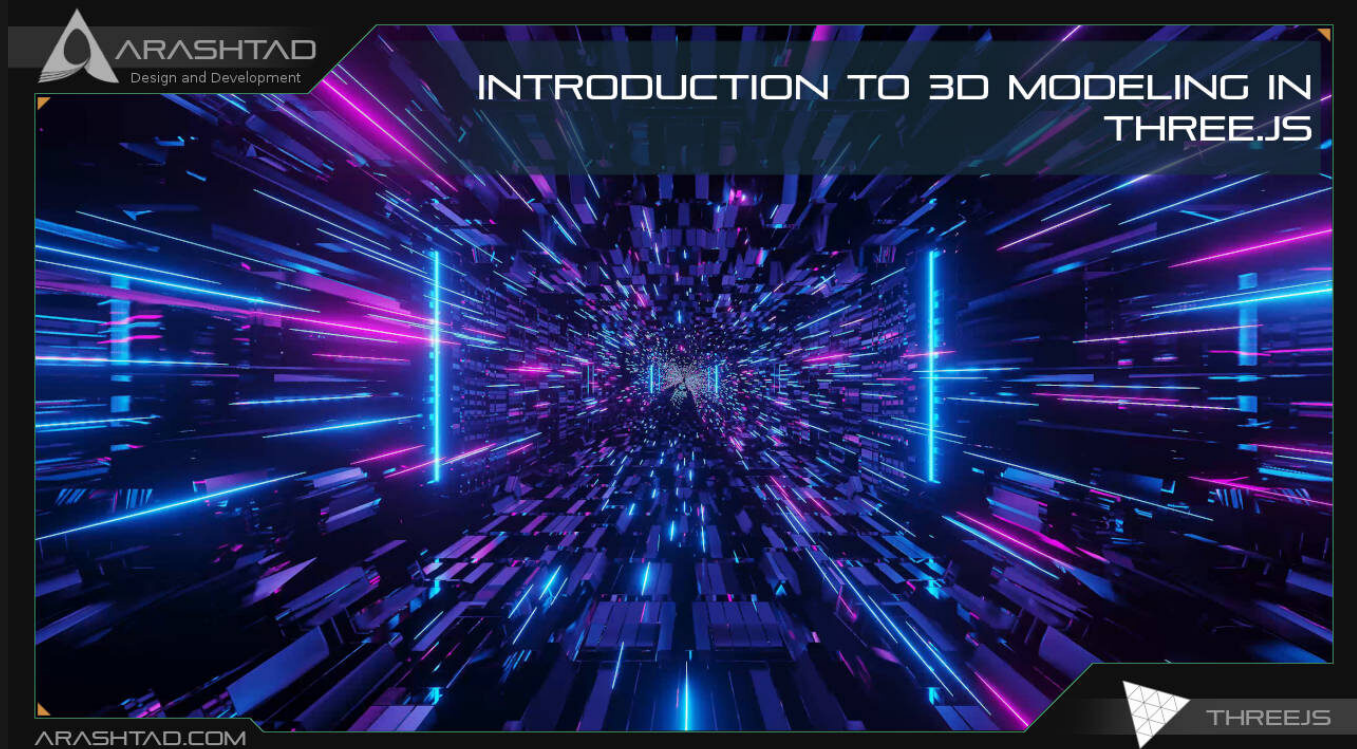


## Introduction To 3D Modeling in Three.js

No comments



*With the Metaverse and virtual reality applications being created these days, 3D modeling and WebGL-based apps are becoming more popular among designers and developers. The concept of creating a virtual world with the closest features to that of the real world seems more and more feasible with the new tools like Blender or Three.js. The whole idea of creating a virtual reality (VR) next to augmented reality (AR) is also becoming the hot passion of the day.*

### Introduction to Three.js 3D Modeling

In this article, we will introduce Three.js and its wide applications in summary and then we will get started with the very first project you can start in Three.js as a beginner. If you have had experience with animations and 3D modeling in Blender, you will find Three.js like a piece of cake and if you haven't, don't worry because we are going to explain all the details of the simplest project you can find in Three.js.

