Three.JS Building Blocks

Alex Mackey
simpleIsBest.co.uk

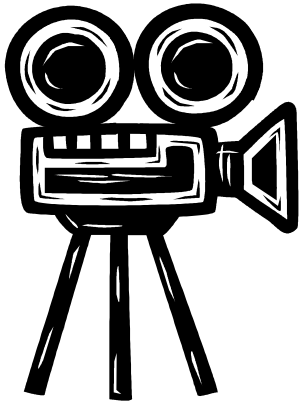
pluralsight
hardcore dev and IT training

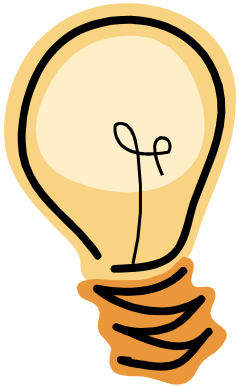
Introduction

- Components of a Three.JS application
- Hello World Demo
- Three.JS coordinate system
- Object3D
- Changing an objects position & size & rotation
- Child objects & transformations
- Common mistakes


HOLLYWOOD
PRODUCTION
DIRECTOR
CAMERA
DATE
SCENE
TAKE

Three.JS Scene Components

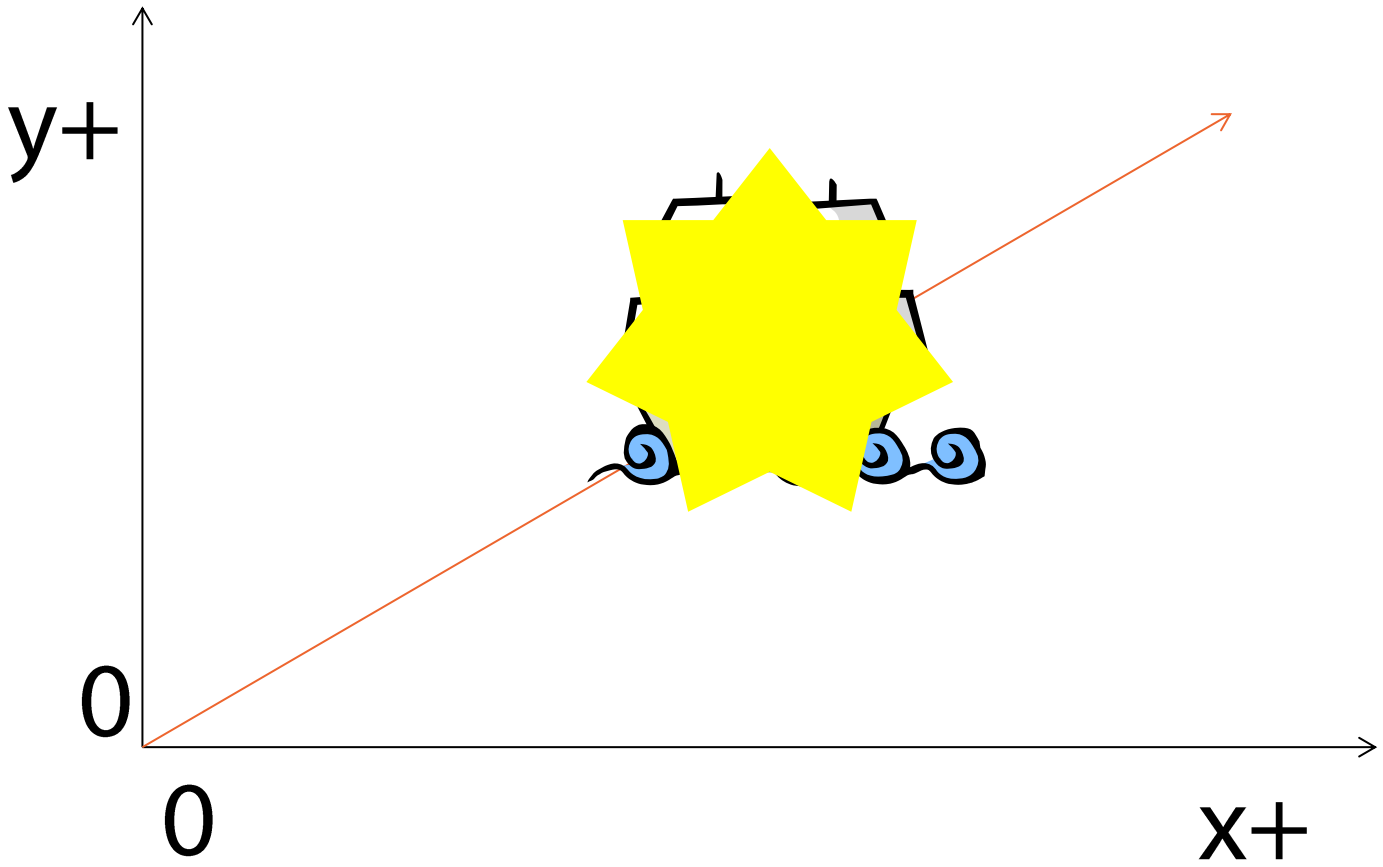




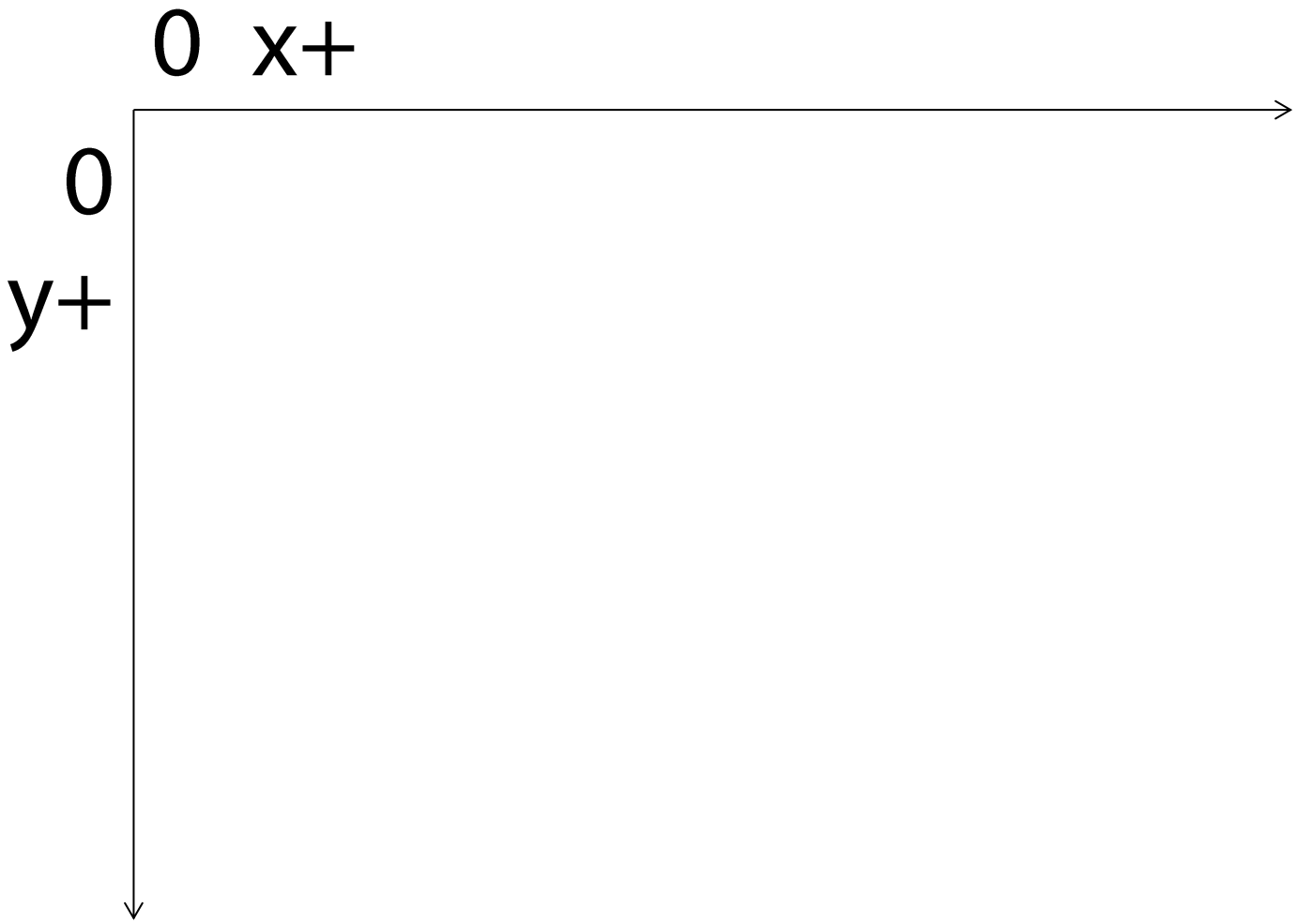
Coordinates

0,0,0

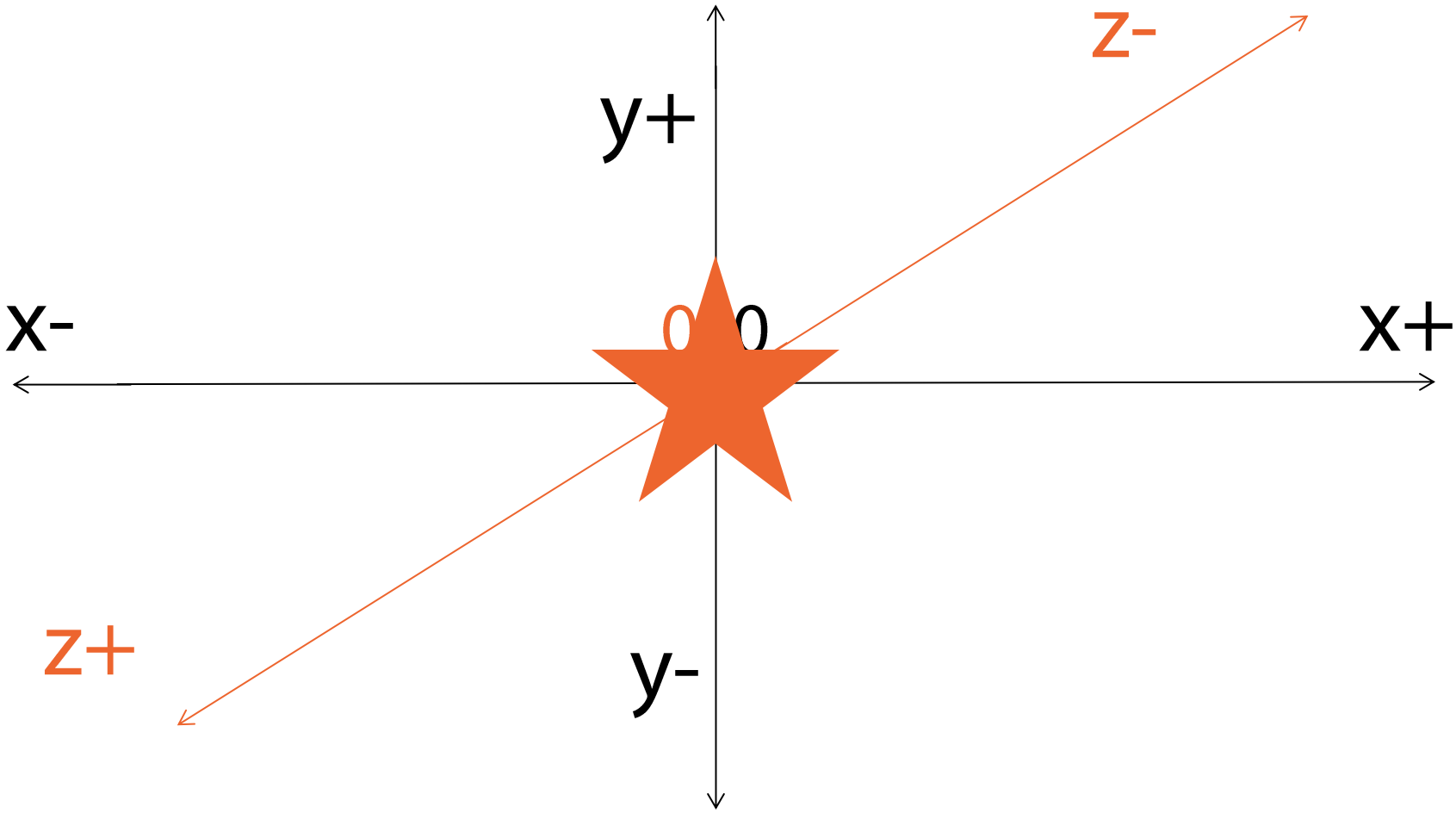
Coordinates (Graph)



