**Link:**

<https://learn.unity.com/tutorial/explore-shaders?uv=2020.3&pathwayId=61a65568edbc2a00206076dd&missionId=619f9b6cedbc2a39aabd7b1e>

A **shader** is a script that applies the properties contained in a material to render the meshes of your 3D objects to the 2D image on your screen. Each shader is written for a specific [render pipeline.](https://learn.unity.com/tutorial/get-started-on-your-guided-project#619fbf21edbc2a39aabd80eb)

## Types of shaders

2

Overall, there are two types of operations that occur in a shader: **fragment shading** and **vertex shading**.

**Fragment shading**, also known as pixel shading, is the shading that represents mesh surfaces to produce the color of each pixel in the 2D image. In this project, we’ll be working with fragment shaders and discussing in detail how they render with the light in the scene.

**Vertex shading** operates on the vertices of the mesh, typically changing their locations to make the surface move or transform. We won’t cover vertex shading in this learning experience, but we will provide more resources in a future tutorial.

**Physically Based Rendering (PBR)**

**Universal Render Pipeline** submenu.

* **2D** > **Sprite-Lit-Default**: Designed for 2D projects, this shader is for flat objects only and will render any 3D object as 2D. As a lit shader, it will render based on the light in the scene that reaches the object.
* **Particles** > **Lit, Simple Lit, and Unlit:** These shaders are for visual effects (VFX). In the Creative Core pathway, you will use these shaders in the VFX mission.
* **Terrain > Lit:** This shader is optimized for use with the Terrain tools in Unity. In the **Creative Core** pathway, you will use this shader in the **Prototyping** mission.
* **Baked Lit:** This shader is automatically applied to lightmaps, which you will encounter in the **Creative Core** pathway’s **Lighting** mission.
* **Complex Lit, Lit, and Simple Lit:** These are all variations on a general-purpose, physically based lit shader.
* **Unlit:** As described above, a shader that does not use light.

## Get started with shader graph:

Shader Graph allows you to create specialized shaders without the need to write code. You can combine textures and make them move in a fragment shader or even change the positions of vertices in a vertex shader. Professional technical artists create custom shaders to implement specialized artistic styles and to create complex substances like flowing lava, storm clouds, and vegetation. The possibilities are nearly endless!

Here are some things you can easily try based on what you’ve learned:

* Create a shader that allows the material artist to enter a 2D texture file. Hint: add a Texture2D input node and a Sample Texture 2D node to manage the channels of the texture file.
* Experiment with Artistic and Procedural nodes. Dither, Blend, Voronoi, and Polygon can be fun!