

Chromium policy on JavaScript dialogs

History of JavaScript dialogs

JavaScript was introduced in 1995, and in the very first version of JavaScript were methods on the window object named alert(), confirm(), and prompt().

While they fit into the JavaScript of the time, their synchronous API is problematic for modern browsers. Because the JavaScript engine needs to pause until a user response is obtained, the JavaScript dialogs are app-modal. And because the dialogs are app-modal, they commonly (and unfortunately) are used to harm our users.

Because of this, the Chromium team highly recommends that you not use JavaScript dialogs.

Alternatives

There are many options for dialog replacement.

There are several choices for `alert()/confirm()/prompt()`. For notifying the user of events (e.g. calendaring sites), the Notifications API should be used. For obtaining user input, the HTML <dialog> element should be used. For XSS proofs-of-concept, devtool's `console.log(document.origin)` can be used.

As for `onbeforeunload`, it should be noted that it is *already* unreliable. As Ilya Grigorik points out, “You cannot rely on `pagehide`, `beforeunload`, and `unload` events to fire on mobile platforms.” If you need to save state, you should use the Page Visibility API.

Changes

The ability for a page to specify the `onbeforeunload` string was removed in Chrome 51. (It was also removed by Safari starting with Safari 9.1 and in Firefox 4.)

`alert()/confirm()/prompt()` dialogs have changed from being app-modal to being dismissed when their tab is switched from. This change took place across all channels at the beginning of May 2017.

`beforeunload` dialogs require a user gesture on the page to allow them to show [starting in Chrome 60](#). (This does not change the dispatching of the `beforeunload` event.) This aligns Chromium with Firefox, which made this change with [Firefox 44](#).

Showing an `alert()`/`confirm()`/`prompt()` dialog while in fullscreen will cause fullscreen to be lost [starting in Chrome 61](#).

`prompt()` dialogs do not activate their tab. If `prompt()` is called from a background tab, the call returns immediately and no dialog is shown. [This change](#) took place across all channels at the beginning of May 2017.

`alert()` dialogs do not activate their tab. If `alert()` is called from a background tab, the call returns immediately. The tab is marked with an indicator and the user will see the dialog when they switch to the tab. This behavior change is seen [starting in Chrome 64](#).

`confirm()` dialogs do not activate their tab. If `confirm()` is called from a background tab, the call returns immediately and no dialog is shown. This behavior change is seen [starting in Chrome 69](#).

Because of these changes, if your site uses dialogs, it is highly recommended that you move to using the earlier-mentioned alternatives so that this will not affect you.

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 June 11, 2018.