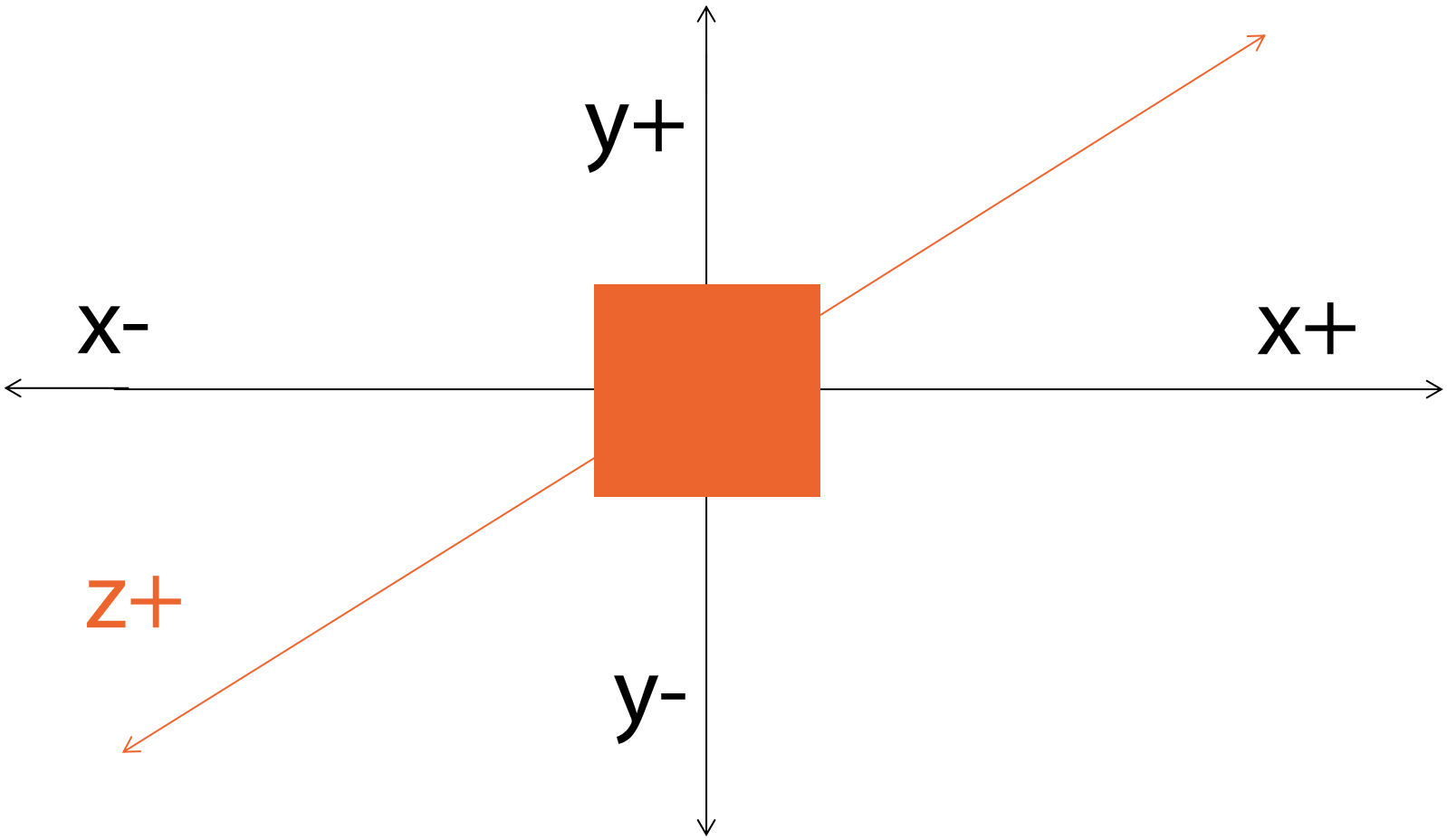
Coordinates (CSS)



