벡터 기초

SGA 프로그래밍 제작 전문가 양성 과정

벡터란?

- 사전적 정의 : 경향성을 가지는 수(혹은 수열)의 집합
- 더 좁은 해석 : 크기와 방향을 가지는 힘(물리적 힘의 양)

- 더 간단히 표현하면? 화살표!

이거요...

벡터란?

- 식으로 정해진 개념은 아니다.
  -> 표현할 수단이 여럿 존재한다

- 가장 흔한 표현 방식 : 기호 ***v***

- 그 밖에 입출력 하나짜리 함수 f(x), 일차 방정식, 기하학적 의미를 담은 행렬의 식, 그 외에도 많은 방법으로 벡터를 표현

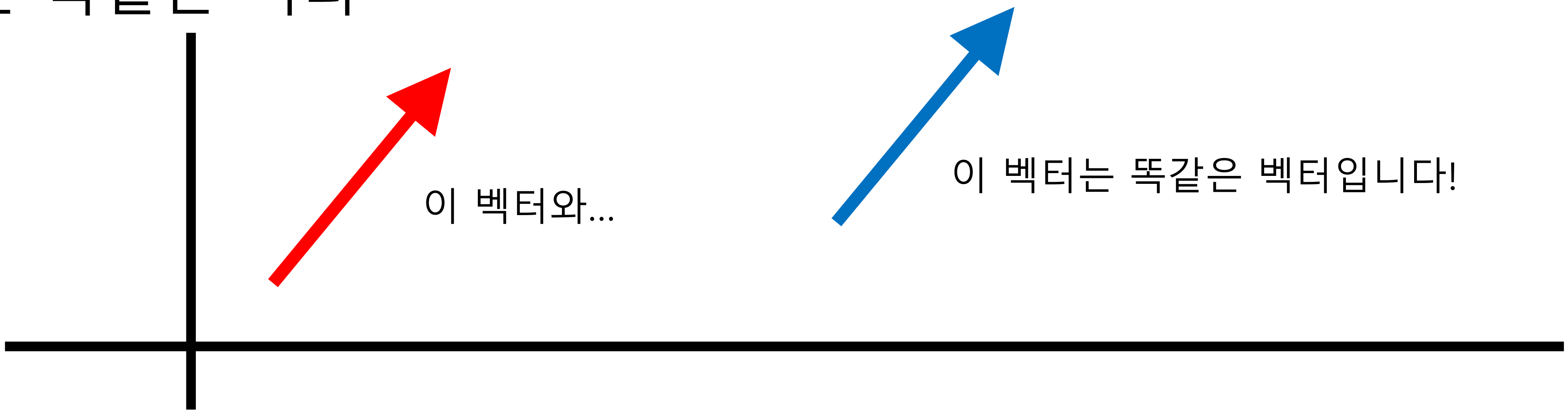
벡터의 특징

- 거의 대부분 직선이다.
  (수식상의 표기가 이차 이상을 사용하지 않는다)
- 크기와 방향이 존재
  = 실체를 이해하려면 최소 2차원 인지가 필요
- 벡터 내 모든 수는 동일한 경향을 갖는다
  (힘이 작용되는 크기가 벡터 내 모든 '수'에서 똑같다)

벡터의 특징

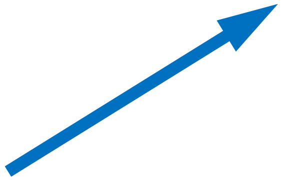
- 위치라는 개념이 없다. (시작과 끝 대신 "시점(원점)"과 "종점"만 존재)

