Types: vec3, mat4
Functions: normalize, dot, max, pow
For more, see here:
http://www.shaderific.com/glsl-functions
https://www.khronos.org/opengl/wiki/Category:OpenGL_Shading_Language
https://www.khronos.org/opengl/wiki/OpenGL_Shading_Language

Steps:

1. Get normal, light, and direction to viewer; everything should be in "view/eye" space.
2. Transform the light position from world space to view space by multiplying view matrix by light position.
3. Calculate the light vector (position in view space to light position in view space, normalized).
4. Calculate the view-direction vector (from vertex position to camera position, in view space).
5. Use the normal, view, and light vectors as below (and Blinn's paper) to calculate the diffuse and specular values.
6. Scale those by the light and material diffuse and specular responses, adding to the ambient, to get the total intensity.

All vectors and positions need to be in "view space."

$$A = K_a * L_a \qquad H = \frac{L+V}{\|L+V\|}$$

$$D = K_d * L_d * N \cdot L$$

$$S = K_s * L_s * (N \cdot H)^s \Rightarrow$$ "shininess" determines specular roll-off

$$I = A + D + S$$

↗ each of these is an R,G,B vec3 in our case.