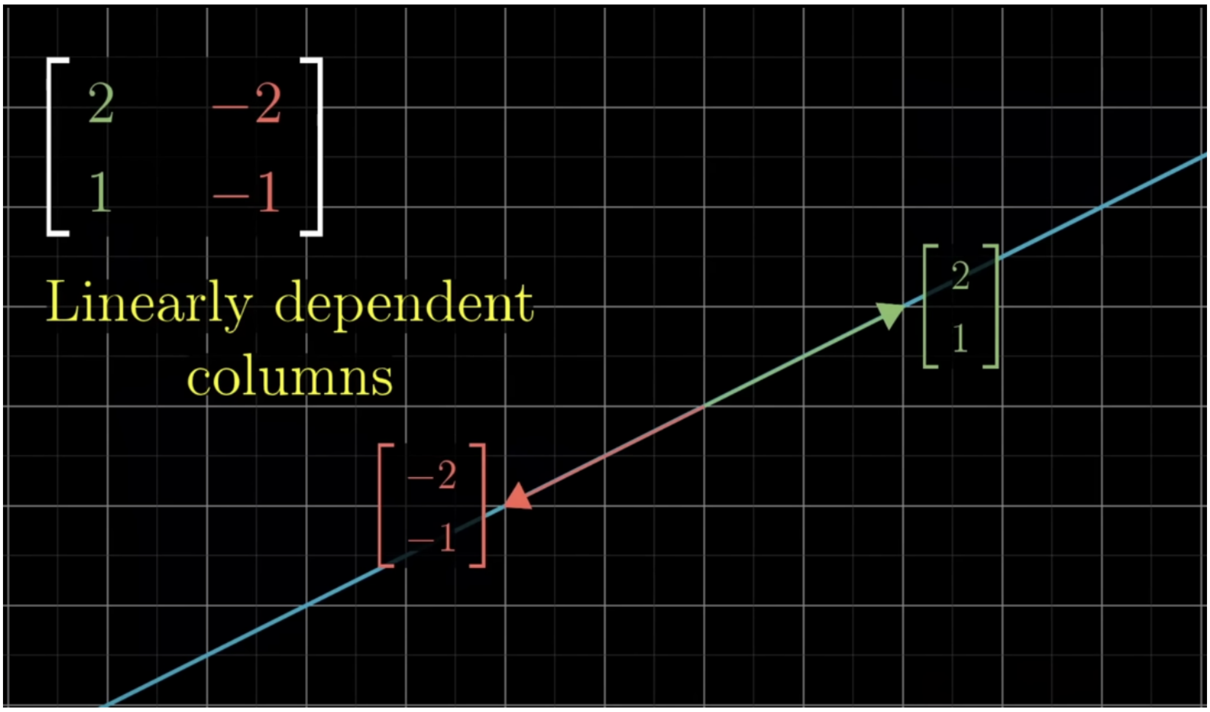


반시계 90도 돌림



shear 변환(i-hat은 그대로, j-hat은 (1,1)로)

선형 종속적인 기저 벡터들로 이루어진 매트릭스



만약 i-hat, j-hat이 선형 종속(linear dependent) 관계에 있다면, (즉 하나가 다른 하나의 스케일링 버전) 이 선형 변환은 2차원 공간을 수축 시켜 하나의 선으로 바뀐다.

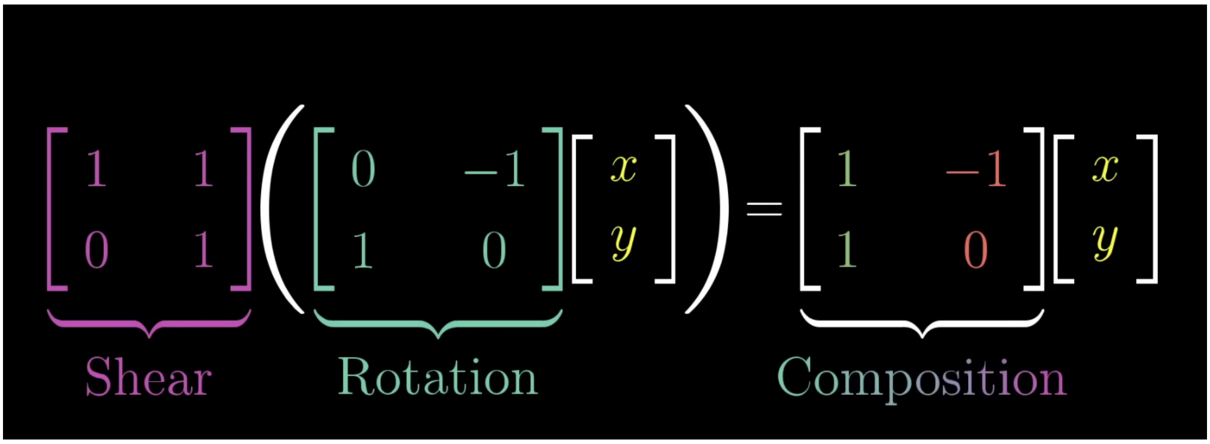
요약

즉 선형 변환은 평행하고 균등 간격의 격자선을 유지하고 원점을 유지하며 공간을 이동하는 방법이다. 이는 숫자적으로 기저 벡터들의 변형 후 좌표값들로 표현이 가능하고 이를 매트릭스라 한다.

