**NOTES\_Basics\_Fundamentals**

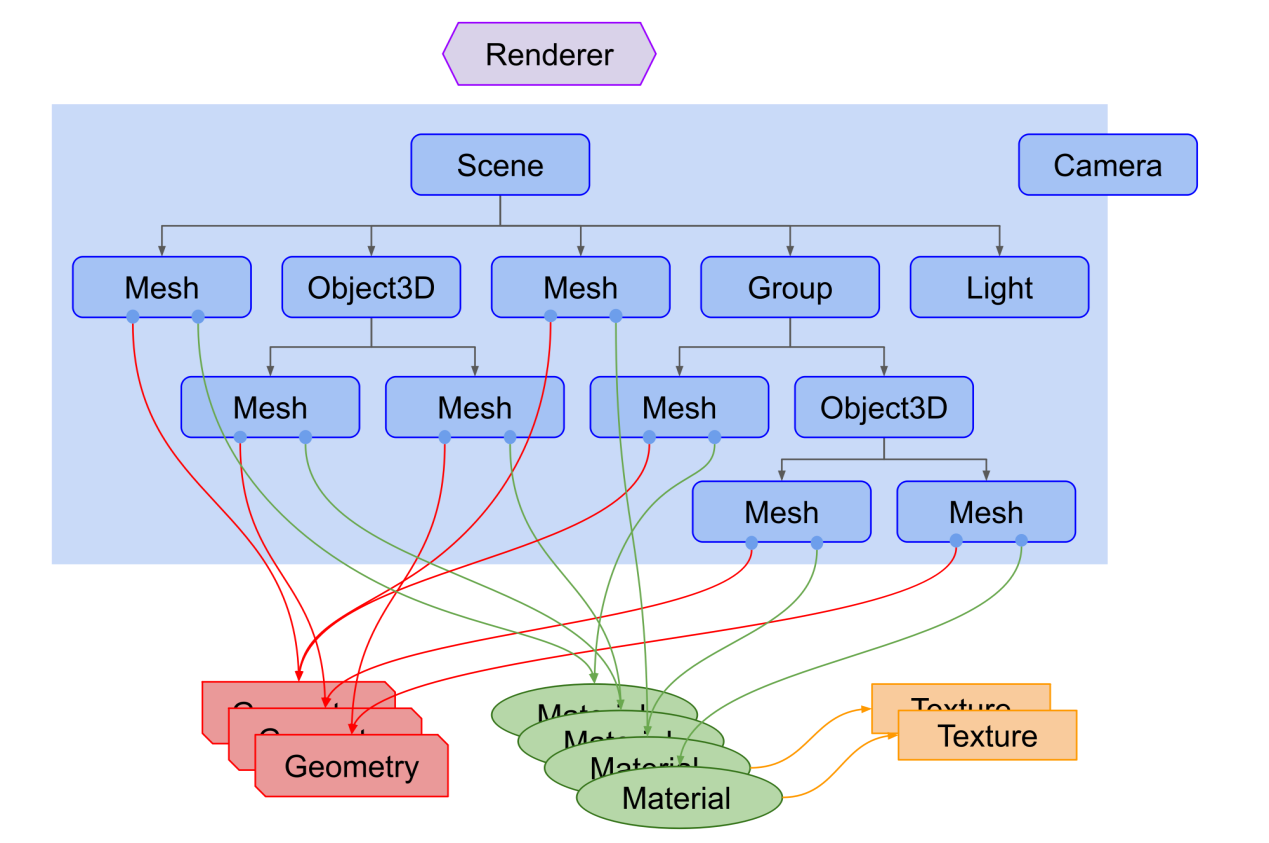
Three.js, a 3D JavaScript library designed to simplify adding 3D content to web pages.

Three.js 是一个 JavaScript 3D 库，旨在尽可能简化在网页上添加 3D 内容的过程。

Although Three.js often uses WebGL for rendering, it is not the same as WebGL. WebGL is a low-level API that only draws basic shapes like points, lines, and triangles, requiring significant coding effort to create complex 3D scenes. Three.js simplifies this process by managing essential features like scenes, lighting, shadows, materials, textures, and 3D math.

尽管 Three.js 经常使用 WebGL 进行渲染，但它们并不相同。WebGL 是一个底层 API，只能绘制点、线和三角形，因此直接使用 WebGL 需要大量编码。而 Three.js 负责处理场景、光照、阴影、材质、纹理和 3D 数学计算等功能，极大地降低了开发难度。

Example: a diagram that represents a small three.js app



1. Renderer: The core object that draws the 3D scene onto a 2D canvas using a Camera.

渲染器： 核心对象，利用摄像机将3D场景绘制到2D画布上。

1. Scenegraph: A hierarchical structure (tree) of objects like Scene, Mesh, Light, Group, etc., where child objects inherit transformations from their parents.

场景图： 层次结构（树形）对象集合，如场景、网格、光源、组等，子对象继承父对象的变换。

1. Camera: Can function outside the scenegraph, but if nested, it follows its parent's movement.

摄像机： 可独立使用，但若嵌入场景图中，则随父对象移动。

1. Mesh: Combines Geometry (shape data) and Material (surface appearance) to represent 3D objects.

网格： 结合几何体（形状数据）和材质（表面外观）的3D对象。

1. Geometry: Holds the vertex data for various shapes; built-in or custom geometries are available.

几何体： 保存各种形状的顶点数据，可用内置或自定义几何体。

1. Material & Texture: Define the appearance of objects, with Textures adding image details.

材质与纹理： 定义对象外观，纹理为材质添加图像细节。

1. Light: Provides different lighting effects within the scene.

光源： 为场景提供不同的照明效果。