

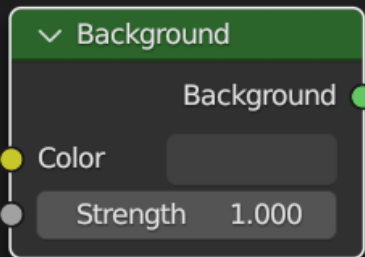
030강 배경과 조명에서의 노드 사용

배경과 조명에서 사용하는 노드
조명 좌표 변환
월드 좌표 변환으로 움직이는 하늘 만들기

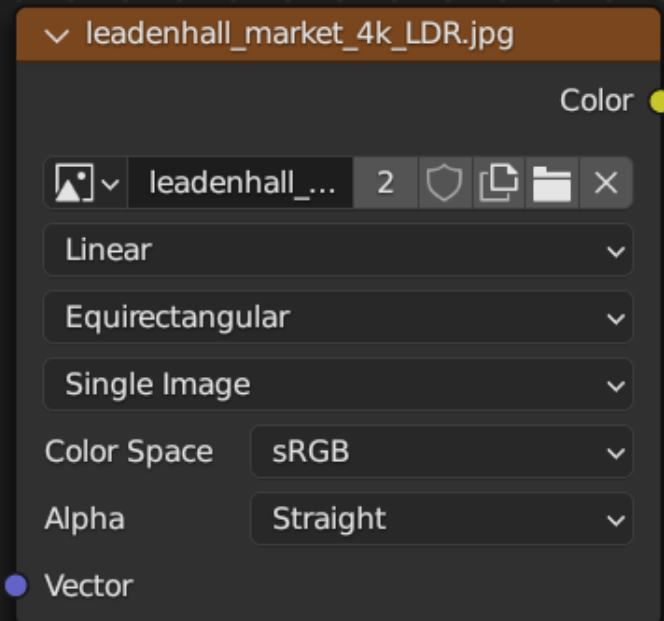


배경에서 쓸 수 있는 노드

배경에서 사용할 수 있는 노드가 몇가지 있습니다.



Background : Emission과 비슷하게 작동합니다.
입력받은 이미지의 밝기를 조절합니다.

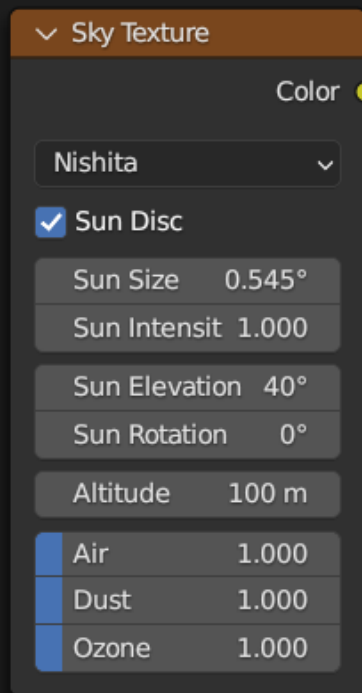


Environment Texture :
이미지 텍스처 노드와 비슷하지만, 입력받는 좌표와 사용하는 이미지의 종류 모두 다릅니다.

기본값으로 Equirectangular 이미지를 받지만, Mirror Ball 이미지도 사용 가능합니다.

배경에서 쓸 수 있는 노드

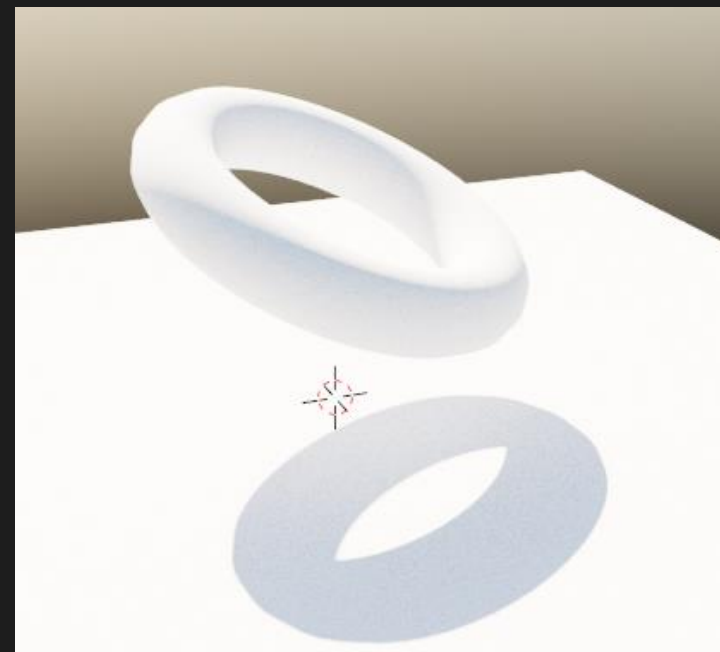
Sky Texture



하늘 배경을 생성합니다.