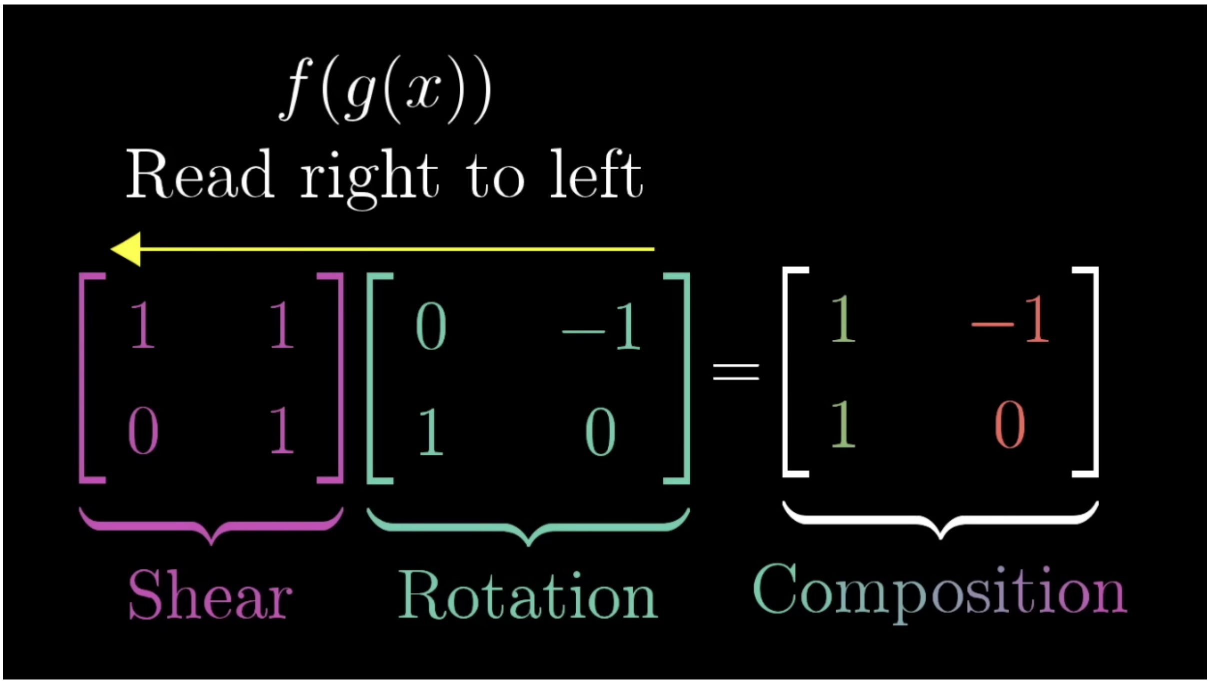
즉 매트릭스를 볼 때 이를 공간의 변환으로 생각해보자!

Matrix Multiplication

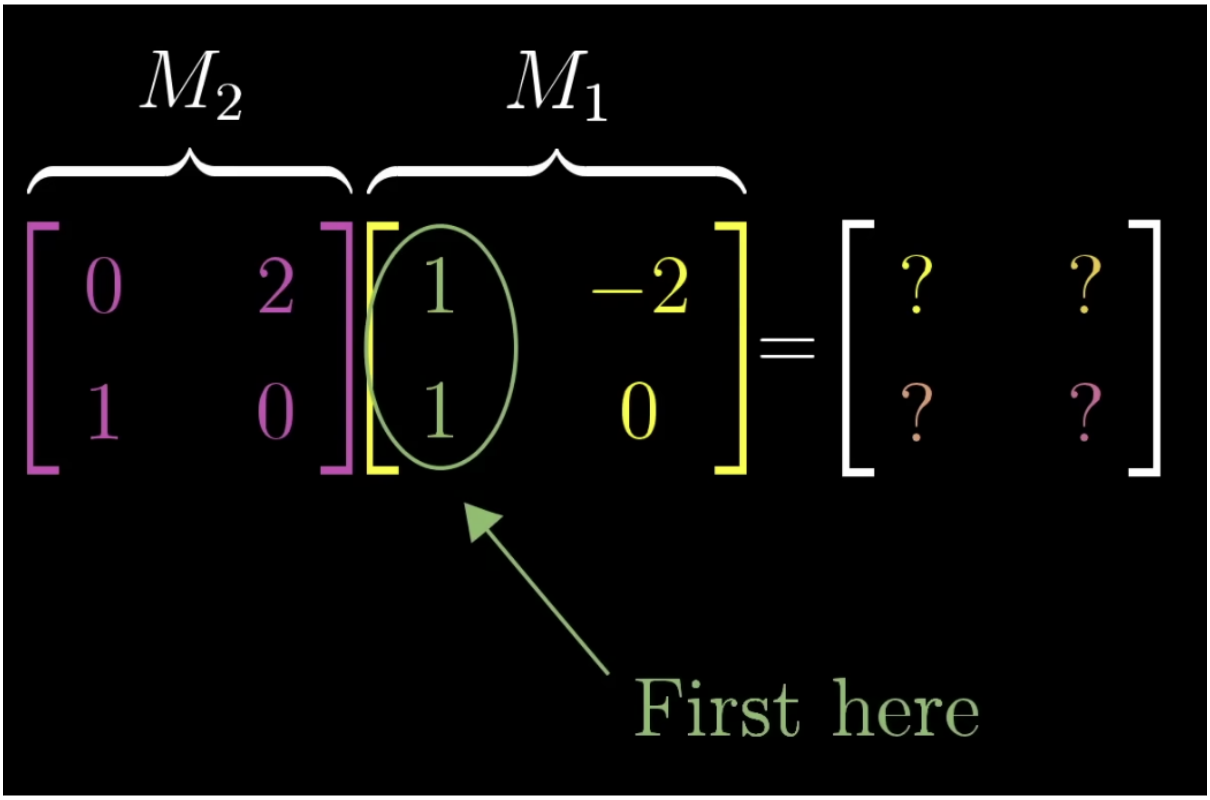
매트릭스에 벡터를 곱하는 것은 수식적으로 벡터를 선형 변환(linear transformation) 하는 것과 같다.



