2021 Web Audio API Survey

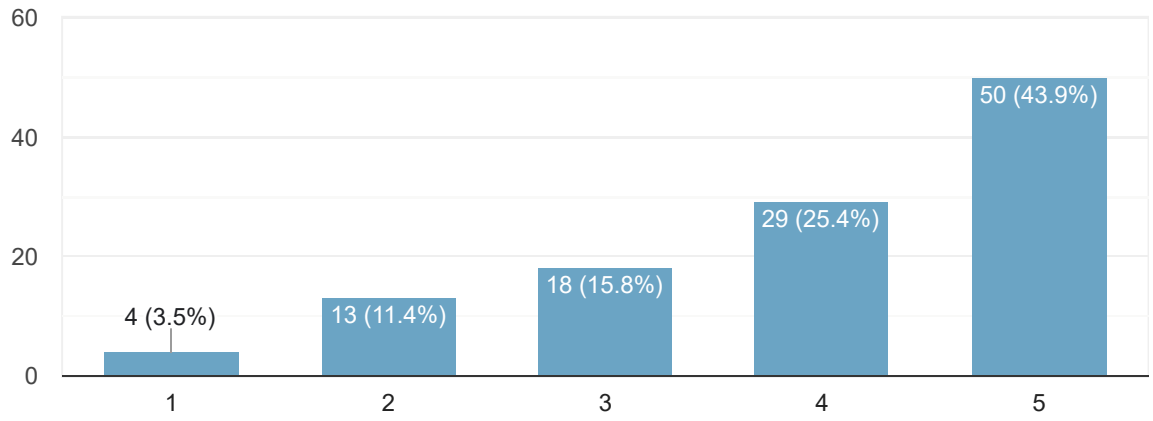
114 responses

What do you need most?

Select an audio device for input or output from an AudioContext.

