

## How to implement scripts

Scripts enable the implementation of dynamic logic in editor-based applications. For each 3D object in the scene graph, it is possible to add one or more scripts.

**Lifecycle methods** Each script can implement the following lifecycle methods:

- **update()**: Executed right before a frame is going to be rendered. Its primary purpose is to update the state of the 3D object which owns the script. The method has an **event** parameter which holds a **time** and **delta** property. **time** represents the elapsed time in milliseconds and **delta** represents the time between two frames in milliseconds.
- **init()**: Executed once after the application has been loaded.
- **start()**: Executed once when the application is ready to start rendering.
- **stop()**: Executed once when the application is stopped.

**Events** It is also possible to implement event listeners for selected browser events. The following events are supported by the editor:

- **keydown**
- **keyup**
- **pointerdown**
- **pointerup**
- **pointermove**

**Script variables** Certain application components are accessible in the scope of scripts as variables:

- **player**: A reference to the application player (a wrapper component which executes the editor application).
- **renderer**: A reference to the renderer.
- **scene**: A reference to the scene graph.
- **camera**: A reference to the application's camera.

## Miscellaneous

- Code outside of lifecycle and event listeners is immediately executed when the script is loaded.
- The **this** reference can be used to refer to the 3D object which owns the script.