

Presentation

What is Strudel?

Showcase

Blog

Workshop

First Sounds

First Notes

First Effects

Pattern Effects

Recap

Workshop in German

Making Sound

Samples

Synths

Audio Effects

MIDI & OSC

More

Recipes

Mini-Notation

Visual Feedback

Offline

Patterns

Mondo Notation

Music metadata

CSound

Hydra

Input Devices

Device Motion

Pattern Functions

Introduction

Creating Patterns

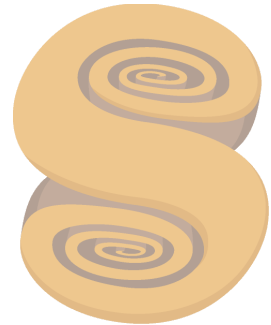
Time Modifiers

Control Parameters

Signals

Welcome

Welcome to the Strudel documentation pages! You've come to the right place if you want to learn how to make music with code.



What is Strudel?

With Strudel, you can expressively write dynamic music pieces.

It is an official port of the Tidal Cycles pattern language to JavaScript.

You don't need to know JavaScript or Tidal Cycles to make music with Strudel. This interactive tutorial will guide you through the basics of Strudel.

The best place to actually make music with Strudel is the Strudel REPL

What can you do with Strudel?

- live code music: make music with code in real time
- algorithmic composition: compose music using tidal's unique approach to pattern manipulation
- teaching: focussing on a low barrier of entry, Strudel is a good fit for teaching music and code at the same time.
- integrate into your existing music setup: either via MIDI or OSC, you can use Strudel as a really flexible sequencer

Examples

Here are some examples of how strudel can sound:

- Accumulation
- Presentation
 - What is Strudel?
 - Showcase
 - Blog
- Workshop
 - First Sounds
 - First Notes
 - First Effects
 - Pattern Effects
 - Recap
 - Workshop in German
- Making Sound
 - Samples
 - Synths
 - Audio Effects
 - MIDI & OSC
- More
 - Recipes
 - Mini-Notation
 - Visual Feedback
 - Offline
 - Patterns
 - Mondo Notation
 - Music metadata
 - CSound
 - Hydra
 - Input Devices
 - Device Motion

```

2 // @version 1.0
3 samples('github:eddyflux/crate')
4 setcps(.75)
5 let chords = chord("<Bbm9 Fm9>/4").dict('i
6 stack(
7   stack( // DRUMS
8     s("bd").struct("<[x*<1 2> [~@3 x]] x>"
9     s("~ [rim, sd:<2 3>]").room("<0 .2>"),
10    n("[0 <1 3>]*<2!3 4>").s("hh"),
11    s("rd:<1!3 2>*2").mask("<0 0 1 1>/16")
12  ).bank('crate')
13  .mask("<[0 1] 1 1 1>/16".early(.5))
14  , // CHORDS
15  chords.offset(-1).voicing().s("gm_epiano
16  .phaser(4).room(.5)
17  , // MELODY
18  n("<0!3 1*2>").set(chords).mode("root:g2
19  .voicing().s("gm_acoustic_bass"),
20  chords.n("[0 <4 3 <2 5>>*2](<3 5>,8)")
21  .anchor("D5").voicing()
22  .segment(4).clip(rand.range(.4,.8))
23  .room(.75).shape(.3).delay(.25)
24  .fm(sine.range(3,8).slow(8))
25  .lpf(sine.range(500,1000).slow(8)).lpq(5
26  .rarely(ply("2")).chunk(4, fast(2))
27  .gain(perlin.range(.6, .9))
28  .mask("<0 1 1 0>/16")
29 )
30 .late("[0 .01]*4").late("[0 .01]*2").size(

```

These examples cannot fully encompass the variety of things you can do, so check out the showcase for some videos of how people use Strudel.

Getting Started

- Pattern Functions
 - Introduction
 - Creating Patterns
 - Time Modifiers
 - Control Parameters
 - Signals

Accumulation

Presentation

What is Strudel?

Showcase

Blog

Workshop

First Sounds

First Notes

First Effects

Pattern Effects

Recap

Workshop in German

Making Sound

Samples

Synths

Audio Effects

MIDI & OSC

More

Recipes

Mini-Notation

Visual Feedback

Offline

Patterns

Mondo Notation

Music metadata

CSound

Hydra

Input Devices

Device Motion

Pattern Functions

Introduction

Creating Patterns

Time Modifiers

Control Parameters

Signals