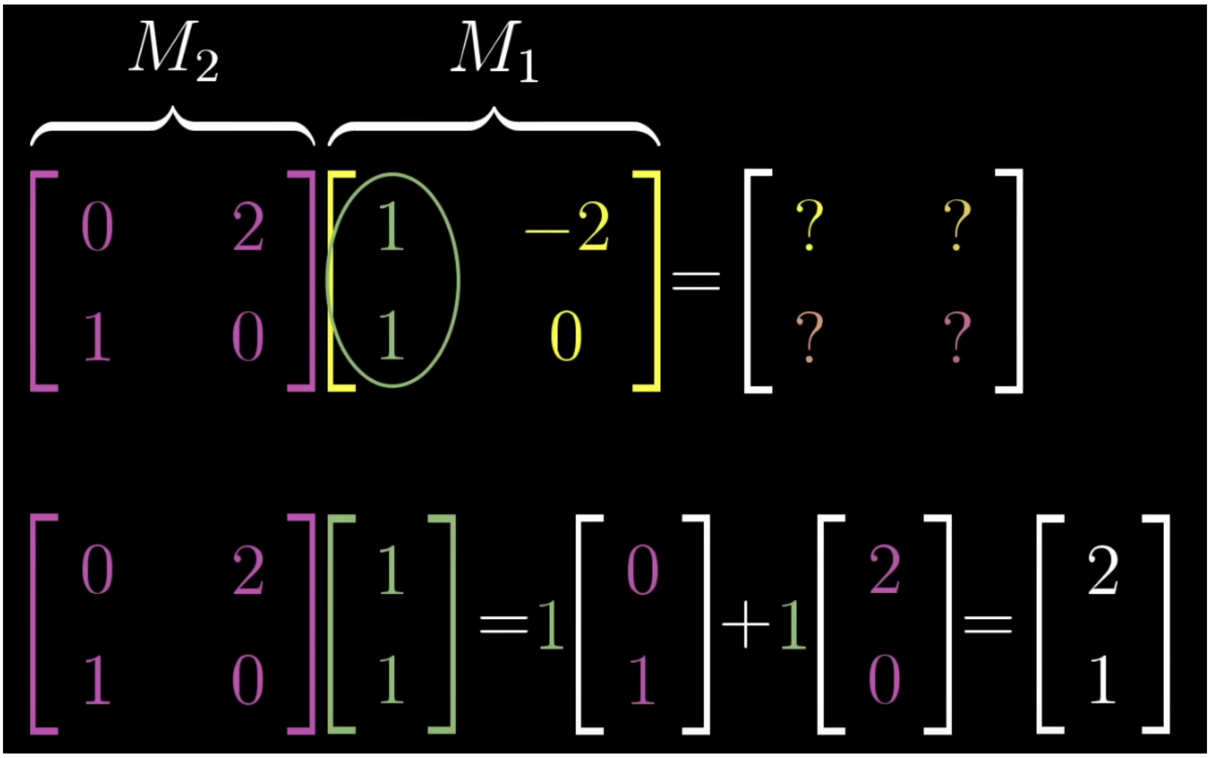
선형 변환 후 다시 선형 변환 하는 것은 결국엔 한 번의 선형 변환으로 표현 할 수 있다. 따라서 행렬의 곱은 언제나 기하학적으로 한 변환을 적용한 후 다른 변환을 적용하는 것과 같다는 것을 알아두자.