Copy

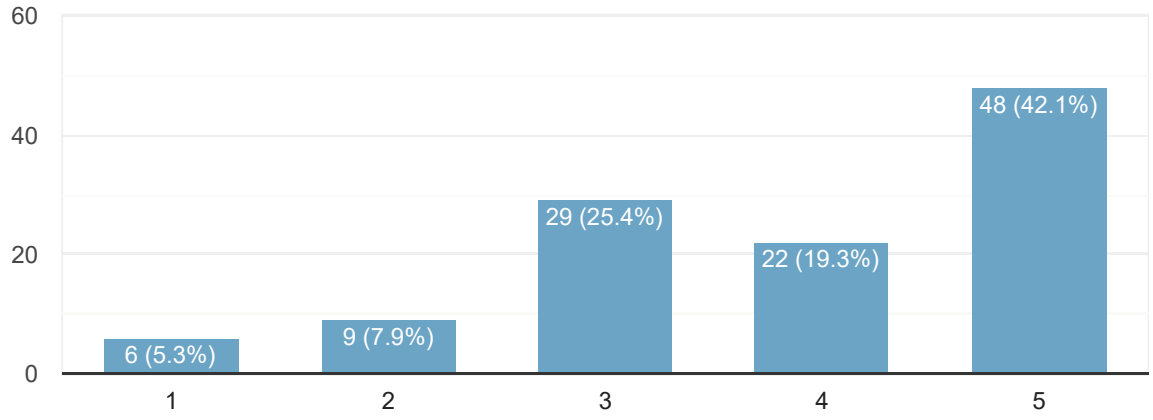
114 responses



Better integration between AudioWorklet and WASM

Copy

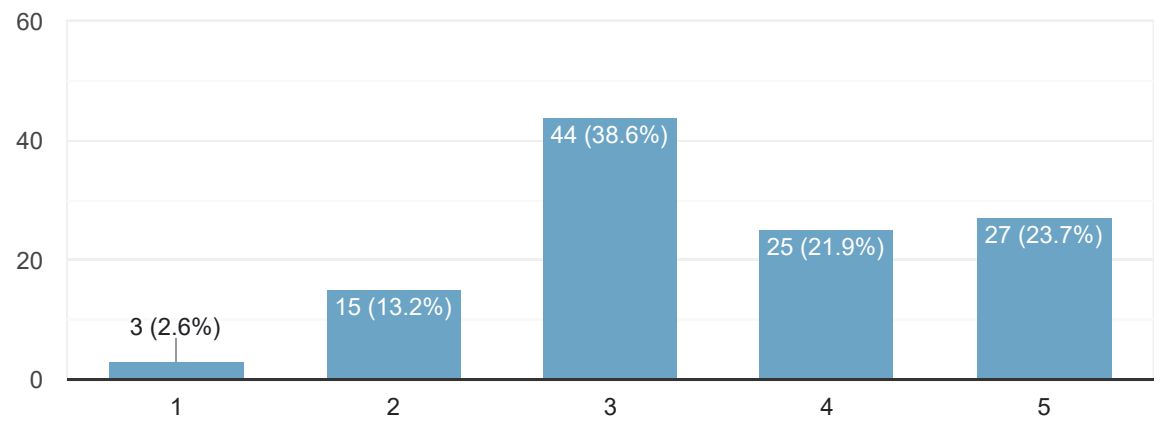
114 responses



Monitor the capacity of the Web Audio renderer. (i.e. CPU meter)

Copy

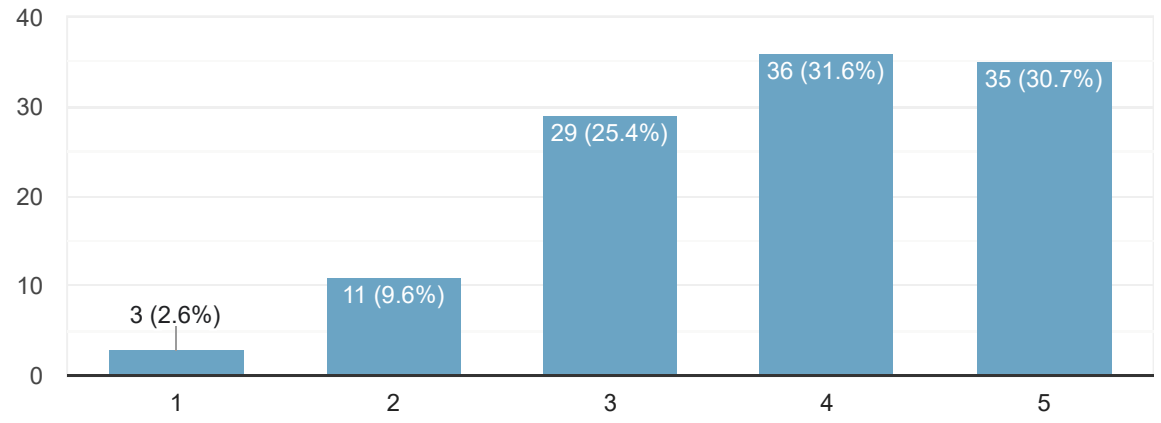
114 responses



Control the render buffer size (render quantum) of Web Audio API.

Copy

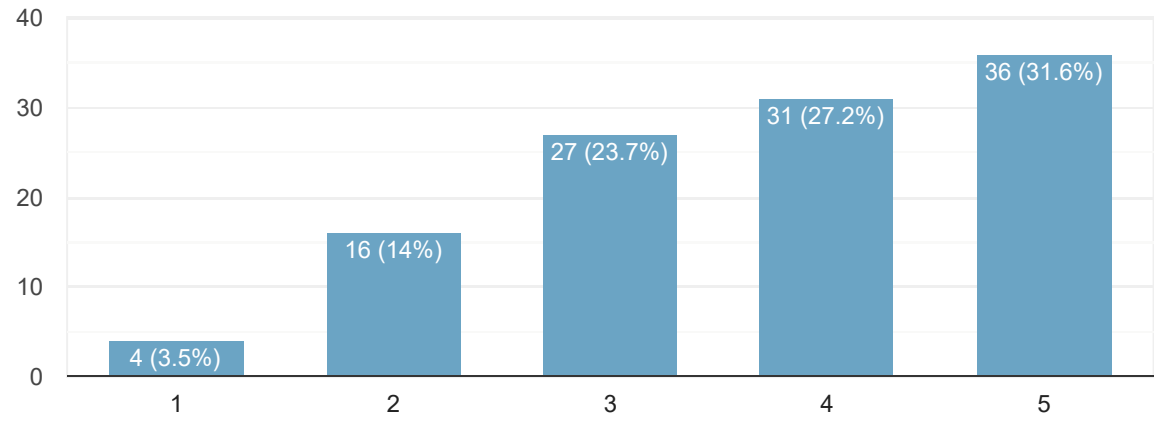
114 responses



Allow AudioContext and OfflineAudioContext in Worker.

