Exemple complet (quasi): vertex shader

```
<script id="shader-vs" type="x-shader/x-vertex">
attribute vec3 aVertexPosition;
attribute vec4 aVertexColor;

uniform mat4 uMVMatrix;
uniform mat4 uPMatrix;

varying vec4 vColor;
void main()
{
        gl_Position = uPMatrix * uMVMatrix * vec4(aVertexPosition, 1.0);
        vColor = aVertexColor;
}
</script>
```

Exemple complet (quasi): fragment shader

```
<script id="shader-fs" type="x-shader/x-fragment">

varying vec4 vColor;
void main()
{
        gl_FragColor = vColor;
}
</script>
```

Exemple complet (quasi)

```
 var shaderProgram;
 function initShaders() {
    var fragmentShader = getShader(gl, "shader-fs");
    var vertexShader = getShader(gl, "shader-vs");
    shaderProgram = gl.createProgram();
    gl.attachShader(shaderProgram, vertexShader);
    gl.attachShader(shaderProgram, fragmentShader);
    gl.linkProgram(shaderProgram);
    gl.useProgram(shaderProgram);

    shaderProgram.vertexPositionAttribute = gl.getAttribLocation(shaderProgram,
"aVertexPosition");
    gl.enableVertexAttribArray(shaderProgram.vertexPositionAttribute);
    shaderProgram.vertexColorAttribute = gl.getAttribLocation(shaderProgram, “aVertexColor”);
    gl.enableVertexAttribArray(shaderProgram.vertexColorAttribute);

    shaderProgram.pMatrixUniform = gl.getUniformLocation (shaderProgram, “uPMatrix”);
    shaderProgram.mvMatrixUniform = gl.getUniformLocation (shaderProgram, “uMVMatrix”);
 }
```

Exemple complet (quasi)

```
var pyramidVertexPositionBuffer;  var pyramidVertexColorBuffer;

function initBuffers() {
    pyramidVertexPositionBuffer = gl.createBuffer();
    gl.bindBuffer(gl.ARRAY_BUFFER, pyramidVertexPositionBuffer);
    var vertices = [
        0.0, 0.5, 0.0,  -0.5, -0.5, 0.5,  0.5, -0.5, 0.5,  // front face
        0.0, 0.5, 0.0,  0.5, -0.5, 0.5,  0.5, -0.5, -0.5,  // right face
        0.0, 0.5, 0.0,  0.5, -0.5, -0.5,  -0.5, -0.5, -0.5,  // back face
        0.0, 0.5, 0.0,  -0.5, -0.5, -0.5,  -0.5, -0.5, 0.5  // left face
    ];
    gl.bufferData (gl.ARRAY_BUFFER, new Float32Array(vertices), gl.STATIC_DRAW);
    pyramidVertexPositionBuffer.itemSize = 3;
    pyramidVertexPositionBuffer.numItems = 12;
    . . .
```

Exemple complet (quasi)

```
    . . .
    pyramidVertexColorBuffer = gl.createBuffer();
    gl.bindBuffer (gl.ARRAY_BUFFER, pyramidVertexColorBuffer);
    var colors = [
        1.0, 0.0, 0.0, 1.0,  1.0, 0.0, 1.0, 1.0,  1.0, 0.0, 1.0, 1.0,
        1.0, 0.0, 0.0, 1.0,  1.0, 0.0, 1.0, 1.0,  1.0, 0.0, 1.0, 1.0,
       1.0, 0.0, 0.0, 1.0,  1.0, 0.0, 1.0, 1.0,  1.0, 0.0, 1.0, 1.0,
       1.0, 0.0, 0.0, 1.0,  1.0, 0.0, 1.0, 1.0,  1.0, 0.0, 1.0, 1.0,
    ];
    gl.bufferData (gl.ARRAY_BUFFER, new Float32Array(colors), gl.STATIC_DRAW);
    pyramidVertexColorBuffer.itemSize = 4;
    pyramidVertexColorBuffer.numItems = 12;
}
```

Exemple complet (quasi)

```
function drawScene() {

   gl.viewport (0, 0, gl.viewportWidth, gl.viewportHeight);
   gl.clear (gl.COLOR_BUFFER_BIT | gl.DEPTH_BUFFER_BIT);

   gl.bindBuffer (gl.ARRAY_BUFFER, pyramidVertexPositionBuffer);
   gl.vertexAttribPointer (shaderProgram.vertexPositionAttribute,
                           pyramidVertexPositionBuffer.itemSize, gl.FLOAT, false, 0, 0);

   gl.bindBuffer (gl.ARRAY_BUFFER, pyramidVertexColorBuffer);
   gl.VertexAttribPointer (shaderProgram.vertexColorAttribute,
                           pyramidVertexColorBuffer.itemsize, gl.FLOAT, false, 0, 0);

   mat4.perspective (45, gl.viewportWidth/gl.viewportHeigth, 0.1, 100.0, pMatrix);
   mat4.lookAt (/*OBS*/ 5.0, 0.0, 0.0, /*VRP*/ 0.0, 0.0, 0.0, /*VUP*/ 0.0, 1.0, 0.0, mvMatrix);

   setMatrixUniforms ();
   gl.drawArrays(gl.TRIANGLES, 0, pyramidVertexPositionBuffer.numItems);
}
```