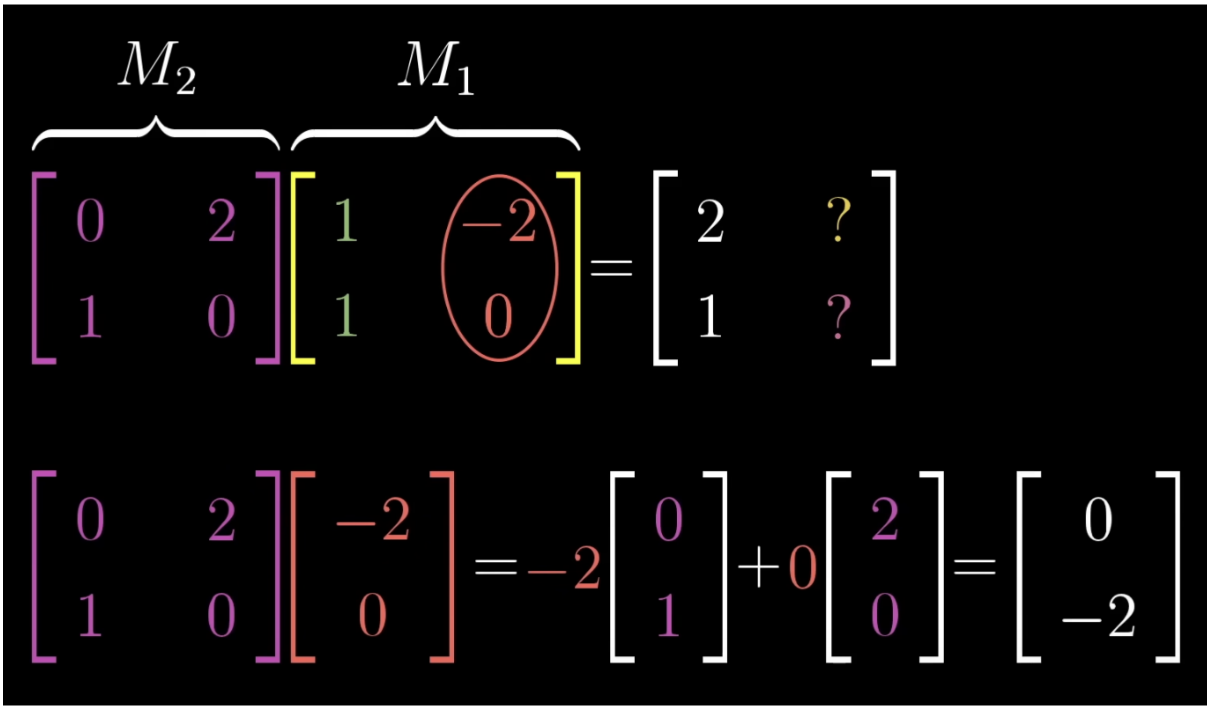
변환은 오른쪽에서 왼쪽 순으로 읽는다. 이는 함수를 읽는 원리와 같기 때문이다.



