

```

// By default, the environment contains four separate output buffers that can
each render different graphics.
// The outputs are accessed by the variables o0, o1, o2, and o3.

// to show all render buffers at once:
render()

// If no output is specified in out(), the graphics are rendered to buffer o0.
s0.initCam()
src(s0).out() // src(s0).out() is the same as saying src(s0).out(o0)

// to render an oscillator to output buffer o1:
osc(20, 0.1, 0.6).out(o1)

render(o1) // show only o1

render(o0) // show only o0

render() // show all four outputs o0, o1, o2, o3

// Blending modes (similar to photoshop) allow you to combine multiple sources
together
// multiply o0 with o1, and render the result to o2
src(o0).mult(o1).out(o2)

// 'multiply' means that the rgb values of each pixel of one image is
multiplied by
// the rgb values of the pixel in the same place in the other image

src(o0).add(o1).out(o2)
src(o0).diff(o1).out(o2)
src(o0).mask(o1).out(o2)

// The composite functions blend(), diff(), mult(), and add() perform
arithmetic operations to combine
// the input texture color with the base texture color, similar to photoshop
blend modes.
// modulate(texture, amount) uses the red and green channels of the input
texture to modify the x and y coordinates of the base texture. More about
modulation at: https://lumen-app.com/guide/modulation/

src(o0).modulate(o1).out(o2)

src(o0).modulateScale(o1).out(o2)

osc(20, 0.1, 0.6).color(1, 0, 0).out(o1)
osc(20, 0.1, 0.6).color(0, 1, 0).out(o1)
osc(20, 0.1, 0.6).out(o1)

src(o0).modulateRotate(o1).out(o2)
src(o0).modulatePixelate(o1).out(o2)

// composite operations can also be added onto each other within the same

```

```
// chain of functions
src(s0).mult(osc(20, 0.1, 0.8).rotate(0, 0.2)).out(o3)

// This can also be written as follows. (ctrl + enter to evaluate a block)
src(s0)
  .mult(osc(20, 0.1, 0.8).rotate(0, 0.4))
  .out(o3)
render(o3)

// more examples
osc(80, 0.1, 0.8)
  .color(1.0, 0.8, -1.0)
  .rotate(0.51, 0.1)
  .modulate(osc(40, -0.1), 0.1)
  .out(o3)

osc(80, 0.1, 0.8)
  .thresh()
  .rotate(1.71)
  .modulateRotate(osc(20, -0.1), 1)
  .out(o3)
```