SoX Audio swiss knife

START HERE

`soxi <file>`
Show file info

`play <file>`
Play audio

`rec <file>`
Record from mic

`sox <in> <out>`
Convert format

`sox <in> <out> trim 10 30`
Cut a piece

`sox <in> <out> rate 16k`
Resample

`sox <in> <out> norm`
Normalize peak

`sox <in> -n spectrogram`
Draw spectrogram

Input file(s) read, decoded → *combine* → **Effects chain** trim, gain, rate... → *encode* → **Output file** type from extension

CONCEPTS

Command shape: sox [global] [format opts] in... [format opts] out [effect args...]. Effects run left to right on the way to the output file.

Format options (-r -c -b -t) apply to the file name that follows them; the output extension picks the type. Effects go after the output name.

The sox on many distros (14.4.2) may lack MP3 support; install the libsox mp3 plugin if mp3 files fail. The sox_ng fork documented here is otherwise compatible.

INSPECT AND PLAY

`soxi <file>`
Show format, rate, channels, duration

`soxi -D <file>`
Duration in seconds

`soxi -r <file>`
Sample rate only

`soxi -c <file>`
Channel count only

`soxi -T -d <files>`
Total duration of several files

`play <file>`
Play through the default device

`play <file> trim 30`
Play starting at 30 seconds

`play -n synth 3 sine 440`
Play a 3 s test tone

`sox <in> -n stat`
Level statistics, no output

`sox <in> -n stats`
Peak, RMS and crest summary

RECORD

`rec <file>`
Record from default input until Ctrl+C

`rec -c 1 -r 16k <file>`
Record mono 16 kHz

`rec <file> trim 0 60`
Record for 60 seconds

`rec <file> silence 1 0.1 1% 1 2.0 1%`
Auto start and stop on silence

`sox -d <file>`
Same as rec, with sox directly

`sox <in> -d`
Same as play, with sox directly

CONVERT AND FORMAT

`sox <in.wav> <out.flac>`
Convert, type from extension

`sox <in> -r 44100 <out>`
Set output sample rate

`sox <in> -c 1 <out>`
Set output channel count

`sox <in> -b 16 <out>`
Set output bit depth

`sox <in> -e signed-integer <out>`
Set encoding

`sox -t raw -r 48k -e signed -b 16 -c 2 <in.raw> <out.wav>`
Wrap raw PCM in a header

`sox <in> -t wav -`
Write to stdout as WAV

`sox -t wav - <out>`
Read from stdin

`sox <in> <out> dither`
Add dither on bit-depth reduction

`sox -V3 <in> <out>`
Show processing details

TRIM, JOIN AND MIX

`sox <in> <out> trim <start> <len>`
Cut a piece by start and length

`sox <in> <out> trim <start> =<end>`
Cut up to an absolute position

`sox <in> <out> trim 0 -5`
Drop the last 5 seconds

`sox <a> <b> <out>`
Concatenate files
Inputs need the same rate and channels, or sox converts to the first file's.

`sox -m <a> <b> <out>`
Mix files on top of each other

`sox -M <l.wav> <r.wav> <out>`
Merge mono files into channels

`sox <in> <out> remix 1`
Keep only the left channel

`sox <in> <out> remix 1,2`
Mix both channels to mono

`sox <in> <out> pad 2 0`
Add 2 s silence at the start

`sox <in> <out> repeat 2`
Play the audio 3 times

`sox <in> <out> reverse`
Reverse the audio

TRIM, JOIN AND MIX (cont.)

`sox <a> <b> <out> splice -q <pos>`
Cross-fade join at position

`sox <in> <out> trim 0 10 : newfile : restart`
Split into pieces of 10 s

`sox <in> <out> channels 2`
Convert to stereo

VOLUME AND FADES

`sox <in> <out> norm`
Normalize peak to 0 dBFS

`sox <in> <out> norm -3`
Normalize peak to -3 dBFS

`sox <in> <out> gain <dB>`
Change level by dB

`sox <in> <out> vol <factor>`
Multiply amplitude by a factor

`sox -v <factor> <in> <out>`
Scale input before combining

`sox <in> <out> gain -l <dB>`
Boost with a simple limiter

`sox <in> <out> fade <in-s> 0 <out-s>`
Fade in and out
Fade-out needs the stop position; 0 means the end of the file. The input length must be known.

`sox <in> <out> fade t 3`
Linear 3 s fade-in only

`sox <in> <out> compand 0.3,1 6:-70,-60,-20 -5 -90 0.2`
Compress dynamic range

`sox --norm <a> <b> <out>`
Normalize whatever the mix produces

RATE, SPEED AND PITCH

`sox <in> <out> rate <hz>`
Resample to a rate

`sox <in> <out> rate -v 48k`
Resample, very high quality

`sox <in> <out> speed <factor>`
Change speed and pitch together

`sox <in> <out> tempo <factor>`
Change speed, keep the pitch

`sox <in> <out> pitch <cents>`
Shift pitch, keep the speed
100 cents are one semitone.

FILTERS AND CLEAN-UP

`sox <in> <out> highpass <hz>`
Cut low rumble

`sox <in> <out> lowpass <hz>`
Cut high hiss

`sox <in> <out> bass <dB>`
Boost or cut the lows

`sox <in> <out> treble <dB>`
Boost or cut the highs

`sox <in> <out> equalizer <hz> <width>q <dB>`
Adjust one frequency band

`sox <in> <out> silence 1 0.1 1% -1 0.5 1%`
Trim silence at start and end

`sox <in> <out> silence -l 1 0.1 1% -1 1.0 1%`
Cap long silences, keep gaps

`sox <noise.wav> -n noiseprof <prof>`
Learn a noise profile

`sox <in> <out> noisered <prof> 0.21`
Reduce noise with the profile

ANALYSE AND GENERATE

`sox <in> -n spectrogram`
Write spectrogram.png

`sox <in> -n spectrogram -o <out.png>`
Spectrogram to a chosen PNG

`sox -n <out> synth 3 sine 440`
Generate a 3 s sine tone

`sox -n <out> trim 0 5`
Generate 5 s of silence

RECIPES

Podcast clean-up and normalize

```
sox raw.wav out.wav highpass 80 \
  silence 1 0.1 1% -1 0.5 1% \
  gain -n -3
```

Speech data for ASR (16 kHz mono)

```
sox in.mp3 -r 16k -c 1 -b 16 out.wav
```

Mix voice over music

```
# music at 30% volume, ends with the voice
sox -m voice.wav -v 0.3 music.wav out.wav trim 0 $(soxi -D voice.wav)
```

Spectrogram of one minute

```
sox in.flac -n trim 0 60 spectrogram -x 1200 -o spec.png
```