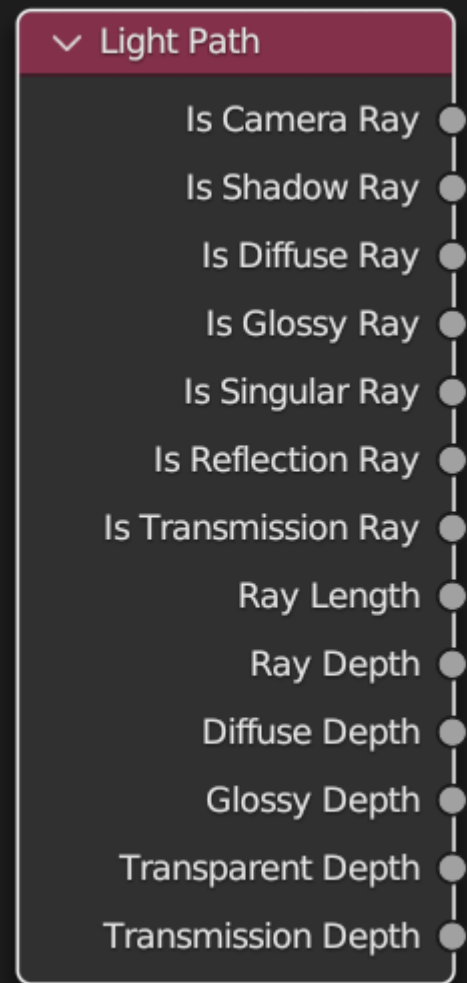


# 028강 Light Path

Cycles 엔진이 이미지를 만드는 원리

Light Path 노드



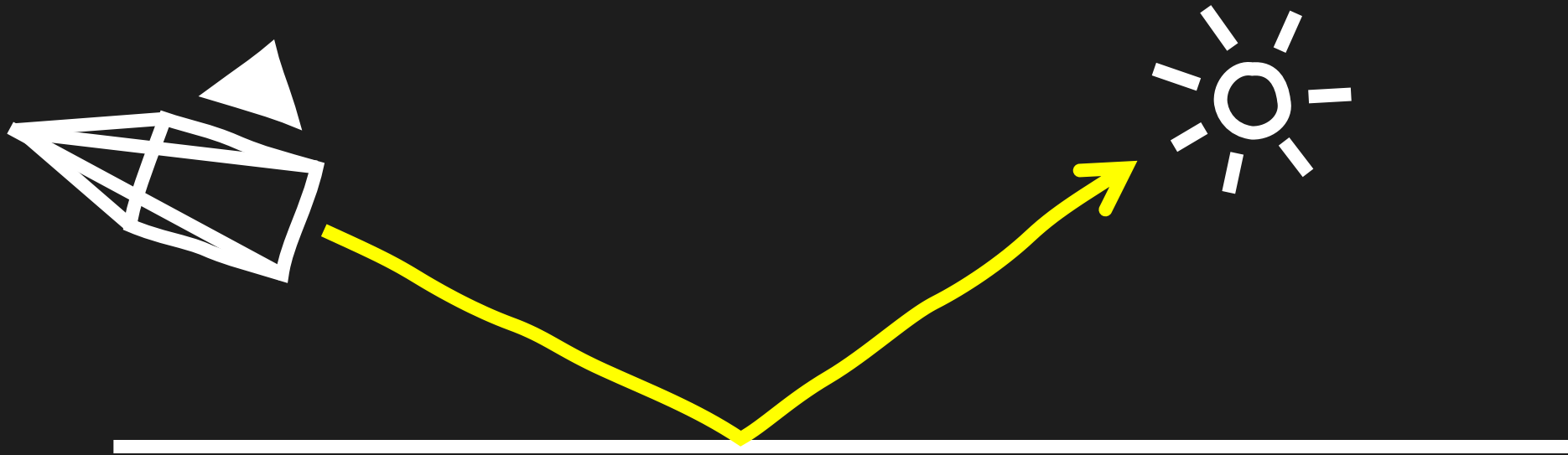
# Cycles의 기본 원리

## 광선은 카메라에서 나옵니다

Cycles를 비롯한 대부분의 Path tracing은 카메라에서 Ray를 쏘서 광원에 닿을때까지를 연산합니다.

-연산은 어딘가에 닿을 때마다 일어납니다.

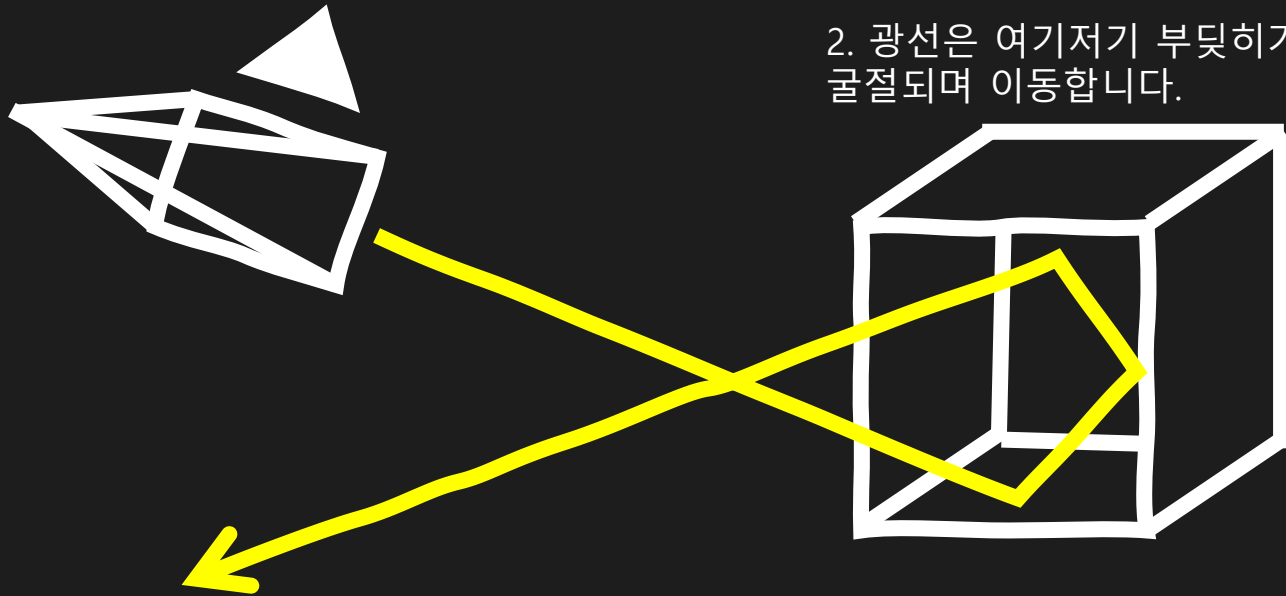
- 표면에 닿아서 일어난 연산은 표면 뿐만 아니라 그 전 단계에도 영향을 미칩니다.



