

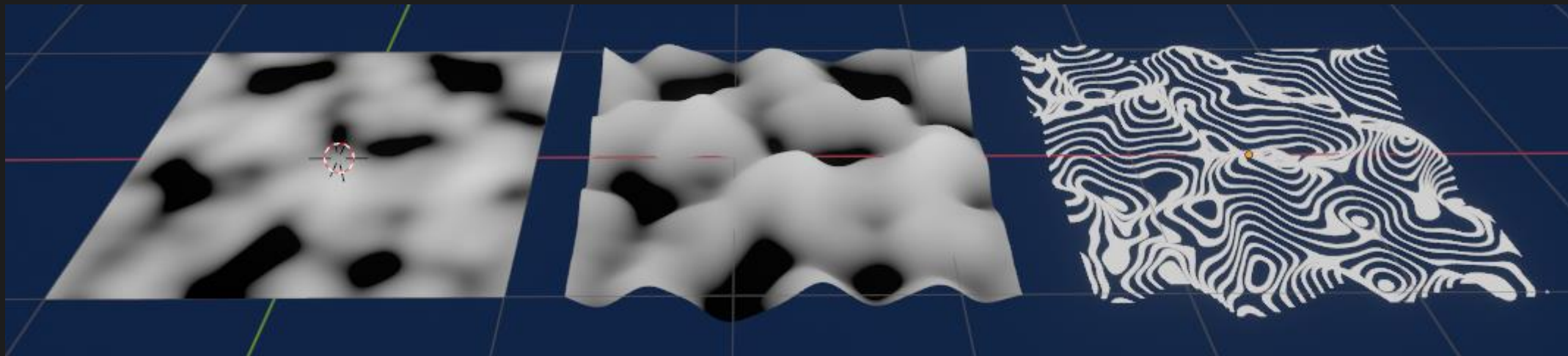
068강 Simulation Node – 그래디언트 (v3.6~)

Gradient와 Level Curves



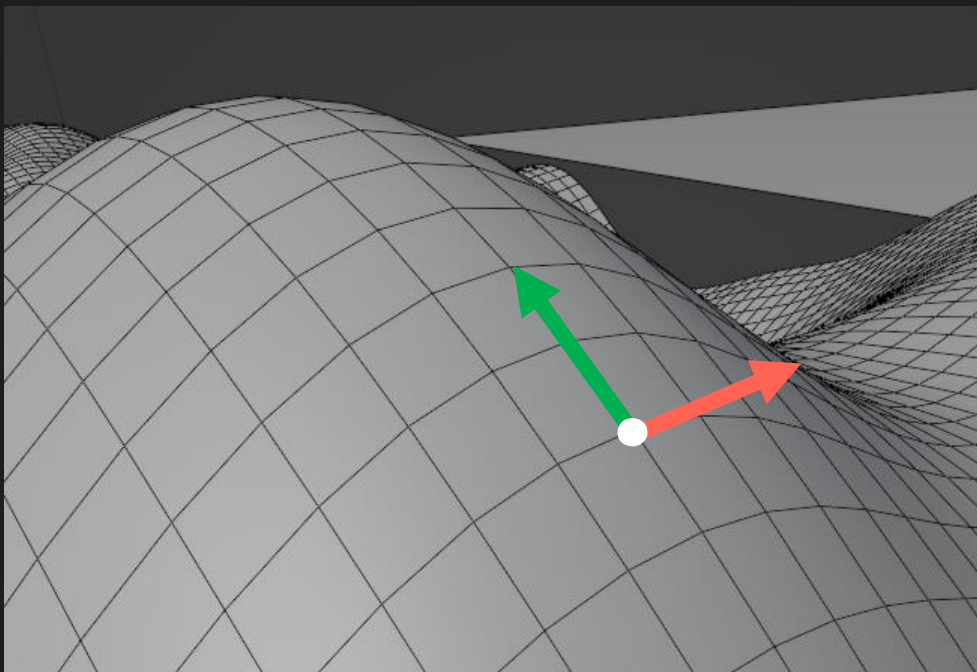
회색 텍스처

블렌더의 텍스처 중 2차원 좌표를 받아 흑백 이미지를 내보내는 $R^2 \rightarrow R$ 를 생각해 봅시다.
텍스처를 높이로 생각한다면, 이것은 $(x,y) \rightarrow z$ 로의 높이 함수입니다.
그렇게 보았을 때, 이것의 **등고선**을 실시간으로 표현해 보는 것이 이번 시간의 목표입니다.