기본값은 Nishita이며, Preetham 등, 이전 버전도 선택 가능하지만 추천하지 않습니다.

Eevee에서도 작동하지만 태양은 생기지 않습니다.
(Eevee에서는 Hdri를 통한 그림자가 생기지 않으므로 어차피 소용 없음.)

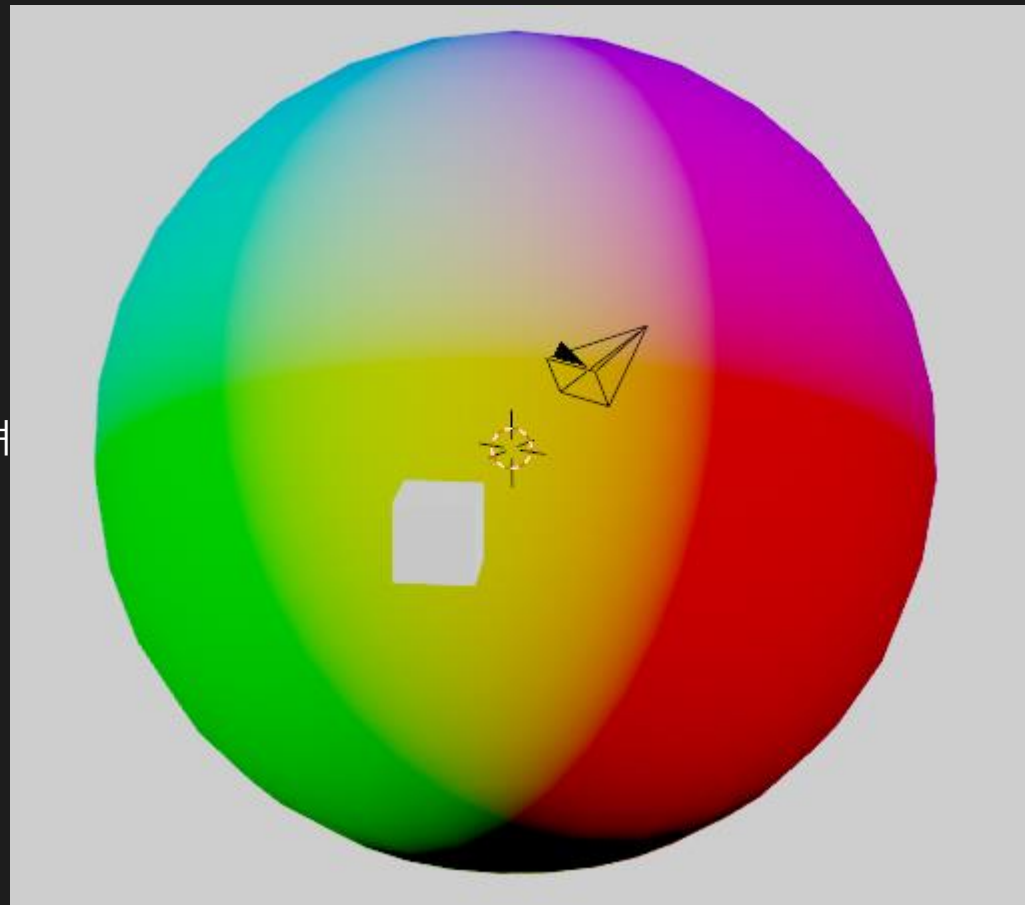


배경 좌표

일반적으로 사용하는 배경 좌표는 방향만 존재하는 출력입니다.

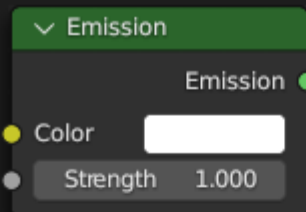
구면 위의 점을 나타낸 것과 비슷하여, 크기는 항상 1입니다.