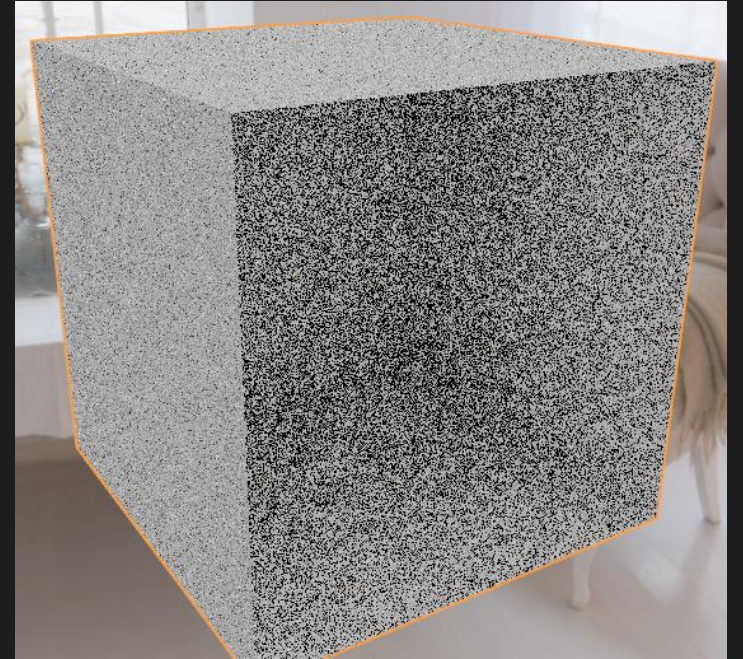
# Cycles의 기본 원리

## 연산의 기본 흐름

1. 광선이 카메라에서 출발합니다.



2. 광선은 여기저기 부딪히거나, 굴절되며 이동합니다.



연산이 종료되면 그 결과를 처음 쏘아올린 방향(위치)에 기록합니다.

3. 광원에 닿으면 연산은 종료됩니다.

연산이 종료되는 시점은

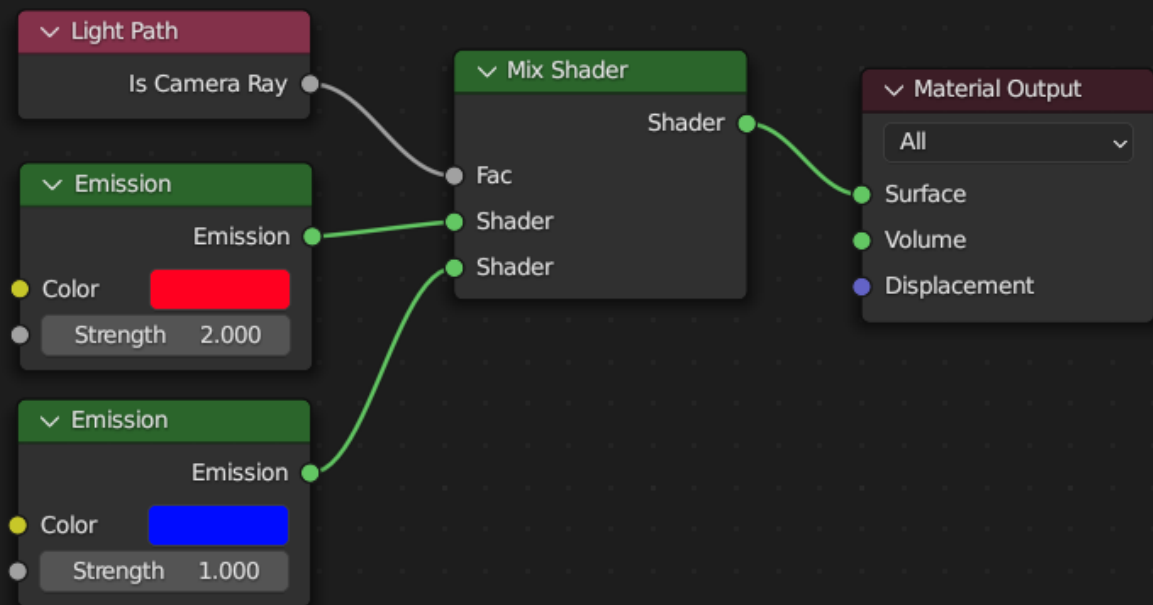
- Light Object에 닿거나
- Emission 재질에 닿거나
- 배경에 닿거나
- 연산 횟수를 초과하는 경우가 있습니다.

# Light Path

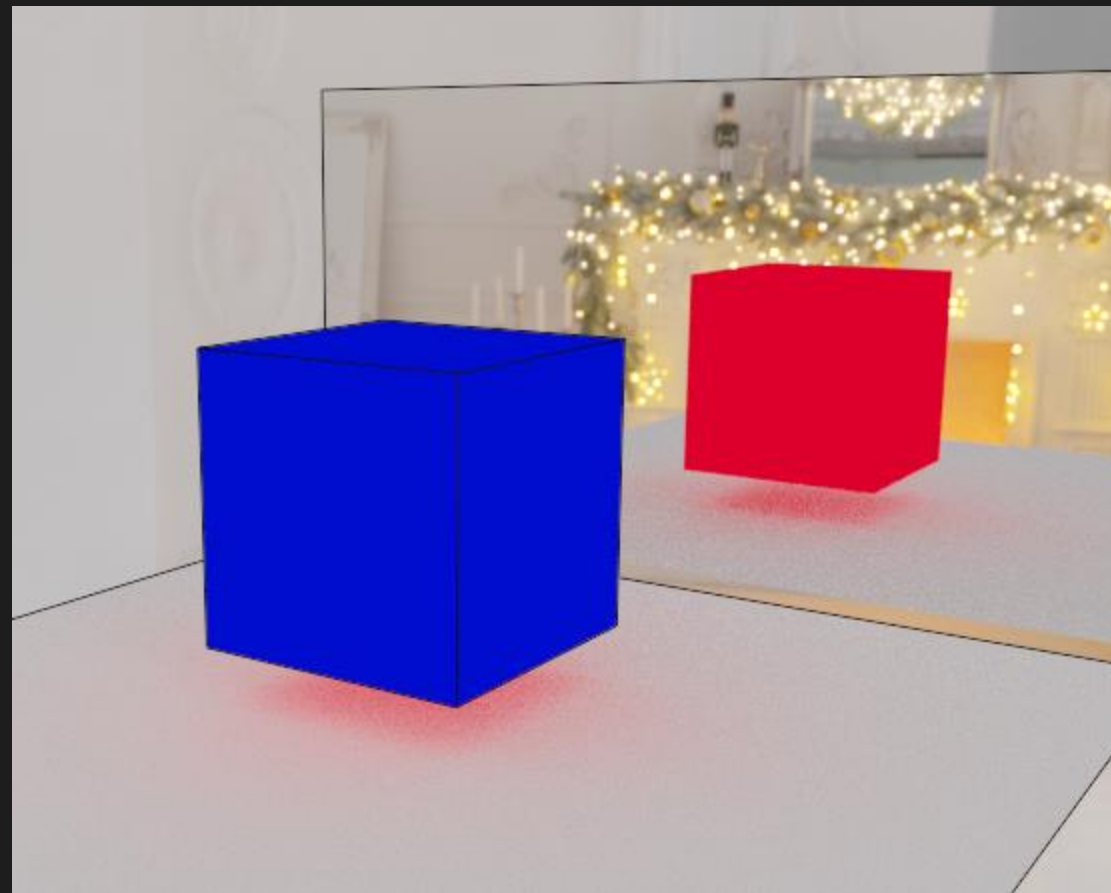
## Camera ray

카메라에서 막 출발했을 때의 광선을 Camera Ray라고 합니다.

