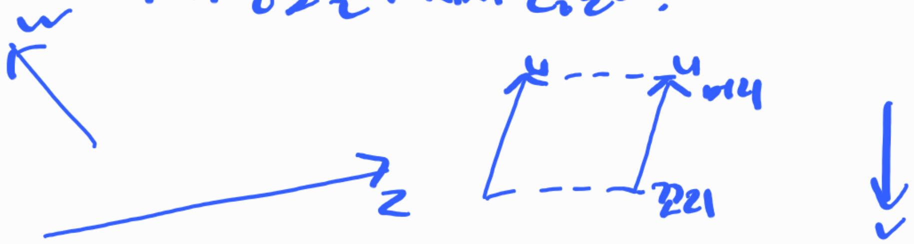
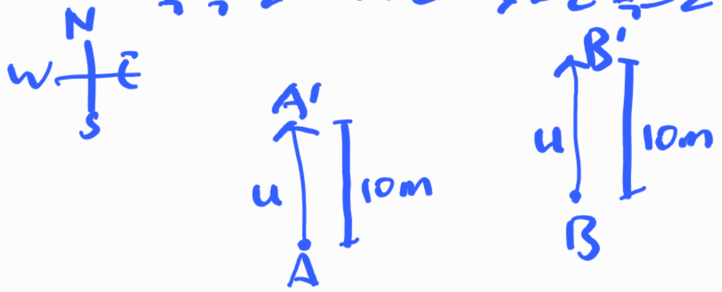
벡터 - 굵은 소문자 사용 (u, v)
행렬은 대문자 사용 (M, T)

벡터(Vector) - 방향과 크기(Direction and Magnitude)
ex) 힘, 이동경로, 속도 등
두 벡터의 방향과 길이가 같으면 같은 벡터다.
-벡터는 위치정보를 가지지 않는다.

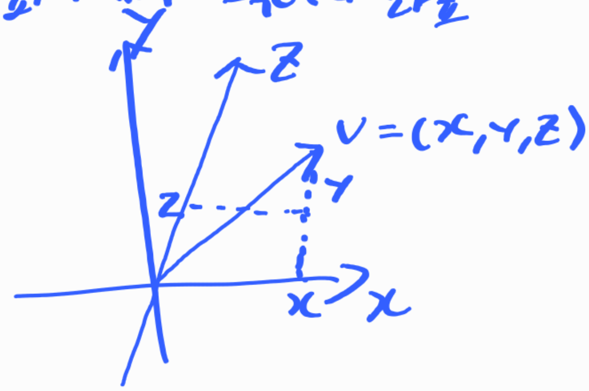


변위(Displacements) - 벡터를 가지고 이동하는 행위
ex) 북쪽으로 10m 간다, 오른쪽으로 연다 등

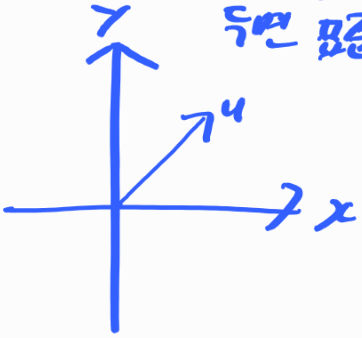


u = 북쪽으로 10m 이동을 의미한다

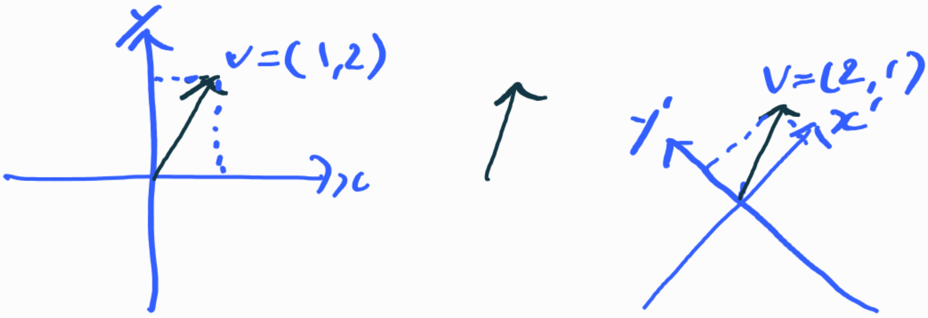
좌표계에서 벡터의 좌표



표준 위치 - 벡터는 위치가 없기에
벡터를 그릴 때 꼬리를 원점에
두면 표준위치에 있다고 한다



좌표계와 같은 용어들 = Frame, frame of reference, space, coordinate System
! 같은 값을 다른 좌표계로 표현가능 ex) 섭씨 온도와 화씨 온도



동일 벡터라도 좌표계에
따라 다른 값으로
표현될 수 있다.