이러한 특수성 때문에 일반적인 image Texture 노드는 정상적으로 작동하지 않으며 Environment texture를 써야 좌표를 제대로 해석해 이미지를 배경에 매핑합니다.



조명에서 쓸 수 있는 노드 (Cycles only)

Cycles일 때만 조명에서 노드를 사용할 수 있습니다.

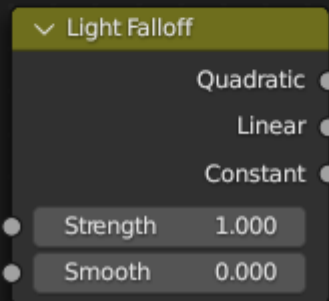


Emission : 조명에서는 Emission만 사용할 수 있습니다.



Wavelength, Blackbody : 노드 자체는 조명이 아니어도 사용할 수 있지만, 그 특성상 조명에서 많이 유용합니다.

Wavelength는 파장에 따른 색을, Blackbody는 온도에 따른 색을 출력합니다.



Light Falloff : 일반적으로 거리 제곱에 반비례하는 빛의 밝기 특성을 변경합니다.

Quadratic : 빛의 세기가 거리 제곱에 반비례합니다. (이는 Light Falloff을 사용하지 않았을 때의 기본값입니다.)

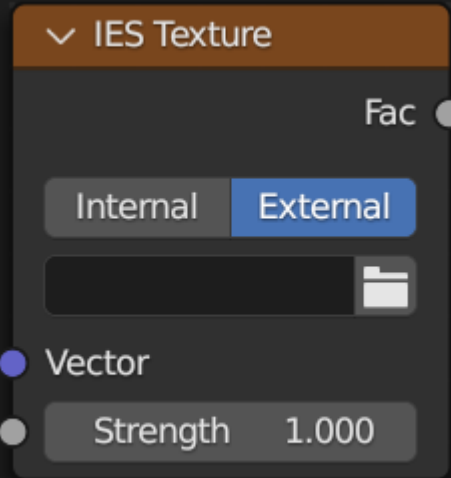
Linear : 빛의 세기가 거리에 반비례합니다. 빛은 더 멀리 퍼집니다.

Constant : 빛의 세기가 거리에 영향을 받지 않습니다. (이는 Sun light의 기본값입니다.)

Smooth : 가까운 거리에서의 밝기를 줄입니다.

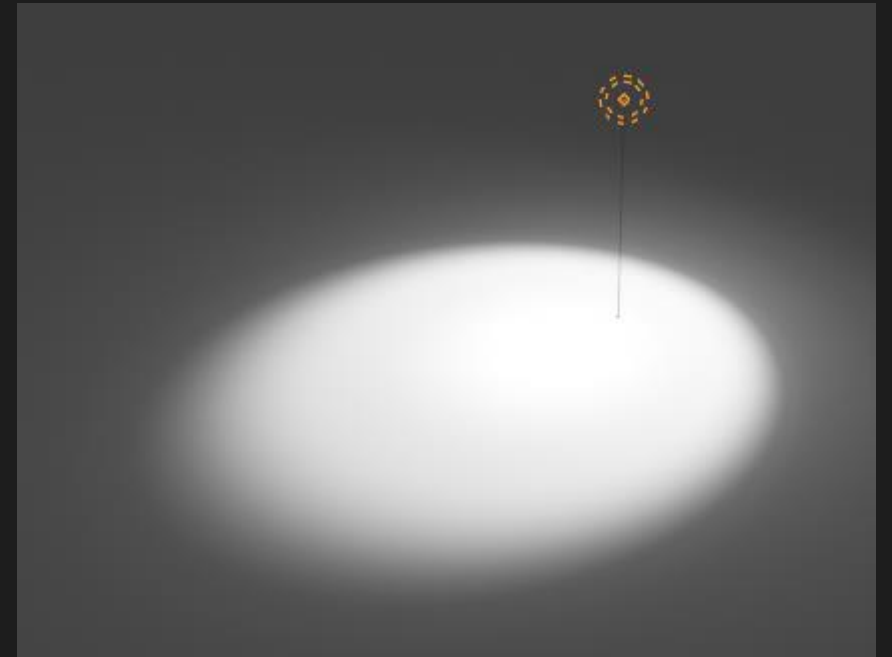
IES Texture

<https://ieslibrary.com/>



조명이 방출하는 빛의 분포 형태를 담고있는
IES 파일을 사용할 수 있습니다.