무언가와 상호작용한 시점부터 Camera ray가 아닙니다. (투명 재질 제외)



오브젝트에 적용하여, 카메라로 볼 때만 파란색으로 만들었습니다.  
그림자나 비치는 모습 등 나머지 연산에선 빨간색으로 작동합니다.

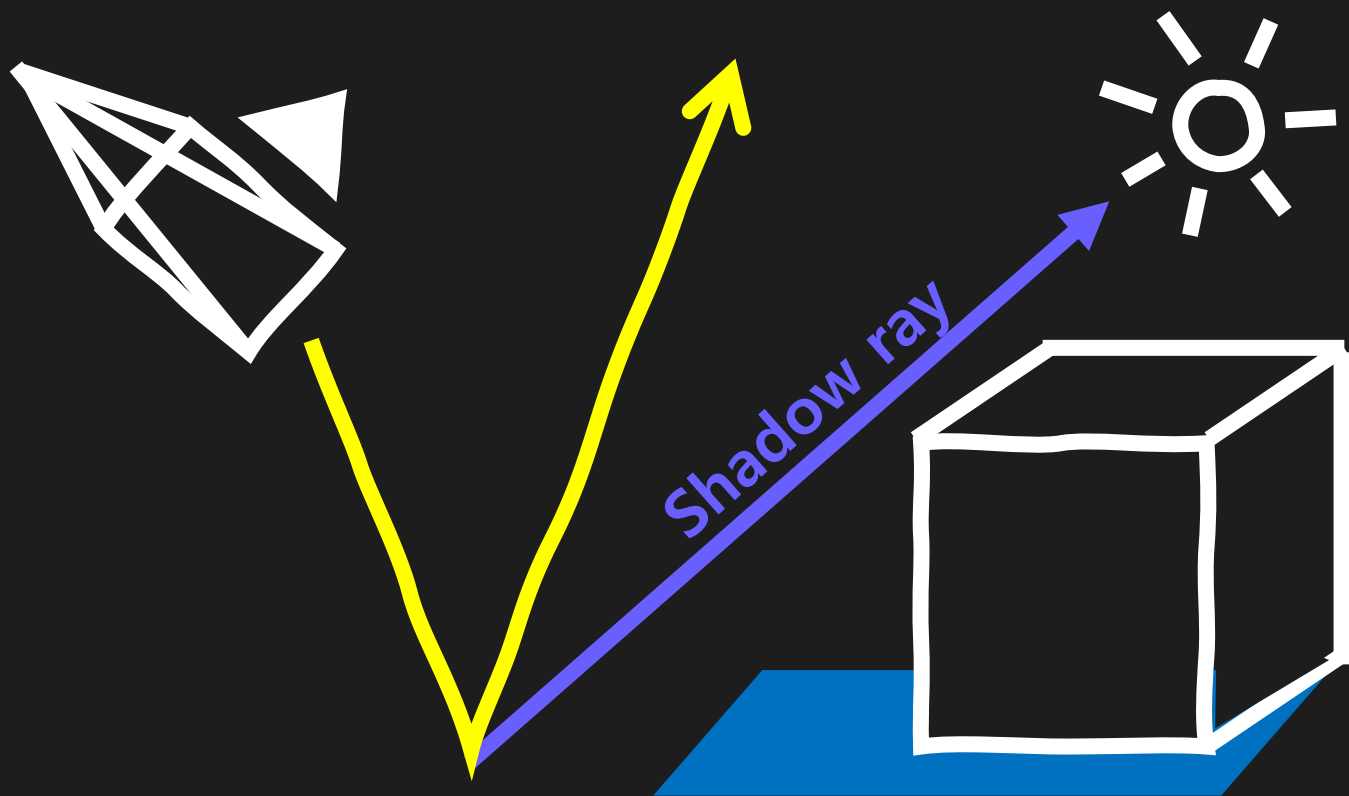


# Light Path

## Shadow ray

바운스할 때마다, 반사/ 굴절의 원리와 상관없이, 광원을 향해 직행하는 광선이 존재합니다.  
이렇게 분리된 광선을 Shadow Ray라고 합니다.

- Shadow Ray는 다시 반사되거나 하지 않고, 오로지 광원에 닿는지/아닌지만 판단합니다.
- 광원에 닿으면 해당 부분을 밝게, 닿지 않으면 어둡게 표시하는 역할을 합니다. 즉, 그림자를 만듭니다.



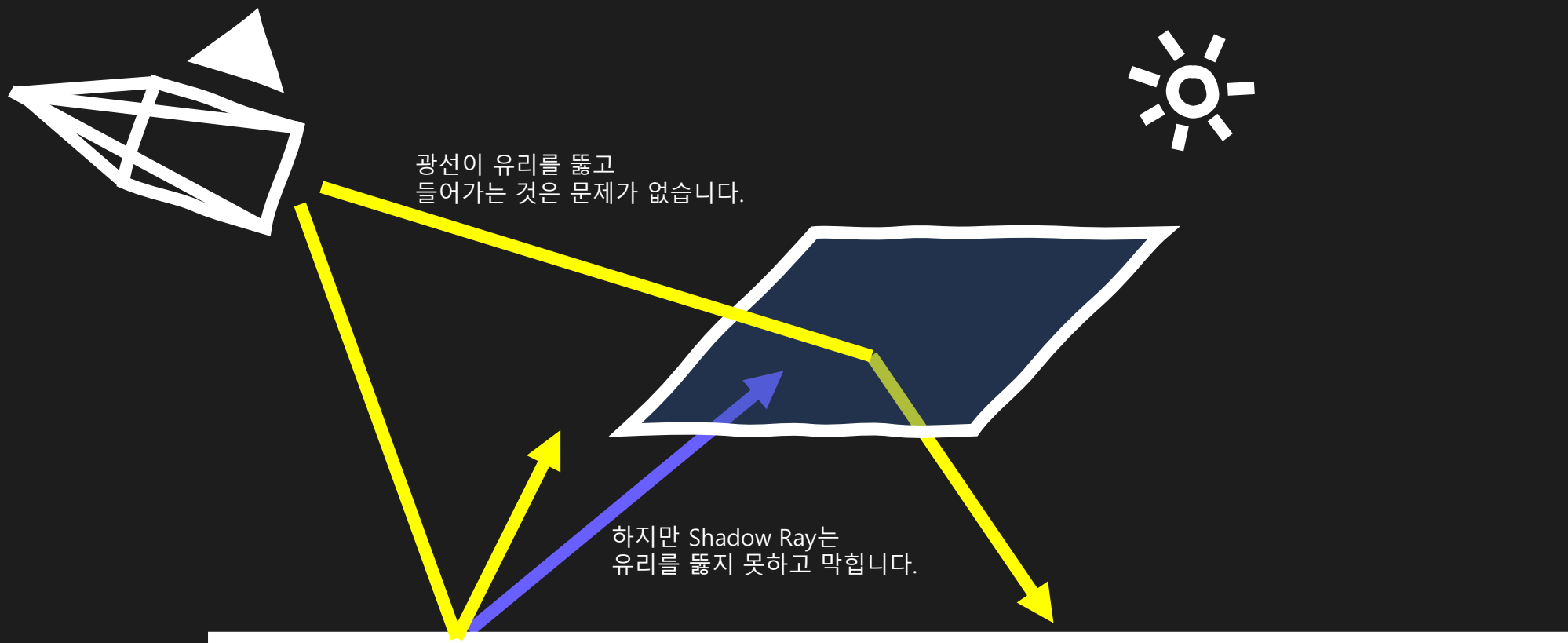
# Light Path

## Shadow ray

Shadow Ray는 오로지 투명 재질 (Transparent) 만 통과할 수 있습니다.

