

015강 Procedural Texture 입문

Procedural Texture의 장점과 단점
기본 수학 지식 훑어보기

Procedural Texture의 장점과 단점

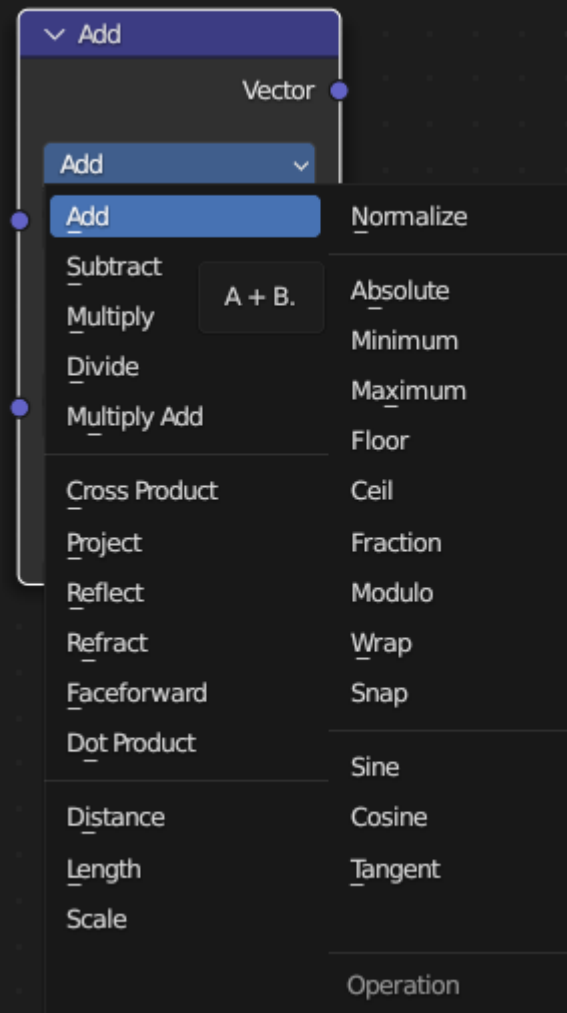
	Image	Procedural
해상도	▼	▲
제작 난이도	(▼)	(▲)
메모리	▲	▼
연산	▼	▲

MATH NODE

The image shows a 'Math Node' interface. At the top, there is a dropdown menu labeled 'Add' with a 'Value' input field. Below this, a list of mathematical functions is displayed, organized into five columns: Functions, Comparison, Rounding, Trigonometric, and Conversion. The functions are listed as follows:

Functions	Comparison	Rounding	Trigonometric	Conversion
Add	Minimum	Round	Sine	To Radians
Subtract	Maximum	Floor	Cosine	To Degrees
Multiply	Less Than	Ceil	Tangent	
Divide	Greater Than	Truncate	Arcsine	
Multiply Add	Sign	Fraction	Arccosine	
Power	Compare	Modulo	Arctangent	
Logarithm	Smooth Minimum	Wrap	Arctan2	
Square Root	Smooth Maximum	Snap	Hyperbolic Sine	
Inverse Square Root		Ping-Pong	Hyperbolic Cosine	
Absolute			Hyperbolic Tangent	
Exponent				

VECTOR MATH NODE



수식 사용

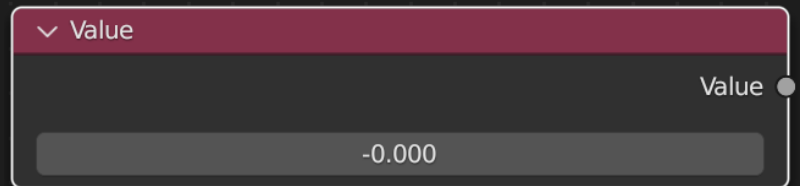
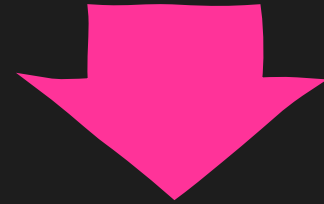
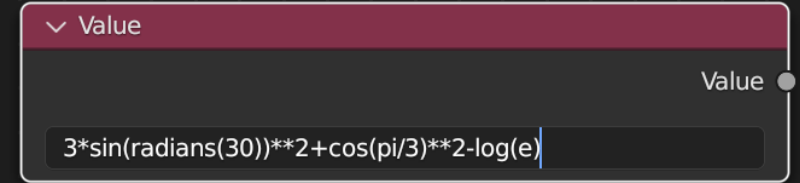
노드의 숫자 입력 창에서 기본적인 계산을 할 수 있습니다.
예를 들어 1/3을 입력하면 자동으로 0.3333... 이 입력됩니다.