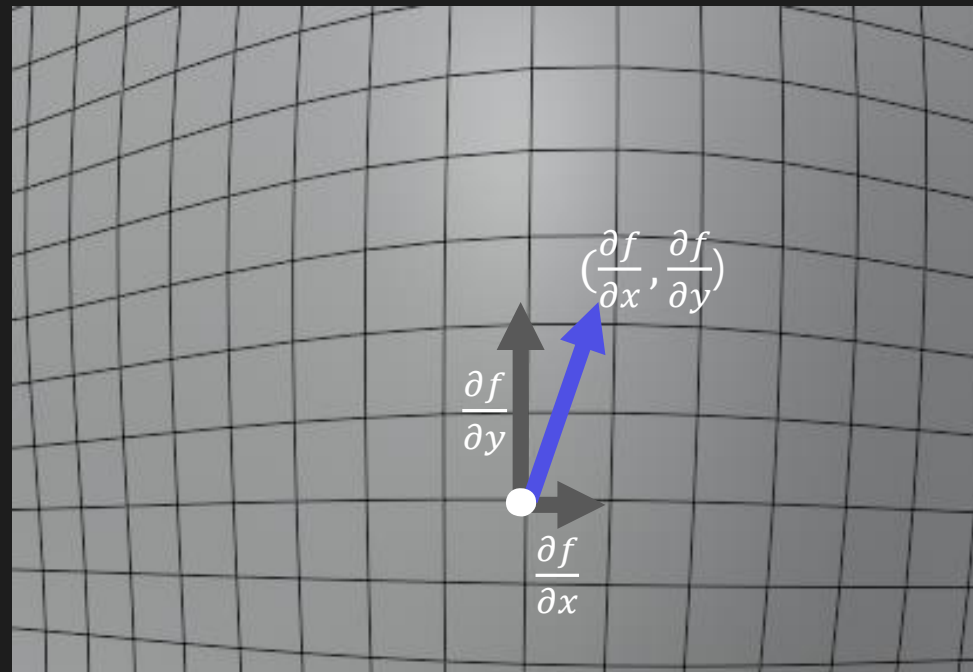


Gradient

$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$ 로 표현되는 그래디언트는 언덕을 오르는 방향의 벡터를 나타냅니다.