조명의 종류에 따라 어떻게 빛이 퍼지는지를 나타내며
스팟 조명을 선택하지 않아도 자동으로 한쪽으로만 빛을 내보냅니다.

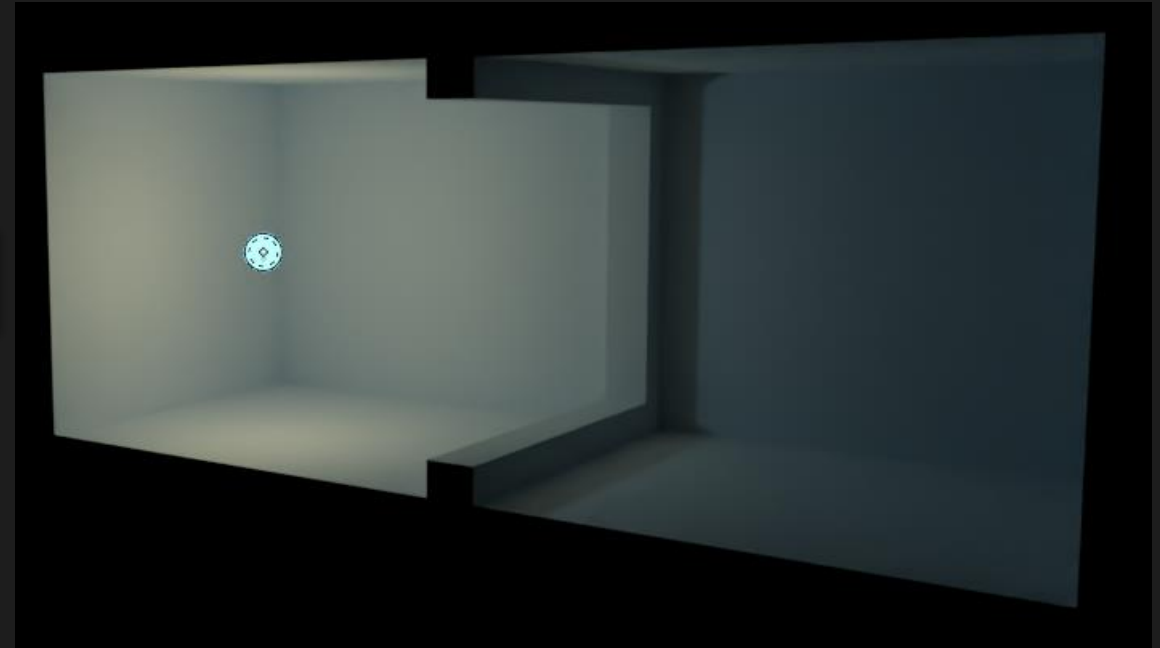
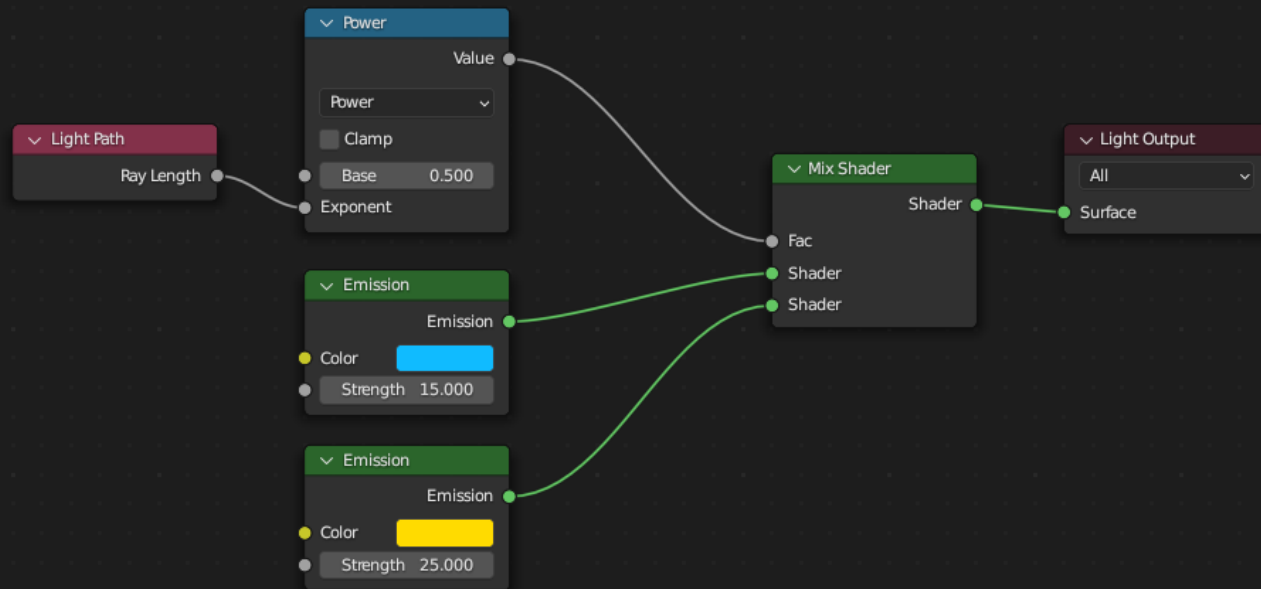