Copy

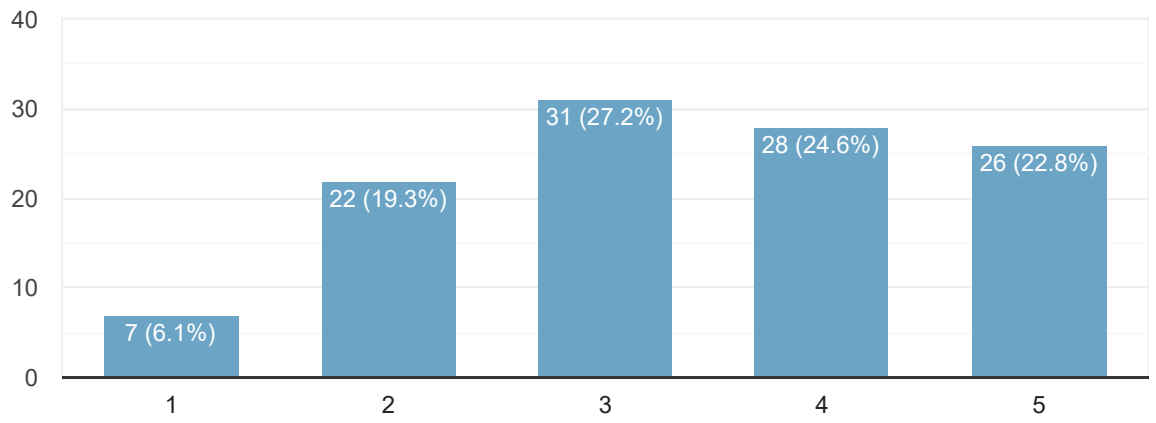
114 responses



Load an audio file with its original sample format. (e.g. 16-bit or less)

Copy

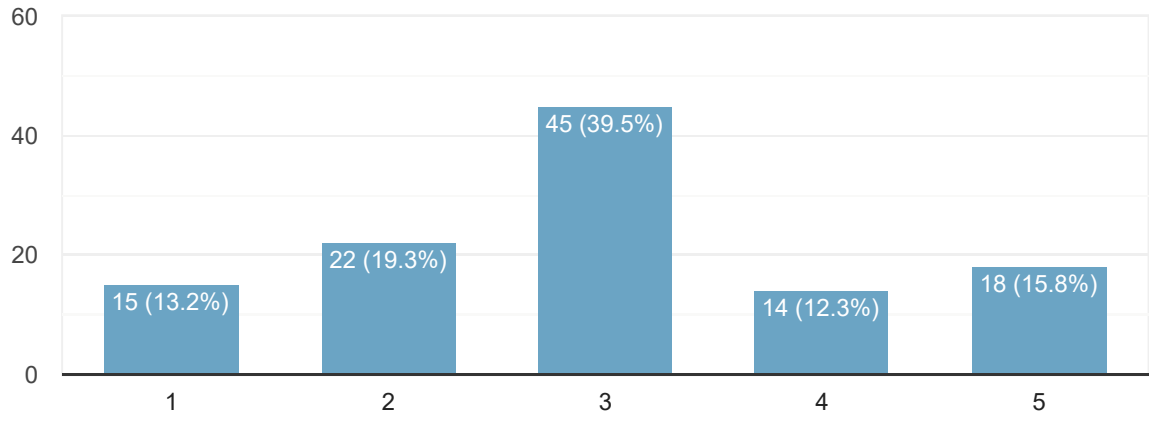
114 responses



Render the audio data with OfflineAudioContext only as much as I need and when I need it.

