

Deprecations and removals in Chrome 68

To see all deprecations and removals for this and previous versions of Chrome, visit the [deprecations page](#).



By Joseph Medley
Technical Writer

Deprecate and Remove Negative Brightness Values in Filter

For compliance with specification, filter's `brightness()` function no longer accepts negative values.

[Chromestatus Tracker](#) | [Chromium Bug](#)

Remove `document.createTouch`

The `document.createTouch()` method is being removed because the `Touch()` constructor has been supported since Chrome 48. This follows a long-standing trend in JavaScript APIs of moving away from factory functions and toward constructors. The closely-related `document.createTouchList()` method is expected to be removed in Chrome 69.

[Intent to Remove](#) | [Chromestatus Tracker](#) | [Chromium Bug](#)

Remove `Document.selectedStylesheetSet` and `Document.preferredStylesheetSet`

The `Document.selectedStylesheetSet` and `Document.preferredStylesheetSet` attributes are removed because they are non-standard and only implemented by Chrome and WebKit. The standard versions of these attributes were removed from the spec in 2016.

`Document.styleSheets` provides some of the same functionality, though not all. Fortunately the risk to websites is low as the use of these items appears to be in single digits. (See the [Intent to Remove](#) for exact numbers.)

WEBGL_compressed_texture_atc

Previously, Chrome provided the `AMD_compressed_ATC_texture_atc` formats. These formats were widely supported at the time the extension was created. Hardware support has since dwindled to near-zero, with implementation currently possible only on Qualcomm devices. This extension has been rejected by the WebGL Working Group and support for it is now removed from Chrome.

[Chromestatus Tracker](#) | [Chromium Bug](#)

Deprecation policy

To keep the platform healthy, we sometimes remove APIs from the Web Platform which have run their course. There can be many reasons why we would remove an API, such as:

- They are superseded by newer APIs.
- They are updated to reflect changes to specifications to bring alignment and consistency with other browsers.
- They are early experiments that never came to fruition in other browsers and thus can increase the burden of support for web developers.

Some of these changes will have an effect on a very small number of sites. To mitigate issues ahead of time, we try to give developers advanced notice so they can make the required changes to keep their sites running.

Chrome currently has a [process for deprecations and removals of API's](#), essentially:

- Announce on the [blink-dev](#) mailing list.
- Set warnings and give time scales in the Chrome DevTools Console when usage is detected on the page.
- Wait, monitor, and then remove the feature as usage drops.

You can find a list of all deprecated features on [chromestatus.com](#) using the [deprecated filter](#) and removed features by applying the [removed filter](#). We will also try to summarize some of the changes, reasoning, and migration paths in these posts.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](#), and code samples are licensed under the [Apache 2.0 License](#). For details, see our [Site Policies](#). Java is a registered trademark of Oracle and/or its affiliates.

Last updated July 2, 2018.