상수

pi : 원주율 3.141..
e : 자연상수 2.718...
tau : $2 \times \pi = 6.283...$

함수

기본적인 사칙연산 : +, -, *, /
거듭제곱 : **
루트 : sqrt(x)
삼각함수 sin(x), cos(x), tan(x) asin(x), acos(x), atan(x), atan2(x,y).....
지수함수 exp(x)
라디안 - 육십분법 변환 degrees(), radians()
최대, 최소 max(x,y,z,.....) min(x,y,z,.....)
등등..



MATH NODE

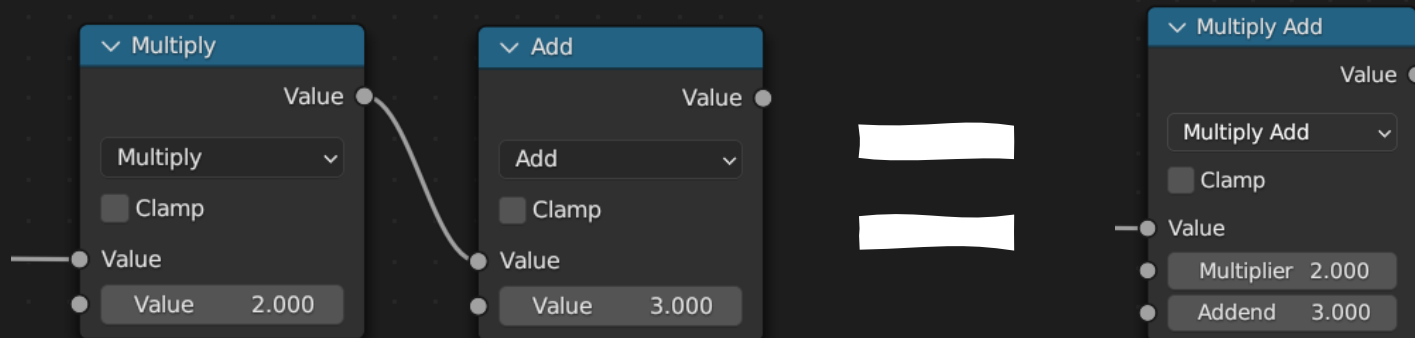
Add

덧셈은 이동의 성질을 가지고 있습니다.
이미지는 0에서 1 사이의 값을 가지기 때문에, 이때 덧셈은 항상 양의 방향의 이동 = 밝아지는 성질을 가집니다.

Multiply

곱셈은 확대/축소의 성질을 가지고 있습니다.
이미지는 0에서 1 사이의 값을 가지기 때문에, 이때 곱셈은 항상 소수점의 곱셈 = 어두워지는 성질을 가집니다.

Multiply Add 곱셈과 덧셈을 한번에 처리합니다.



※Multiply Add는 곱셈을 먼저 함에 유의하세요.

MATH NODE

Subtract

블렌더에서 대부분의 값의 범위는 0에서 1 사이이기 때문에, Subtract를 이용하여 $1-x$ 를 해주면 Invert와 동일한 효과를 얻을 수 있습니다.

Logarithm

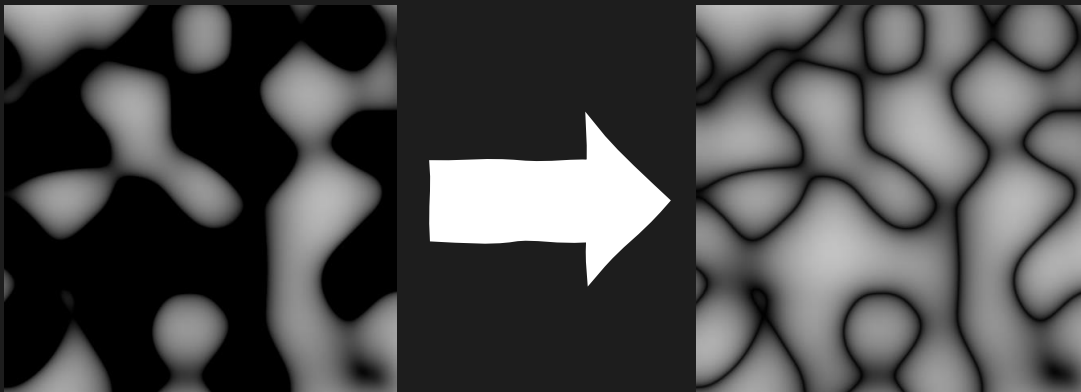
로그를 계산합니다.

Square Root

제곱근 (=0.5제곱) 을 계산합니다.

Absolute

음수의 부호를 바꾸어 양수로 만듭니다.



Inverse Square Root

제곱근의 역수를 계산합니다.

Exponent

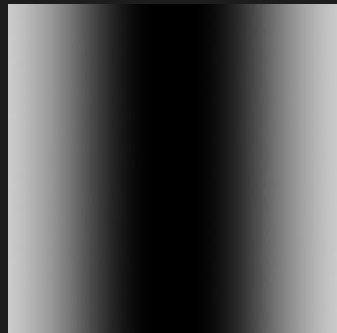
밑이 2인 지수함수를 계산합니다.