10강에서 알아본 유리의 문제점이 여기에 있습니다.

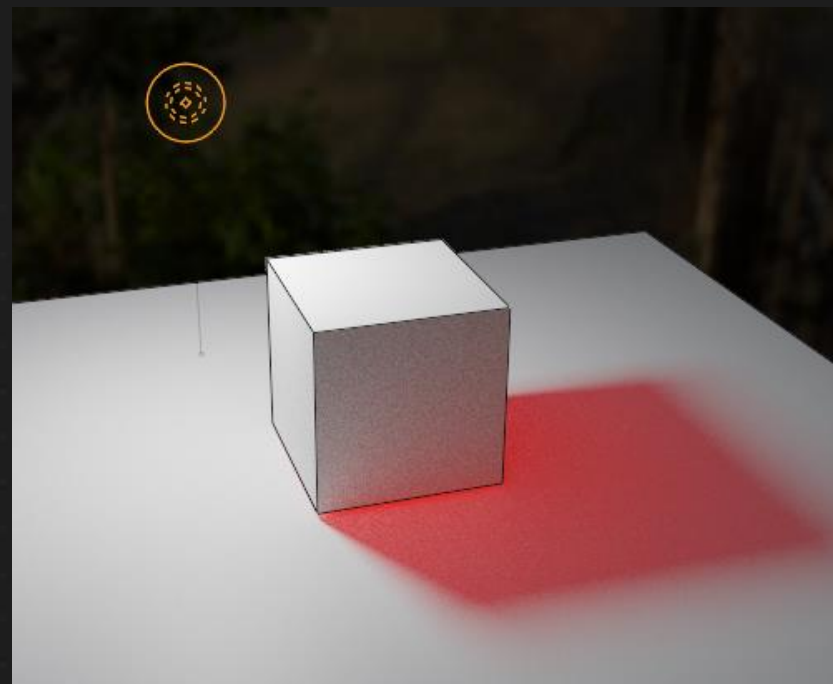
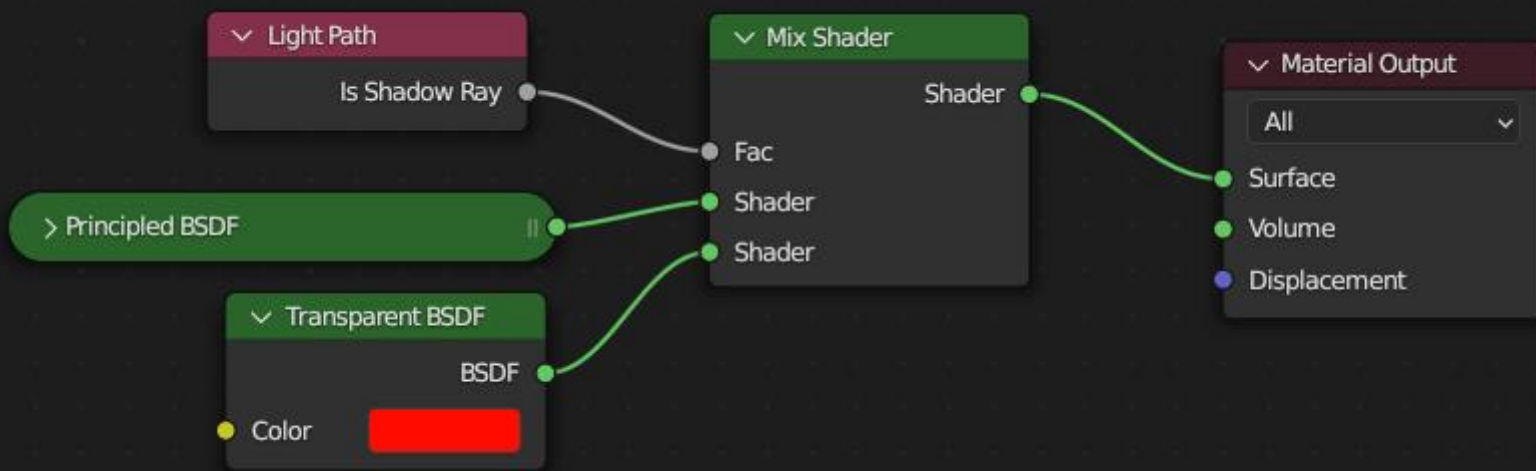
유리는 transparent가 아니기 때문에 Shadow Ray가 통과할 수 없고, 그에 따라 그림자가 생깁니다.



# 그림자 색을 바꾸려면?

그림자는 바닥에 생기지만, 그림자가 생기는 물체 쪽을 아래처럼 바꾸어야 합니다.

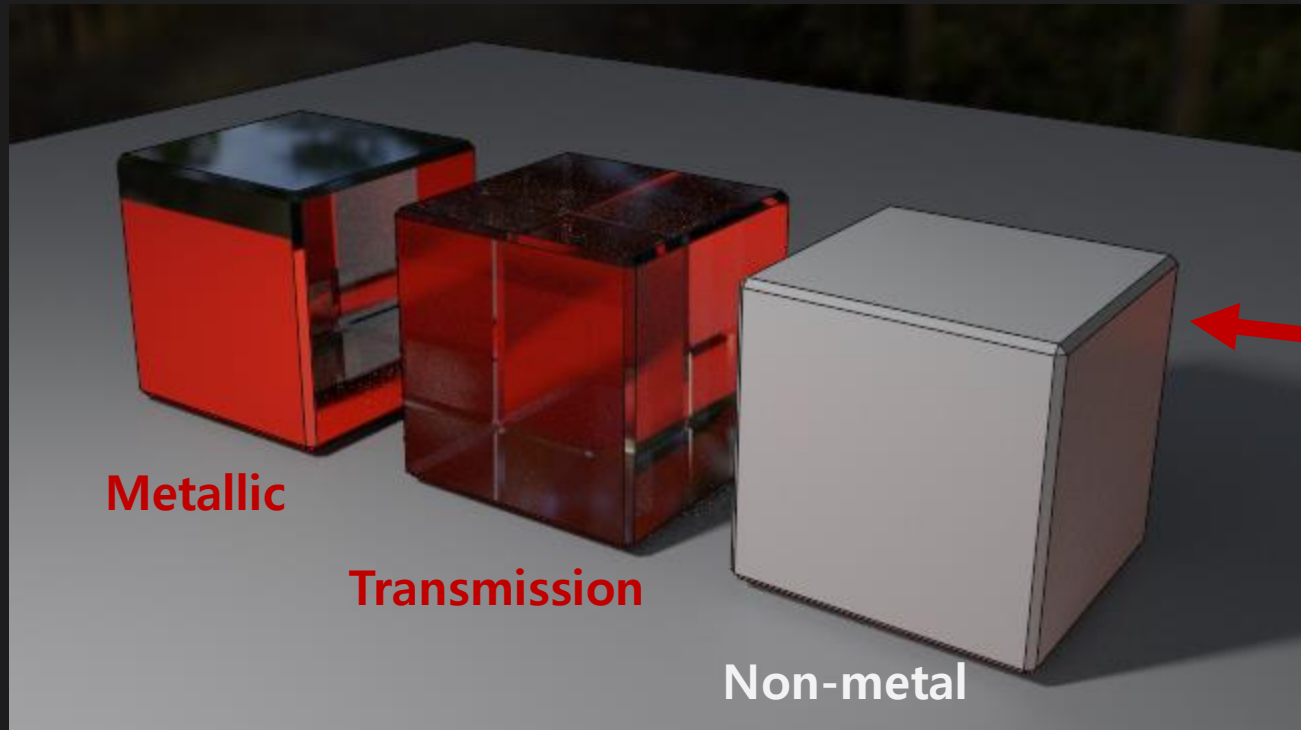
1. 카메라에서 나온 광선이, 바닥에 닿아 Shadow Ray가 분리됩니다.
2. Shadow Ray는 바닥에서 광원을 향해 날아가다 물체에 닿습니다.
3. 물체의 셰이더 연결은 아래와 같아서, Shadow Ray는 통과해서 광원에 닿습니다.
4. 해당 계산 결과는 처음 광선이 향했던 곳 : 바닥에 적용됩니다.