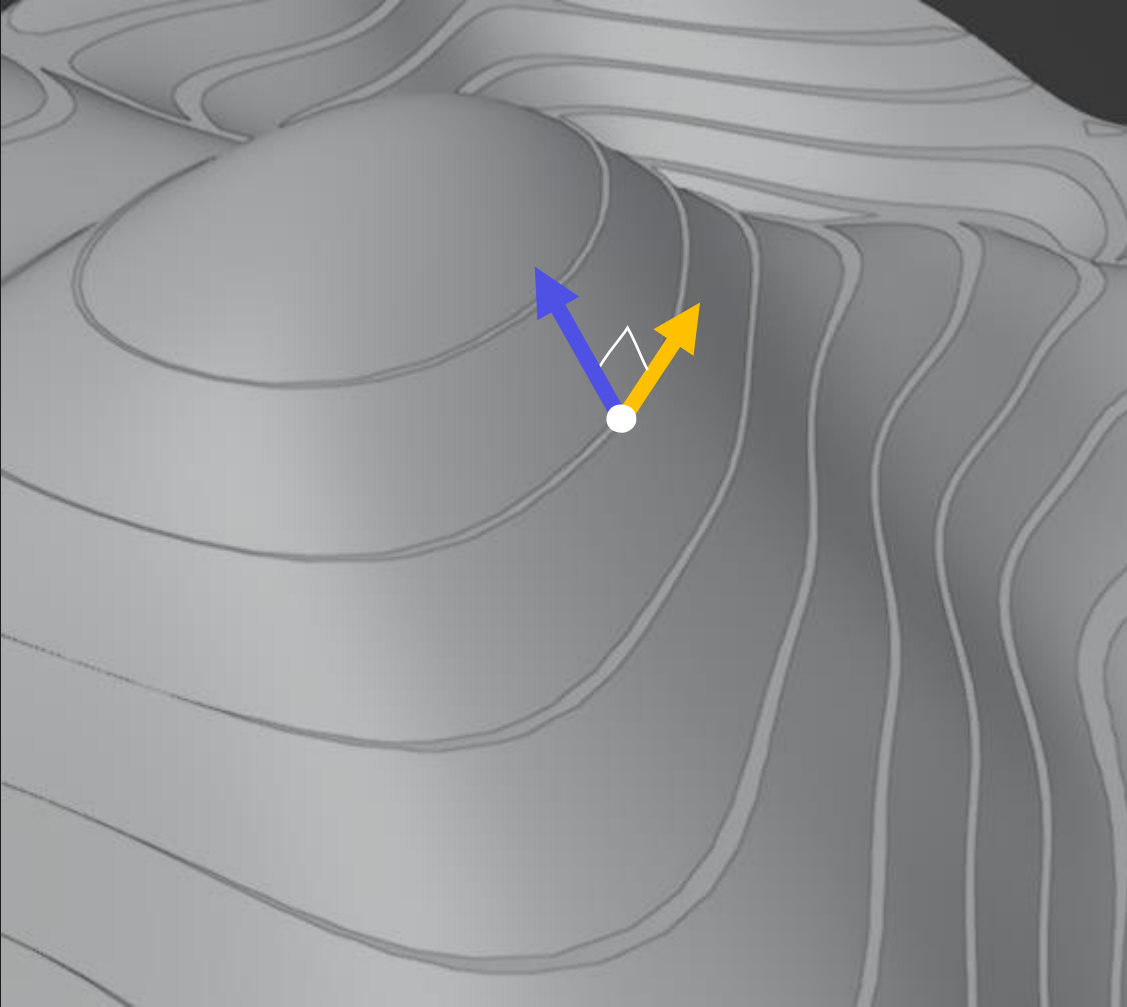


$\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}$ 는 각각 x,y방향으로의 기울기입니다.
흰색 점에서 x축 방향(빨간색)보다 y축 방향(초록색)이 더 가파릅니다.
따라서 $\frac{\partial f}{\partial y}$ 가 $\frac{\partial f}{\partial x}$ 보다 큼니다.



