



Web Audio API

By James Zhang

What is it?

The Web Audio API is a high level Javascript API developed by W3C and is used for processing and synthesizing audio in web applications.

Browser Support

Web Audio API - WD

High-level JavaScript API for processing and synthesizing audio

Global 66.6%
unprefixed: 52.86%

Current aligned Usage relative Show all

IE	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
		31						
		33						
8		35	5.1				4.1	
9	31	36	7		7.1		4.3	
10	32	37	7.1		8		4.4	
11	33	38	8	25	8.1	8	4.4.4	38
	34	39		26			37	
	35	40		27				
	36	41						

Notes Known issues (1) Resources (7) Feedback

Firefox versions < 25 support an alternative, deprecated audio API.

What can you do with it?

Anything audio related in the browser is fair game.

Games, music, synthesis, visualizations, media enhancements, sound effects, etc.

More examples: <https://chromium.googlecode.com/svn/trunk/samples/audio/samples.html>

How do you use it?

AudioNodes



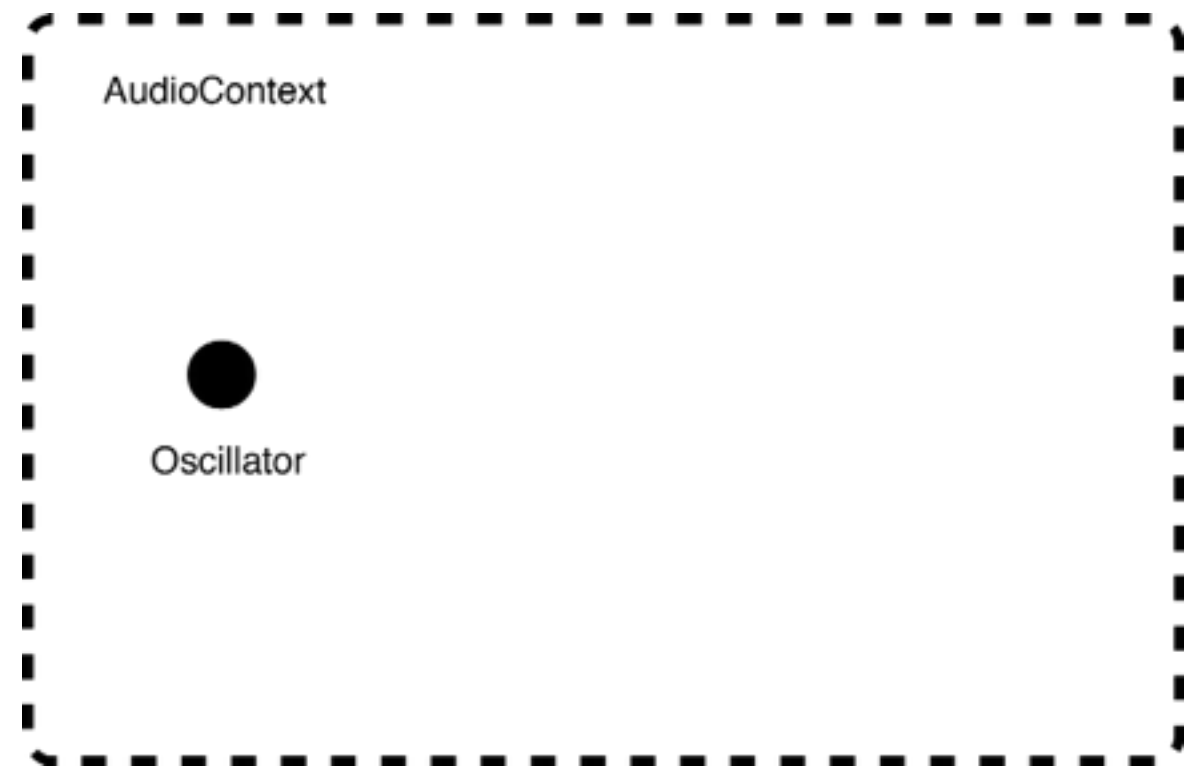
AudioContext

```
var context = new AudioContext();
```