# Light Path

## Diffuse vs Glossy

Diffuse에 반사된 광선을 Diffuse Ray  
Glossy에 반사된 광선을 Glossy Ray라고 합니다.



**Glossy의 종류 : Metallic , Transmission, Specular**

바닥을 'Glossy'일 때 빨간색' 으로 했을때의 결과입니다.

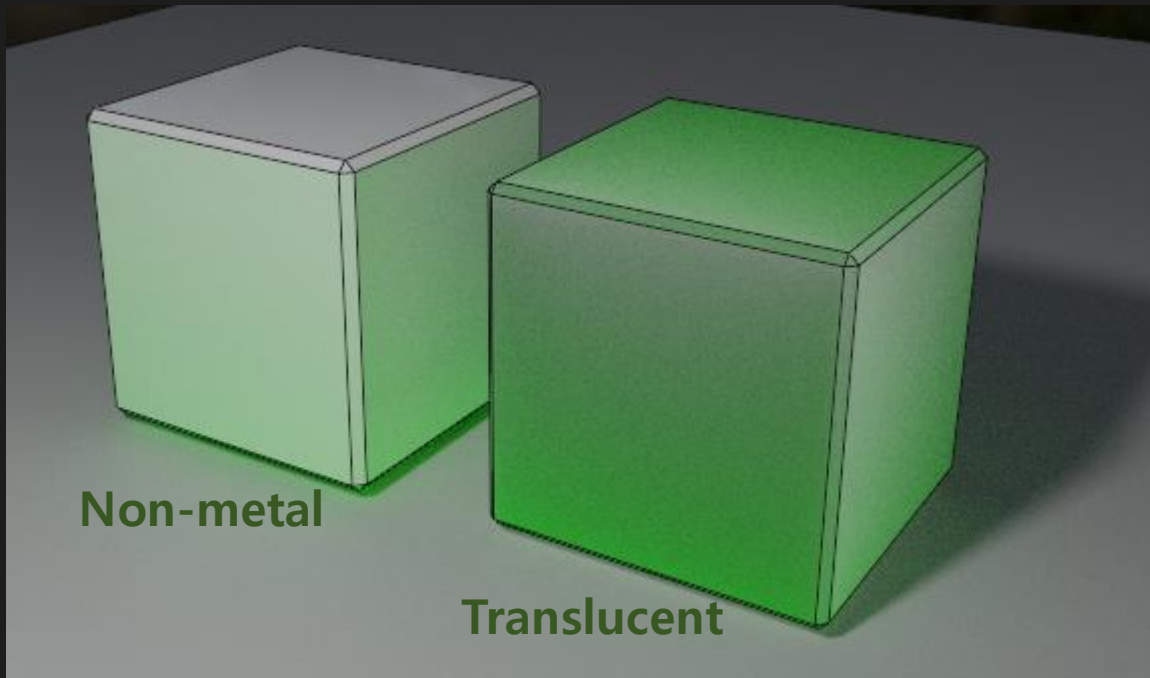
- Glossy에는 유리재질도 포함됩니다.

Principled BSDF의 Metallic을 꺼도, 광택이 있는 부분은 Glossy입니다.



# Light Path

## Diffuse vs Glossy



Diffuse의 종류 : Non-Metal, Translucent

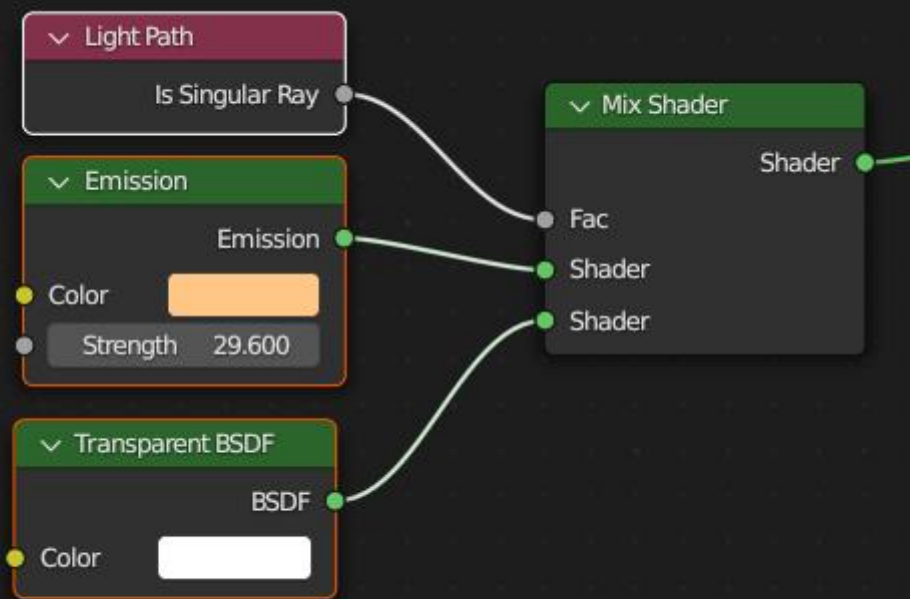
바닥을 'Diffuse일 때 초록색' 으로 했을때의 결과입니다.

- Diffuse에는 Translucent도 포함됩니다.

