A buffer object to hold vertex data is also called a Vertex Buffer Object(VBO)
VAO: Vertex Array Object: descriptor of the vertex data, describes how the vertex are stored in the VBO.
Some Discussion about VAO: https://stackoverflow.com/questions/11821336/what-are-vertex-array-objects

Buffer Memory Allocation:

1. Ask WebGL for a buffer
2. Bind the buffer to a binding point
3. Fill the buffer with data

Browser supports WebGL: https://get.webgl.org

WebGL tutorial: https://www.tutorialspoint.com/webgl/index.htm

One feature of WebGL-Automatic memory management: no need for manual allocation of memory

Frame buffer: portion of graphics memory, hold the scene data.

Vertex Buffer Objects: store vertices data

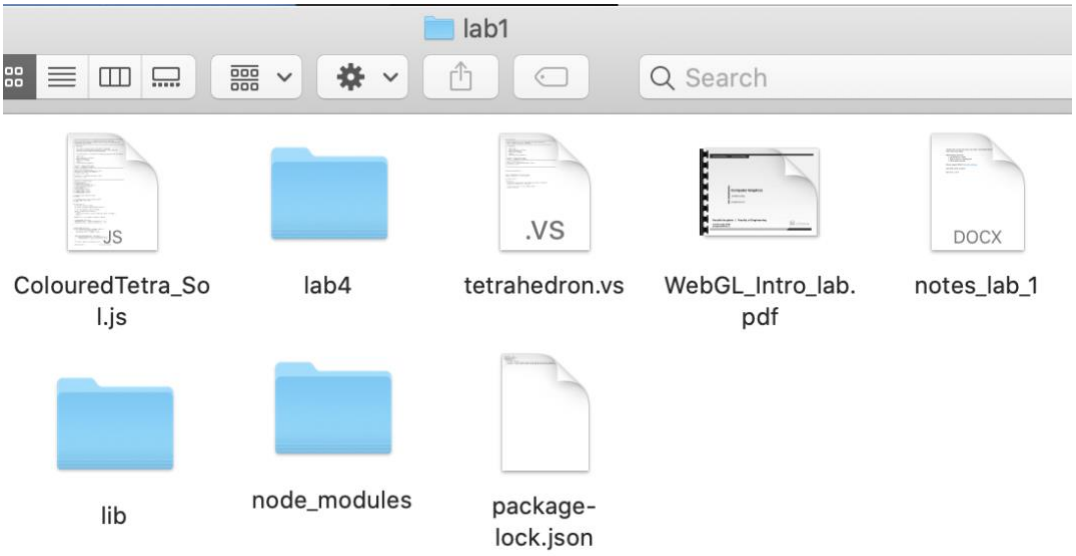
gl.STATIC_DRAW – Data will be specified once and used many times.

OpenGL Shader Language:

vec2, vec3, vec4 n-component floating point vector
mat2, mat3, mat4 2x2, 3x3, 4x4 float matrix

Uniform: read-only

files structure(Note: create a folder named as 'lab1')
go into lab1 folder and install gl-matrix package
npm install –prefix . gl-matrix
http-server . -p *port*



Note: If there is an error of starting code, like unsupported shader version, just try different browsers or update your browser. (In my case, Safari does not work, but Chrome works). Must use http-server to run these files in this lab.

Notepad++: open .vs .fs files