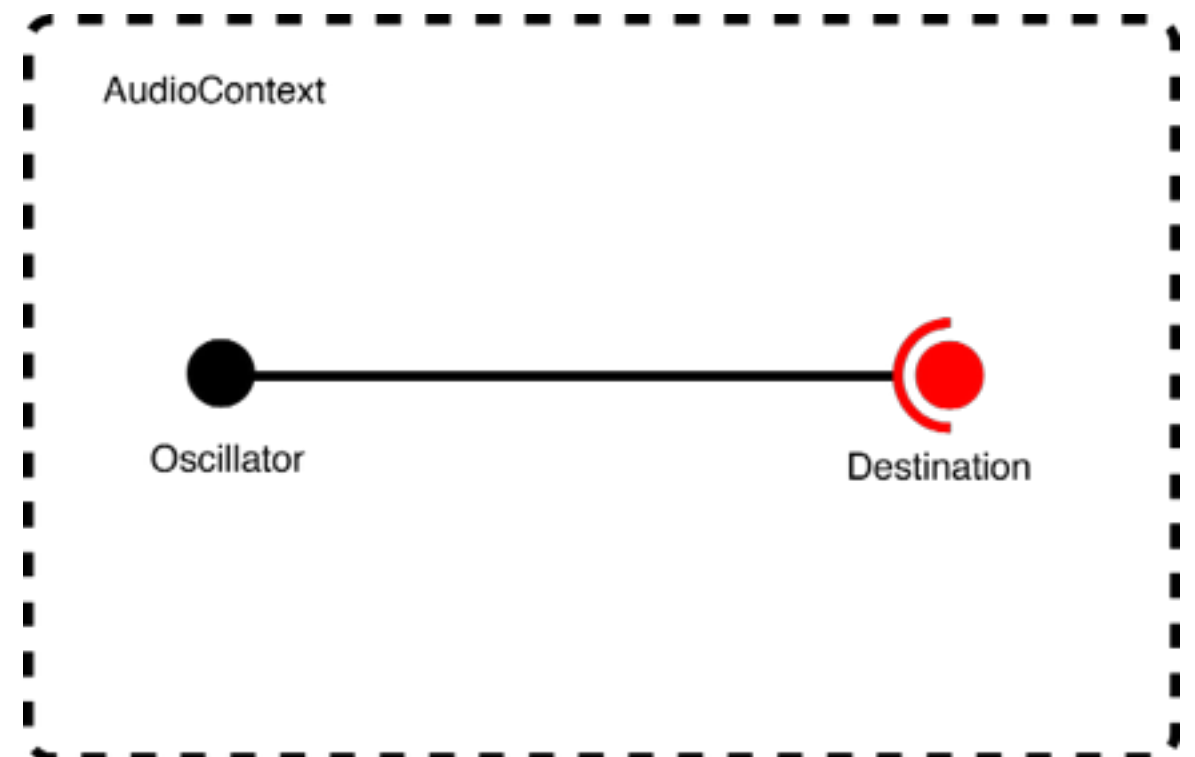
Oscillator

```
var context = new AudioContext();  
var oscillator = context.createOscillator();
```



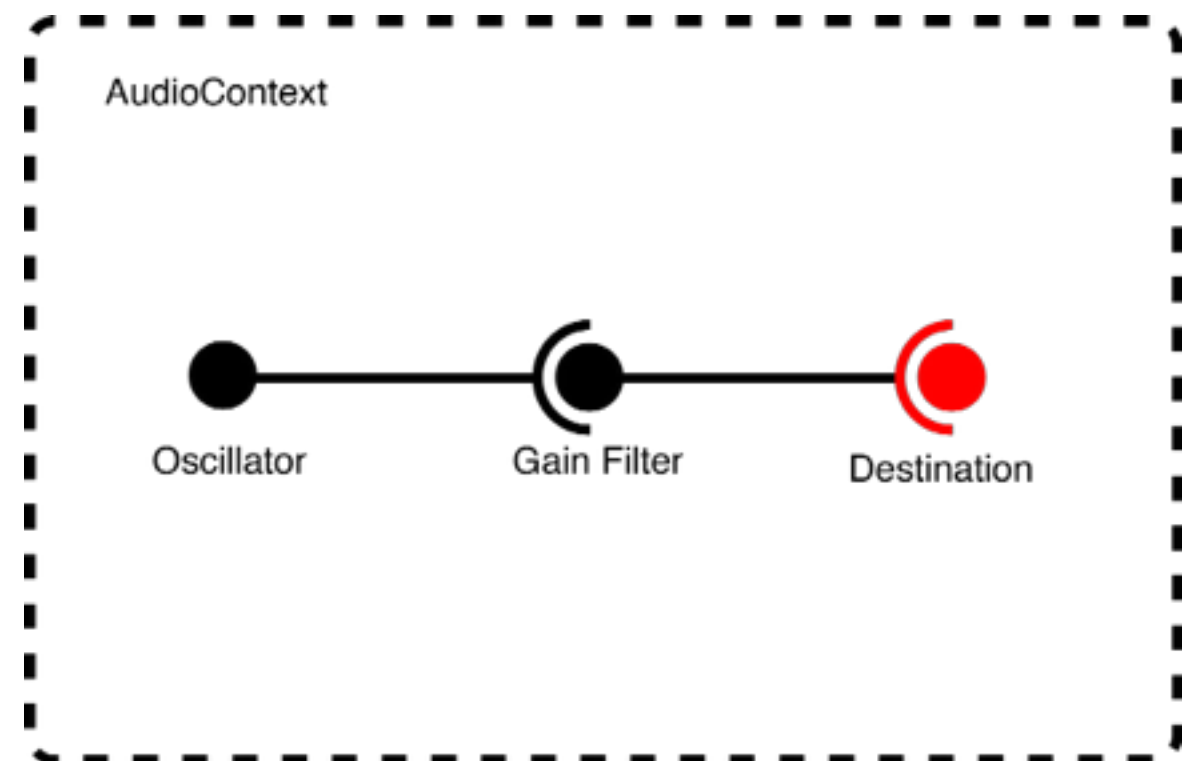
Connecting Oscillator to output

```
var context = new AudioContext();  
var oscillator = context.createOscillator();  
oscillator.connect(context.destination);  
oscillator.start(0);
```