# Light Path

## Singular

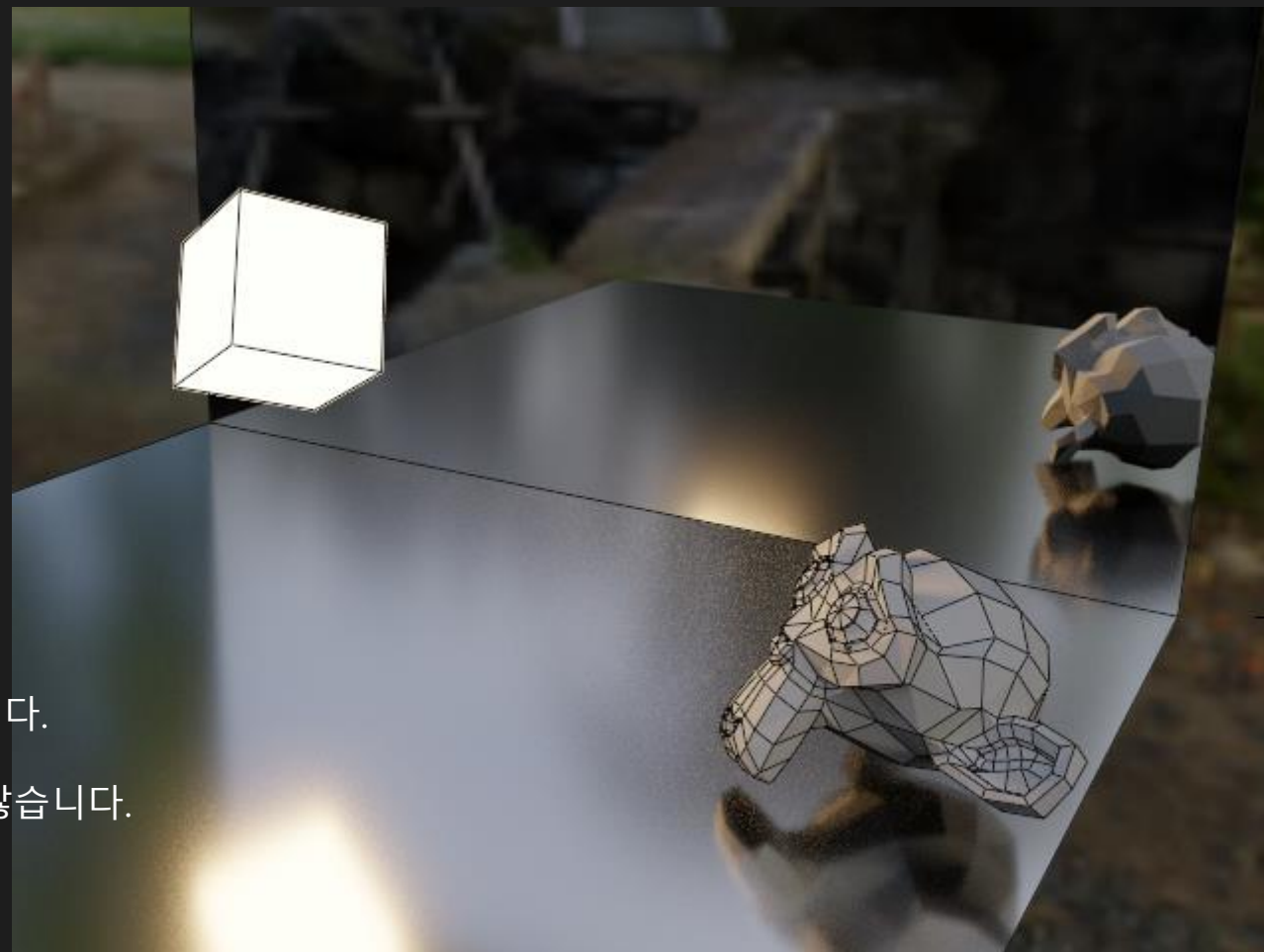
러프니스 0으로 반사된 광선을 Singular Ray라고 부릅니다.



박스를 Emission으로 사용하되, Singular일 때 투명으로 둔 예시입니다.

바닥은 박스를 반사하지만, 러프니스 0인 정면의 거울은 반사하지 않습니다.

※조명에는 사용할 수 없습니다. 자세한 것은 30강 참고.



# Light Path

## Transmission

굴절된 광선을 Transmission Ray 라고 부릅니다.

굴절 공식은 현실과 같지만, 3D에서 굴절은 매질에 의해 생기는 것이 아닙니다.

하나의 굴절 면만 존재하면 됩니다.



블렌더에선 하나의 면만 있어도 굴절이 일어납니다.

앞쪽 면을 소한 매질, 뒤쪽 면을 밀한 매질로 보고 계산합니다.

그에 따라 노멀이 뒤집히면 바깥 전체가 물속에 있는 것처럼 작동하기 때문에, 스넬의 창이 생기고 부자연스러워집니다.

※Eevee의 경우, 여러 면이 겹쳐도 한번의 굴절만 일어납니다.  
Eevee에서 유리로 창문을 만들면 안되는 이유입니다.

# Light Path

## Transmission

연산의 특성때문에 생기는 문제가 하나 있습니다.



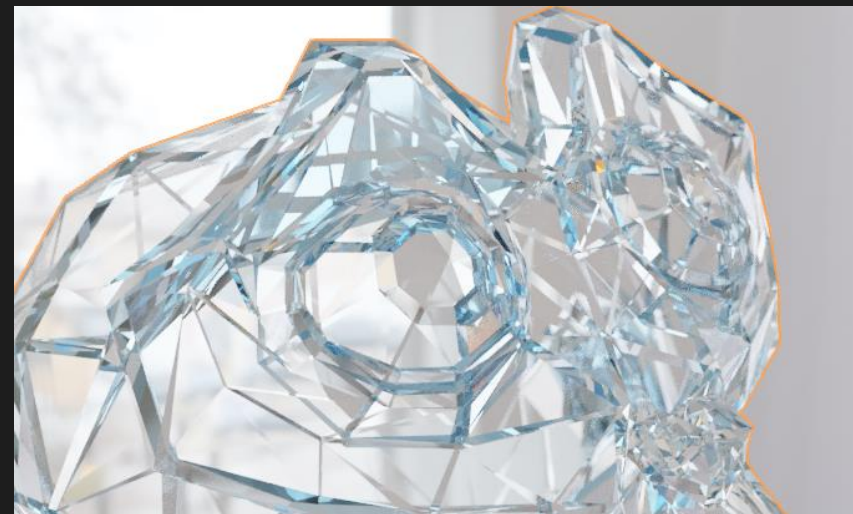
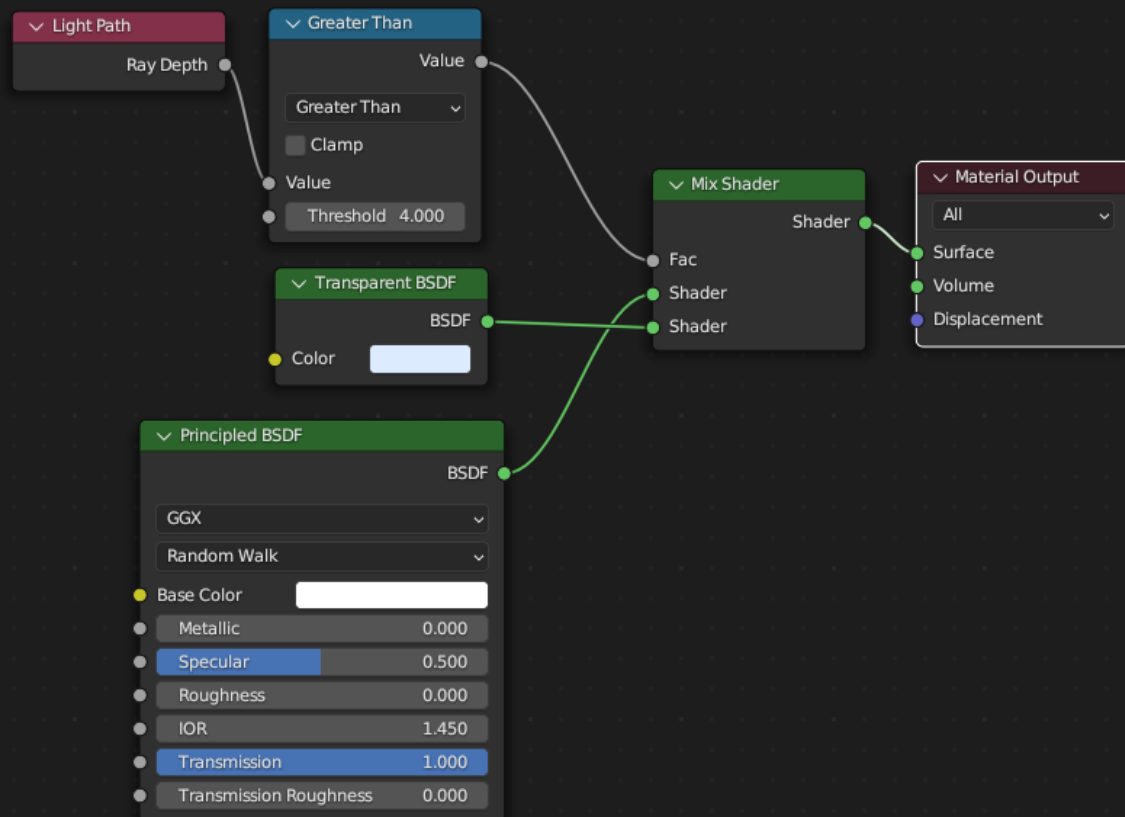
만약 유리컵 안에 물 오브젝트가 존재한다면..



