



# Clock - Les bases

# La classe Clock de Three.js

**Clock** est une classe de Three.js permettant de mesurer le temps écoulé lors de l'exécution de notre application 3D

Le constructeur de **Clock** ne nécessite pas de paramètre

```
var clock;  
clock = new THREE.Clock();  
  
function render()  
{  
    var delta = clock.getDelta(); //get delta time between two frames  
    var elapsed = clock.elapsedTime; //get elapsed time  
  
    [...]  
}
```

# getDelta et elapsedTime

Pour mesurer le temps écoulé, on utilisera principalement deux concepts : **getDelta** et **elapsedTime**

La méthode **getDelta** est généralement utilisée dans la boucle principale d'animation

Dans ce cas, elle permet d'obtenir le nombre de secondes écoulées depuis la dernière *frame*

La propriété **elapsedTime** conserve une trace du temps total écoulé depuis que l'horloge fonctionne

```
function render()
{
    var delta = clock.getDelta(); //get delta time between two frames
    var elapsed = clock.elapsedTime; //get elapsed time

    [...]
}
```

# Pourquoi utiliser Clock ?

Dans l'exemple *Hello World*, nous avons créé un cube en rotation, en incrémentant sa propriété **rotation** à chaque *frame*

La vitesse du cube va donc dépendre du nombre de FPS (*Frames* par secondes) de l'appareil affichant notre cube 3D, ce n'est pas acceptable !

```
// Hello World code Chapter 2
function render()
{
    cube.rotation.y += 0.01; // raw value increment at each frame
    cube.rotation.x += 0.005; // raw value increment at each frame

    renderer.render(scene, camera);
    requestAnimationFrame(render);
}
```

# Pourquoi utiliser Clock ?

Avec la classe **Clock**, nous allons utiliser un nouveau facteur : Le temps écoulé depuis la dernière *frame*

Ainsi, la vitesse du cube sera indépendante du nombre de FPS : La vitesse de rotation sera toujours la même, peu importe le nombre de FPS !

```
// Clock update
function render()
{
    var delta = clock.getDelta(); // Elapsed time since last Frame

    cube.rotation.y += 0.6 * delta; // delta
    cube.rotation.x += 0.3 * delta; // delta

    renderer.render(scene, camera);
    requestAnimationFrame(render);
}
```

# Clock - Les bases

Si vous êtes prêts, passons à la pratique !

