

# norns studies

Comprehensive lessons on individual elements of norns scripting. Through these resources, you'll learn how to translate your musical ideas into scripts and how to extend those scripts to speak with all kinds of friends.

## ▼ sections

- [core concepts](#)
- [contributed knowledge](#)
- [learning SuperCollider](#)
- [learning Lua](#)

## core concepts

Tutorials written by monome.

*Note: all of these resources assume you're on the latest version of the norns software. if you have any questions about updating, please see [this section of the docs](#). note that if you are running software version `220129` or earlier, you will need to complete a fresh installation – [more details here](#).*

**For the best learning experience, we recommend one lesson per week.** A slower pace makes it easier to experiment and uncover the simplicity + musicality of the underlying code.

### norns: introductory studies

- [first light](#) // learning to see. edit a script.
- [many tomorrows](#) // variables, simple maths, keys + encoders
- [patterning](#) // screen drawing, for/while loops, tables
- [spacetime](#) // functions, parameters, time
- [physical](#) // incorporating grids, midi, clock syncing
- [streams](#) // system polls, osc, file storage

### norns: extended API techniques

- [softcut](#) // a multi-voice sample playback and recording system built into norns
- [clocks](#) // create timed and clock-synced function calls: loops, repetition, and delays
- [grid recipes](#) // short snippets of code for canonical grid interactions

### norns: SuperCollider

- [rude mechanicals](#) // introduction to building norns engines with SuperCollider and Lua
- [skilled labor](#) // extended study of norns engine development: polyphony and realtime parameter changes
- [transit authority](#) // deep-dive into audio busses, FX, and polls (to communicate from SuperCollider to Lua)

## contributed knowledge

Resources developed by other learners, script authors, and designers from our community.

- [norns study group](#) // a community run discord server for asynchronous and video study sessions
- [athenaeum](#) // a repository of study, spike, and sample scripts from [Tyler Etters](#)
- [norns: tutorial](#) // chunked examples that reveal the idiosyncrasies of the core norns concepts from [Devine Lu Linvega](#)
- [norns development](#) // general development notes from Jordan Besly's [awesome monome norns](#) repo
- [lua libs index](#) // overview of norns library extensions (ibid.)

## learning SuperCollider

SuperCollider is a free and open-source platform for making sound, which powers the synthesis layer of norns. Many norns scripts are a combination of SuperCollider (where a synthesis engine is defined) and Lua (where the hardware + UI interactions are defined). SuperCollider can be run on most any computer – you do not need a norns to dig into these resources!

For those who want to explore creating new synthesis engines for norns, we've developed [rude mechanicals](#), [skilled labor](#), and [transit authority](#). These studies walk through building norns engines from scratch with SuperCollider and Lua, as well as showcasing extended techniques for audio routing and bi-directional communication between SuperCollider and Lua.

We also highly recommend:

- Zack Scholl's video series, produced in partnership between monome and Music Hackspace: [Tone to Drone](#) and [Ample Samples](#).
- Eli Fieldsteel's [YouTube series](#)
- Nathan Ho's [collected SuperCollider tips](#)

## SuperCollider language fundamentals

---

- [written tutorial](#)
- [first steps to colliding atoms](#)
- [how to get your code to play](#)
- [what is a function?](#)
- [if , while , for , etc](#)
- [supercollider server explained](#)
- [j concepts in supercollider](#)
- [how to use an interpreter with supercollider](#)
- [node](#)
- [bus](#)

## learning Lua

---

While you don't need to master the entire Lua language to make the most of norns, you eventually may be interested in checking out more traditional programming texts.

- [lua cheatsheet](#)
- [programming in lua \(first edition\)](#)
- [lua 5.3 reference manual](#)
- [lua-users tutorials](#)
- [lua in 15 mins](#)

Ask questions and share what you're making at [llllllll.co](#)

Edits to these pages welcome, see [monome/docs](#)

---

[help](#)