행렬을 곱할테는 i-hat과 j-hat의 위치를 생각한다. i-hat의 경우 처음엔 [1,1]에 위치한다

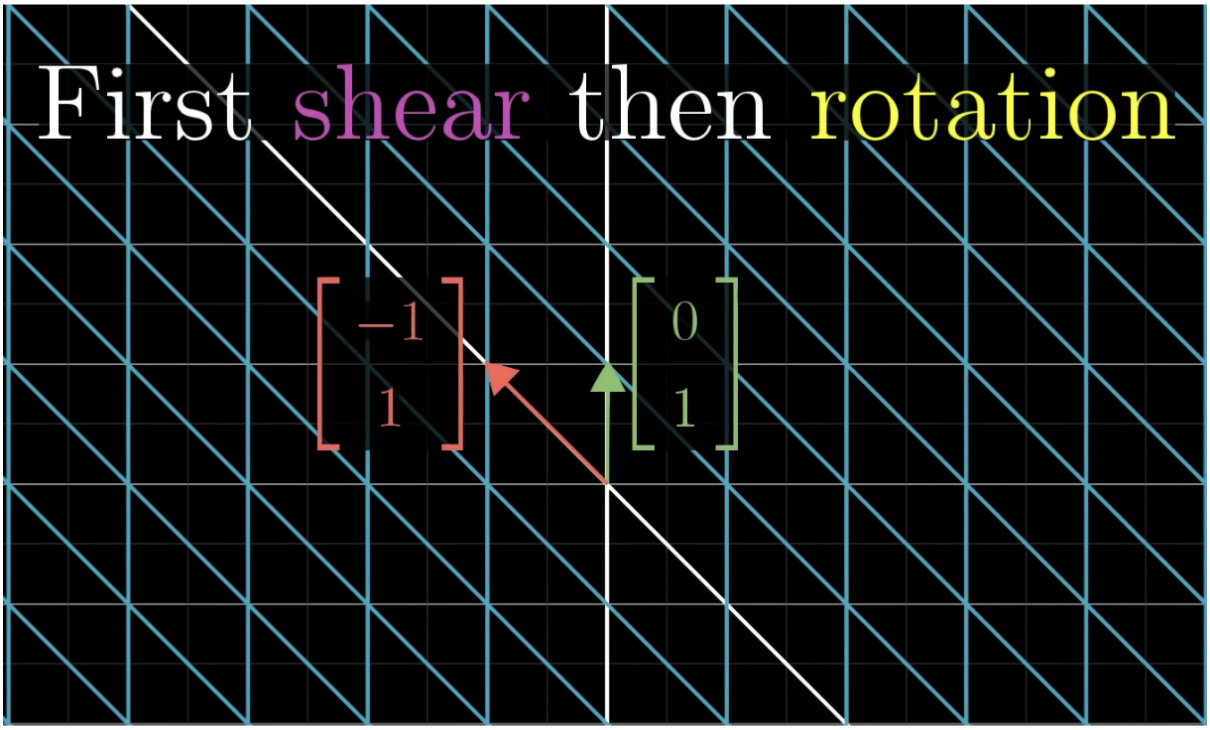


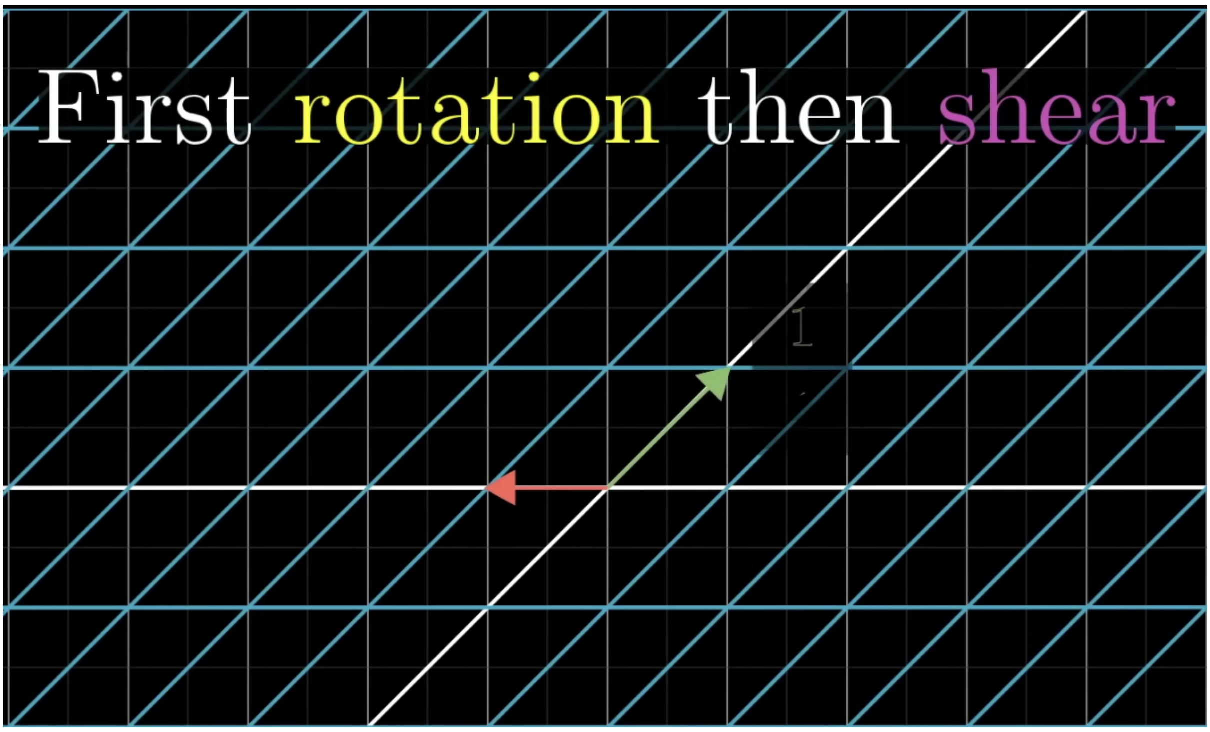
그다음 M2 행렬로 선형 변환을 한 위치[2,1]에 존재하게 된다.



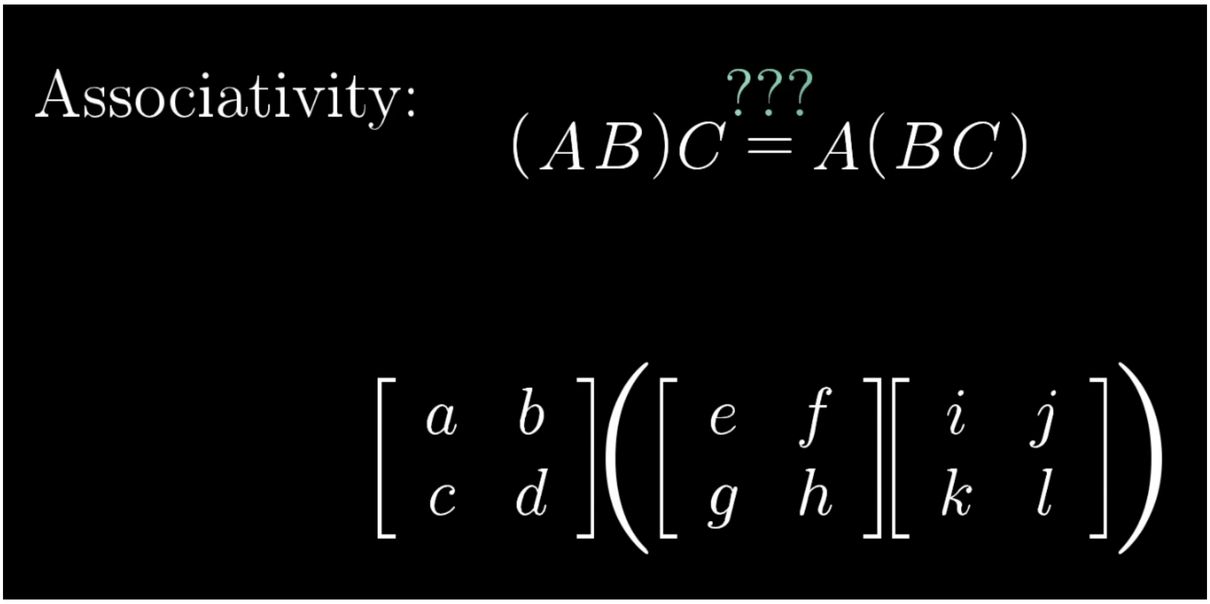
j-hat도 마찬가지 원리로 처음 위치 [-2,0] 이었다가 M2 행렬로 선형변환을 한 후 [0,-2]에 위치하게 된다.