- 그러므로 좌표상의 어디에 그려졌든 아무 상관없이 길이(=크기)와 방향(모양과 화살표 위치)이 똑같은 두 벡터가 있다면, 이 둘은 똑같은 벡터



벡터의 계산 : 덧셈

벡터 A

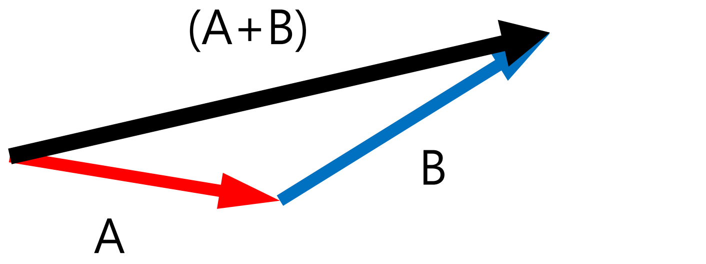


벡터 B

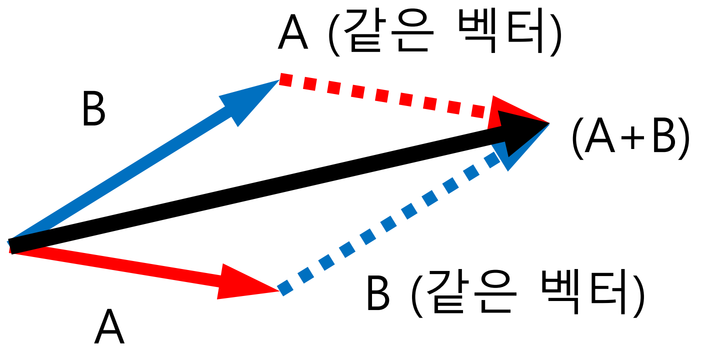
벡터 A + B :

두 벡터를 이어서 배치한 다음
삼각형이 되는 선을 그리거나,

같은 원점에서 출발한 두 벡터를
변으로 삼는 사각형을 계산해서
역시 같은 원점에서 출발하는
대각선을 그리면 됩니다.



혹은....



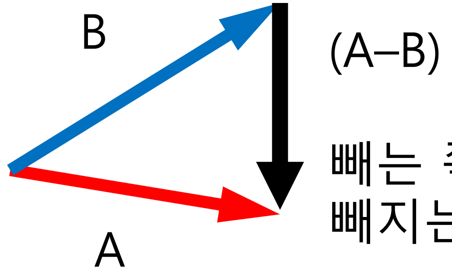
벡터의 계산 : 뺄셈

벡터 A

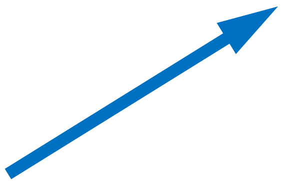
벡터 A - B :

같은 원점에서 출발한 두 벡터를
그린 다음, 기준 벡터(A, 먼저 나온 벡터)
의 종점으로 가기 위해, 빼는 벡터(B)의
시점에서 출발하는 벡터를 그리면 됩니다.

빼는 쪽 종점에서, 빼지는 쪽 종점으로.



빼는 쪽 종점에서 출발
빼지는 쪽 종점에서 도착



벡터 B

벡터의 계산 : 덧셈과 뺄셈

- 벡터 덧셈과 뺄셈 실습 랜덤 퀴즈 주소

  덧셈 : https://www.geogebra.org/m/fsecn352

  뺄셈 : https://www.geogebra.org/m/axrcruq4

- 위 주소에서 벡터 문제를 풀어보면서 감을 잡아봅시다!

