



# Exploring WebXR

WebXR is the spiritual successor to the WebVR 1.0 and 1.1 API, sitting on top of native WebGL APIs. Where WebGL only allowed for 3D VR experiences WebXR, when fully specified and implemented in browsers, will allow for both 3D and [Augmented Reality](#) experiences using either phones or [Head Mounted Displays](#) to interact with virtual objects in the real world.

I'm not advocating for everyone to learn WebGL, at least not initially; there will be times when you will need shaders to do specific work that can't be done with the higher level APIs but that's far down the line from where we will start this exploration.

## Creating content for VR and XR

The objectives for this reasearch are:

- Create content using Three.js and A-Frame
- Explore the difference between 2-D, 3D, and WebXR content
- What additional constraints, if any, do the polyfills impose.
- This is also a good time to see how we can use `model-viewer` as a Magic Window type tool for displaying 3D content in a 2D environment
- Explore what it would be like to get 2D environments inside a 3D world

## Links and resources

- Libraries
  - [Three.js](#)
  - [Babylon.js](#)
  - [a-frame](#)
- Polyfills and extensions
  - [model-viewer](#)
  - [WebXR Polyfill](#)
  - [three.ar.js](#)
- Intel Open Source — Creating responsive VR experiences
  - [VR Concepts And The Immersive Web](#)
  - [Rendering immersive web experiences with Three.JS and WebXR](#)
  - [Adding support for VR inputs with WebXR and Three.JS](#)

- Examples
  - [SponzaVR](#)