$\left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right)$ 벡터(파란색) 은 평면상에서 언덕을 오르는 방향을 가리킵니다.

Gradient에 수직인 것



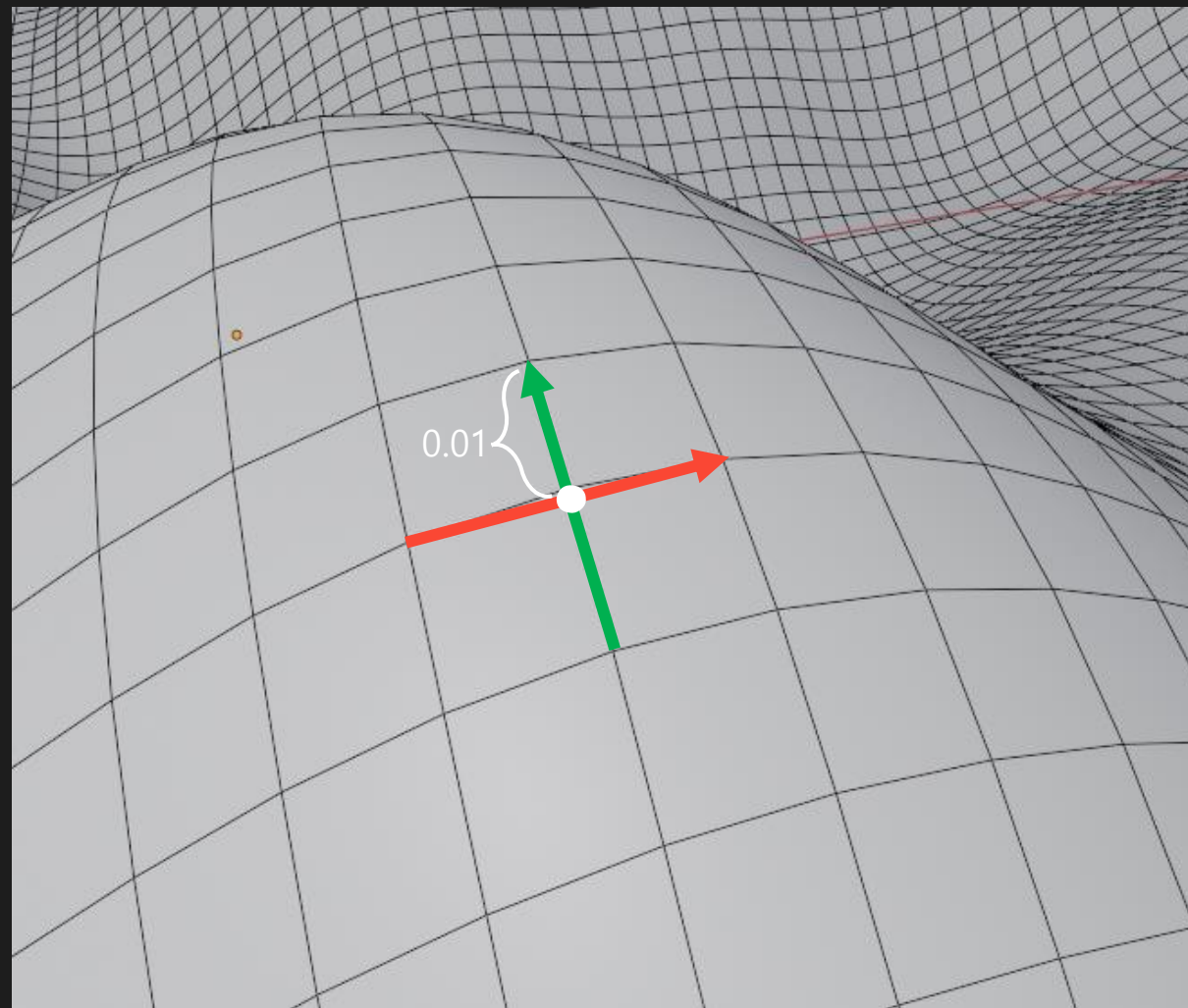
그래디언트 벡터(파란색)는 언덕을 가장 빠르게 올라가는 방향이고, 그래디언트의 반대 방향은 가장 빠르게 내려가는 방향입니다.