Three.js is a cross-browser JavaScript library and API which uses WebGL to create and display 3D animated computer

graphics in a web browser. The range of applications you can create using Three.js is wide. You can use this library for creating online games, animated websites, user-interactive 3D widgets, Metaverse, and so on. Imagine creating a 3D online shopping center where you can virtually see the items you want to purchase or seeing a 3D animated architecture of a building before it is built or wandering inside a virtual world with your 3D animated Avatar. The variety of use cases for Three.js is so wide that we can confidently say "Sky is the limit".

## Getting Started with a Simple Project of Three js 3D Modeling

First off, make sure you have npm installed. You can start your first project by git cloning one of the simple Three.js projects to find a boilerplate for further development. Most of these projects can be found on Github, animate simple objects like a cube, sphere, torus, and so on. The animation is usually rotation. Running these projects and developing them step by step by changing some of the features about rendering, lighting, camera perspective, the color of the objects, textures, animations, and so on will help you find the necessary tools to design more complex objects.

```
mkdir Starting_Three
```

```
cd Starting_Three
```

```
git clone https://github.com/designcourse/threejs-webpack-starter
```

 Then,

enter the following commands one by one to install the dependencies: `npm install`

```
npm install webpack-dev-server -g
```

 Now, change the directory to `threejs-webpack-starter`: `cd threejs-webpack-starter`

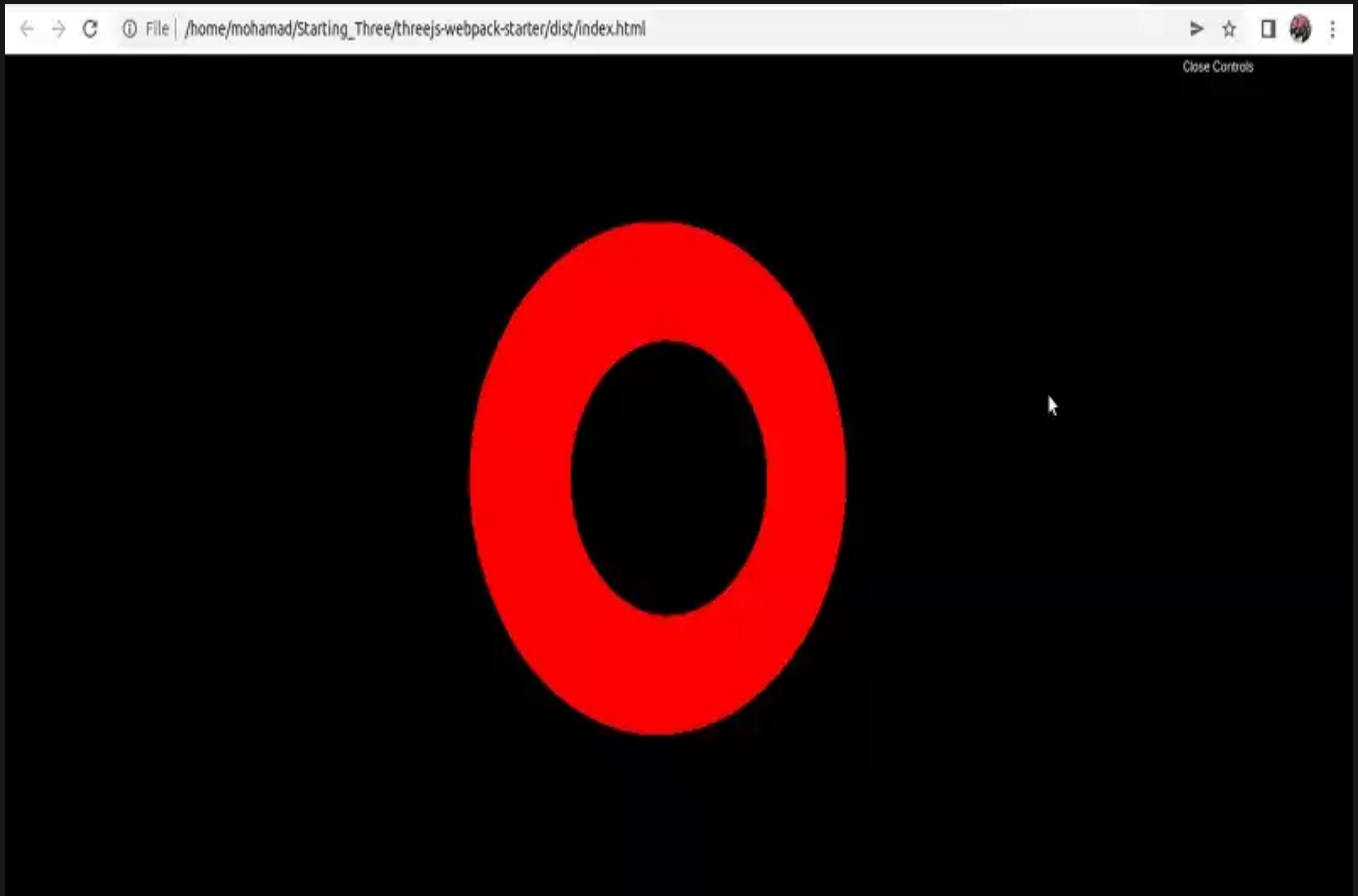
Then, enter the followings one by one to get a pre-