Copy

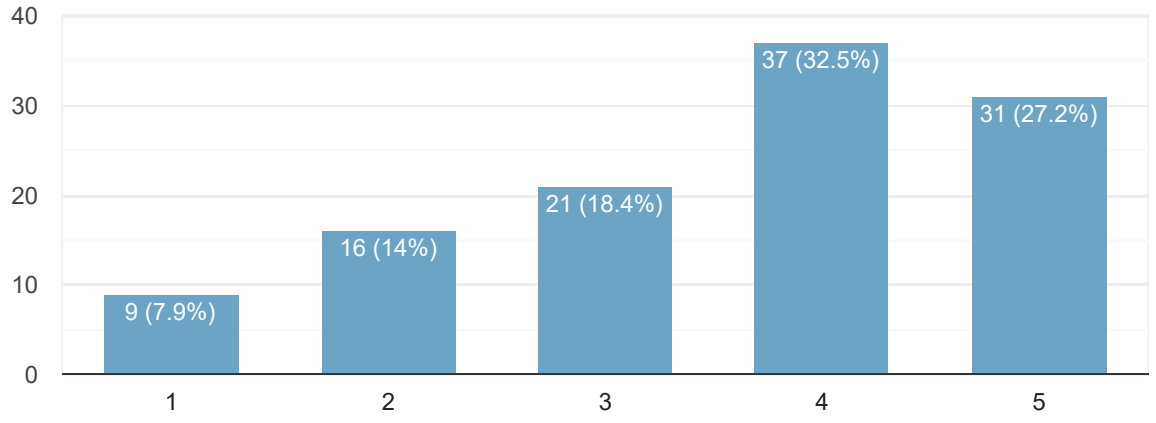
114 responses



AudioBufferSourceNode.playbackPosition

Copy

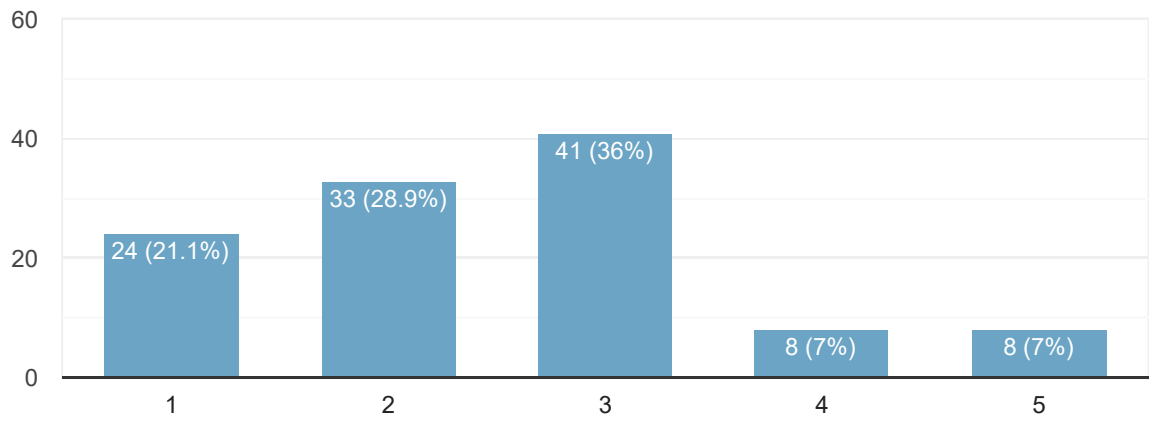
114 responses



Allow a custom windowing function for AnayserNode.

