

## Code a VR world and view it on your phone!


You can code a VR scene on your computer then view it on your phone. The only other things you need are:

- A modern web browser, like Chrome or Firefox
- A VR viewer like Google Cardboard


Note: Both Android and Apple have recently changed the default settings for Safari and Chrome browsers on phones. You may need to update the settings for Motion and Orientation Access to get WebVR to work on your phone

### Step 1 – Get Started


To remix our VR example, go to <https://glitch.com/~lead-turret> Then click on the **View Source** button.



**lead-turret**




Base for creating virtual reality experiences on the Web using A-Frame. Based on code from <https://learn.framevr.io/project1/part4> #aframevr #webvr

Show 

View Source

### Step 2 – Get Started

This will take you to a new page with a description of the project and a button saying **Remix to Edit** in the top right corner. Click this and you'll get your own copy of the code that you can make changes to.

Remix to Edit 

### Step 3 – Get Started

There's a list of filenames on the left of the screen. Click on the one that says index.html to get the code.

New File ▾

assets

LICENSE.md

README.md

**index.html** ⋮

```
1<
2<
3<  <html>
4<
5<  <head>
6<    <script src
7<  </head>
8<
9<  <body>
10<
11<    <a-scene>
12<
13<  <!--      3D n
14<    <a-gltf-n
15<  </a-gltf-n>
16<  </body>
17<
18<  </html>
19<
```

## Step 4 – Get Started

You should now be able to see the code that creates our VR scene. This is **HTML**, the language used to describe the structure of web pages and tell browsers what to display.

The words inside pointed brackets are called **tags**. They describe a `<html>` particular element on the page.

We've added a link to a file called *aframe.min.js* that lets us use lots of extra tags that Mozilla's Aframe project has defined to allow people to code 3D VR objects.

## Step 5 – Change the colour of 3D geometric objects

To start with, try changing the colour of one of the geometric objects in the scene, for example the Torus Knot sculpture. This is the code for it:

```
13<!-- torus knot sculpture -->
14   <a-box position="4 1 -8" scale="2 2 2" color="white"></a-box>
15   <a-torus-knot position="4 4 -8" color="blue"></a-torus-knot>
16
```

Pick a colour from the list of colour names that HTML knows about and replace the words in quotes after **color=** with the names of new colours.

Note: HTML always uses the American spelling "color"

Try this with the Geometric sculptures and Floating Spheres as well.

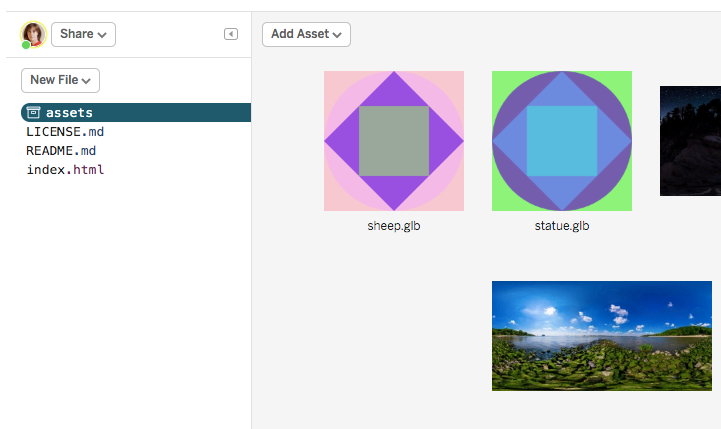
## Step 6 – Change the background picture

The background picture is defined by this line of code, beginning with the tag **<a-sky>**

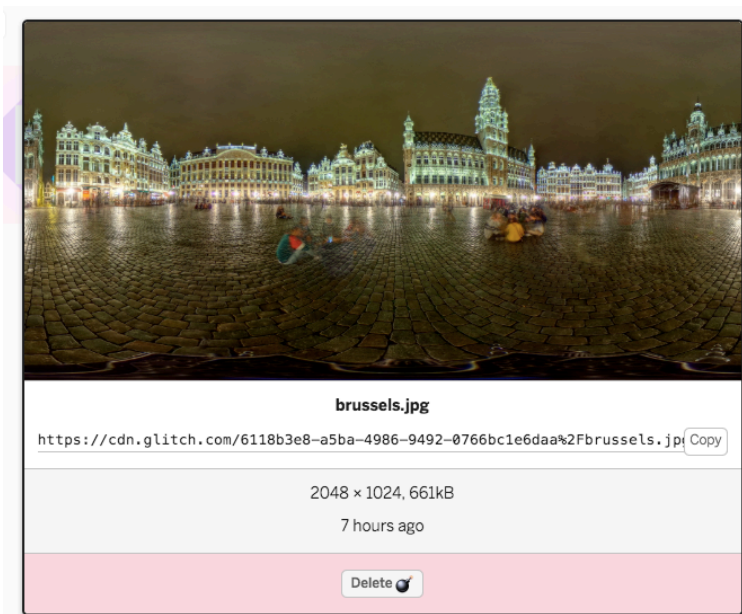
```
<!-- Background picture -->
<a-sky src="https://cdn.glitch.com/0ad5e2ab-7d80-408e-b7ed-2cc874f56810%2Fberlin.jpg?v=1563365457241"></a-sky>
```

To change it, we need to change the URL inside quotes after **src=**

Glitch has a folder called **assets** where you can upload Equirectangular photos to use as backgrounds. If you click on the assets folder on the left of the code panel you'll see



several that we've already uploaded.