조명에서의 Light Path

배경에서는 Ray Length를 제외하곤 자유롭게 쓸 수 있는 반면,

조명에서는 Ray Length, Ray depth만 정상 작동합니다.

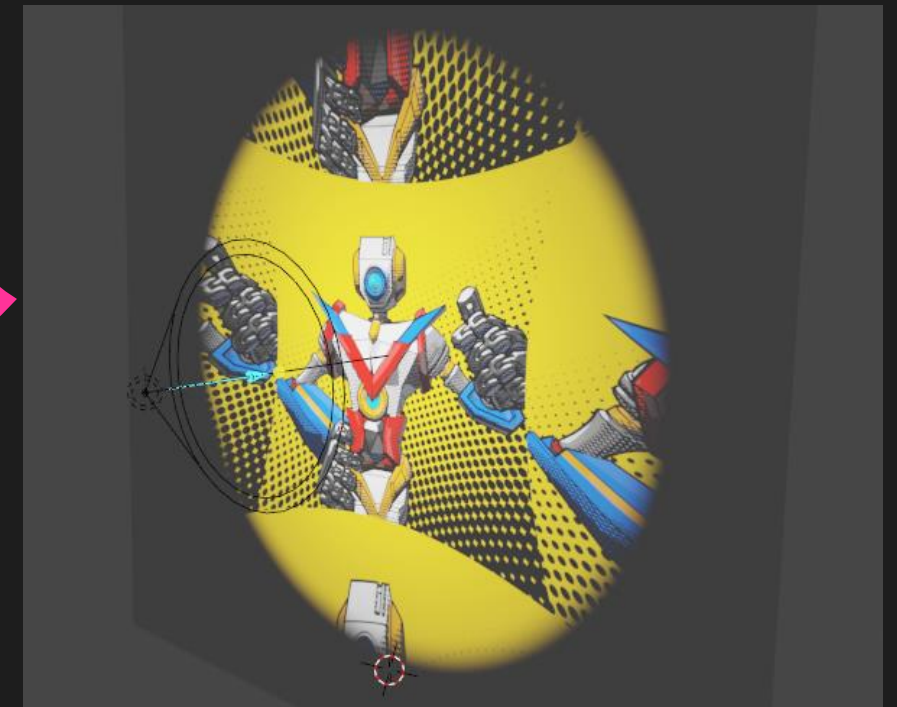
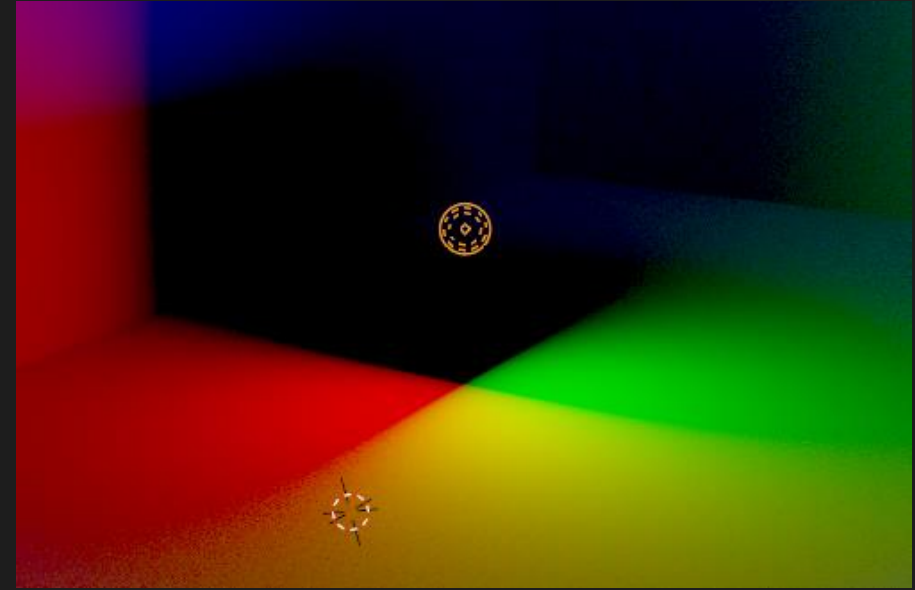