Copy

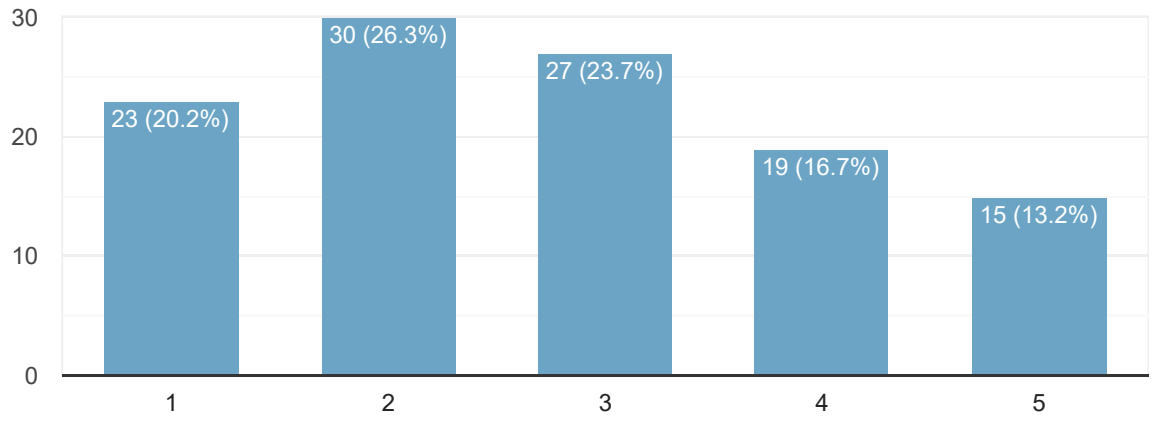
114 responses



OscillatorNode.phaseOffset

Copy

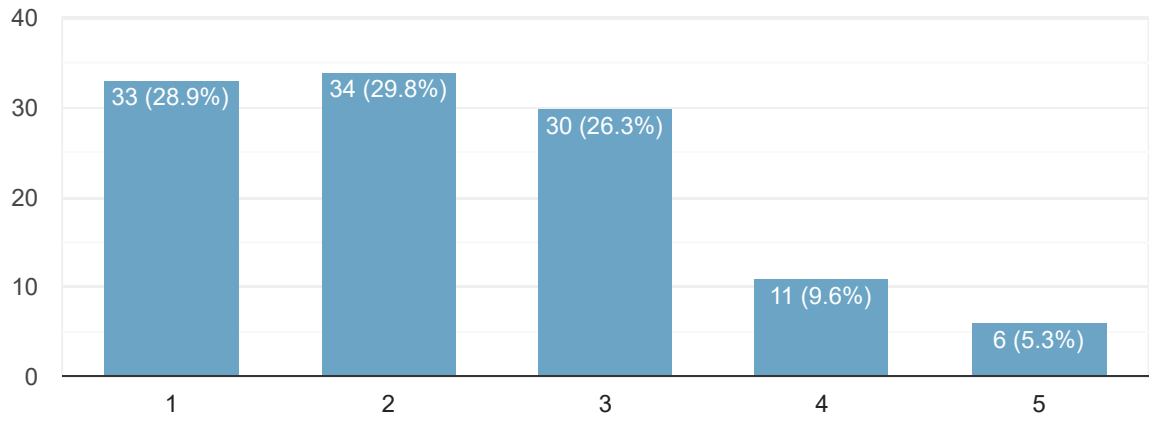
114 responses



Load a custom HRTF database for PannerNode.

Copy

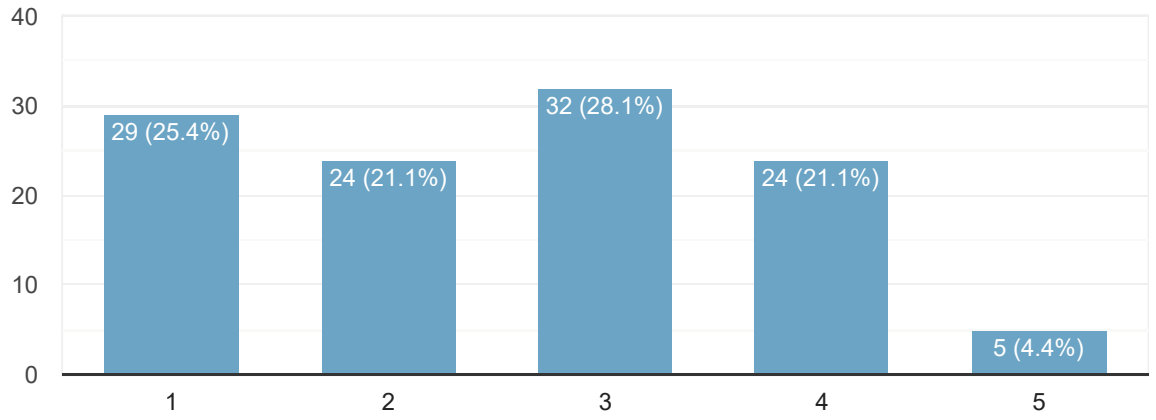
114 responses



Allow multichannel convolution more than 2 channels on ConvolverNode.