If you click on one, for example the one called `brussels.jpg`, a window will open that includes a very long URL that starts <https://cdn.glitch.com/> and a button marked **Copy** beside it.

Click on the **Copy** button to copy the file's URL in the assets folder. Now close the window and click on `index.html` to see the code again.

Now delete the URL between the quotes and paste in the new one:

```
<a-sky src="https://cdn.glitch.com/6118b3e8-a5ba-4986-9492-0766bc1e6daa%2Fbrussels.jpg?v=1563352510967"></a-sky>
```

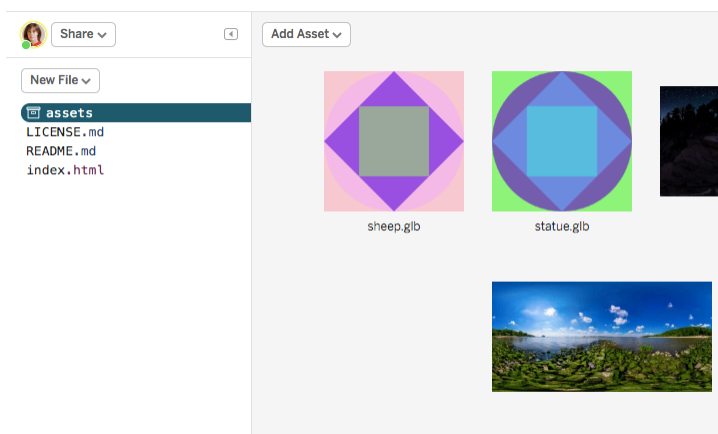
When you view the scene now the background should have changed to the new version.

## Step 7 – Add a 3D object

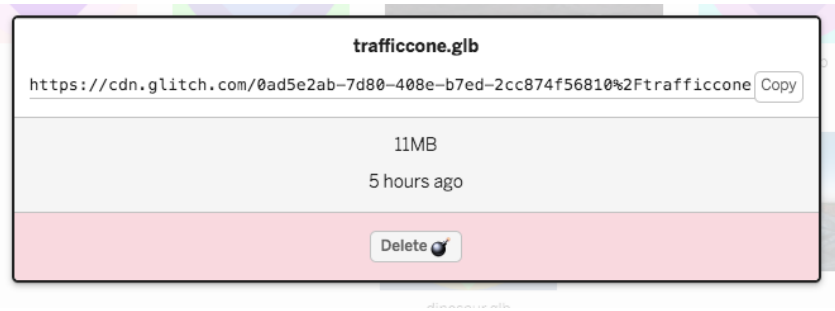
You can see there are several objects in our scene that aren't geometric figures: an octopus, a sheep, and an Easter Island statue. The code that puts an object in the scene starts with the tag `<a-gltf-model>`. Copy and paste one of these lines, but change the `id` value to "traffic-cone" and delete the URL between the quotes after `src=`

Just like background photos, the models are stored in the assets folder.

All the object file names end in **.gltb**



To add the 4<sup>th</sup> object in the assets folder, a traffic cone, click on the **trafficcone.glb** file, then click the **Copy** button in the window that opens.



Back in the index.html file, paste the URL of the traffic cone you just copied in between the quotes after src=. The sheep should now show up in your scene (although you may want to change its size, by editing the numbers in the scale attribute).

### Step 8 – Use the Inspector

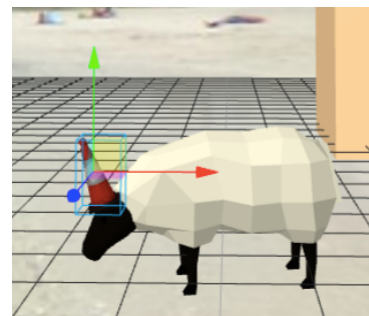
Your traffic cone probably showed up in the middle of the sheep, with a bit sticking out of its back. Which isn't ideal (particularly from the sheep's point of view). But it can be quite hard to get your objects positioned properly by editing the code alone. This is where the **Aframe Inspector** comes in.

Go to the tab that shows you the scene and press the following key combination:

**Ctrl + Alt + i**

This should start up the inspector window. Clicking on the line that says `<a-gltf-model traffic cone>` in the list on the left side of the screen should display a framework round the cone, along with some coloured arrows.

Clicking on the arrows should allow you to reposition the cone, in a tribute to GoMA's Duke of Wellington statue.



When you're happy with the cone's new position, click the icon at the furthest left hand corner of the positioning window (it's shown in blue here). This copies a line of code with the new position into the clipboard. Now, back in the index.html file, delete the original `<a-gltf-model id="traffic-cone">` line and paste in the line with the new position for the sheep.