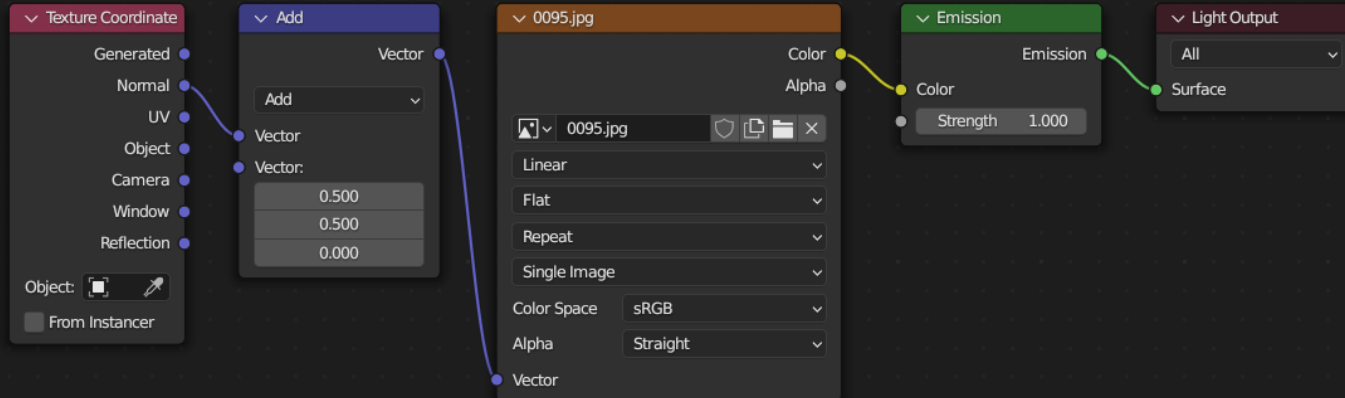
Ray Length를 이용할 때는 Power의 Exponent 연결이 도움됩니다.



조명의 텍스처좌표

조명도 배경처럼 텍스처 좌표가 방향을 출력합니다.
(따라서, hdri를 꽂으면 잘 작동합니다..)