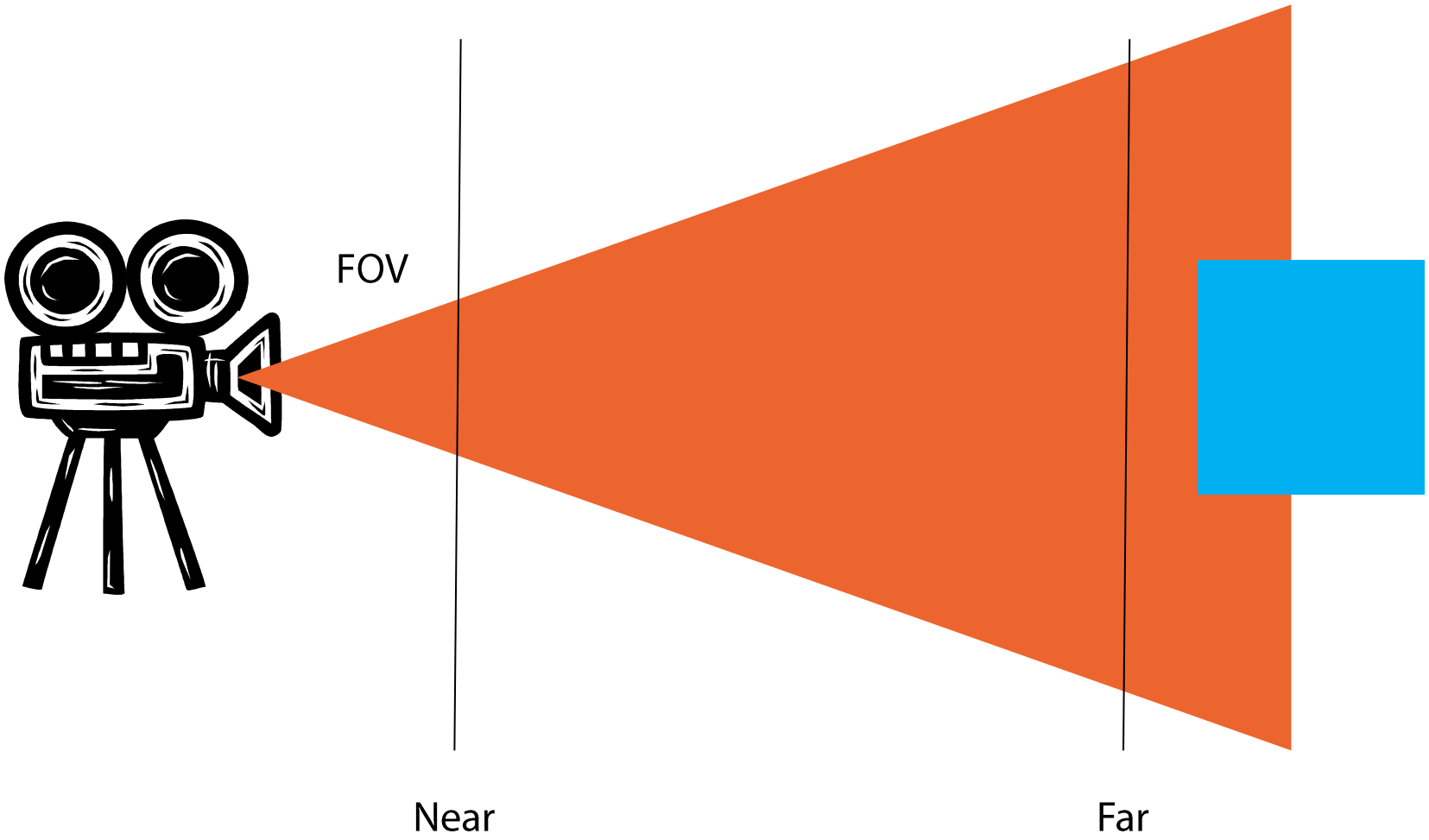
Coordinates (Three.JS)

