

# Three.js Learning Roadmap

## UI Basic (Compulsory for Three.js):

HTML basics, follow from [W3Schools HTML Tutorial](#) through to this section [HTML Versus XHTML](#)

CSS basics, follow from [W3Schools CSS Tutorial](#) through to [w3schools.com: CSS Specificity](#) also cover flexbox in advanced section (flex is now a standard, no more an advanced thing though) [Introduction to CSS layout - Learn web development | MDN](#)

[Create a Complete Responsive Website Using Bootstrap](#)

[HTML5 and CSS3 Responsive design with media queries \(basic understanding\)](#)

Difference in %, px, em, rem, vh, vw etc.

## Git Training (Optional for Three.js):

Have a basic how know of GitHub and how to use git commands and GitHub desktop. Below are two helping links in this regard:

<https://gitexercises.fracz.com/exercise/master>

<https://learngitbranching.js.org/>

## JavaScript + ES6 Tutorials (Compulsory for Three.js):

<https://www.freecodecamp.org/learn/javascript-algorithms-and-data-structures/es6/>

## Basic (Compulsory for Three.js)

Reference Book for JavaScript: [Eloquent JavaScript](#)

- [Variables](#)
- [Data Types](#)
- [Conditional Statements](#)
  - if else
  - switch
  - ternary
- [Loops](#)
  - While
  - for
  - for of
  - for in
- [Comparison Operators](#)

- [Functions](#)
- [Functions with default parameters](#)
- [Callbacks](#)
- [Higher Order Functions](#)
- [Arrays](#)
- [Objects](#)
- [null vs undefined](#)
- [truthy / falsy](#) concept
- [Error Handling](#)

## Advanced (Optional for Three.js):

Reference video lectures for advanced concepts of JavaScript: [Advance Concepts](#)

- [Hoisting in JavaScript](#)
- [Variable](#) and [Function](#) Hoisting
- [Scope Chain and Lexical Scoping](#)
- [IIFE](#)
- ["use strict" mode](#)
- [Closures in JavaScript](#)
- [Bind\(\) method](#)
- [this](#) keyword
- [Iterators and Generators](#)
- [Classes, Objects,](#) and [Inheritance](#)
- [Callback Hell](#)
- [Promises](#)
- [Async / Await](#)
- [Event Loop](#)

## Arrays and Object Functions (Compulsory for Three.js):

Reference to learn and practice array functions: [w3schools array functions](#). Practice all the given functions:

- [find\(\)](#)
- [findIndex\(\)](#)
- [indexOf\(\)](#)
- [filter\(\)](#)
- [includes\(\)](#)
- [push\(\)](#)

- [pop\(\)](#)
- [map\(\)](#)
- [reduce\(\)](#)
- [slice\(\)](#)
- [splice\(\)](#)
- [shift\(\)](#)
- [unshift\(\)](#)
- [valueOf\(\)](#)
- [forEach\(\)](#)
- [some\(\)](#)
- [Object.keys\(\)](#)
- [Object.values\(\)](#)
- [Object.entries\(\)](#)

## ES6 / ES7 and latest versions (Optional for Three.js):

ECMAScript is nothing but a collection of some new features added to JavaScript that obviously are designed to make our life easier. Have a look at the following points and see how it solves the problem in a friendly way.

- [Object destructuring](#)
- [Object literal](#)
- [Spread operator](#)
- [Rest operator](#)
- [Arrow Functions](#)
- [Nullish coalescing operator](#)
- [Optional chaining operator](#)
- [var vs let vs const](#)
- [Trailing commas](#)

## Three.js Official Documentation:

Below is the link to official Three.js documentation:

<https://threejs.org/docs/index.html>

# Three.js Basics

## Basics of WebGL:

WebGL (short for Web Graphics Library) is a JavaScript API for rendering interactive 2D and 3D graphics within any compatible web browser without the use of plug-ins. Just learning the basics of WebGL would be enough but if you want to learn more here is helping link:

[https://developer.mozilla.org/en-US/docs/Web/API/WebGL\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API)

## Basic:

- [Scene](#)
- Webpack bundling
- Transforming Objects
- [Animations](#)
- [Cameras](#) (All types of cameras especially [Orthographic](#) and [Perspective](#) Camera)
- Fullscreen Responsiveness and Resizing
- [Geometries](#) (All types of Geometries)
- Debug UI ([dat.gui](#))
- [Controls](#) (All types of Controls Orbit, Trackball, Drag etc.)
- LookAt method and it's usage
- [Textures](#)
- [Materials](#)
- [Lights](#) and Light Helpers
- [Shadows](#)
- Particles
- [Raycaster](#)
- Importing and using 3D Models
- [Loaders](#)

## Mini Project:

Make a mini project that includes all the basics of Three.js and get your grip stronger on Three.js.

## Shaders:

Helping link: <https://www.shadertoy.com/>

- GLSL Basics
- Shaders Basic
- Vertex Shader
- Fragment Shader
- Attributes
- Uniforms
- Built-in Functions
- Noises
- Texturing with Shaders
- Post Processing

## Extra Techniques

- Physics
- Realistic Render
- Modified Materials
- Mixing HTML and WebGL
- Importing and Optimizing Scene
- Baking and Importing scenes
- Performance Tips