물 오브젝트가 컵보다 약간 커서 면이 겹쳐지면 이렇게 됩니다.  
현실은 이쪽에 더 가깝습니다.

# 유리가 너무 어두울 때

유리 내부에서 전반사가 일어나 빛이 제대로 빠져나오지 못하면, 위쪽 이미지처럼 부분적으로 어두워집니다. 이것은 현실에서도 일어나는 일이지만, 만약 너무 과하게 어둡다면 **Depth가 충분히 클 때 Transparent로 바꿔** 통과시켜 주면 됩니다. Transparent의 색을 조절하면 유리다운 색으로 만들어줄 수도 있습니다.





# Light Path

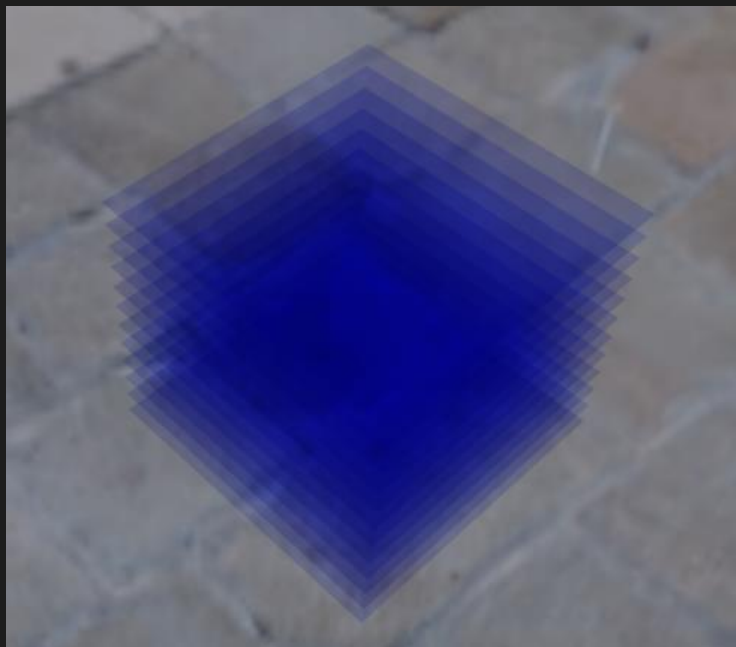
## Transparent

Transparent 특성은 오로지 Transparent BSDF에 의해서만 생깁니다. (유리재질 안됨!)

-Transparent는 Shadow ray가 통과할 수 있는 유일한 재질이며,

-바운스 횟수를 셀 때도 따로 셉니다.

-Camera Ray가 Transparent를 통과해도 Camera Ray의 특성이 변하지 않습니다.



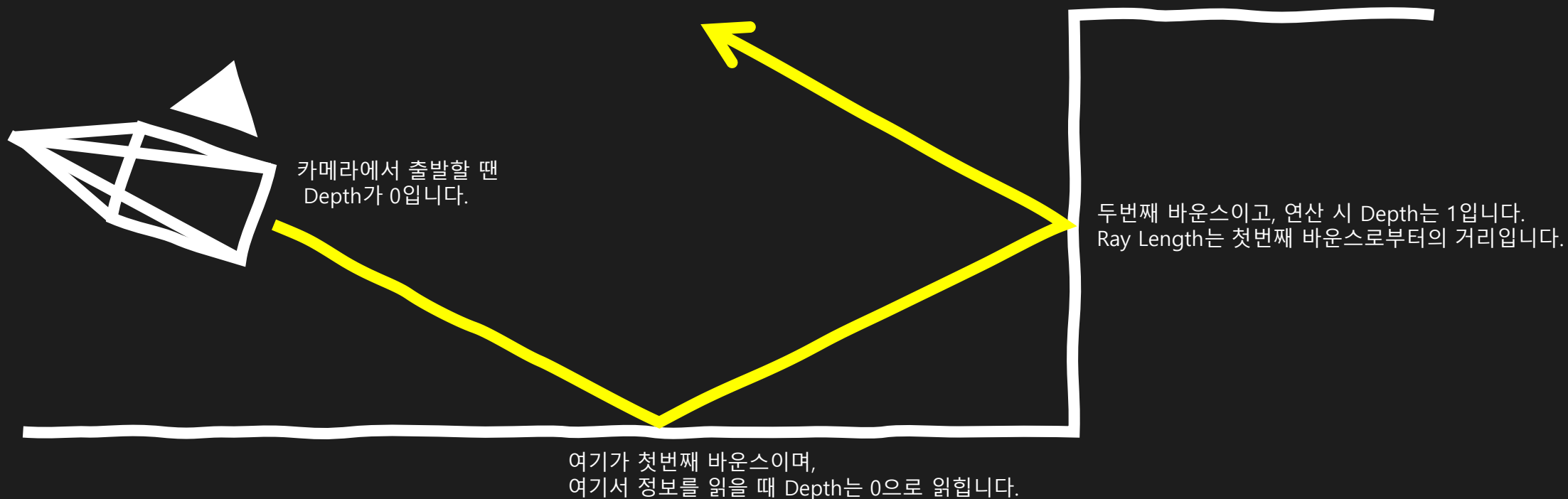
Light Paths	
Max Bounces	
Total	12
Diffuse	4
Glossy	4
Transmission	12
Volume	0
Transparent	12

# Light Path

## Ray Depth, Length

Ray Depth는 지금까지 바운스한 횟수를 의미합니다. (그러므로, Camera Ray는 Depth가 0입니다.)