Copy

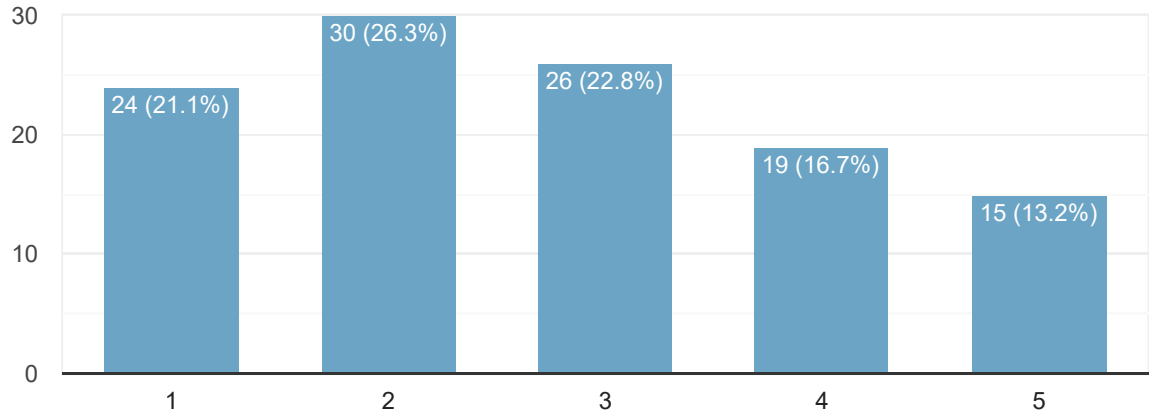
114 responses



Pulse Oscillator with PWM (Pulse Width Modulation) capability

Copy

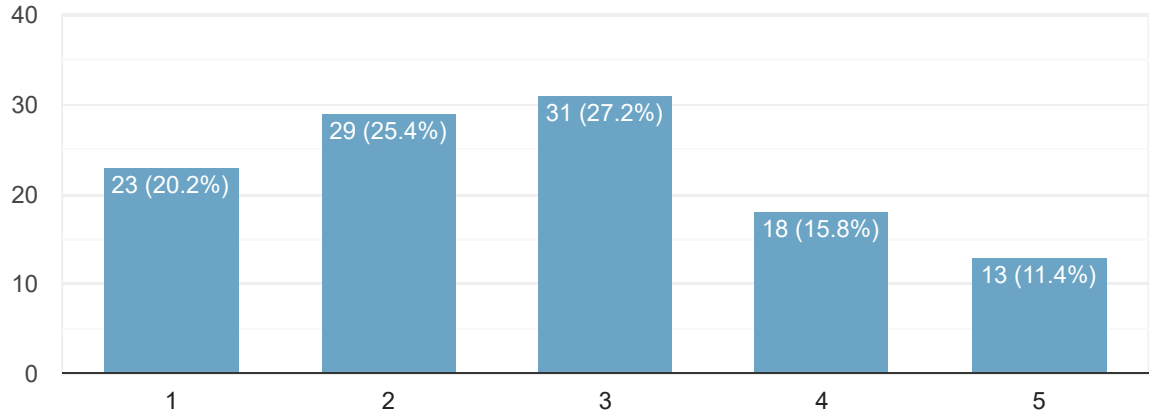
114 responses



NoiseGateNode & ExpanderNode

Copy

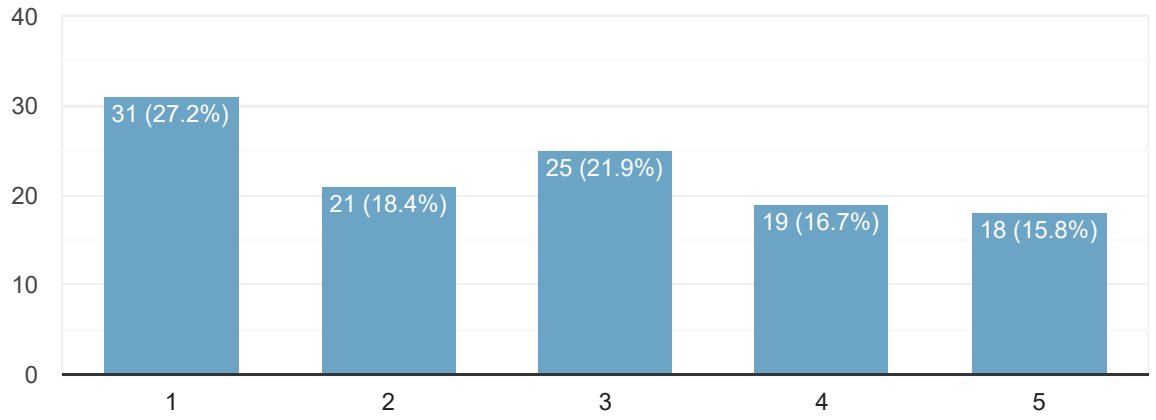
114 responses



NoiseGeneratorNode

Copy