기본적인 벡터 연산

$u = (u_x, u_y, u_z) \quad v = (v_x, v_y, v_z)$

- 각각의 요소가 같으면 같은 벡터다.

$u_x = v_x,\ u_y = v_y,\ u_z = v_z$

- 벡터 더하기

$u + v = (u_x + v_x,\ u_y + v_y,\ u_z + v_z)$

- 스칼라 곱하기

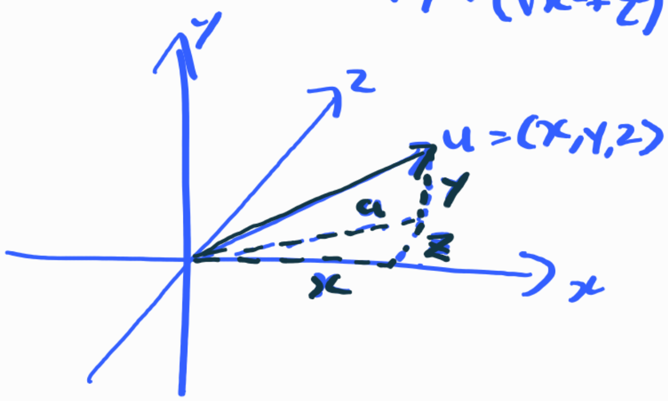
$ku = (ku_x,\ ku_y,\ ku_z)$

- 빼기는 -를 곱한 후 더하기

$u - v = u + (-1 \cdot v) = (u_x - v_x,\ u_y - v_y,\ u_z - v_z)$

벡터의 길이 = 피타고라스의 정리

$\|u\| = \sqrt{y^2 + a^2} = \sqrt{y^2 + (\sqrt{x^2 + z^2})^2} = \sqrt{x^2 + y^2 + z^2}$



단위 벡터 = Normalizing

$\hat{u} = \frac{u}{\|u\|} = \left(\frac{x}{\|u\|}, \frac{y}{\|u\|}, \frac{z}{\|u\|}\right)$

$\|\hat{u}\| = \sqrt{\left(\frac{x}{\|u\|}\right)^2 + \left(\frac{y}{\|u\|}\right)^2 + \left(\frac{z}{\|u\|}\right)^2}$

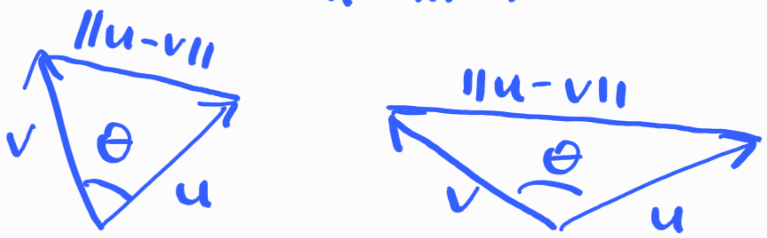
$= \frac{\sqrt{x^2 + y^2 + z^2}}{\sqrt{\|u\|^2}} = \frac{\|u\|}{\|u\|} = 1$

내적 (Dot Product) = 결과로 스칼라 값이 나온다.

$u = (u_x, u_y, u_z)$ $v = (v_x, v_y, v_z)$

$u \cdot v = u_x v_x + u_y v_y + u_z v_z$ $u \cdot v = v \cdot u$

$= \|u\|\|v\| \cos\theta$



u, v가 모두 Unit Vector일 때 Dot Product로 각도계산을 많이 한다

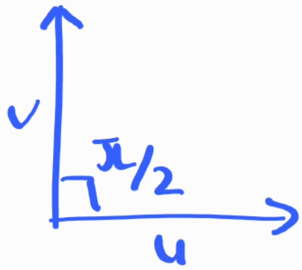
Radian을 주로 사용한다.

