

firefox yt better audio scaling for when im studying

```
// ==UserScript==
// @name      Youtube Music fix volume ratio
// @namespace  http://tampermonkey.net/
// @version   0.4
// @description Makes the YouTube music volume slider exponential so it's easier to
select lower volumes.
// @author    Marco Pfeiffer <git@marco.zone>
// @icon      https://music.youtube.com/favicon.ico
// @match     https://music.youtube.com/*
// @run-at    document-start
// @grant     none
// ==/UserScript==

(function() {
    'use strict';

    // manipulation exponent, higher value = lower volume
    // 3 is the value used by pulseaudio, which Barteks2x figured out this gist here:
https://gist.github.com/Barteks2x/a4e189a36a10c159bb1644ffca21c02a
    // 0.05 (or 5%) is the lowest you can select in the UI which with an exponent of 3
    becomes 0.000125 or 0.0125%
    const EXPONENT = 3;

    const storedOriginalVolumes = new WeakMap();
    const {get, set} = Object.getOwnPropertyDescriptor(HTMLMediaElement.prototype,
'volume');
    Object.defineProperty(HTMLMediaElement.prototype, 'volume', {
        get () {
            const lowVolume = get.call(this);
            const calculatedOriginalVolume = lowVolume ** (1 / EXPONENT);
```

```

        // The calculated value has some accuracy issues which can lead to problems
for implementations that expect exact values.

        // To avoid this, I'll store the unmodified volume to return it when read
here.

        // This mostly solves the issue, but the initial read has no stored value and
the volume can also change though external influences.

        // To avoid ill effects, I check if the stored volume is somewhere in the same
range as the calculated volume.

        const storedOriginalVolume = storedOriginalVolumes.get(this);
        const storedDeviation = Math.abs(storedOriginalVolume -
calculatedOriginalVolume);

        const originalVolume = storedDeviation < 0.01 ? storedOriginalVolume :
calculatedOriginalVolume;

        // console.log('manipulated volume from', lowVolume.toFixed(2), 'to ',
originalVolume.toFixed(2), storedDeviation);

        return originalVolume;
    },
    set (originalVolume) {
        const lowVolume = originalVolume ** EXPONENT;
        storedOriginalVolumes.set(this, originalVolume);
        // console.log('manipulated volume to ', lowVolume.toFixed(2), 'from',
originalVolume.toFixed(2));
        set.call(this, lowVolume);
    }
});
})();

```

Revision #1

Created 6 January 2024 11:31:44 by naruzkurai

Updated 6 January 2024 11:32:11 by naruzkurai