벡터의 곱셈

- 벡터 곱셈에는 세 가지 종류가 있다

  1) 벡터곱 (크로스 연산) : 두 벡터가 만들 수 있는 공간, 혹은 해당 공간으로부터 도출 가능한 +1차원 수직선을 계산

  2) 텐서곱 (텐서 연산) : 텐서(식으로 표현한 선)화한 벡터의 성분을 수학 식으로 곱한 결과를 계산

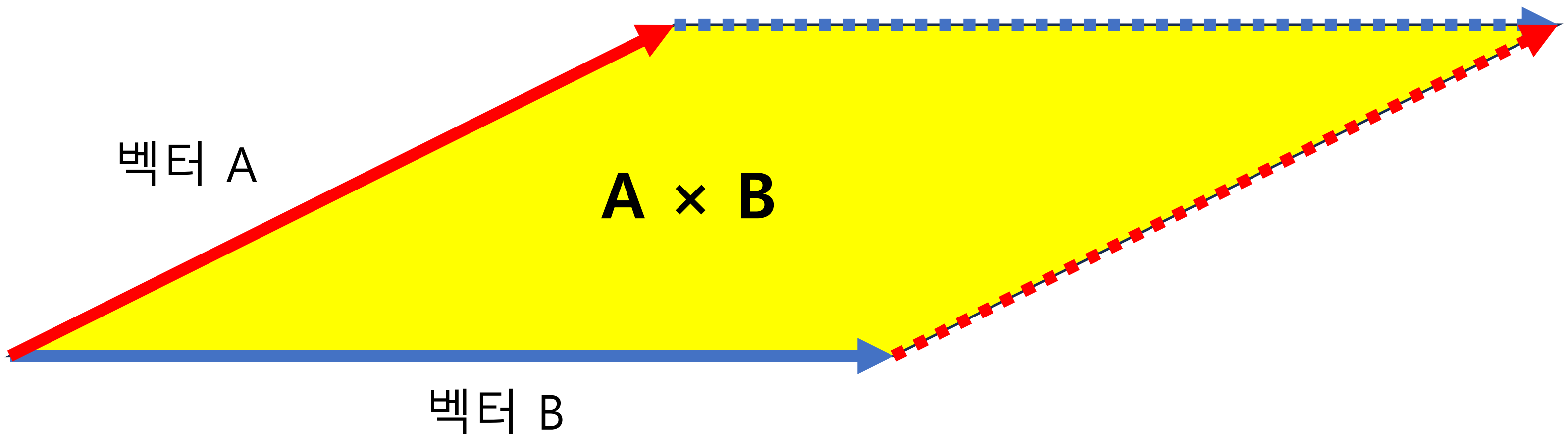
  3) 점 곱 (스칼라 연산) : 두 벡터를 삼각함수를 통해 겹치고, 한 벡터가 다른 벡터에 비하여 갖는 비중을 계산

벡터의 곱셈

- 두 벡터를 서로 겹친다(수렴시킨다)는 모양의 특성에서 두 벡터의 점 곱셈을 **내적**이라고 부르기도 한다
- 두 벡터 사이의 벡터곱과 텐서곱을 합쳐 **외적**이라고 부르기도 한다.

벡터의 곱셈 : 벡터곱

- A × B 로 표시
- 두 벡터가 만나 이루는 공간의 크기를 도출 (최초 개념)

