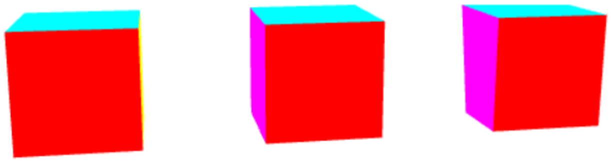
Lab 12

Here, we want to render three versions of the same cube,



and we will do this three different ways:

First, we try successive `gl.drawArray` calls.

But rather than duplicating the cube definitions, we only define one cube (with a color for each face), and we will reuse this cube definition. This one cube uses our standard definition of 8 `vertices` and 8 `vertexColors` (although we only use 6 of the defined colors, one for each face).

Since we have only one set of vertices and face colors defined, we only need to create one color buffer, `cBuffer`, and one vertex buffer, `vBuffer`.

Also, the cube is rendered in object coordinates, rather than directly in clip coordinates, so we will use three different modelview matrices (with the same projection matrix), one for each cube.

Each modelview matrix is initialized with:
`modelViewMatrix = lookAt(eye, at , up);`
which uses the pre-defined vectors `eye`, `at`, and `up`.

We scale and translate (the scale is "first" since it is lower in the code) the left cube, then just need to scale the cube in the middle, since its vertex definitions are scaled, and then scale and translate to render the right cube.