acat

October 2026

2 min read • 494 words

`cat` prints files. `acat` plays them.

That's the whole idea. I built it today, and I've been smiling about it all afternoon.

Why

I had a recording of an X Space I'd hosted, sitting in my home directory as an mp4. I wanted to listen to it. Not open an app, not drag it into a window, not wait for something with a library and an account to wake up. I just wanted to hear it, from the place I already was: my terminal.

So I typed what felt natural:

```
acat song.mp3
```

And then I made it work.

> The first thing it ever played was that recorded Space. It's a good test. It's long, it's a video file, and I actually wanted to hear it.

What It Does

It behaves the way you'd expect something named after `cat` to behave. Give it a file, and it plays. Give it several, and it plays them one after another, the same way `cat` concatenates. Pipe something into it, and stdin works too.

Videos play their audio track: mp4, mov, mkv, webm, avi, and friends. That was the case I cared about most, and it's the one most players make awkward.

While it plays, you get a progress bar with elapsed and total time and the volume, and a single line of controls underneath:

```
[←/→] ∓5s  [a/d] ∓30s  [space] pause  [↑/↓] volume  [n] next  [p] prev  [q] quit
```

If you don't want any of that, `-q` gives you plain playback.

This is the for humans idea again: sensible defaults, obvious behavior, nothing to configure. The command you'd guess is the command that works.

How It Works

The player is modeled on the one I wrote for Interpretations, my album written in Python. It uses sounddevice and numpy for playback, with ffmpeg streaming the decoded audio in.

The part I'm happiest with is seeking. Rather than loading a whole file into memory, a seek just restarts ffmpeg at the new position. Memory stays flat, even for a 47-minute recording.

> It's one of those choices that sounds lazy and turns out to be the right one. The simplest thing that could possibly work, working.

When the fancy path isn't available, acat gets out of the way. If it's piped, running quiet, or there's no ffmpeg, it falls back to whatever your system already has: afplay, ffplay, mpv, paplay, or aplay. And if you need ffmpeg, it tells you how to install it, kindly, for the platform you're actually on.

Getting It

The code is at github.com/kennethreitz/acat, ISC licensed.

The name `acat` on PyPI was already taken by an unrelated chemistry toolkit, so the package is `acat-player`. The command is still `acat`:

```
uv tool install acat-player
```

It's at v0.1.1.

That's It

It's a small tool. It does one thing. I like small tools that do one thing, and I like it when the thing they do is exactly what you'd guess from the name.

Audio and video in your terminal, for humans.