

Open <https://strudel.cc/> and replace the code in the code area with:

```
$: s("bd").beat("0,4,8", 16)
```

```
$: s("sd").beat("0,6,12", 16)
```

Press the Play button in the top right. Change something. Press the play button again!

Beat haps

bd and **sd** are drumkit sounds. You can also try: **hh oh rd cr lt mt ht rim**

beat specifies how many timeslots a cycle should be split into, and where beats should land.

Strudel is a live coding platform to write dynamic music pieces in the browser. It's free and open-source and made for beginners and experts alike. I encourage you to explore!

Open <https://strudel.cc/> and replace the code in the code area with:

```
$: s("bd").struct("x - [- x] -")
```

```
$: s("sd").struct("- x - x")
```

Press the Play button in the top right. Change some code. Press the play button again!

Struct haps

x makes a hap

- is a rest.

[and] collect the contents into one timeslot

Remember to close any brackets that you've opened!

Strudel is a live coding platform to write dynamic music pieces in the browser. It's free and open-source and made for beginners and experts alike. I encourage you to explore!

Open <https://strudel.cc/> and replace the code in the code area with:

```
$: s("bd")
```

```
$: n(run(8)).s("bongo")
```

Press the Play button in the top right. Change some code. Press the play button again!

Sample runs

The first line, with bd is just there to show where the cycle loops

The second line makes **8** different samples within the bongo sound play in succession. They all have different tones, and make pleasing run together.

Instead of **bongo**, try: **cowbell**, **cabasa**, **conga**, **darbuka**, **guiro**, **slapstick**, **woodblock**

Strudel is a live coding platform to write dynamic music pieces in the browser. It's free and open-source and made for beginners and experts alike. I encourage you to explore!

Open <https://strudel.cc/> and replace the code in the code area with:

```
$: s("saw*4").lpf("200 500 1000 2000")
```

Press the Play button in the top right. Change some code. Press the play button again!

Filters

Here we're running a low-pass filter over the saw sound. It removes the top end of the frequency spectrum to make the sound less buzzy. You can try removing the filter function to see how that sounds.

We also have **.bpf(...)** which is a band-pass filter, selecting a chunk of the frequency spectrum, and **.hpf(...)** which cuts out the bottom end.

Strudel is a live coding platform to write dynamic music pieces in the browser. It's free and open-source and made for beginners and experts alike. I encourage you to explore!

Open <https://strudel.cc/> and replace the code in the code area with:

```
$: s("piano*8").mask("1 0 0 1 0 1 0 0")
```

Press the Play button in the top right. Change some code. Press the play button again!

Masking

A mask allows you to specify an open or shut gate for the sound.

Here we have a piano triggering **8** times in a cycle, but the mask only lets **3** of them through.

If you remove the mask, all the triggers will play.

Strudel is a live coding platform to write dynamic music pieces in the browser. It's free and open-source and made for beginners and experts alike. I encourage you to explore!

Open <https://strudel.cc/> and replace the code in the code area with:

```
$: chord("CM Am Fm7 GM").voicing()
```

Press the Play button in the top right. Change some code. Press the play button again!

Chords

Strudel has notation for chords.

You'll notice that if you miss out **voicing**, you'll get a flat tone instead. Voicing is needed to turn the chord symbols into notes. Just one of those things to remember!

Strudel is a live coding platform to write dynamic music pieces in the browser. It's free and open-source and made for beginners and experts alike. I encourage you to explore!

Open <https://strudel.cc/> and replace the code in the code area with:

```
$: run(8).s("piano").scale("<C:major F:pentatonic>")
```

Press the Play button in the top right. Change some code. Press the play button again!

Scales

Strudel has a selection of build-in scale definitions. The **run(8)** declaration sends a sequence of **8** increasing index numbers into the specified scales.

Scales you can try: C:major C:minor C:major:blues C:mixolydian C:ritusen

This is one of many ways to use a scale, firing through random indices between 0 and 9:

```
$: s("piano*4").n(irand(10)).scale("C:major")
```

Strudel is a live coding platform to write dynamic music pieces in the browser. It's free and open-source and made for beginners and experts alike. I encourage you to explore!

Open <https://strudel.cc/> and replace the code in the code area with:

```
$: note("c3,e3,g3").s("piano")
```

Press the Play button in the top right. Change some code. Press the play button again!

Simltaneous Sounds

A comma in between items specifies that they should be played together.

You'll notice that if you remove the two commas, the notes will play in sequence rather than making a chord.

Strudel is a live coding platform to write dynamic music pieces in the browser. It's free and open-source and made for beginners and experts alike. I encourage you to explore!

Open <https://strudel.cc/> and replace the code in the code area with:

```
$: s("piano").note("c3,e3,g3").arp("0 2 1 2")
```

Press the Play button in the top right. Change some code. Press the play button again!

Arpeggio

arp picks out the **n**th value from the collection of items sent in to it.

You can see how the arp pattern lights up the note pattern when you press play.

Without the **arp**, the comma between notes groups them together, so they would otherwise all play simultaneously. You can try that out too.

Strudel is a live coding platform to write dynamic music pieces in the browser. It's free and open-source and made for beginners and experts alike. I encourage you to explore!

Open <https://strudel.cc/> and replace the code in the code area with:

```
$: s("piano*8").pan(sine.range(.1,.9).slow(4))
```

Press the Play button in the top right. Change some code. Press the play button again!

Pan

Panning (setting where a stereo sound should be between left and right ears) add some nice depth.

Here, we're playing a piano sound repeatedly, and panning it almost fully between left (.1) and right (.9) over 4 cycles.

Strudel is a live coding platform to write dynamic music pieces in the browser. It's free and open-source and made for beginners and experts alike. I encourage you to explore!

Open <https://strudel.cc/> and replace the code in the code area with:

```
$: note("<c2 e4>").s("piano").echo(3,1/4,.5)
```

Press the Play button in the top right. Change some code. Press the play button again!

Echo

The echo function takes three arguments:

The first is how many echoes you'd like.

The second is the time delay for each echo.

The third is how you'd like the echo to decay.

Be careful – if you put a value larger than **1** for the third argument, the echoes will increase in volume!

Strudel is a live coding platform to write dynamic music pieces in the browser. It's free and open-source and made for beginners and experts alike. I encourage you to explore!

Open <https://strudel.cc/> and replace the code in the code area with:

```
$: note("c1 e2 f2 g2 - a2 - c3").s("handbells").jux(rev)
```

Press the Play button in the top right. Change some code. Press the play button again!

Jux

.jux(...) is fun. It applies a function to a pattern, but only in the right ear.

So the example above is playing the pattern forwards on one side, and in reverse in the other. You could also try a slight echo effect by specifying that you wish the right side to be played a little later, i.e.

```
$: note("c1 e2 f2 g2 - a2 - c3").s("handbells").jux(late(1/8))
```

Strudel is a live coding platform to write dynamic music pieces in the browser. It's free and open-source and made for beginners and experts alike. I encourage you to explore!