114 responses

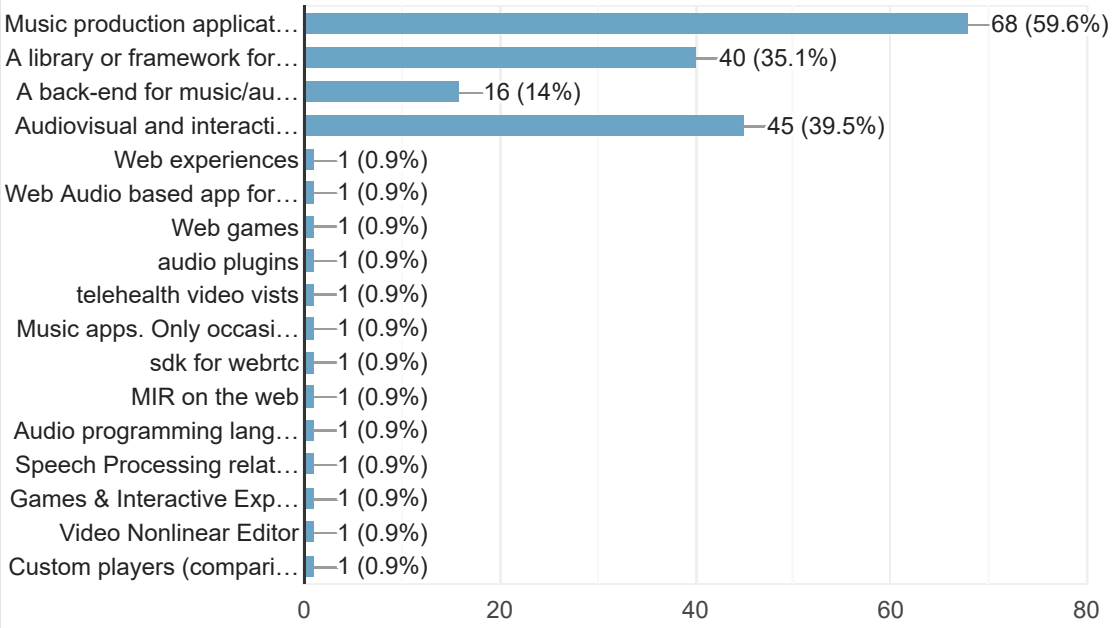


Developer Experience

What are you working on?

Copy

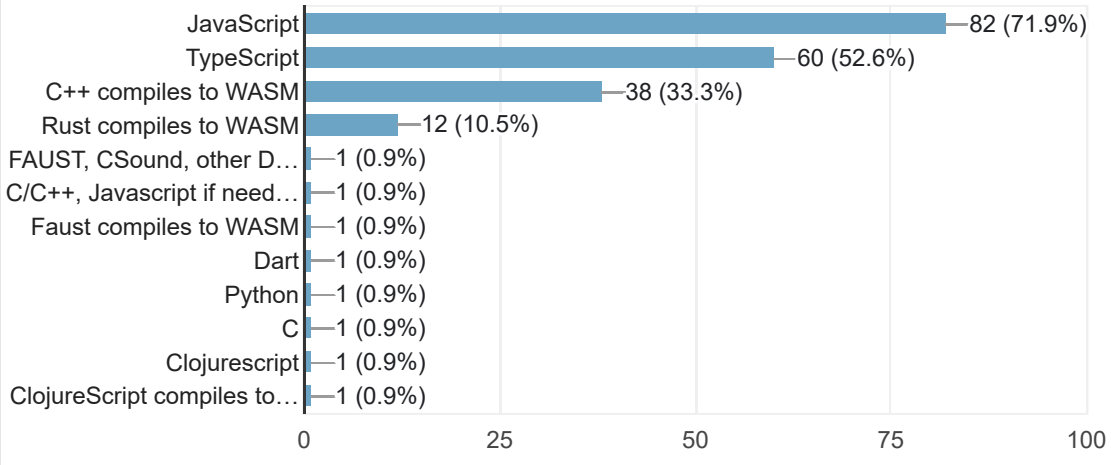
114 responses



What programming language do you use?

Copy

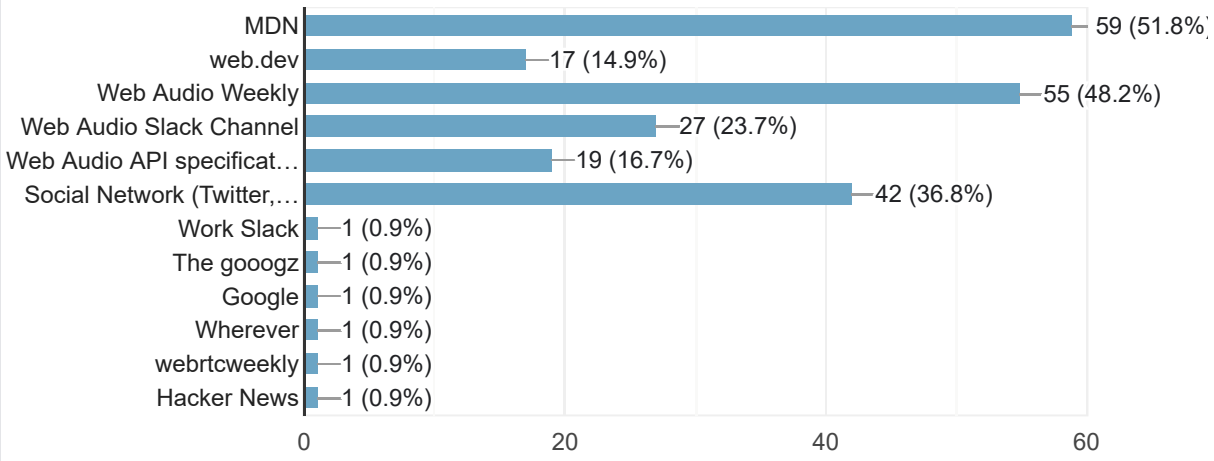
114 responses



Where do you get news and updates of Web Audio API?