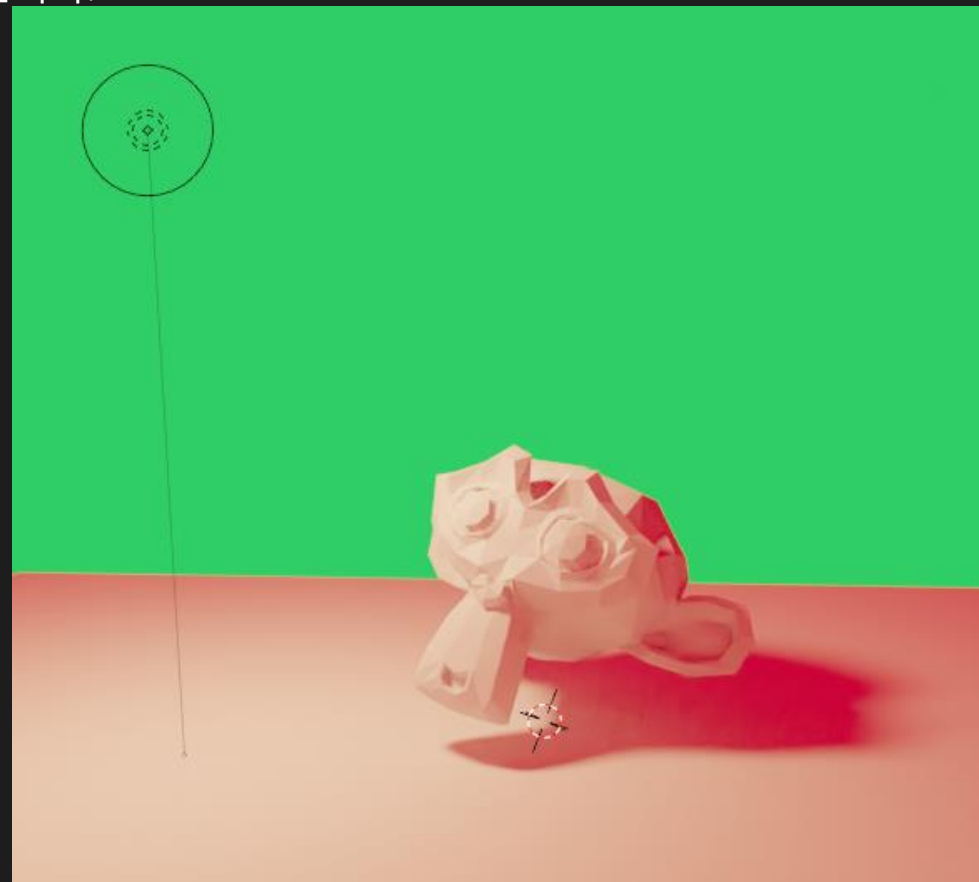
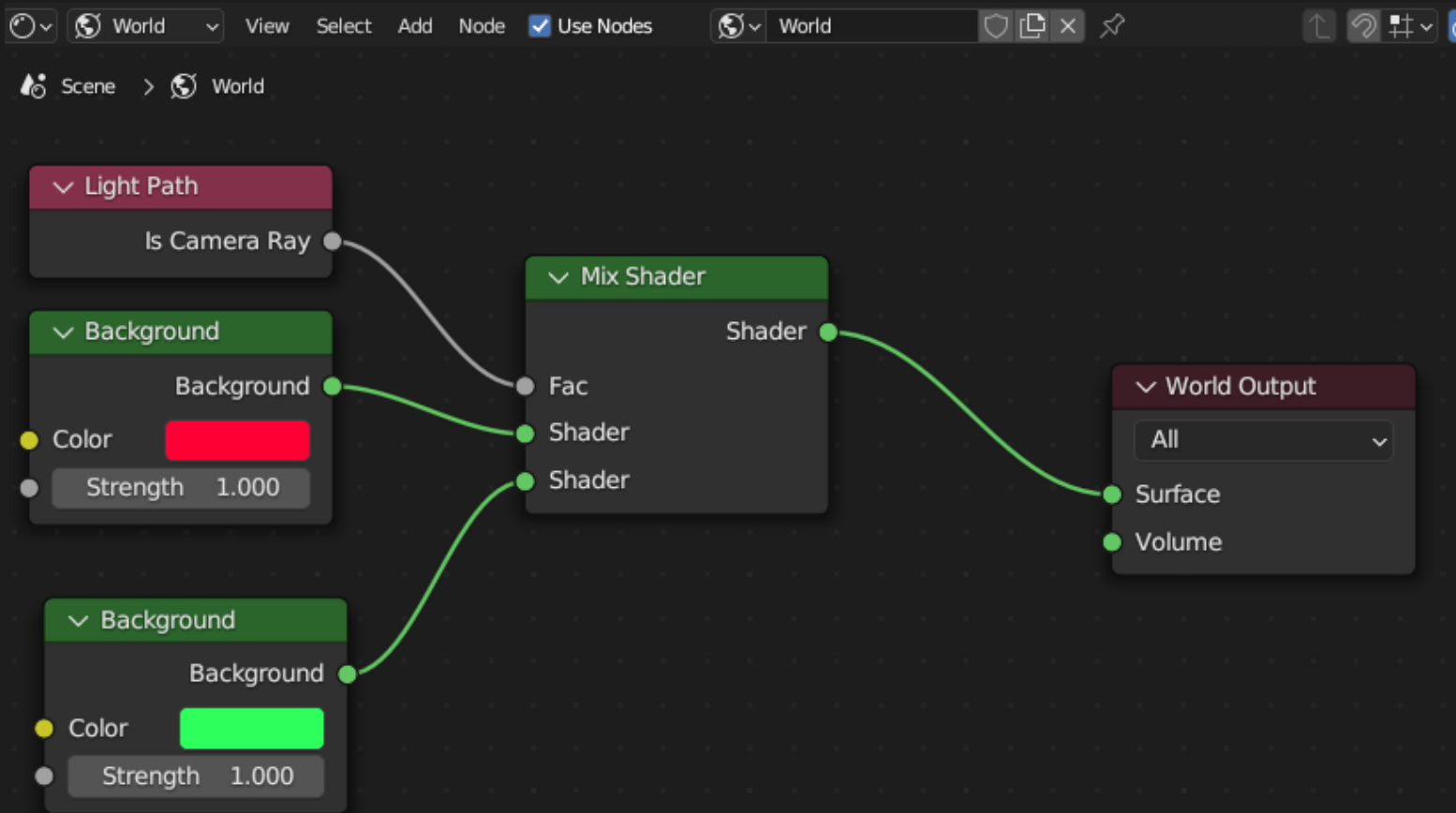
Ray Length는 마지막 바운스로부터의 거리를 의미합니다. (전체 거리가 아님에 유의하세요!)



# 예시 - 배경에서의 활용

## 배경 분리

Camera Ray를 이용해서, 눈으로 볼 때의 배경과 상호작용하는 배경색을 분리시킬 수 있습니다.  
HDRi를 쓰고 싶은데, 눈으로 볼 땐 단색으로 하고 싶을 때 자주 사용하는 방법입니다.





# 예시 - 배경에서의 활용

## 물체의 기본색 강화

Diffuse는 물체의 기본색 역할을 합니다.

만약 Diffuse일 때 **배경의 채도를 낮춘다면**, 배경 광원 효과와 반사효과는 그대로 사용하면서 물체 기본색을 살릴 수 있습니다.

