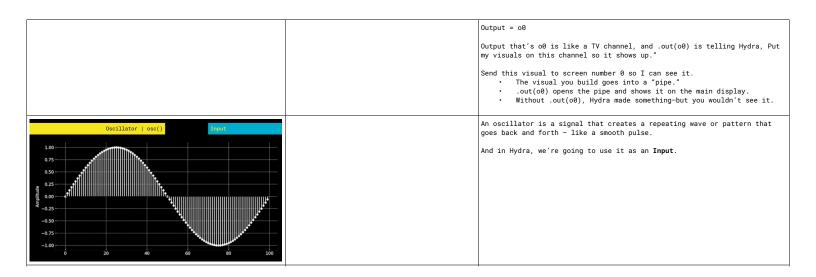
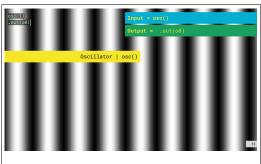
Slide	Code	Notes
Made by Olivia audit 2. Beas inforces pumpada off. Trages edition 10.7. 69  hydra the common visitor owners  light as less note atth violes yeth and coding environment that most directly in the browser.  It is the and organism accurate and made for beginners and experts alles  Toper actived.  1. Clinic entit survivol.  2. Type Carlo With Violes  3. Type Carlo With Violes  (In the common violes of the coding of the coding owners and organism and experts alles  1. Clinic entit survivol.  2. Type Carlo With Violes  (In the coding owners administration).		Hydra Website https://hydra.ojack.xyz/ Pro Tip Workshop display: Two-screen split; one screen with Hydra, the other with workshop slides.







osc () .out(o0) Hydra expects an input and an output argument.

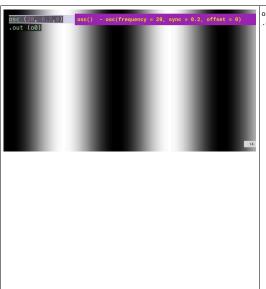
osc () // Input .out(o0) // Output

Hydra is written in **JavaScript**, a programming language and a core technology of Websites, alongside HTML and CSS.

It enables  $\ensuremath{\mathsf{dynamic}}$  and  $\ensuremath{\mathsf{interactive}}$  content on websites and web applications.

 $\ensuremath{\mbox{Syntax}}$  is the  $\ensuremath{\mbox{grammar}}$  of  $\ensuremath{\mbox{code}}$  . It's how we tell the computer what to do step by step.

In Hydra, the code is read from left ot right, top to bottom.



osc (20, 0.2,0) out(00)

We have 3 arguments in osc()

osc() - osc(frequency, sync, offset)

Oscillator ( frequency = 20, sync = 0.2, offset = 0)

In coding, an argument is information you give to a **function** so it knows how to behave.

- osc() is a function it creates something. The dot . chains actions together. Each pair of parentheses () passes arguments (values) that control

Think of a function like a machine, and arguments are the settings or ingredients you give it. Each function has its own argument.

A parameter is a blank spot a function expects, and an argument is the actual value you fill that spot with when you use the function.

The first argument 20 is the frequency of the lines,

osc (20, 0.2,0)

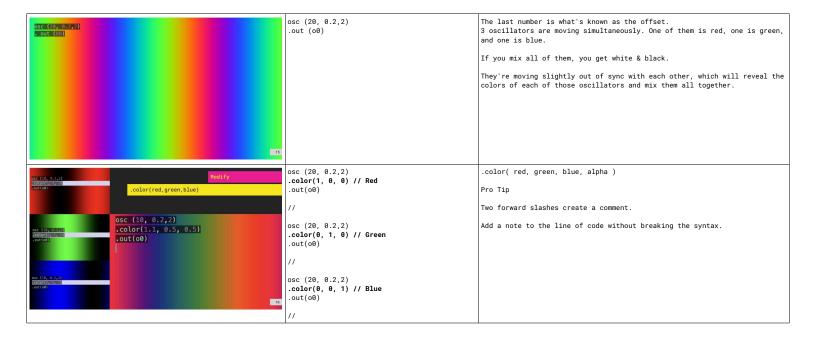
.out (00)

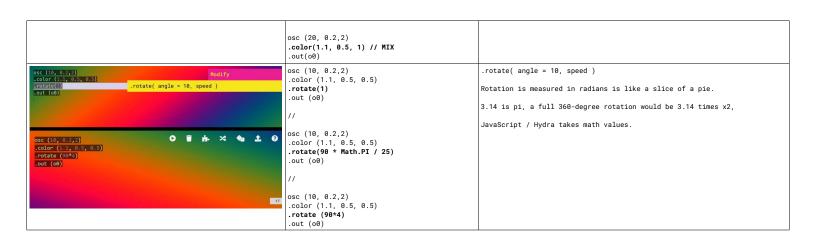
The second number is the synchronization, known as speed.

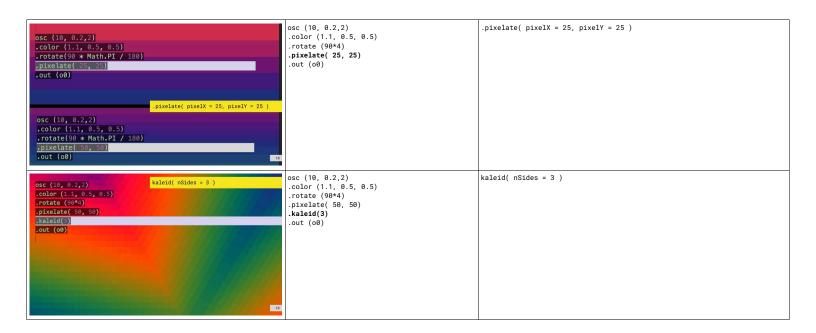
Put 0.2; they're moving quite slowly.