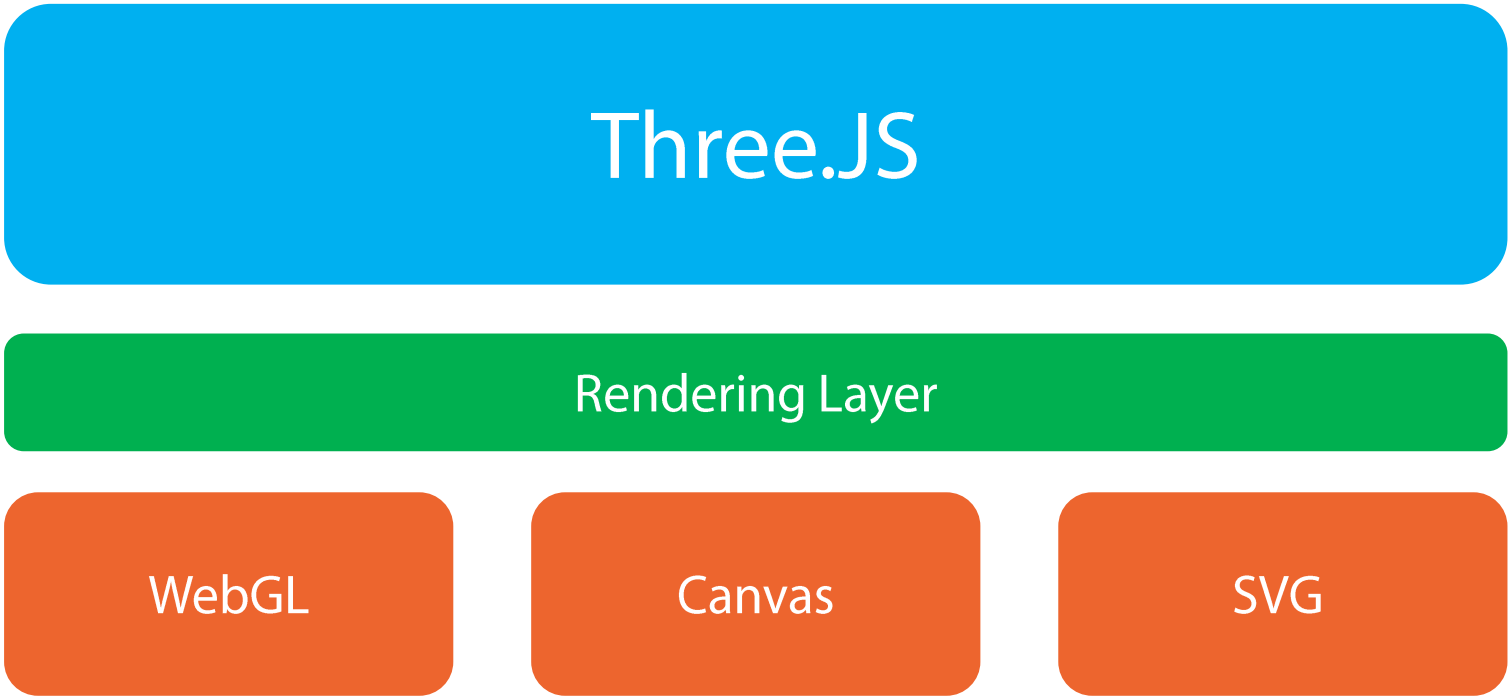


Coordinates (Three.JS)



Camera




Three.JS
Rendering Layer
WebGL
Canvas
SVG

Object3D Important Properties

- Id
- uuid
- name
- position
- userData
- parent/children

Object3D Important Methods

- getObjectByName
- getObjectById
- lookAt

Manipulating Objects

- Position
- Scale
- Rotation

Position

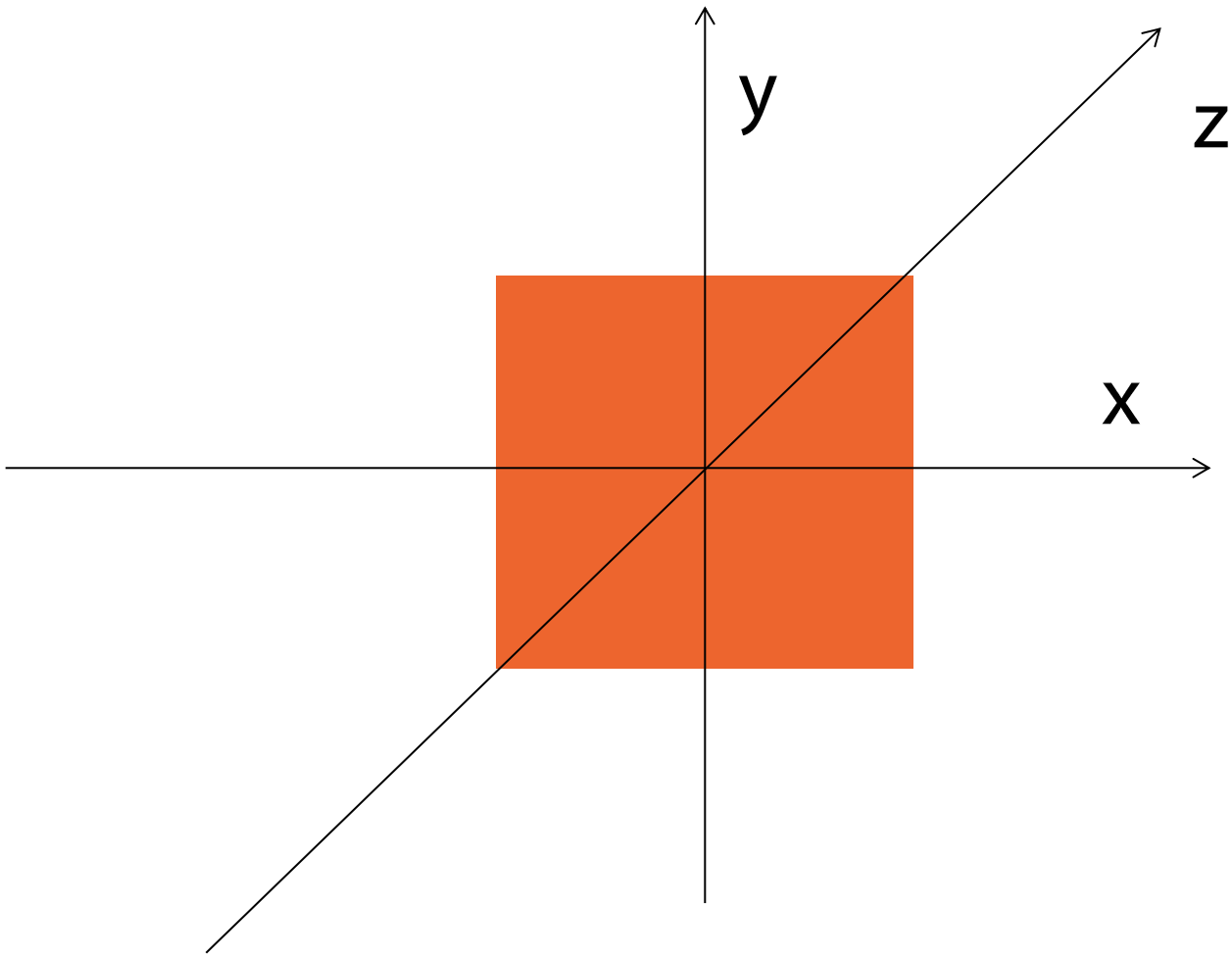
```
Object.position.x = value

Object.position.set(x, y, z)

Object.position =
new THREE.Vector3(0, 0, 0);
```

