**Three.js Materials**

Three.js provides various materials to control object appearance. Choosing the right material depends on your needs.

**1. Setting Material Properties**

Properties can be set during creation or modified later.

Set at creation:

const material = new THREE.MeshPhongMaterial({

color: 0xFF0000, // Red

flatShading: true,

});

Modify after creation:

const material = new THREE.MeshPhongMaterial();

material.color.setHSL(0, 1, 0.5); // Red

material.flatShading = true;

Color formats:

material.color.set(0x00FFFF); // Hex

material.color.set('red'); // CSS color

material.color.setRGB(1, 0, 0); // RGB

material.color.setHSL(0.5, 1, 0.5); // HSL

**2. Common Materials**

MeshBasicMaterial: ignores lighting, ideal for UI or backgrounds.

MeshLambertMaterial: calculates lighting per vertex, good for low-end devices.

MeshPhongMaterial: calculates lighting per pixel, supports specular highlights.

MeshToonMaterial: uses gradient shading for a cartoon effect.

MeshStandardMaterial: is a PBR material with roughness and metalness.

MeshPhysicalMaterial: adds clearcoat for glossy effects.

ShadowMaterial: captures shadows.

MeshDepthMaterial: renders depth data.

MeshNormalMaterial: visualizes normals.

ShaderMaterial and RawShaderMaterial: allow custom GLSL shaders.

**3. Choosing the Right Material**

For no lighting, use MeshBasicMaterial.

For simple lighting, use MeshLambertMaterial.

For specular reflections, use MeshPhongMaterial.

For realistic surfaces, use MeshStandardMaterial with roughness and metalness.

For glossy coatings, use MeshPhysicalMaterial with clearcoat.

Example:

const phong = new THREE.MeshPhongMaterial({ shininess: 100 });

const standard = new THREE.MeshStandardMaterial({ roughness: 0.5, metalness: 1 });

const physical = new THREE.MeshPhysicalMaterial({ clearcoat: 0.5 });

MeshToonMaterial creates a cartoon-like shading.

MeshNormalMaterial and MeshDepthMaterial are useful for debugging.

**4. Additional Properties**

flatShading controls the smoothness of shading.

const material = new THREE.MeshPhongMaterial({ flatShading: true });

side determines which faces are rendered.

const material = new THREE.MeshBasicMaterial({ side: THREE.DoubleSide });

**5. Updating Materials**

Some properties require material.needsUpdate = true to apply changes.

material.flatShading = true;

material.needsUpdate = true;

Switching between textured and non-textured materials also requires an update.

material.map = new THREE.TextureLoader().load('texture.jpg');

material.needsUpdate = true;

**6. Performance Optimization**

MeshBasicMaterial is the fastest, as it doesn't calculate lighting.

MeshLambertMaterial is efficient for basic lighting.

MeshPhongMaterial requires more GPU power due to specular calculations.

MeshStandardMaterial is more realistic but computationally expensive.

MeshPhysicalMaterial is the most resource-intensive, best for high-end devices.

For mobile or low-end devices, use MeshBasicMaterial or MeshLambertMaterial instead of MeshStandardMaterial or MeshPhysicalMaterial.