MATH NODE

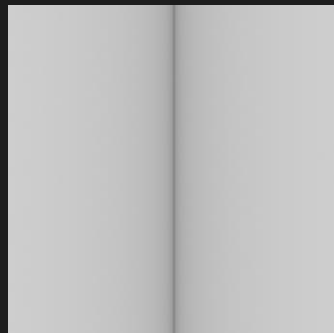
Power 거듭제곱입니다.
0~1사이의 값에 대한 경향성(밝은 쪽, 어두운 쪽) 을 결정합니다.



원본



4제곱



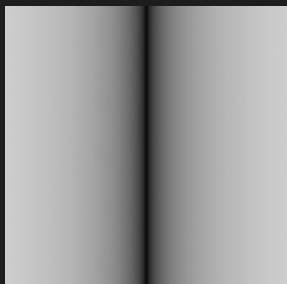
$\frac{1}{4} = 0.25$ 제곱

Power의 exponent연결

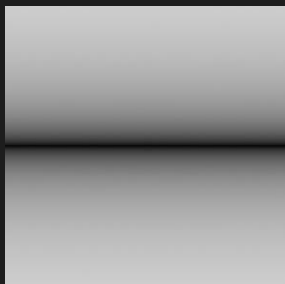
MATH NODE

Minimum, Smooth Minimum

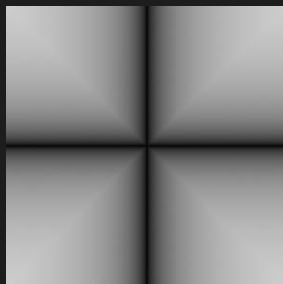
입력받은 두 값중 작은 쪽을 내보냅니다. Smooth Minimum 은 두 값 사이의 경계를 더 부드럽게 만들어줍니다.



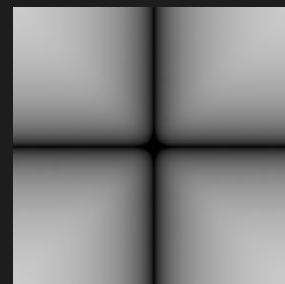
A



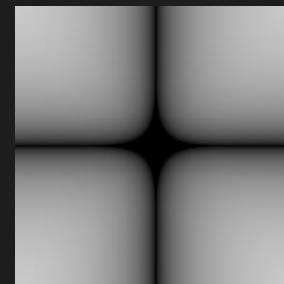
B



Minimum (A,B)



Smooth Minimum
Distance 0.2



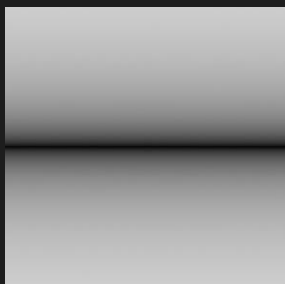
Smooth Minimum
Distance 0.5

Maximum, Smooth Maximum

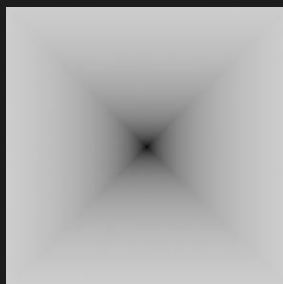
입력받은 두 값중 큰 쪽을 내보냅니다.



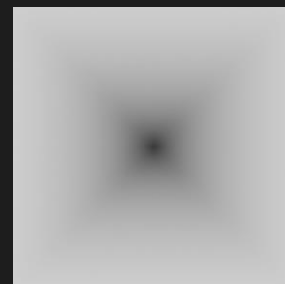
A



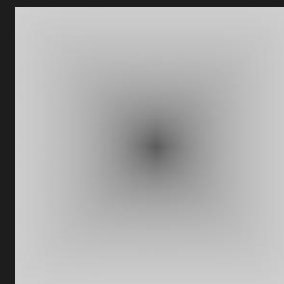
B



Maximum (A,B)



Smooth Maximum
Distance 0.2

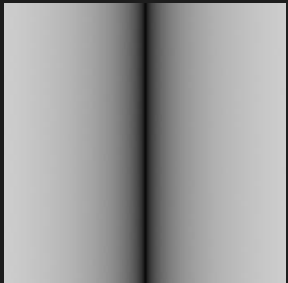


Smooth Maximum
Distance 0.5

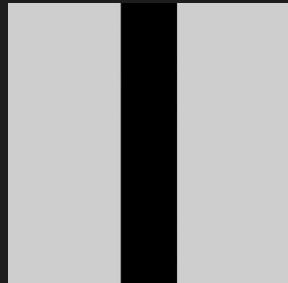
MATH NODE

Less Than, Greater Than : $<$, $>$

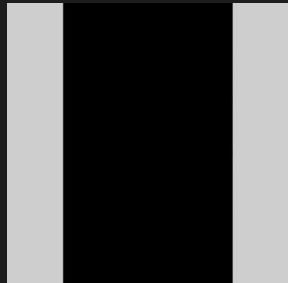
Greater than : Threshold 기준으로 큰 쪽을 1, 작은 쪽을 0으로 나타냅니다. Less than은 그 반대로 작동합니다.