Scale

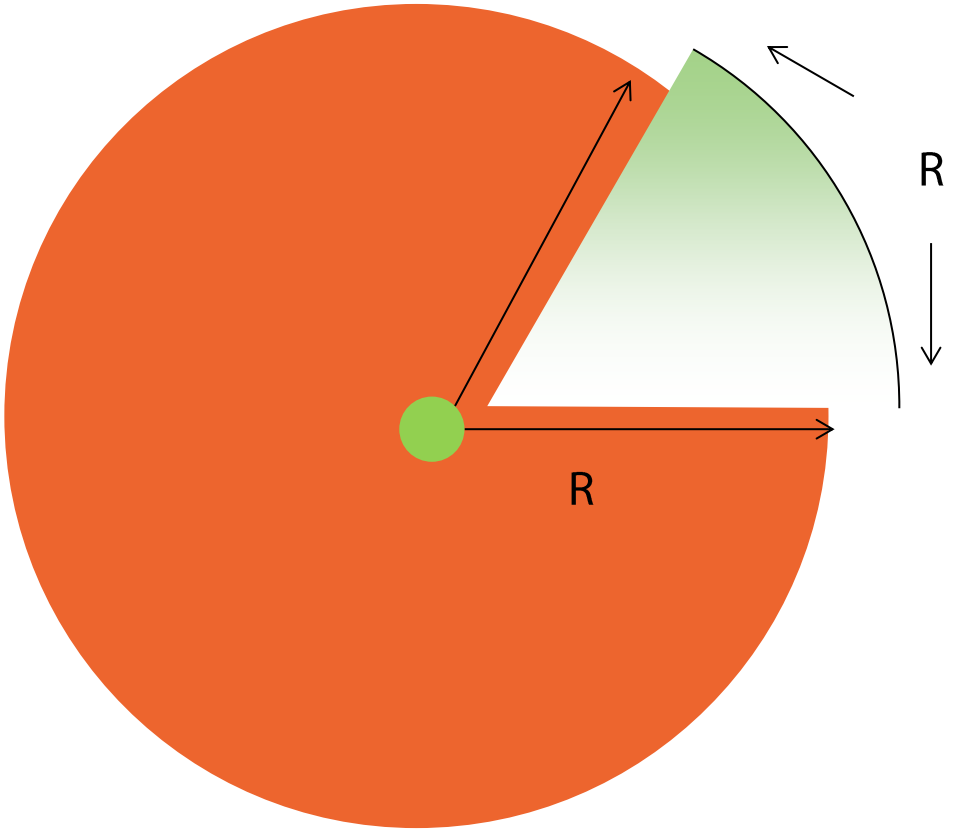
Object.scale.x = value

Object.scale.set(x, y, z)

Rotation



Radians



$2 * \pi$ (or 2 x 3.14) =

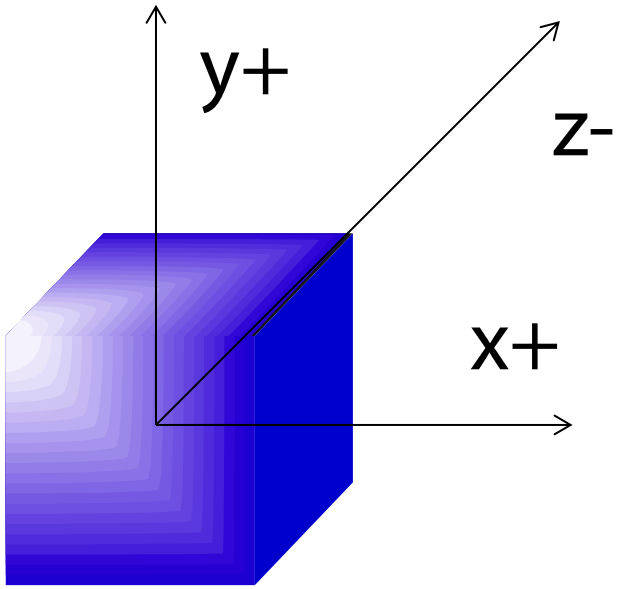
6.28 radians

Converting Degrees to Radians

radians = degrees * (pi/180)

degrees = radians * (180/pi)