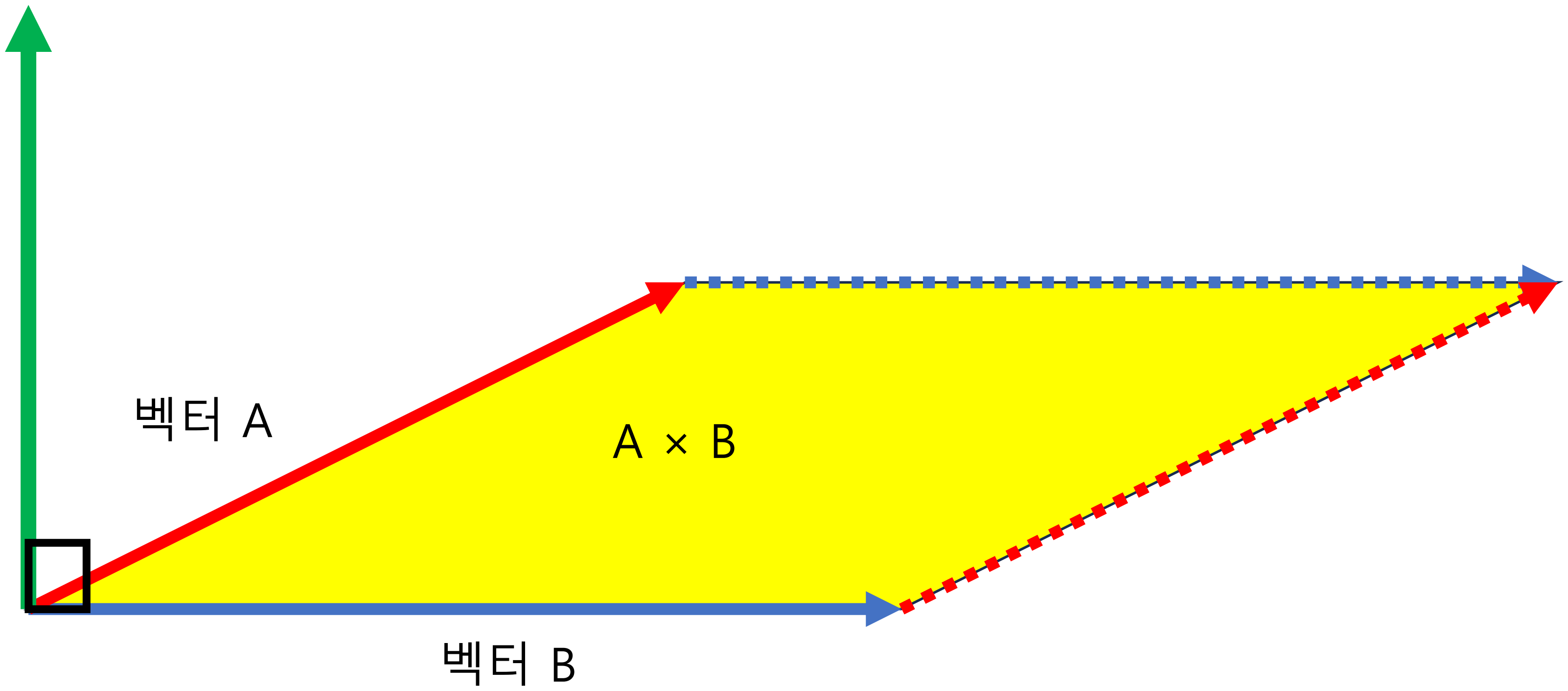


벡터의 곱셈 : 벡터곱

- A × B 공간도 행렬로 표현이 가능함 (x와 y의 곱)
  -> 해당 행렬을 나타내는 새로운 벡터 표시가 가능하다

A × B
이 벡터를 도출하는 것이
실제로 벡터곱을 행하는
더 큰 목적.

도출된 벡터는 A와 B가
만든 평면에 **수직**이다.