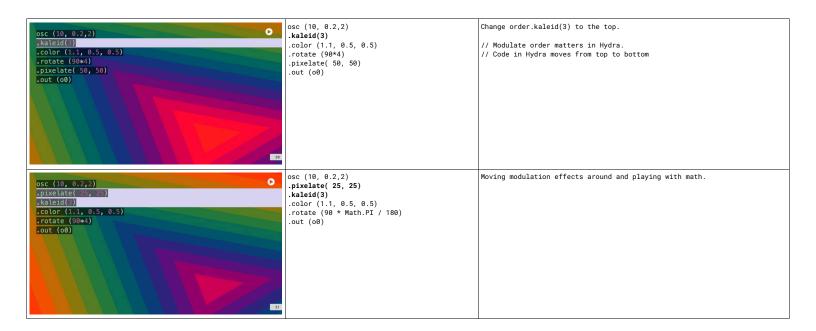
Now put number 0.8, you'll see it begins to move a lot faster.

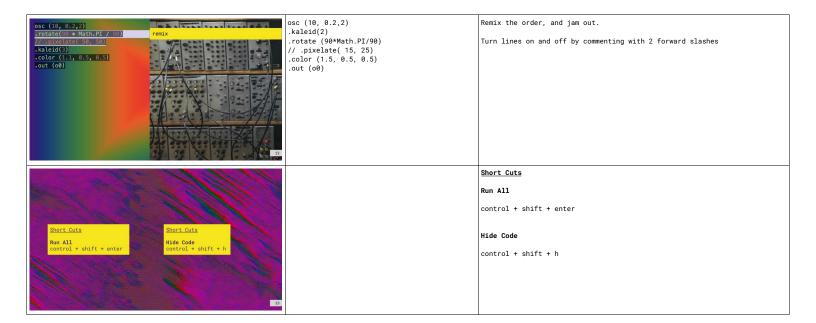
Warning too fast and it will flash!

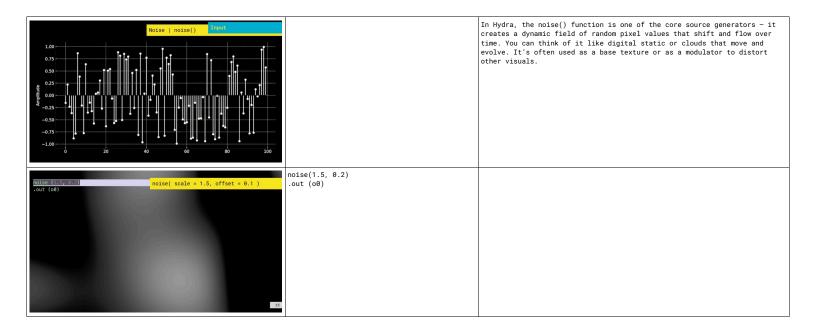


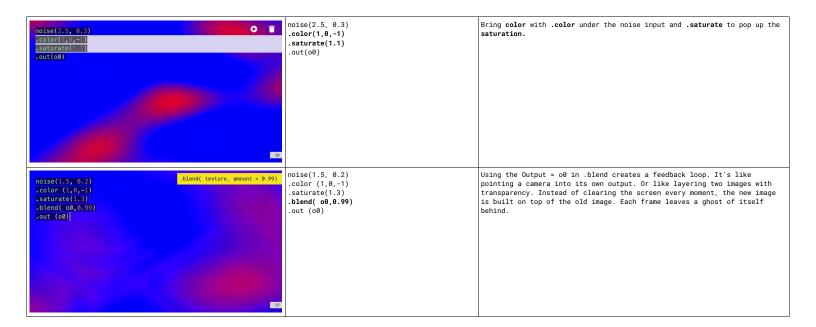


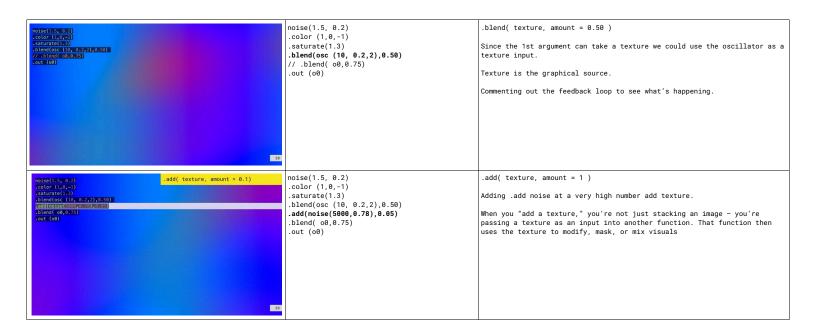














noise(1.5, 0.2)
.color(1,0,-1)
.saturate(1.3)
.blend(osc(10, 0.2, 2), 0.50)
.modulate(osc(15,0.3), 0.4)
// <- more modulation (0.1-1 range)
.add(noise(5000,0.78), 0.05)
.blend(o0, 0.75)
.out(o0)

For more of a water feel and less repetition we can use .modulate

Modulate functions use the colors from one source to affect the geometry of the second source. This creates a sort of warping or distorting effect.

.  $\mbox{modulate()}$  does not change color or luminosity but distorts one visual source using another visual source.

An analogy in the real world would be looking through a texture glass window or water

You can add a second parameter to the modulate() function to control the amount of warping: modulate(o1, 0.9).

In this case, the red and green channels of the oscillator are being converted to  $\boldsymbol{x}$  and  $\boldsymbol{y}$  displacement of the camera image.

usually by shifting or warping its pixels.
Think of it like bending one image with another image's energy — the
second texture becomes a map that tells Hydra how to distort the first
one.