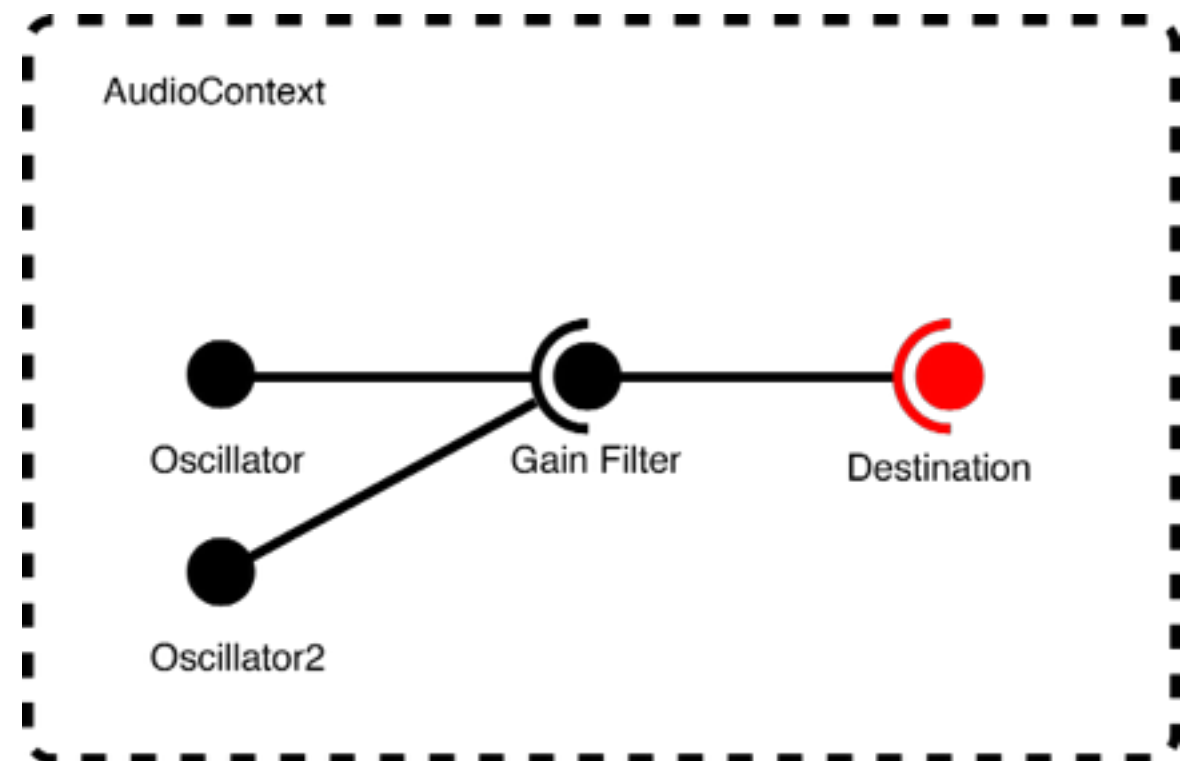
Adding a filter node

```
var context = new AudioContext();  
var oscillator = context.createOscillator();  
var gainFilter = context.createGain();  
  
gainFilter.gain.value = 0.5;  
  
oscillator.connect(gainFilter);  
gainFilter.connect(context.destination);  
  
oscillator.start(0);
```