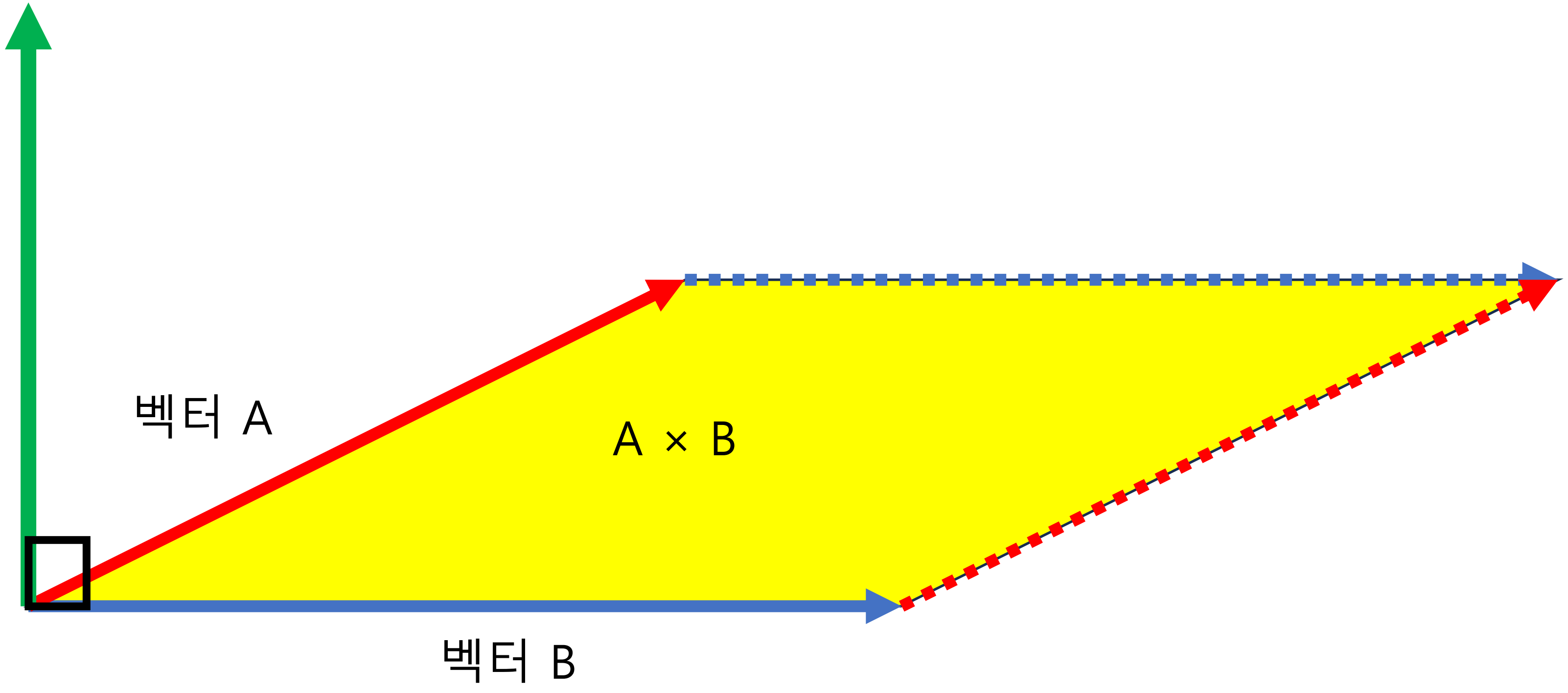
벡터의 곱셈 : 벡터곱

- ~~기하학에서 벡터곱의 결과는 반시계 원칙 (오른손 좌표)~~

그러나 다이렉트X는 반대!

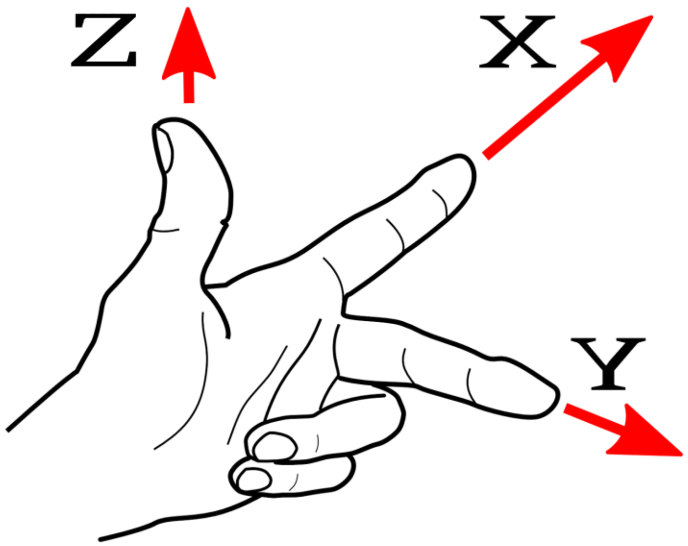
기준선(곱셈에서 먼저 나오는 벡터)을 왼쪽으로,
곱하는 벡터를 오른쪽으로,
도출되는 벡터를 위로 그리는

왼손 좌표계를 사용한다.