원본



Greater than
Threshold 0.2

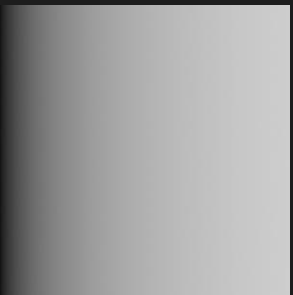


Greater than
Threshold 0.6

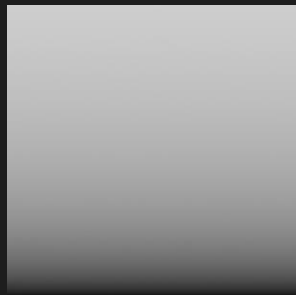
Compare : $=$

입력받은 두 값이 같은지 판단합니다.

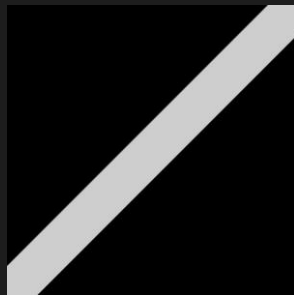
'정확히 같은지' 판단한다면 많은 경우 표현이 곤란하므로, epsilon을 이용하여 '얼마나 비슷한지'를 판단합니다.



X



Y



Y=X (epsilon 0.1)

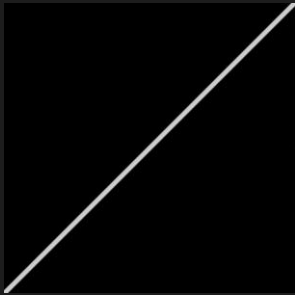
Sign

양수는 1, 음수는 -1을 내보냅니다.

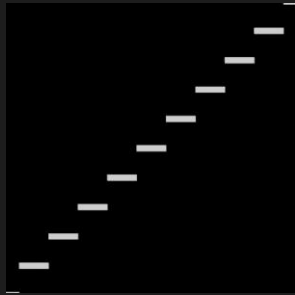
MATH NODE

Round, Floor, Ceil

차례대로 반올림, 내림, 올림입니다.
기준에 따라 정수 부분을 출력합니다.



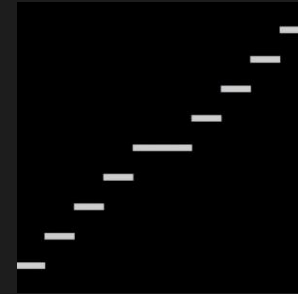
$$Y=X$$



$$Y=\text{Round}(X)$$

Truncate

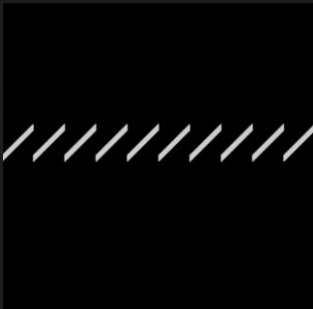
입력받은 값의 정수 부분을 출력합니다.
만약 음수라면, 예컨대 -1.2라면 -1을 출력합니다.
그에 따라 그래프의 모양은..



양수 범위에서는 Floor와 같습니다.

Fraction

입력받은 값의 소수 부분을 출력합니다.
만약 음수라면, 예컨대 -1.2라면 -2+0.8로 보아 0.8을 출력합니다.



그렇기에 $\text{Truncate} + \text{Fraction}$ 은 - 정수부분+소수부분 이니까 - 원래 숫자가 나올 것 같지만 그렇지 않습니다.
만약 Truncate 처럼, -1.2의 소수부분을 -0.2로 인식하고 싶다면 Modulo를 사용하세요!

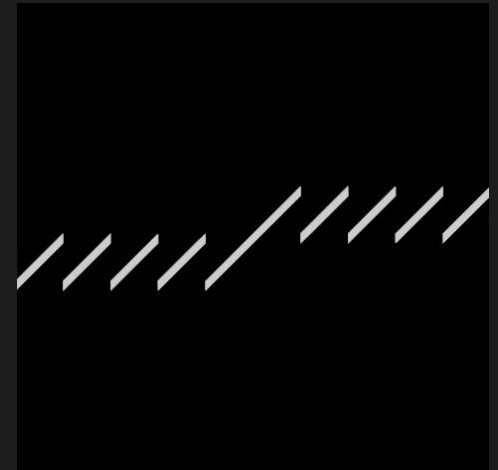
MATH NODE

Modulo

Modulo(A,B)는 A를 B로 나눈 나머지입니다.
예컨대, $\text{Modulo}(5.3, 2) = 5.3$ 을 2로 나눈 나머지 = 1.3입니다.

