// create a value of 0. or 1. from the value of x with respect to the location of an edge:

float t = step(float edge, float x);

// create a value in the range 0. to 1. from the value of x with respect to the location of edge0 and edge1:

float t = smoothstep(float edge0, float edge1, float x);
Using that Mixing Parameter to Blend Two Quantities

// use the returned value from step() or smoothstep() to blend value0 to value1:

\[ T \text{ out} = \text{mix}( T \text{ value0}, T \text{ value1}, \text{float } t); \]

where T can be just about any type: float, vec2, vec3, vec4, …

\[ \text{out} = (1.-t) * \text{value}_0 + t * \text{value}_1 \]

One would expect \( 0 \leq t \leq 1 \).
but that doesn’t have to be true. After all, these are just numbers.

For a fun exercise with this, change the morphing slider to go beyond 0.-1.

As we will see later, there are really good uses for going beyond the range 0.-1.

---

“SmoothPulse” in a Fragment Shader

in float vX, vY;
in vec3 vColor;
in float vLightIntensity;
uniform float uA;
uniform float uP;
uniform float uTol;

const vec3 WHITE = vec3(1., 1., 1.);

void main()
{
    float f = fract(uA*vX);

    float t = smoothstep(0.5-uP-uTol, 0.5-uP+uTol, f) - smoothstep(0.5+uP-uTol, 0.5+uP+uTol, f);

    vec3 rgb = vLightIntensity * mix(WHITE, vColor, t);
    gl_FragColor = vec3(rgb, 1.);
}
**Fun With One**

**Moral:** There are many ways to turn \([ 0. - 1. ]\) into \([ 0. - 1. ]\).

Sidebar: Why Do These Two Curves Match So Closely?

The Taylor Series expansion of \( y = \sin\left(\frac{\pi}{2} x\right) \) around \( x=0.5 \) is:

\[
y = \frac{1}{2} \pi^2 96 x + \frac{\pi^4}{16} x^3 + \frac{\pi^6}{8} x^5 - \frac{\pi^8}{12} \]

\[
= 0.038 - 0.37x + 3.88x^2 - 2.58x^3
\]

which is pretty close to:

\[ y = 3x^2 - 2x^3 \]
Both go from 0. to 1.
Both have initial and final slopes of 0.
The quintic has initial and final curvatures of 0.

\[ y = 10x^3 - 15x^4 + 6x^5 \]

\[ y = 3x^2 - 2x^3 \]