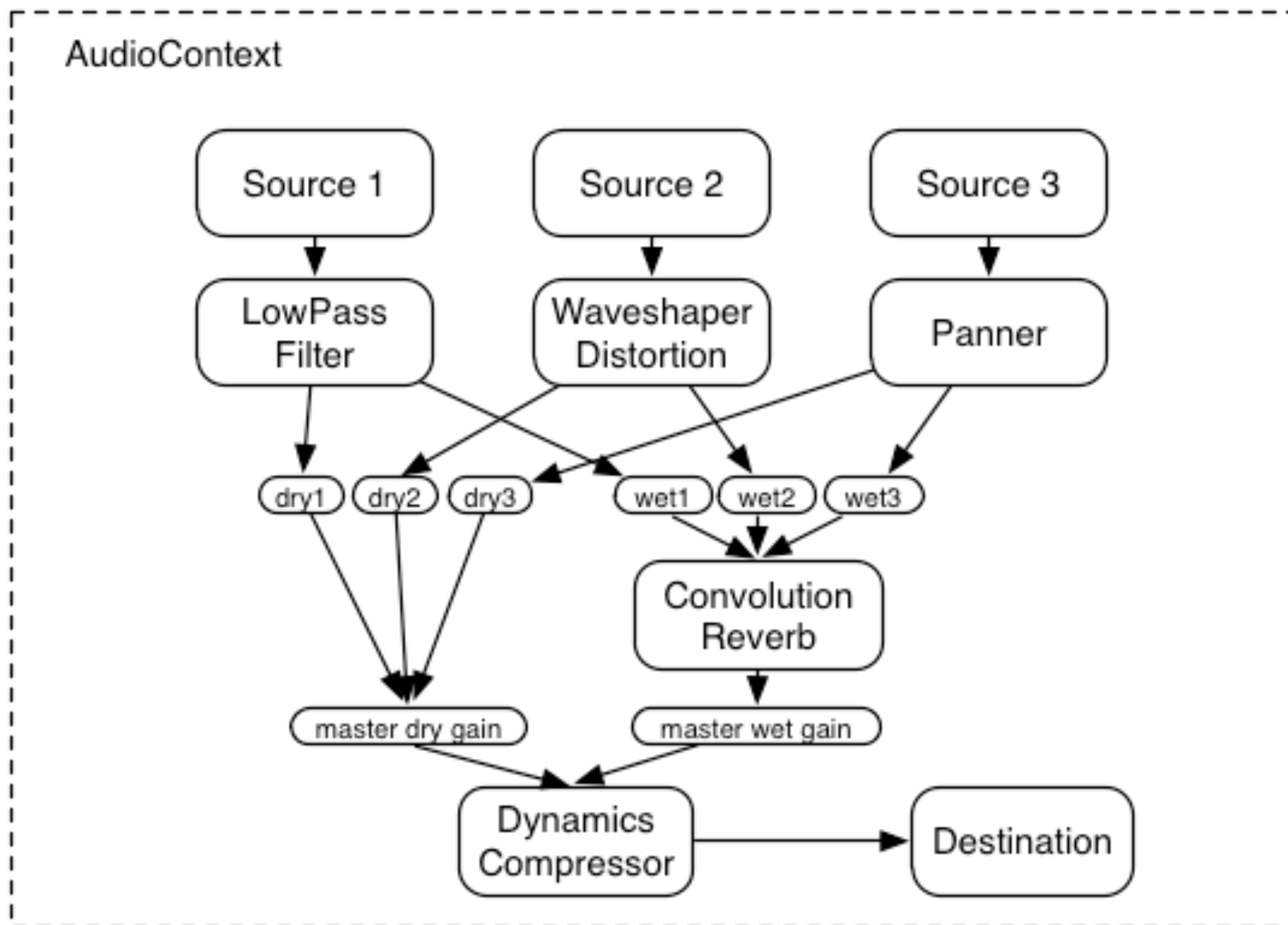


Adding another sound

```
var context = new AudioContext();  
var oscillator = context.createOscillator();  
var oscillator2 = context.createOscillator();  
var gainFilter = context.createGain();  
  
oscillator2.type = "triangle";  
oscillator2.frequency.value = 500;  
gainFilter.gain.value = 0.5;  
  
oscillator.connect(gainFilter);  
oscillator2.connect(gainFilter);  
gainFilter.connect(context.destination);  
  
oscillator.start(0);  
oscillator2.start(0);
```



It can eventually get complex...



Other Cool Features

- Loading local audio files
- Analyser
 - Extracts frequency data from sample byte data
- Access Audio from `<audio>` and `<video>` html elements
- Custom Oscillators
 - Essentially means you can create your own sound effects from scratch
- Interpolate filters over time
 - Such as interpolating the volume from 0 to .5 in a span of 4 seconds when the sound starts playing
 - You can also control how the filter will interpolate by specifying curve function

Questions?