API for the library implemented in `js/render/core/cg.js`

A vector is represented as a an array with 3 values:

```
[x,y,z]
```

A 4x4 matrix is represented as a flat array with 16 values:

```
[x0,x1,x2,x3, y0,y1,y2,y3, z0,z1,z2,z3, w0,w1,w2,w3]
```

```
// VECTOR METHODS

cg.cross(a,b)     // return cross product of two vectors
cg.dot(a,b)       // return dot product of two vectors
cg.mix(a,b,t)     // return a+t*(b-a), where a and b are vectors
cg.mix(a,b,t,u)   // return a*t + b*u, where a and b are vectors
cg.norm(v)        // return norm (geometric length) of a vector
cg.normalize(v)   // return v scaled to unit length
cg.scale(v, s)    // return v scaled by s

// NOISE METHOD

cg.noise(x,y,z)   // continuous pseudo-random noise from [Perlin1985]

// MATRIX METHODS

cg.mAimX(vec)        // return matrix that rotates X to vec
cg.mAimY(vec)        // return matrix that rotates Y to vec
cg.mAimZ(vec)        // return matrix that rotates Z to vec
cg.mIdentity()       // return identity matrix
cg.mInverse(m)       // return inverse of matrix m
cg.mMultiply(a,b)    // return product of two matrices a and b
cg.mRotateX(theta)   // return matrix that rotates about X
cg.mRotateY(theta)   // return matrix that rotates about Y
cg.mRotateZ(theta)   // return matrix that rotates about Z
cg.mPerspective(fl)  // return perspective transformation along Z
cg.mScale(s)         // return matrix that scales by [s,s,s]
cg.mScale(vec)       // return matrix that scales by vec
cg.mScale(x,y,z)     // return matrix that scales by [x,y,z]
cg.mTranslate(vec)   // return matrix that translates by vec
cg.mTranslate(x,y,z) // return matrix that translates by [x,y,z]
cg.mTranspose(m)     // return transpose of matrix m
```