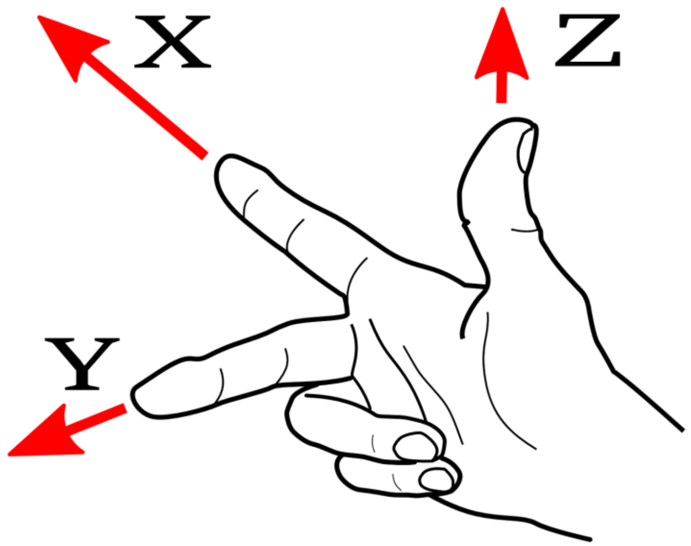


벡터의 곱셈 : 왼손 좌표 (왼손잡이 규칙)