Copy

114 responses



What are your pain points with Web Audio API? (optional)

46 responses

Fragmented, difficult to learn, not a clear guide or full examples of how to use and what could be made

Would be cool to be able to cancel a play or stop event. Also better performance analyze tools.

I've been missing a .getValueAtTime for audio parameters since the beginning. Would make working with changing parameter automation much easier.

IO selection not possible, drift of oscillators, when using web media API for sinkID output selection without getting hacky, really useful LFO nodes would be cool

Not Optimised !

Getting consistent results (without glitches) across browser and OS types

iOS

Latency management (i.e compensate latency with recorded multi tracks)

Control and reduce latency. Pinpoint web audio performances in DevTools performance tab.

transport and schedule audio events for a long timelines

Few recent intermediate/advanced code samples to study

AudioWorklet requires a secure context, complicating local testing and making the low latency API unavailable over less traditional networks, such as a server on a home LAN.

No build in FFT or Inverse FFT for audio processing.

debugging & profiling

I use a lot of nodes, and it pushes the cpu really hard, causing underruns. I need to start looking at AudioWorklet and WASM, but I just wish the native nodes were more performant. Also I'd love for cancelAndHoldAtTime to be implemented in Firefox.

Too focused on web devs dabbling in audio, should also be for audio devs trying web things.

slow input audio latency. getUserMedia Sucks

Wasm, threads, shared memory

Ease of making complex synthesizers without getting bogged down in unnecessary abstractions and with the output that is without jitter, popping or other artifacts

Speed and reliability when using custom logic (circle buffer where I do all float calculations) compared to C++

Still some audio glitches under high load.

latency

Not being able to select and stream from multiple audio input devices

Some builtin support to avoid crack/pop on suspend/resume would be nice, it would make pause/resume code quite a bit simpler

Needs clearer error messages when something goes wrong

We lack a developper tool dedicated tab to web audio which allows to see current audio graph

In a pro audio environment one of the most important things is predictability aka. being glitch free - so things like sudden V8 optimizations or garbage collections can be very problematic.

It is a poor low-level API for glitch-free playback of audio algorithms coded in C++/WASM

Audio Worklets requiring a secure context to work makes testing kinda painful, especially on mobile devices.

Does not work well with ReadableStream

Some of references(MDN) are outdated.

AnalyserNode would be more insightful, i.e. include more tools for power/amplitude analysis

Sample rate matching, especially when using live microphone input

lack of implementation of `audioContext.outputLatency`

stuff that's probably outside the scope of WAA like browser per-tab memory limits (https://github.com/WebAudio/web-audio-api/issues/2396 would go a long way towards alleviating this though!)

Safari (the browser) - more accurately, differences in implementation between browsers

Major pain points all relate to not being able to monitor thread overload, which results in unexpected audio drops.

Interacting between context and audio worklet can still be a bit complicated.

Please provide a modern builtin way to stream audio from the main thread (Audio Worklets make sense when the audio data can be generated in the audio thread, but they're too cumbersome if the audio data must be generated on the browser thread and then streamed to the audio thread).

Basically a modern replacement for the deprecated ScriptProcessorNode.

Only hacky solutions available to sync playback with visuals. No side chain compressor.

Debugging WASM-based AudioWorklets

Having separate gain nodes for buffers and any signal routable content such as Convolver or a filter. Having a ready to go gain node in such objects are quite essential I believe. Because controlling gain in a signal path is the most essential stage when dealing with audio. Original specs were therefore showed some respect to that workflow.

audioworklet support, backgrounding / jittery / mobile support

Would like to use the various ramp to value over time functions but concerns for cross-browser compatibility mean I often use tweening libraries instead

If I use a particular node from the WebAudio API, say, the Compressor or a Biquad filter, I have no way (that I know of) to query the latency of such DSP. So if I'm trying to play multiple streams of audio, and one of the streams has the effect prior to being mixed for playback, I don't know how to time align them in a sample accurate way.

No synchronization to video possible, bad/inflexible multichannel support