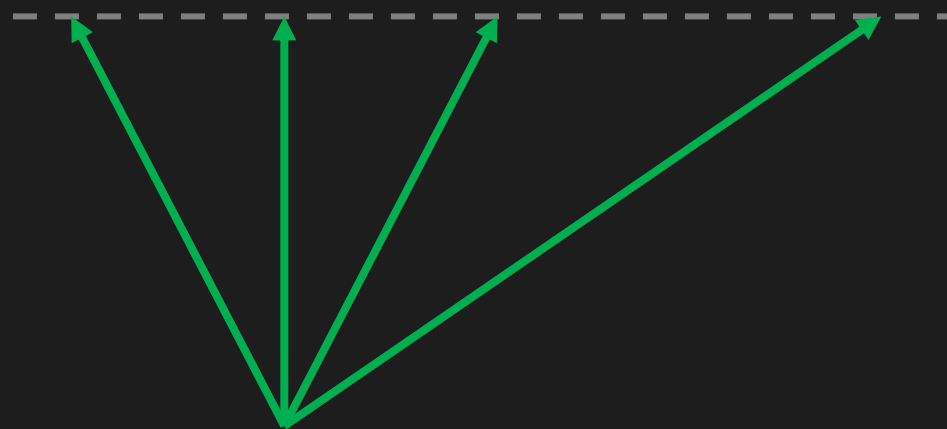
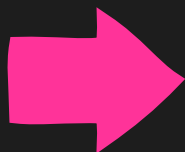
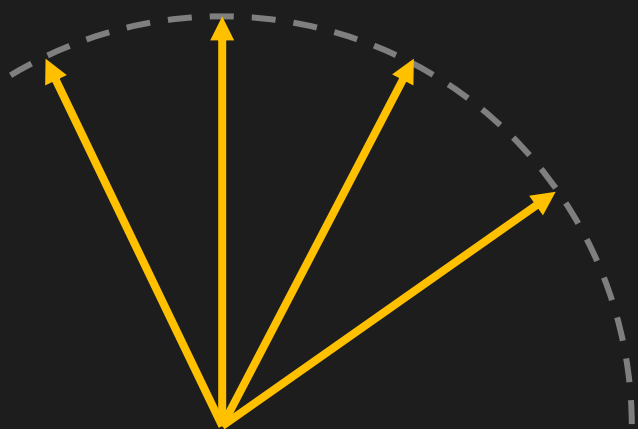
이 출력은 좁은 범위에서는 보통의 좌표처럼 보이나
범위가 넓어지면 왜곡이 발생합니다.
이를 해결하기 위해서는 좌표를 수정해야 합니다.



좌표 변환

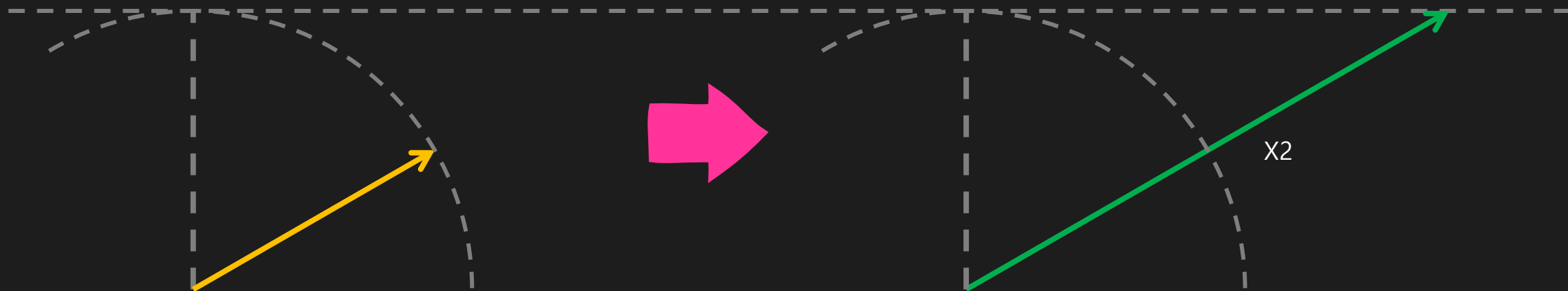
이 부분은 노드 연결만 알아두셔도 괜찮습니다.

우리가 하고자 하는 건 구형 좌표를 평면으로 펼치려는 것입니다.