- 다이렉트 X의 좌표계 규칙은 왼손



다이렉트X의 좌표계



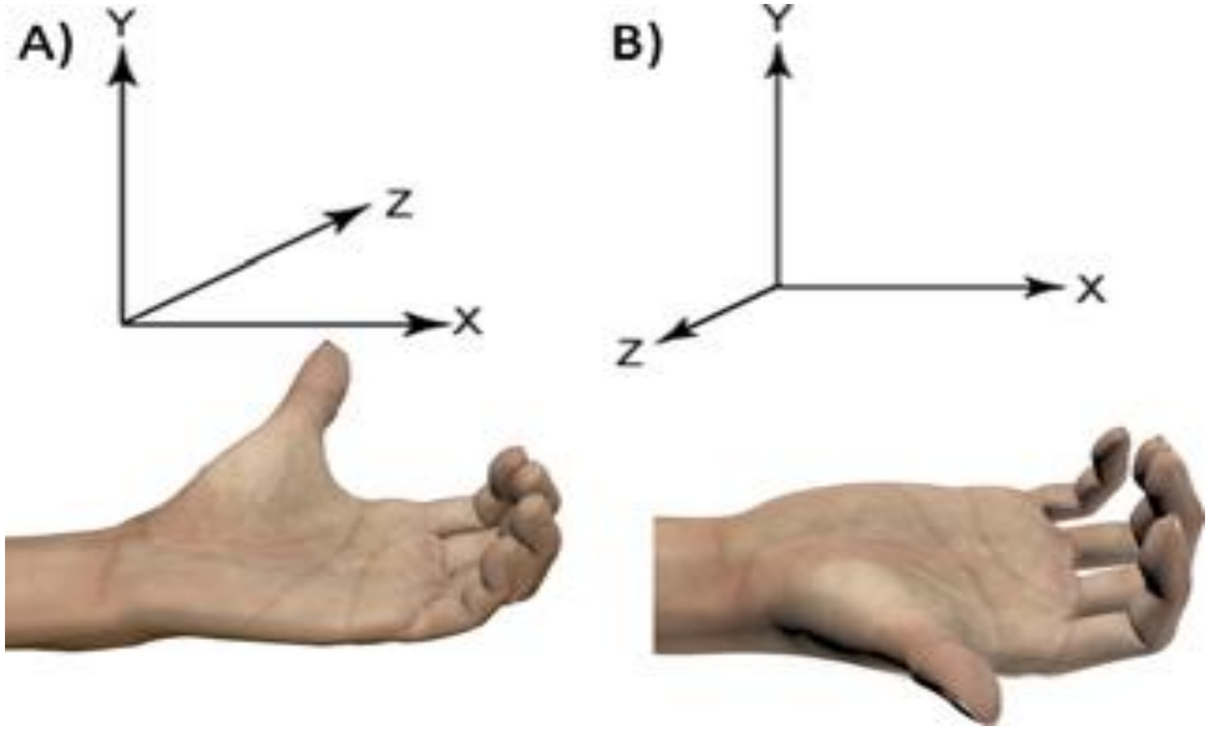
일반적인 벡터곱

벡터의 곱셈 : 왼손 좌표 (왼손잡이 규칙)

- 다이렉트 X의 좌표계 규칙은 왼손

- X가 오른쪽, Y가 위쪽을 향하게 배치를 바꾸었을 경우,
  왼손 좌표계는 Z(=AxB)를

  앞 으로 보낼 수 있기 때문
  (게임개발 등에서 계산이 편리)



손 좌표계를 표현하는 다른 방식
손바닥을 X, 손가락 끝을 Y로 이해하면
엄지손가락이 향하는 방향이 Z(=AxB)가 된다

벡터의 곱셈 : 텐서곱

- A ⓧ B 로 표시
- ~~어려워서 생략이 아니고~~… 게임 개발에서 사용하는 벡터의 곱셈은 방향 도출(Z축 방향), 힘의 크기 계산(Z축 길이)이 주 목적
- 1차 방정식이 사용된 벡터는 벡터곱 및 텐서곱 결과가 동일하다(그래서 둘을 뭉뚱그려 외적이라고 부르는 것이 용인되는 것)
- 2D 및 3D 게임을 만들 때는 불필요한 계산 ~~사실은 어려워요~~

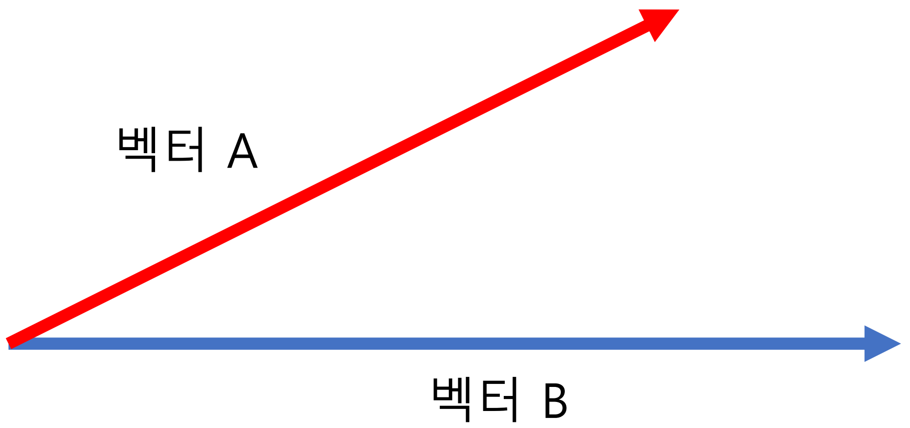
벡터의 곱셈 : 텐서곱

- 이후 인공지능 등의 심화된 프로그래밍을 게임에 접목시킬 경우, 한정적으로 계산 함수를 호출할 가능성은 있음

- 이름을 알아만 둡시다. 텐서곱이라는 것도 있구나... 하고요.