한편, 그래디언트에 수직인 방향(노란색)은 오르지도, 내리지도 않는 수평 방향이 됩니다. 따라서 그래디언트에 수직으로 이동하면 **등고선**이 됩니다.

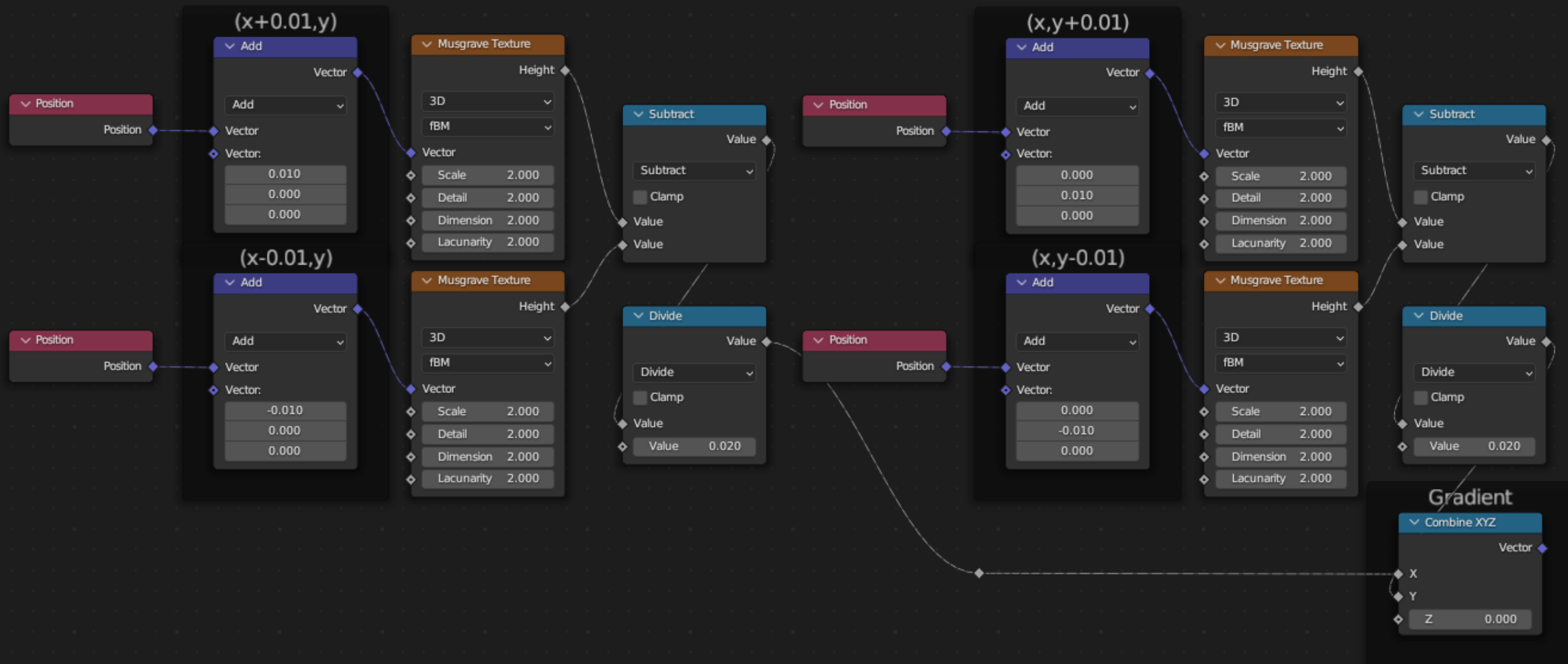
Gradient의 구현

그래디언트를 정확히 구하려면 편미분을 해야 하지만,
충분히 가까운 거리의 기울기를 구하면 근사적으로 구할 수 있습니다.

오른쪽과 같이 x, y 상하좌우로 0.01만큼 떨어진 지점의 높이를 이용하여
기울기를 구하면, 상당히 정확한 그래디언트가 구해집니다.