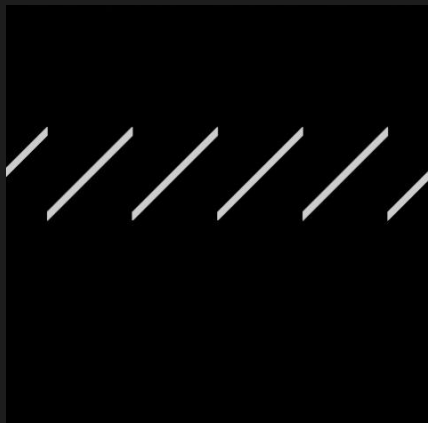
음수일 때 계산에 유의하세요! 피젯수(dividend)가 음수일 때의 작동방식은,
예컨대 $\text{Modulo}(-5, 4)$ 라면 -5를 4로 나눈 나머지 : $-5 = 4 \times (-1) + (-1)$ 로 -1로 계산합니다.

만약 음수일 때도 양수일때와 같은 규칙을 갖게 하고 싶으시면, Wrap을 이용합니다.



MATH NODE

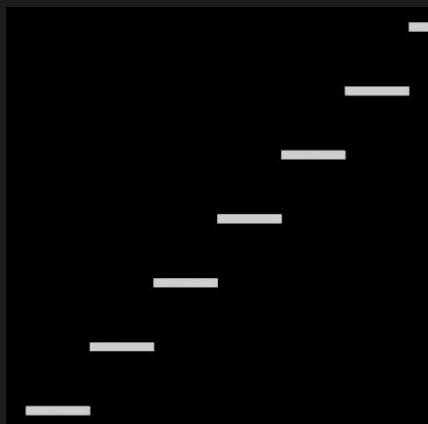
Wrap



Wrap (A, max, min)은 A를 min~max 사이에서 반복하게 만듭니다.
만약 min을 0으로 둔다면 Modulo와 같습니다.
다만, 음수일 때도 같은 규칙을 갖는다는 점은 Modulo와 다릅니다.

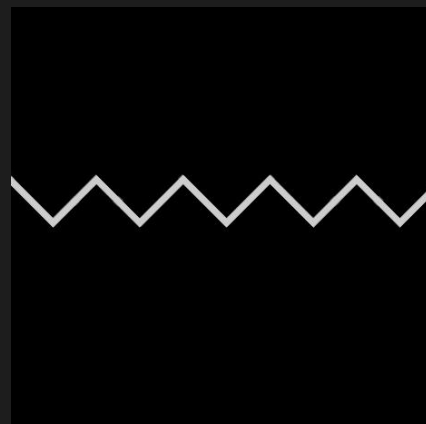
Snap

Snap은 Floor와 비슷하지만
increment에 따라 내림의 범위를 정할 수 있습니다.



Ping-Pong

Ping pong은 입력받은 값을 왔다갔다
반복하는 주기함수로 만듭니다.



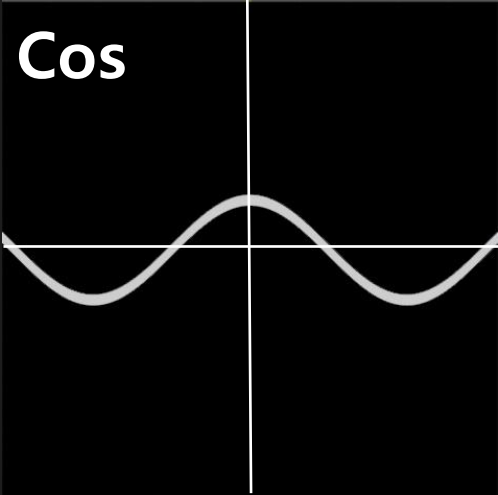
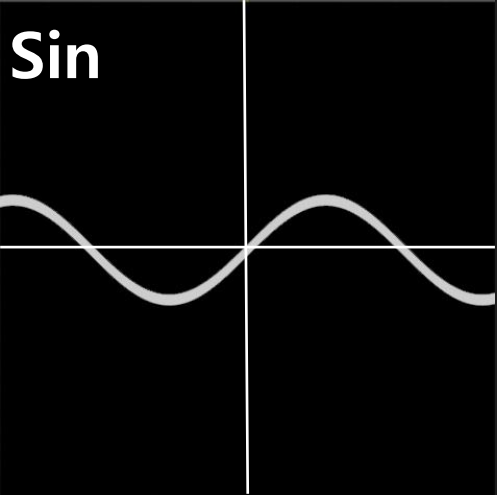
기준은 Scale이며, Scale까지 올라갔다
내려감을 반복하므로 주기는 Scale의
두배입니다.

예컨대, Scale = 0.5라면 0.5까지
올라갔다 내려가므로 주기는 1입니다.

MATH NODE

Sine, Cosine

삼각함수는 기하학적인 의미도 중요하지만, 일단 그래프 모양만 확인하겠습니다.



사인과 코사인은 $2\pi = 6.283...$ 마다 반복되는 주기함수이며, -1에서 1 사이에 놓입니다.