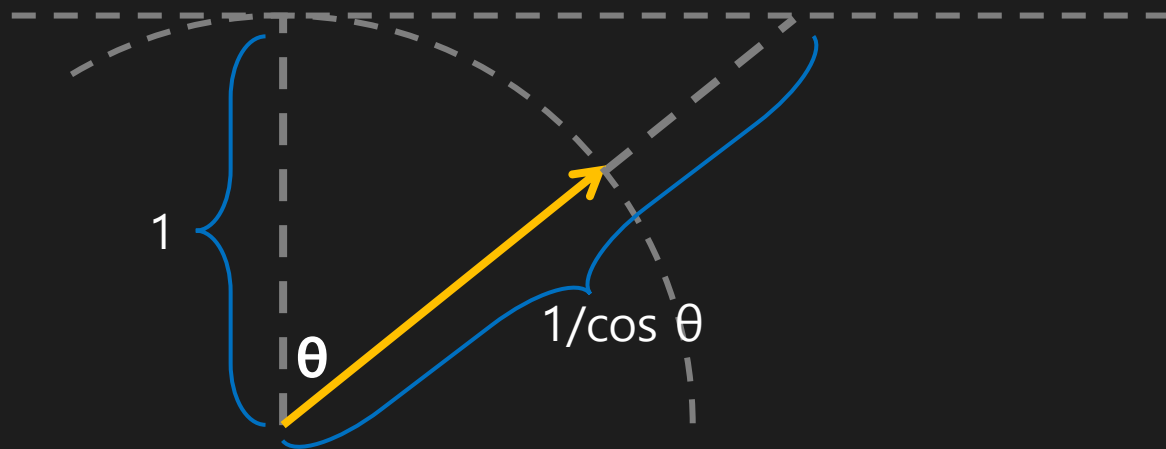
좌표 변환

예를 들어 이 경우 원래 좌표에 2배 정도 곱해줘야 됩니다.



좌표 변환

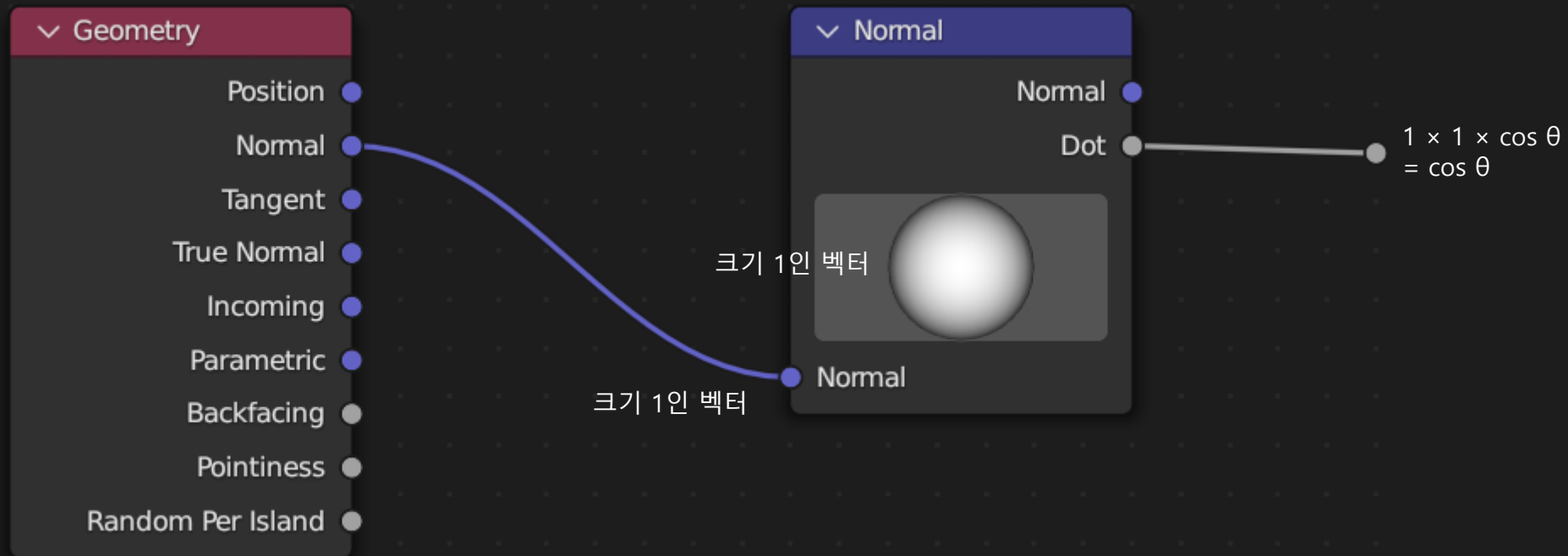
구체적으로, 각도 θ 만큼 기울어진 벡터는, $1/\cos \theta$ 배 커져야 합니다.



즉 θ 를 구하면 문제를 풀 수 있습니다.

좌표 변환

$\cos \theta$ 는 벡터의 내적에서 구할 수 있습니다. 내적은 두 벡터의 크기에 코사인을 곱한 값이기 때문입니다.
그런데 우리의 계산에서 사용하는 벡터는 모두 크기 1이므로, 내적 그 자체로 코사인 값이 됩니다.



좌표 변환

최종 연결은 다음과 같습니다.

※배경 좌표도 같은 원리이므로, 이와 동일하게 사용할 수 있습니다!