designed 3D animated torus: `npm i`

```
npm run dev
```

```
npm run build
```

 And, you will be able to see the animated (rotating) torus like this in your browser:



Now, let's head over to our the JavaScript code behind the scenes and see the elements of this 3D design and rendering. In src folder, script.js file, you will see:

```
import './style.css';
import * as THREE from 'three';
import { OrbitControls } from
'three/examples/jsm/controls/OrbitControls.js';
import * as dat from 'dat.gui';

// Debug
const gui = new dat.GUI();
// Canvas
const canvas = document.querySelector('canvas.webgl');
// Scene
const scene = new THREE.Scene();
```

Up to here, you can write the code and forget it because you are not going to modify it. But from this part on, you should begin to look closer at the features of the design.

### **3D Modeling Features in Three js**

The first important feature is the geometry of the object or in other words the very first shape that you start your design with. This shape could be a cube, a cylinder, a sphere, and so on. Here, for instance, we have a torus.

The second one is the materials which are the type of the mesh and its color. As you can see the color can be set by a hexadecimal number. In your code, the number is `0xff0000`. However, we have changed the number to `0xff0ff0` which will turn the color to pink.

After defining the objects and the materials, It is time to create the object out of these features. To do so, we can use the new `THREE.Mesh(geometry, material)` to apply the defined geometry and material to the object and then use the `scene.add()` to add the object to the scene. There are other features of the scene such as lighting, camera, animation, and so on.

```
// Objects
const geometry = new THREE.TorusGeometry(0.7, 0.2, 16, 100);
// Materials
const material = new THREE.MeshBasicMaterial();
material.color = new THREE.Color(0xff0ff0);
// Mesh
const sphere = new THREE.Mesh(geometry, material);
scene.add(sphere);
// Lights
const pointLight = new THREE.PointLight(0xffffffff, 0.1);
pointLight.position.x = 2;
pointLight.position.y = 3;
pointLight.position.z = 4;
scene.add(pointLight);
```

Of course, we are not going to talk about the rest of the code as most of the parts are not the subject of our article and for now, we can set and forget them. We are going to talk about these details in the next articles where the animation and other details are going to be modified.

### **Blender and Three.js**

If you have ever worked with Blender, you are most probably familiar with many concepts of 3D modeling, designing, rendering animations, shadings, and so on. The good news is that Three.js is pretty much the same as Blender with the same geometry objects and meshing formations. Another good news for Blender developers is that you can import the Blender 3D models into Three.js. So if you have a good experience in 3D modeling in Blender you can export your UV mapped 3D models as glTF and then import these glTF files in Three.js. We will explain the detailed procedures in the next tutorials.

### **Final Word**

In this article, we have introduced Three.js as a tool to create 3D web applications, 3D games, Virtual Reality, Augmented Reality, and other 3D-based productions leading to the development of the Metaverse. We have also compared Blender

with Three.js, explaining the similarities and the capability of importing the 3D models from Blender to Three.js.

Moreover, we have provided the guidelines for setting up the simplest the project you can run in Three.js and explain the important scripts that you can modify to change the 3D model features. Hopefully, we will explain more details about the lighting, shading, animations, and other useful details of decorating a model in Three.js.