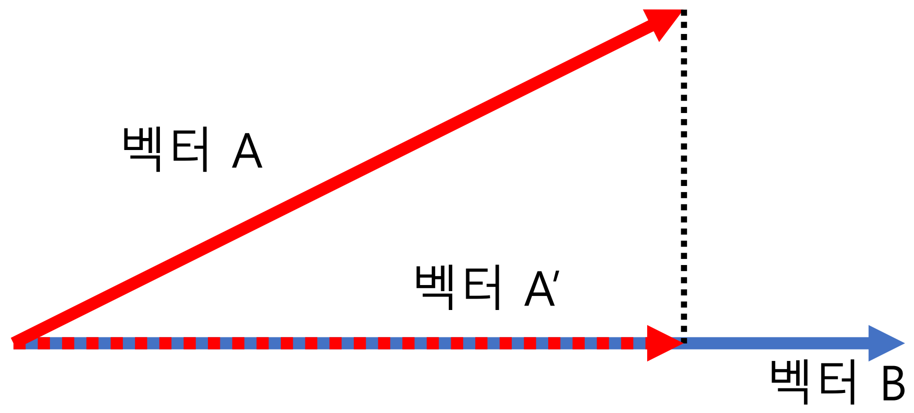
벡터의 곱셈 : 점 곱

- A • B 로 표시
- 두 벡터를 삼각함수 계산을 통해 동일한 방향에서 적용되도록 수정했을 때, 두 벡터 간의 비중 혹은 힘의 크기 관계를 계산하는 것



벡터의 곱셈 : 점 곱

- 곱해질 벡터(먼저 나온 A)의 종점에서 곱하는 벡터(뒤에 나온 B) 직선에 수직이 되는 수직선을 계산하고, 수직선과 B가 만나는 지점까지를 새로운 A로 계산
- 해당 새로운 A와 B의 관계를 곱셈

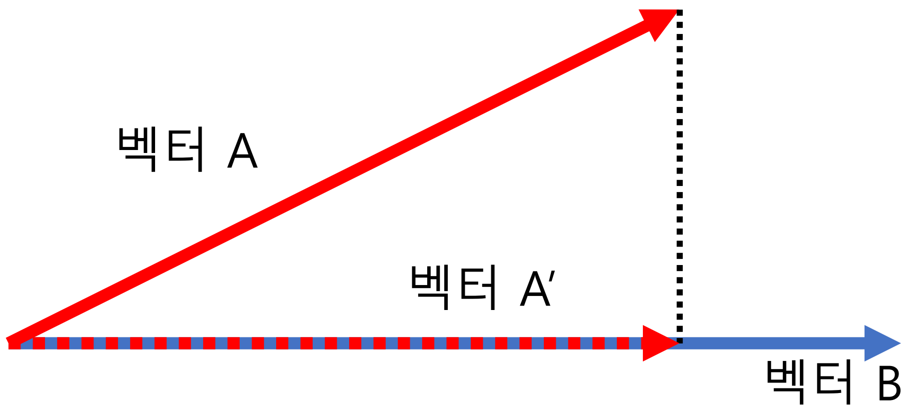


벡터의 곱셈 : 점 곱

- 벡터 A′는 A의 길이를 빗변으로, A와 B 사이의 각도를 기준 각도(Θ)로 가지는 삼각함수의 계산 대상

- 따라서 A와 B의 점곱은 (A cos Θ)B =AB cos Θ

- A, B 모두 이 식에서는 벡터 본체가 아니라 벡터의 크기 (선의 길이)



벡터의 곱셈 : 점 곱

- 점 곱은 식에 들어가는 모든 숫자가 벡터가 아닌 그냥 숫자라는 특징이 있다 (그래서 안으로 계산한다고 내적)
- 점 곱의 다른 이름 : "벡터 말고 그냥 숫자" 연산
  여기서 방향이 없고, 정해진 크기만 있는 흔히 쓰는 보통 숫자를 수학에서는 "스칼라"라고 한다
- 그래서 점 곱의 또 다른 이름은 "스칼라 연산"