- 이때 각도의 범위는 $0 \le \theta \le \pi = 180$
- 두 벡터가 수직이면 $u \cdot v = 0$
- $u \cdot v > 0$ 이면 각도가 90도보다 작다(예각)
- $u \cdot v < 0$ 이면 각도가 90도보다 크다(둔각)

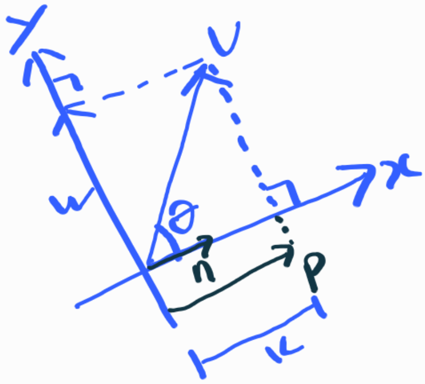
ㄴ이 성질을 이용해 $u \cdot v > 0$이면 u와 v는 같은 방향을 가르키고 $u \cdot v < 0$이라면 반대 방향을 가르킨다고 할 수 있다

(ex: 삼각형의 앞면, 뒷면 판별)

(두 벡터의 Dot Product가 0이면 Orthogonal)



Orthogonal Projection (수직으로 투영) - 어떤 벡터를 다른 벡터에 투영



$P = proj_n(v)$, v를 n에 투영

$= k\hat{n}$, k = Dot Product로 음수도 가능

$= (\|v\|\|\hat{n}\| \cos\theta)\hat{n}$, $\|\hat{n}\| = 1$이기에 $\|\hat{n}\|$이 곱해진 것처럼 적을 수 있음

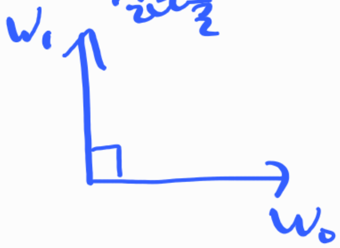
$P = (v \cdot \hat{n})\hat{n}$

$$= \frac{(v \cdot n)}{\|n\|^2} n$$

Orthogonalization

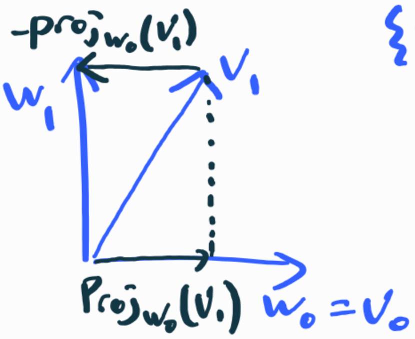
- Orthogonal Set :
  - $\{v_0, \cdots, v_{n-1}\}$의 모든 벡터들이 수직인 경우,
  - Dot Product가 0일 경우
  - 모든 벡터들이 단위 벡터이면 orthonormal

결과물



2D example) 임의의 벡터 집합 $\{v_0, v_1\}$을 Orthogonal Set $\{w_0, w_1\}$로 변환.

벡터가 모두 서로 달라야한다. 그리고 0 벡터가 없어야한다.



$$w_0 = v_0$$

$$w_1 = v_1 - proj_{w_0}(v_1)$$

2D는 집합 안의 벡터가 2개, 3D는 3개가 된다.

Gram-Schmidt Orthogonalization

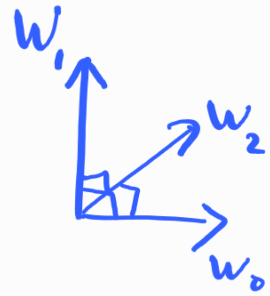
- 3D로 확장
  - 임의의 벡터 집합 $\{v_0, v_1, v_2\}$을 Orthogonal 집합 $\{w_0, w_1, w_2\}$로 변환

$$w_0 = v_0$$

$$w_1 = v_1 - proj_{w_0}(v_1)$$

$$w_2 = v_2 - proj_{w_0}(v_2) - proj_{w_1}(v_2)$$

결과물 (좌표계 같다)



- 임의의 차원 $\{v_0, \cdots, v_{n-1}\}$을 $\{w_0, \cdots, w_{n-1}\}$로
  1) $w_0 = v_0$
  2) For $1 \le i \le n-1$, $w_i = v_i - \sum_{j=0}^{i-1} proj_{w_j}(v_i)$
  3) Orthonormal Set을 구한다면 $\hat{w}_i = \frac{w_i}{\|w_i\|}$

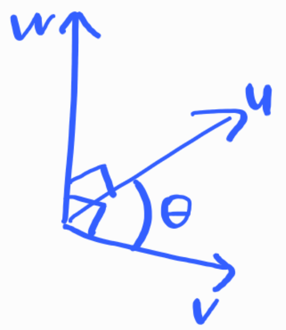
     ex) $\hat{w}_0 = \frac{w_0}{\|w_0\|}, \hat{w}_1 = \frac{w_1}{\|w_1\|}, \cdots$

외적 (Cross Product) - 결과가 벡터로 나온다.