그러면 M1*M2 와 M2*M1은 같은 결과일까? 즉 두 행렬을 곱할때 순서가 상관이 있을까?





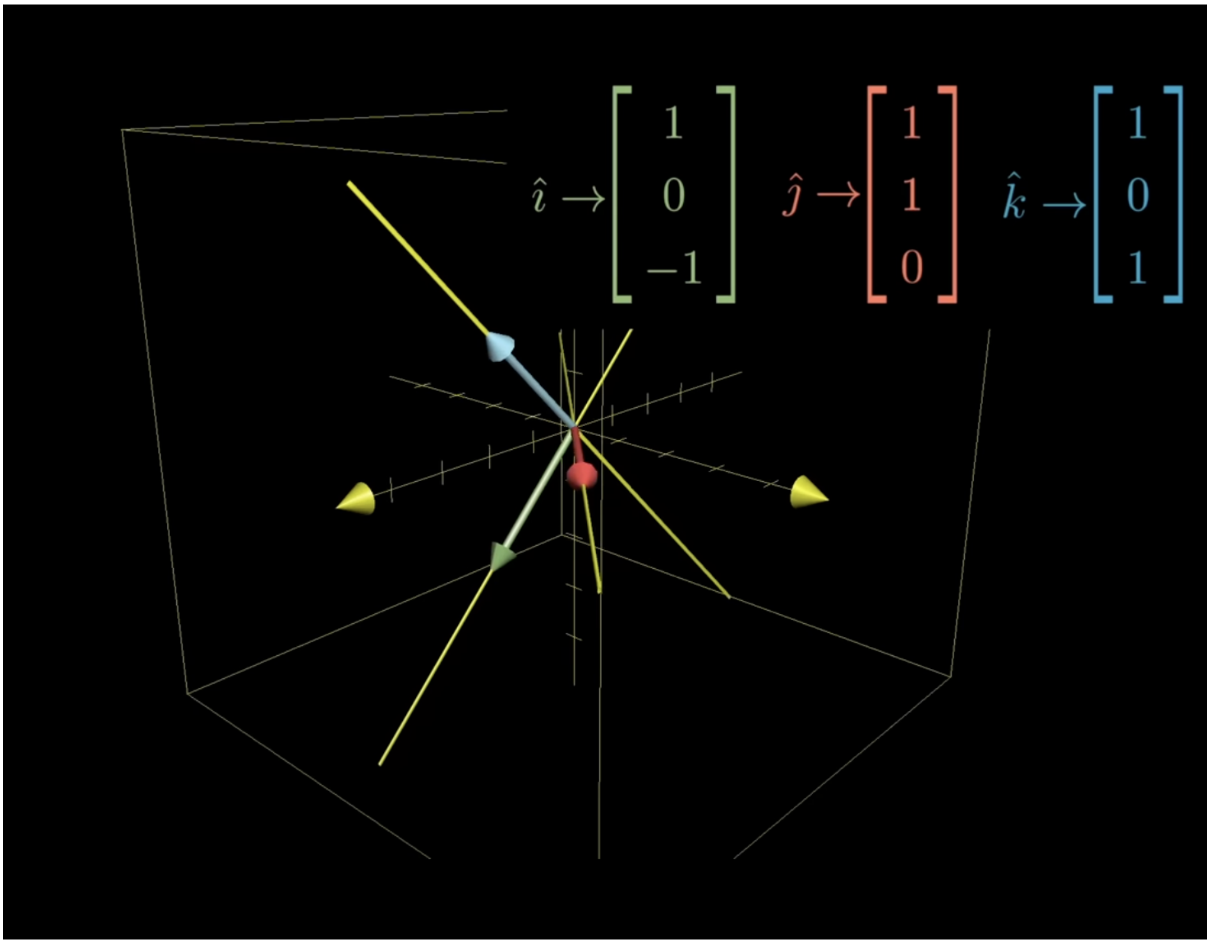
두 결과는 다르다.