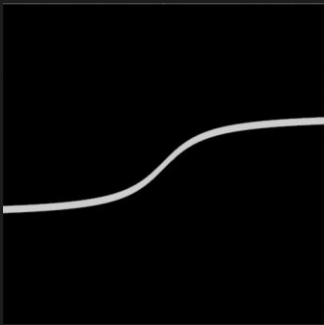
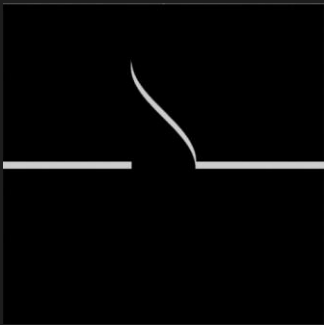
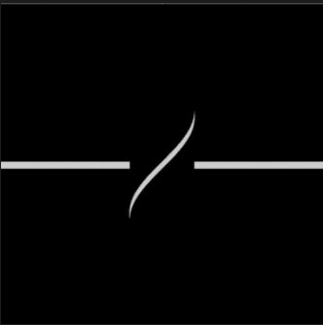
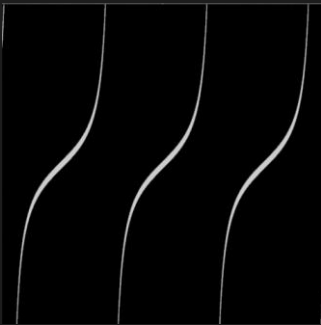
Tangent,

Arcsine,

Arccosine,

Arctangent

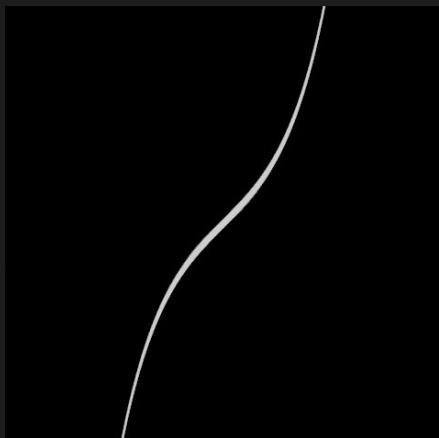
Atan2



Atan2 (y,x)는 y/x의 역탄젠트 각을 출력합니다.

MATH NODE

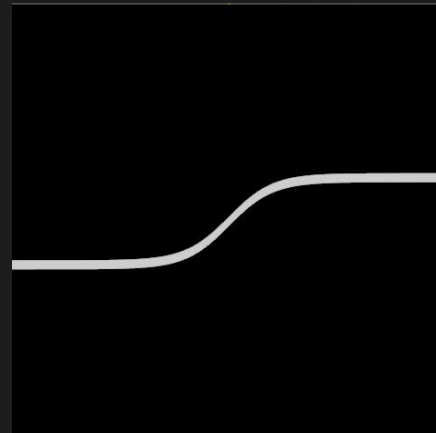
sinh



cosh



tanh



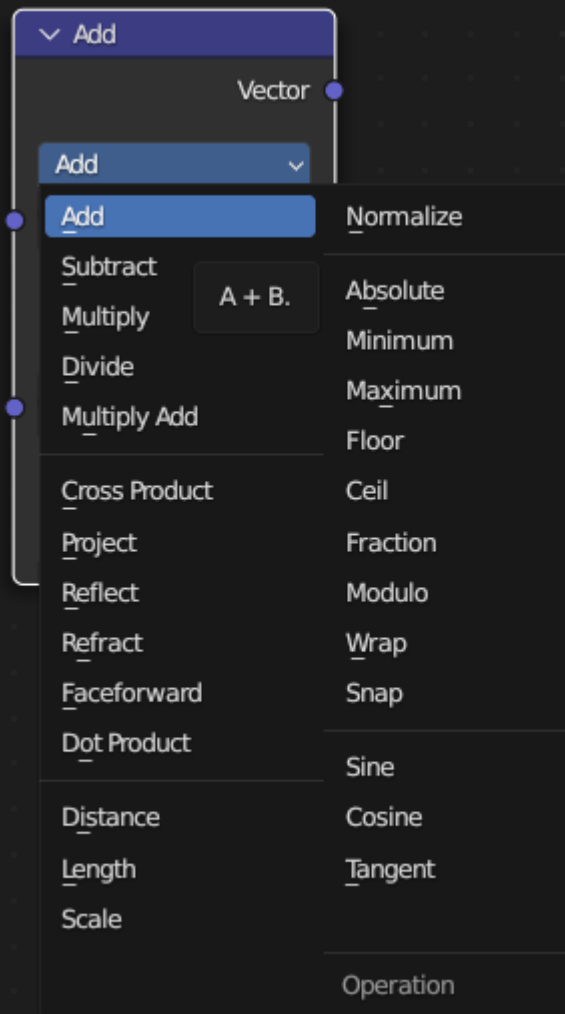
To Radians, To Degrees

육십분법과 호도법을 변환합니다. 예컨대 $\text{To radians}(180) = \pi$, $\text{To degrees}(\pi/3) = 60$ 입니다.