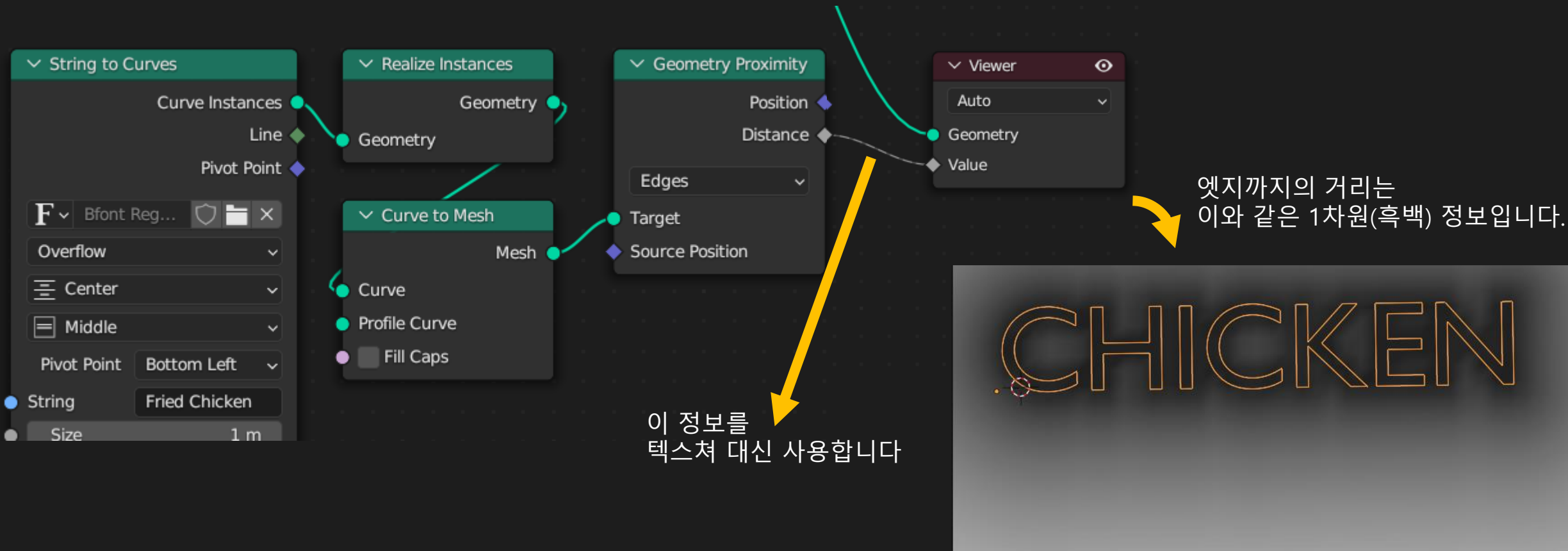


Gradient의 구현



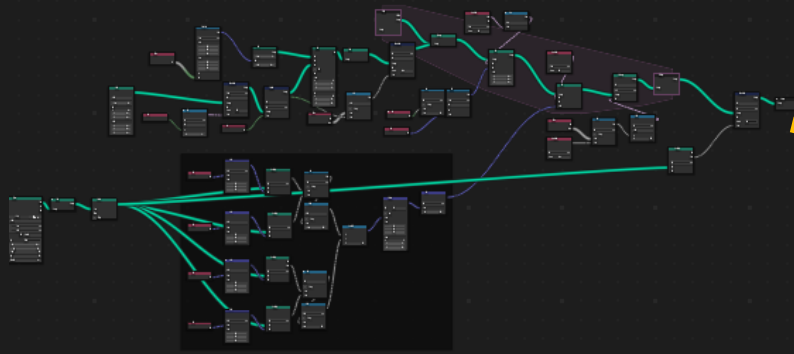
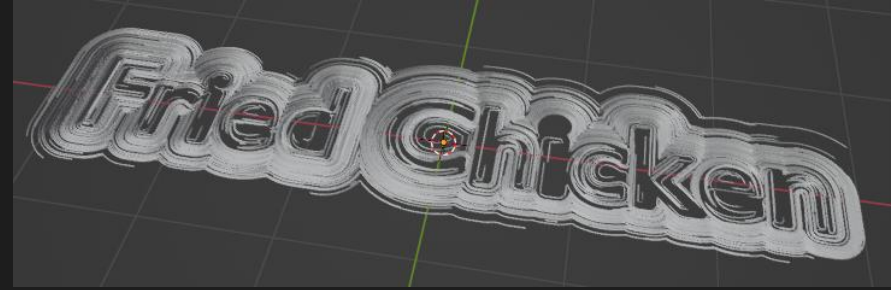
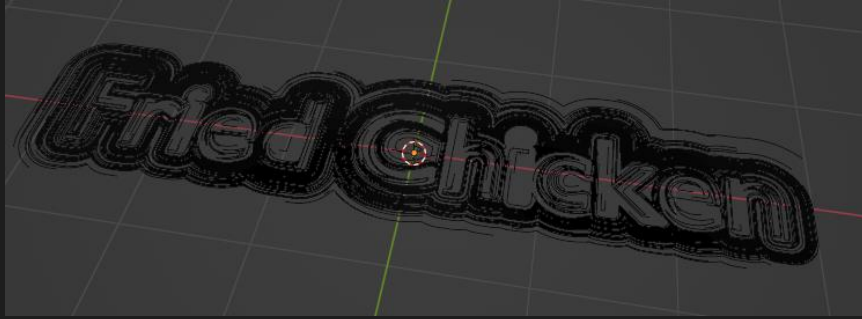
String을 이용한 함수

String to Curves까지의 거리를 이용하면, 문자 주위를 회전하는 등고선도 만들 수 있습니다.

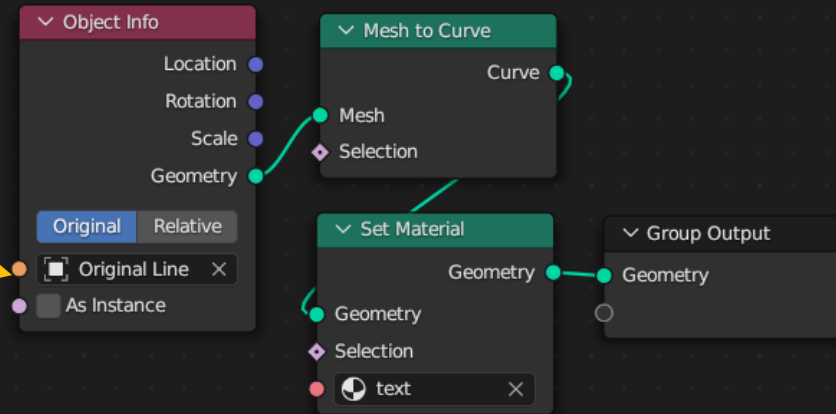


Curves를 이용한 시각화

그냥 Mesh to curve -> Curve to Mesh를 이용한 방법은 많은 폴리곤이 필요하므로 무겁습니다.
지금은 커브의 디테일이 필요하지 않으므로 Curves를 이용하는 것이 좋습니다.



원본 (mesh)



Curves를 만들고 원본을 불러옵니다.

※Curves를 만들 땐 어쩔수 없이 curves가 붙을 오브젝트가 하나 필요합니다.
우리는 헤어를 만드는 것이 아니기 때문에 필요가 없으므로 나중에 지워도 됩니다.