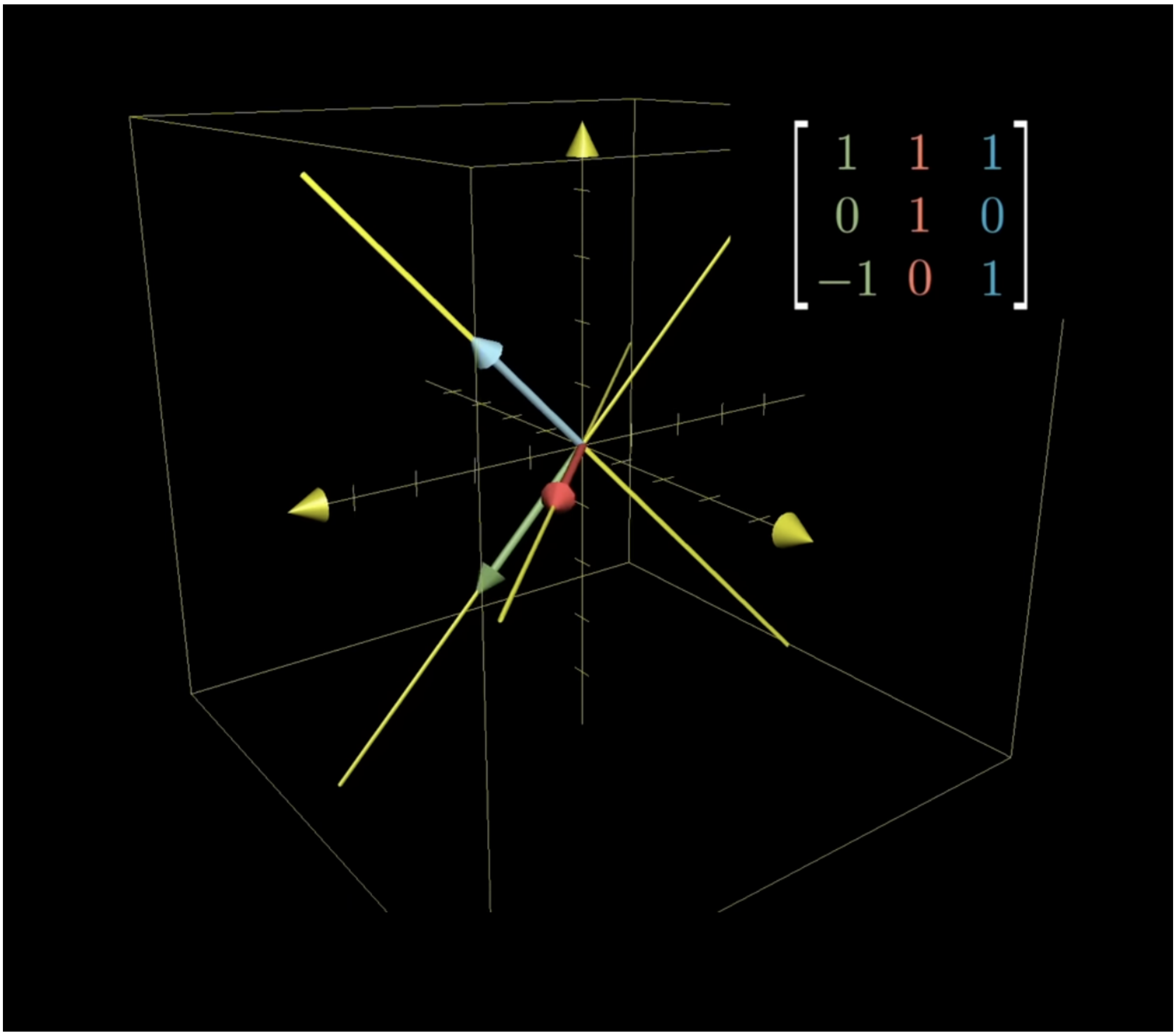


그러면 결합법칙은 어떻게 성립하는 것일까?
가령 A(BC) 와 (AB)C는 같은데, 왜냐면 선형 변환이 C→B→A 순으로 일어나기 때문에 괄호가 어디 위치하는 지 상관이 없기 때문이다.