$u = (u_x, u_y, u_z)$ $v = (v_x, v_y, v_z)$

$w = u \times v = (u_y v_z - u_z v_y,\ u_z v_x - u_x v_z,\ u_x v_y - u_y v_x)$

이때 $w \cdot u = w \cdot v = 0$



w = u와 v 둘 모두와 수직인 벡터

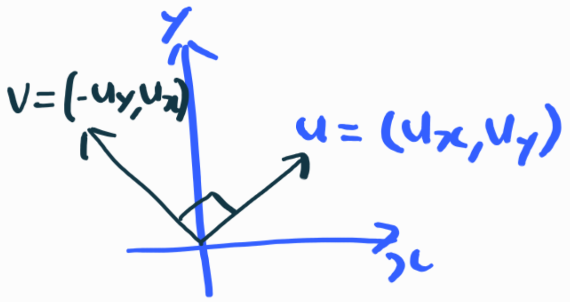
$\theta \neq 0,\ \theta \neq 90,\ \theta \neq 180$

즉 θ는 0°, 90°, 180° 이면 안된다.

두 벡터 u와 v가 있을 때 평면을 정의할 수 있다.
이때 평면에 수직인 벡터를 구하기위해 사용한다.

! $u \times v \neq v \times u$ 방향이 다른 수직 벡터가 계산된다.

Pseudo 2D Cross Product - 한 벡터에 수직인 벡터 구하기



$u = (u_x, u_y)$

$v = (-u_y, u_x)$, 벡터 u를 90로 회전한것과 같다.

$u \cdot v = u_x \cdot -u_y + u_y \cdot u_x = 0$

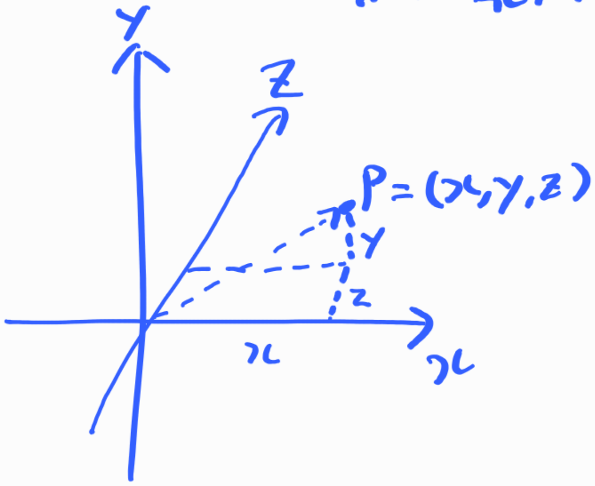
Orthonomalization with the Cross Product

Set $\hat{w}_0 = \frac{v_0}{\|v_0\|}$

Set $\hat{w}_2 = \frac{\hat{w}_0 \times v_1}{\|\hat{w}_0 \times v_1\|}$, $\hat{w}_2$와 $\hat{w}_0$이 수직, $\hat{w}_2$과 v_1도 수직

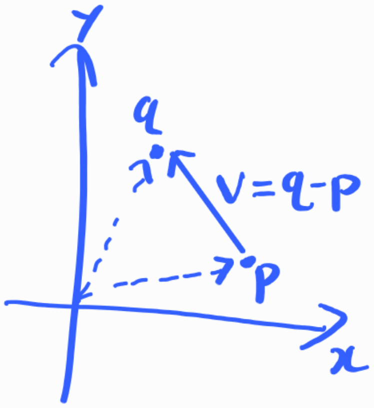
Set $\hat{w}_1 = \hat{w}_2 \times \hat{w}_0$, $\hat{w}_1$과 $\hat{w}_2$가 수직, $\hat{w}_1$과 $\hat{w}_0$이 수직

포인트(Point) - 특정 좌표계에 대해서 포인트를 표현할 때
위치 벡터 사용



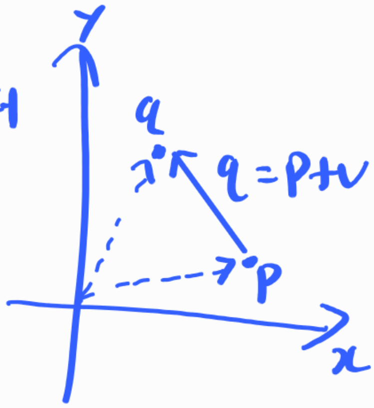
좌표계 안에서 포인트의 상대적인 위치를 표현할 때 위치 벡터를 사용
(원점에서 시작)

포인트 연산



v=변위

포인트 - 포인트 = 벡터
(점)　(점)



점+벡터=점