VECTOR MATH NODE

같은 이름의 연산은 물론 Vector Math 에서도 동일하게 작동합니다.

Multiply는 각 성분을 개별적으로 곱할 수 있고,
Scale은 모든 성분에 하나의 값을 한번에 곱할 수 있습니다.



VECTOR

Vector에 대한 개념은 두 가지 다른 방식으로 생각할 수 있습니다.

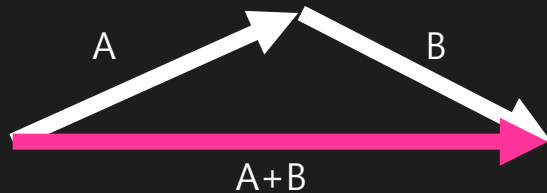
1. 단지 숫자 3개를 묶어놓은 것 : 벡터의 표현에 숫자 3개 이상은 필요하지 않습니다.
벡터 자체에는 첫번째, 두번째, 세번째 숫자가 각각 무엇을 의미하는지 언급되어 있지 않습니다.
그렇기 때문에, 좌표에 꽂으면 위치로, 컬러에 꽂으면 색으로 인식합니다.
2. 원점으로부터 뻗어나온 방향 : 예컨대 $(1,3,2)$ 라는 것은 x축 위의 한 점으로 볼 수도 있지만, 원점으로부터 $(1,3,2)$ 까지 이은 화살표로 생각할 수도 있습니다. 이런 개념은 유용할 때도 있고, 그렇지 않을 수도 있습니다.



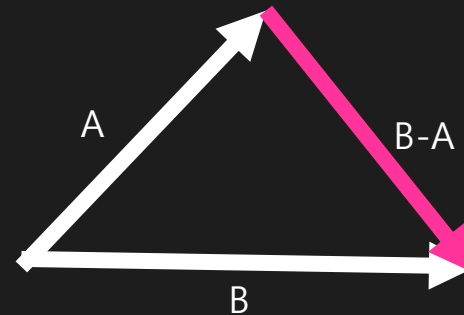
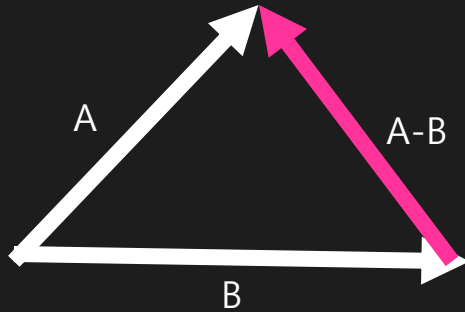
VECTOR

벡터를 방향으로 인지한다면, 다음의 기하학적 의미가 생깁니다.

1. 덧셈 : 두 벡터의 시작점과 끝점을 이은 것이 됩니다.



2. 뺄셈 : 두 벡터의 끝점을 이은 것이 됩니다.

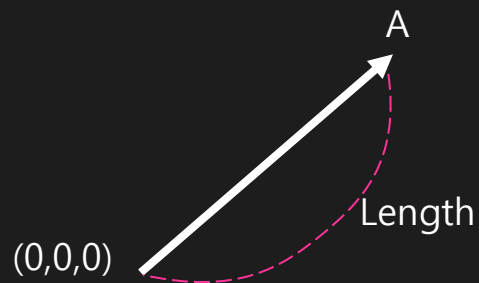
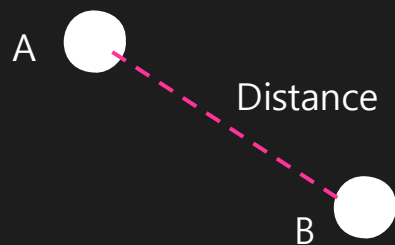


※ 각 성분끼리 곱하거나 나눈 값은 기하학적 의미를 찾기 힘듭니다.

VECTOR MATH NODE

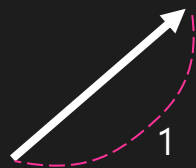
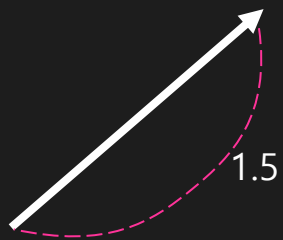
Distance, Length

Distance 는 두 점 사이의 거리, Length는 단일 벡터의 크기 (=화살표의 길이) 를 의미합니다.



Normalize

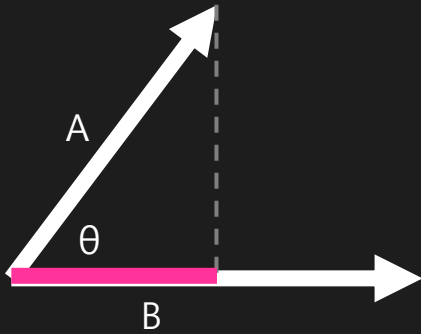
(벡터의 방향만 생각하기 위해,) 크기를 1로 만듭니다.



