



# Free OP-XY Presets, Made from Python

APRIL 2026

3 min read • 777 words

---

I love my OP-XY. It's one of the most elegant pieces of hardware I've ever used — Teenage Engineering at their best. Opinionated, beautiful, immediately playable. You turn it on and you're making music in seconds.

But here's the thing: the built-in synth engines are great — reviewers weren't kidding when they said it's hard to make an OP-XY patch sound bad — but the device really comes alive when you load your own samples. The multisampler engine lets you spread sounds across the keyboard with automatic crossfading between zones. The drum sampler gives you individual samples per key, which is a huge step up from the OP-Z's sliced-file approach. Suddenly this little sequencer becomes whatever instrument you want it to be.

So naturally, I went looking for sample packs.

## The Preset Landscape

The OP-XY preset ecosystem is still young. There are a handful of creators doing good work — [Rephazer](#) has several packs at around €19 each, Metropolis Border and SoundGhost have offerings in the \$20-25 range, and there are bundles that run higher. [NearTao](#) put out a generous free pack with 325 presets. The community has also built some great tools — [OP-PatchStudio](#) is a free web app for building your own presets from scratch.

But overall, the selection is thin for a device this capable. And I happened to have a synthesis engine sitting right there.

## So I Made Some Presets

[pytheory-opxy](#) — 69 multisampled instruments and 12 drum kits for the OP-XY (and OP-1), generated entirely from code. No samples were recorded. No microphones were involved. Every sound was synthesized from scratch using [PyTheory](#), my Python music theory and synthesis library.

Piano, strings, brass, woodwinds, guitars, mallets, synths, world instruments — sitar, koto, didgeridoo, harmonium, tabla, mridangam, djembe — the works. Each instrument is multisampled at six points across the keyboard (C2, C3, C4, A4, C5, C6) so the OP-XY can pitch them intelligently without that chipmunk-at-C6 problem.

Sustained instruments — strings, winds, organs, pads — loop seamlessly while you hold a key and release naturally when you let go. Plucked and percussive instruments just play through. It all works the way you'd expect, which is the hardest thing to get right.

One thing to keep in mind: the OP-XY has a 42MB active sample memory limit (a number Teenage Engineering doesn't publish, naturally — the community had to figure that out). At 44.1kHz that's roughly 4 minutes of audio across your whole project. These presets are lean by design, but you'll still want to be thoughtful about how many you load at once.

## Python All the Way Down

Here's where it gets fun. The entire generation pipeline is two Python scripts. That's it. `generate.py` handles the 69 instruments. `generate_drums.py` handles the 12 drum kits. Run them and you get a folder full of OP-XY-ready `.preset` directories.

Under the hood, PyTheory's synthesis engine is doing all the heavy lifting. I've written about this before — [how NumPy becomes a synth engine](#) when you stop thinking of arrays as data and start thinking of them as audio. Karplus-Strong for

plucked strings. Physical modeling for drums. Additive synthesis for organs. Subtractive for synths. All rendered to WAV with NumPy and SciPy, no audio libraries required.

The loop point detection alone was a journey. The script analyzes each sample's RMS energy to find matching sustain regions, snaps to zero crossings to avoid clicks, and applies crossfade overlap for seamless looping. It's the kind of problem that sounds simple until you're debugging why your violin sounds like a broken CD player at 2am.

If you want the full story of how PyTheory went from a dormant music theory library to a full synthesis engine, I wrote about [breaking through five years of creative block with AI](#) and [building a mini DAW in the Python REPL](#). The short version: Claude helped me finally finish what I started in 2019, and now it synthesizes tabla drums from first principles. Life is strange and wonderful.

## Get Them

Grab the presets from [GitHub](#). Download the ZIP, drag the folders onto your OP-XY using [Field Kit](#), and you're done. OP-1 users get single-sample WAVs that work with the sampler engine.

If you're a developer and want to regenerate everything yourself or tweak the presets:

```
pip install pytheory numpy scipy
python generate.py                # all 69 instruments
python generate.py piano sitar    # just the ones you want
python generate_drums.py          # all 12 drum kits
```

It's all open source. Fork it, modify it, make it weird. I just love open source — always have. If I can generate these sounds from code, there's no reason not to share them.

Go make something with your OP-XY.