# **Quaternion**

**Rotation and Orientation in Unity:**

Rotation in 3D are usually represented in Quaternions or Euler Angels

**EULER ANGLES:**

Euler angles are represented by 3 angle values for x,y,z that are applied simultaneously

To apply euler rotation for a particular gameobject,each rotation value is applied in turn,as a rotation around its corresponding axis

**QUATERNIONS:**

Quaternions are used to represent rotations.

They are compact, don't suffer from gimbal lock and can easily be interpolated. Unity internally uses Quaternions to represent all rotations.Quaternion components (x,y,z,w); These components are independent and we should never adjust them individually.

Quaternion.LookRotation

Quaternion.Angle

Quaternion.Euler

Quaternion.Slerp

Quaternion.FromRotation

Quaternion.identity

**Quaternion.LookRotation:**

Creates a rotation with the specified upward and downward directions.

**Quaternion.Angle:**

Returns the angle in degrees between two rotations a and b.

**Quaternion.Euler:**

Returns a rotation that rotates z degrees around the z axis, x degrees around the x axis, and y degrees around the y axis; applied in that order.

**Quaternion.Slerp:**

Quaternion A quaternion spherically interpolated between quaternions a and b.

**Quaternion.FromRotation:**

Creates a rotation which rotates from fromDirection to toDirection.

**Quaternion.identity:**

This quaternion corresponds to "no rotation" - the object is perfectly aligned with the world or parent axes.