VECTOR MATH NODE

Dot Product

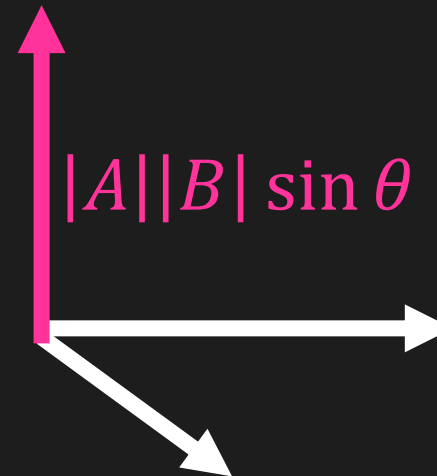
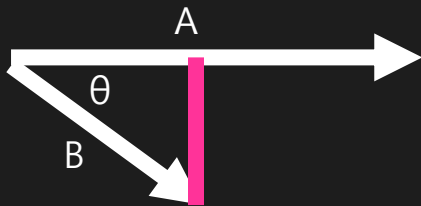
.....결과적으로, 입력받은 두 벡터가 얼마나 **방향**이 다른지 알게 해 줍니다.



$$|A||B| \cos \theta$$

Cross Product

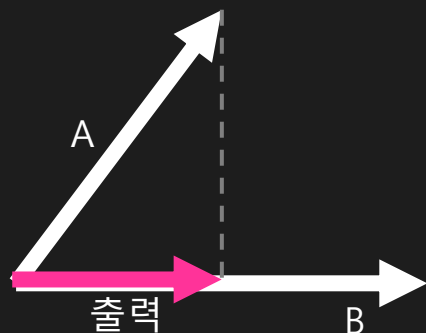
.....결과적으로, 입력받은 두 벡터와 **수직인 벡터**가 만들어집니다.



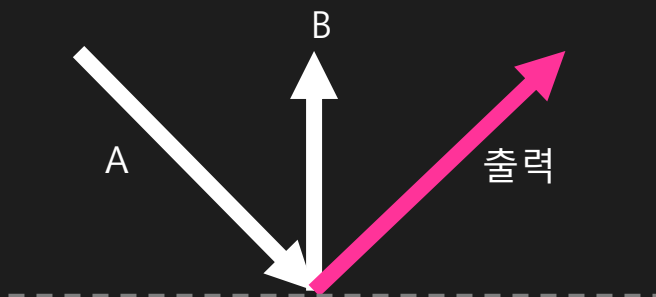
$$|A||B| \sin \theta$$

VECTOR MATH NODE

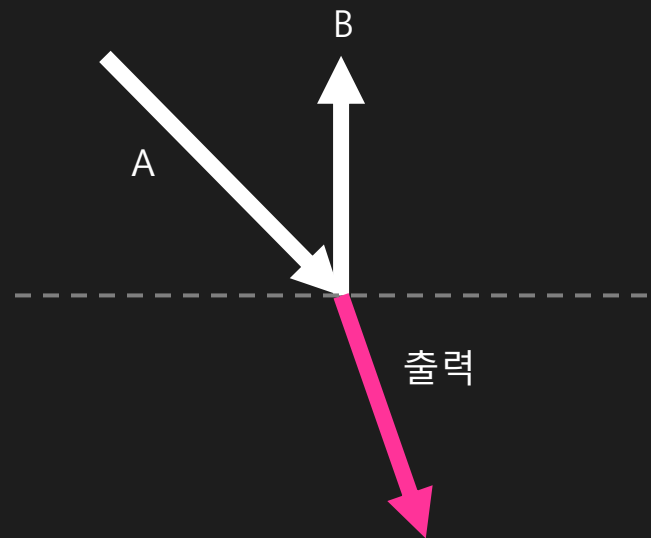
Project



Reflect



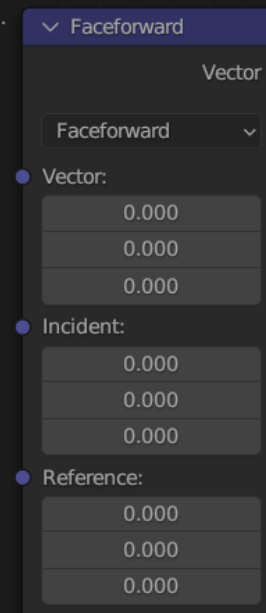
Refract



Faceforward

Incident가 Reference가 만드는 면을 바라보는 방향이면, Vector를 그대로 출력, 바라보는 방향이 아니라면, Vector에 -1을 곱해 방향을 바꿉니다.

이 강의에서는 쓰지 않습니다.



Refract의 IOR은 밀(Dense)한 매질 기준입니다. 왼쪽과 같이 되기 위해서는 IOR이 1보다 작아야 됩니다.