# 👌Developing With the 8th Wall Cloud Editor (Live Learning: Session 1)

We cover some image tracking and world tracking updates, then a full walkthrough of how to develop with 8th Wall Cloud Editor.

In this section, we'll cover:

\* 3D Model Placement

\* Add Animation to your 3D Model

\* Gesture Controls

\* Using World Tracking + Image Targets

\* Adding Media Recorder

\* Source Control Publishing

Add 3D model onto plan

Make them animate

Make them Interactable

Through the use of:  xrextras-gesture by raycaster component

Different types of image targets:

Flat

Cylindrical

Conical

xrextras-one-finger-rotate gltf-model

photo and video-capture ability

head.html is optional; elements will be added to your html head before app.js is loaded.

Use "8thwall:" meta tags to hook into 8th Wall's build process and developer tools.

in head.html is where you include meta tags to packages that are required for your application, such as xrextras, aframe and more, into your project.These give you the ability to render and perform powerful features in webxr.

in app.js we are registering our components before body.html is loaded. Its important to note what a component is and why we need a component.

Components of A-Frame’s [entity-component framework](https://aframe.io/docs/1.3.0/introduction/entity-component-system.html) are JavaScript modules that can be mixed, matched, and composed onto entities to build appearance, behavior, and functionality. We can register component in JavaScript and use it declaratively from the DOM. Components are configurable, reusable, and shareable. Most code in an A-Frame application should live within components.

// app.js is the main entry point for the app. Code here will execute after head.html

// is loaded, and before body.html is loaded

// Register custom A-Frame components in app.js before the scene in body.html has loaded.

**Property Types**

export const tapPlaceComponent = {

schema: {

min: {default: 6},

max: {default: 10},

},

init() {

\_\_\_\_\_\_;

}

From Aframe documentation - [Components in aframe](https://aframe.io/docs/1.3.0/core/component.html) [click on link]

Property types primarily define how the schema parses incoming data from the DOM for each property. The parsed data will then be available via the data property on the component’s prototype.

**AFrame.Components.Shadow**

**Shadow**

See AFrame.Components.Light for corresponding light component.

type\_ : Shadow -> Property

Defines shadow map type (basic, pcf, pcfsoft) with varying appearance and performance characteristics.

Default : pcf (*percentage closer filtering*)

**Cast / Receive**

cast : Bool -> Property

Whether the entity casts shadows onto the surrounding scene.

Default : true

receive : Bool -> Property

Whether the entity receives shadows from the surrounding scene.

Default : true