Rotations Continued

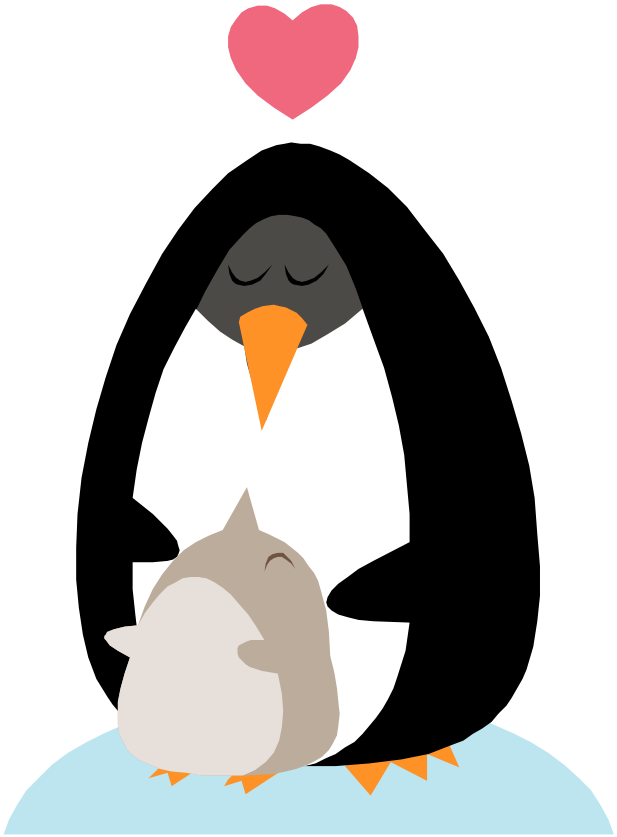


Rotation Syntax

Object.rotation.x = value

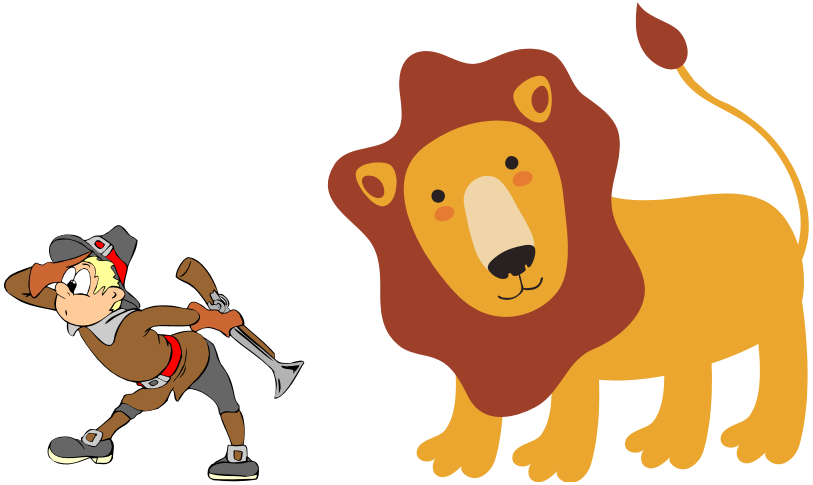
Object.rotation.set(x, y, z)

Child Objects & Transforms

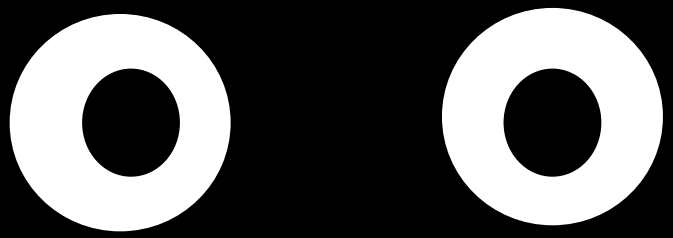


Common Problems

It’s Behind You!



Forgetting to Add Lights



Scale

Scale

Summary

- All Three.JS applications contain a renderer, scene, camera & light
- Object3D
- Objects are hierarchical
- Objects can be positioned, scaled, rotated
- Common issues and how to avoid them