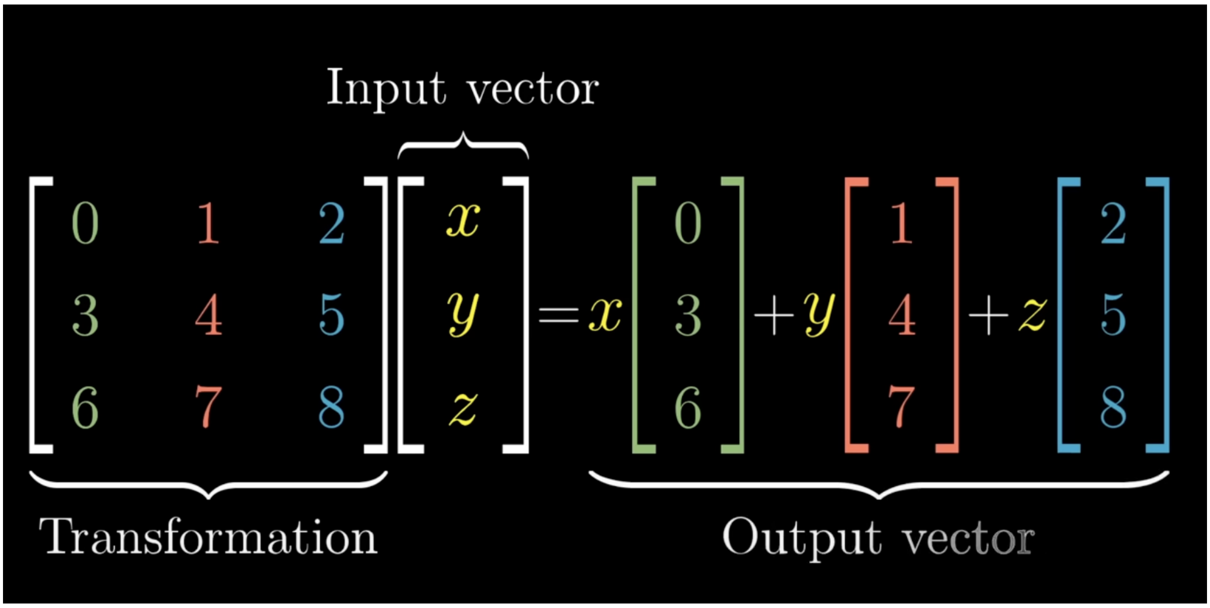
Three-Dimensional Transformation

3차원에서도 2차원과 마찬가지로 행렬은 입력 벡터를 움직여서 출력 벡터를 만든다. 역시 마찬가지로 이는 기저 벡터의 위치를 알면 파악이 가능하다.

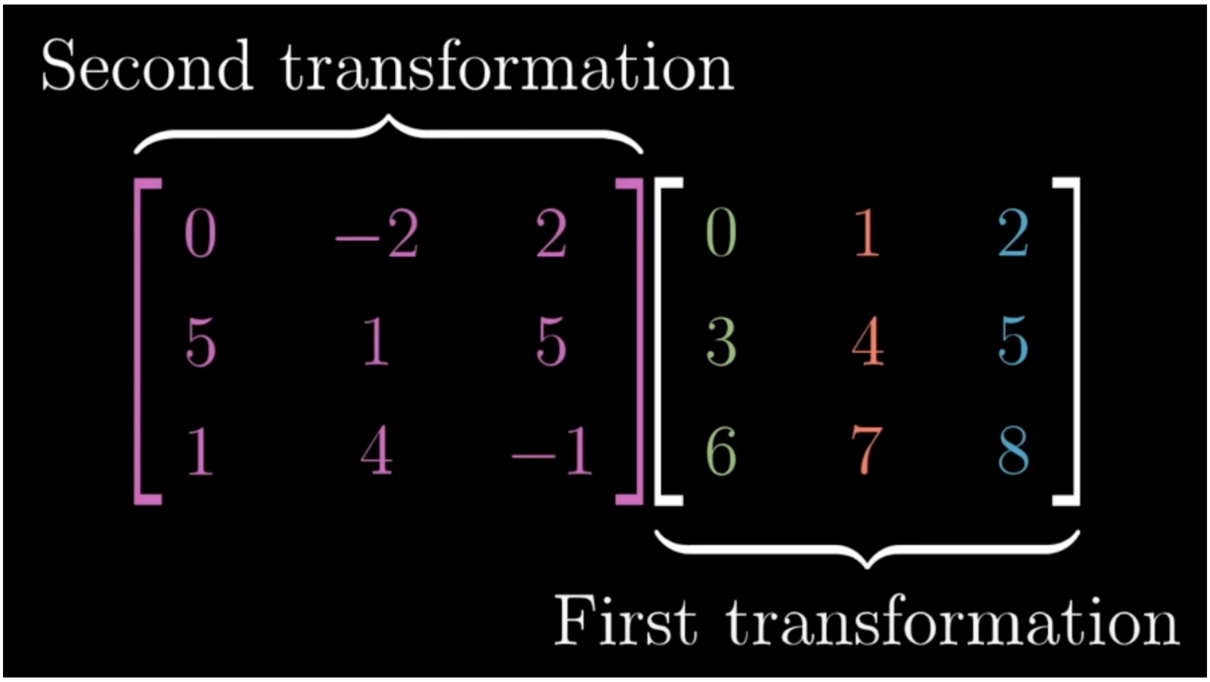




3차원의 기저벡터들 i-hat, j-hat, k-hat의 좌표를 3*3 행렬에 적어 표시 할 수 있다.



2차원때와 마찬가지로 벡터가 3차원 행렬로 선형변환된 이후의 결과 벡터를 얻으려면 벡터의 좌표를 각각에 대응하는 열과 곱하면 된다.



3차원 행렬의 곱도 2차원과 마찬가지로 연속된 선형 변화로 생각할 수 있다.