# Tutorial 6

Florian (florian.maushart@inf.ethz.ch)

November 2, 2020

## Intro

The goal of this tutorial is to get exposed to the basics of WebGL. Open **tut6.html** in any web browser to run the assignment. In Chrome you can press F12 to open the console and work from there. Alternatively, have your text-editor of choice open next to a browser window which runs the tutorial.

HINT: Remember to take baby steps in coding and refresh your browser often. If you write a new function, try to run it empty. If you modify a parameter, check if it influenced only what you wanted it to do. This makes debugging much easier for yourself and for us.

The bulk of the assignment is contained in **tut6.js**. You are given code which will draw a colorful triangle.

NOTE: The code was written to be minimal. It does _not_ represent the best practices of WebGL. This is fine. For a more correct implementation please see this tutorial: `https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Tutorial/Adding_2D_content_to_a_WebGL_context`
NOTE II: The Mozilla tutorials are amazing, and I recommend that you go through all of them in addition to the regular assignments for the next couple weeks.

## 1 Open up a few tabs of documentation

- WebGL uses Javascript: `https://developer.mozilla.org/en-US/docs/Web/JavaScript/A_re-introduction_to_JavaScript`
- This code uses **gl_matrix.js** as its math library. The way it works is kind of funky, but the library is good and very fast. Go find the docs at `http://glmatrix.net/`.

## 2 Get to know the code

In the recap we went over different parts of the code already. Can you explain:

- Why we call a WebGL context **stateful** or a state machine?
- What the getReadyToDraw() and setupShaderProgram() are doing?

Once you went over the code for a bit try to play with it and modify it such that:

- The triangle is red or upside down
- The triangle is no longer moving

## 3 Extend the code

Write a function called drawCircle() which draws a circle centered at the origin. Try to think about how you would go about drawing a circle before diving into documentation or tutorials. Hint: You can do this using gl.TRIANGLES.

## 4 "Extra credit"

You aren't actually graded on the tutorials, so the distinction between the actual assignment and the extra credit is somewhat arbitrary. This is an introduction, and there is obviously a lot more you can do with WebGL. Get creative!

- Color the triangle per vertex following this tutorial: `https://developer.mozilla.org/en-US/docs/Web/API/WebGL_API/Tutorial/Using_shaders_to_apply_color_in_WebGL`
- Take a look at three.js @ `http://davidscottlyons.com/threejs-intro`
- Get intimidated by how much you can do with two triangles and a fragment shader @ `https://www.shadertoy.com/view/Ms2SD1` (you will learn more about shaders in the following weeks).
- Have the canvas resize dynamically when you change the size of the web browser window. Hint: You can quite easily do this from within your .js, as you have access to the canvas.
- Load the shaders from file (a bit advanced, but ultimately cleaner). Note: Because this is out of the scope of this tutorial we will not be able to answer any questions on this during the exercise session.
- Try to implement a triangle or circle version of the old DVD screensaver, changing direction and color when bumping into the screen edges, as in `https://www.youtube.com/watch?v=Kxms-OtUXS0`. Note: Watching the full video is not required.