

```
// To start, follow the instructions in README.md

// shift + enter runs a line of code
// ctrl + enter runs a block of code

osc(20, 0.1, 0.8).out()

// render an oscillator with parameters frequency, sync, and rgb offset.
// change the numbers and run ctrl + enter again to update the visuals
osc(22, 0.1, 0.8).out()

// rotate the oscillator 0.8 radians:
osc(20, 0.1, 0.8).rotate(0.8).out()

// pixelate the output of the above function:
osc(20, 0.1, 0.8).rotate(0.8).pixelate(20, 30).out()

// show webcam output:
s0.initCam(0) // initialize a webcam in source buffer s0

src(s0).out() // render source buffer s0

// If you have more than one camera connected, you can select the camera using
an index:
s0.initCam(1) // initialize a webcam in source buffer s0

// webcam kaleidoscope:
s0.initCam() // initialize a webcam in source buffer s0

src(s0).kaleid(4).out() // render the webcam to a kaleidoscope

// change hue
s0.initCam() // initialize a webcam in source buffer s0

src(s0).rotate(0, 0.1).hue().saturate(5).out() // render the webcam to a rotate
and also hue, and then saturate.

// You can also use the screen or any open window as an input to hydra. When
using the desktop as an input, this
// can create feedback within the visuals. See the feedback.js file for more
examples.

s0.initScreen()

src(s0).out()

src(s0).rotate(0.1).out()

// Changing the number in 'initScreen' lets you choose which window becomes the
input

s0.initScreen()
```

```
src(s0).out()  
// Go to black  
solid().out()  
// is the same as  
hush()
```