
Keyboard Savior Xtreme Documentation

Release 1

Joe Ulfers

Apr 19, 2018

Contents

1	Whitelist	3
1.1	What the savior blocks	3
1.2	Literature review	4
1.3	Related add-ons and scripts	5
1.4	About	6

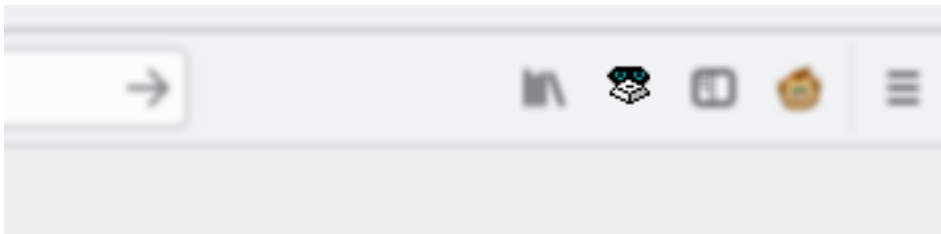
Keyboard Savior Xtreme is a Firefox add-on that blocks websites from hijacking your keystrokes. It takes the extreme approach of blocking sites from seeing any keyboard event except when focus is in *certain editable areas*.

Too extreme for you? Check out the *Related add-ons and scripts*.

[Install for Firefox](#)

If you want to allow custom shortcuts on particular sites, you can whitelist them. To whitelist a site, either:

- Add its domain to the list in add-on preference
- Or click the savior's toolbar icon:



Click the icon again to remove a site from the whitelist.

1.1 What the savior blocks

In general, Keyboard Savior Xtreme blocks pages from detecting keystrokes, thus, they can't interfere with normal browser behavior.

1.1.1 Input fields

There is no good reason sites should hijack keystrokes on checkboxes and other non-textual `<input>` tags, so the savior blocks this kind of thing:

But sites can sometimes use keystroke handlers on textual inputs for useful things like character counters,¹ so this is allowed:

Likewise, the savior permits `contenteditable` areas to capture keystrokes. Rich-text editors need this to implement word-processor shortcuts like `ctrl+s`:

¹ New code should prefer the `input` event.

1.1.2 Media

The savior allows pages to override keyboard shortcuts on `<audio>` and `<video>` elements, so in this example, the `P` key pauses and plays the video²:

But the media exception only applies when not using the browser's built-in controls, so the `P` shortcut won't work on this video:

1.1.3 Escape

The savior can't stop popups, but it allows sites to handle the `escape` key on the off chance that they designed their modals correctly:

1.1.4 Fullscreen

Games usually have good reason to implement custom shortcuts, so the savior allows full screen pages to do whatever they want:

1.1.5 Pointer lock

Pointer lock lets a page trap your mouse, so it may as well be allowed to trap your keyboard too:

1.2 Literature review

Plenty of ink has already spilled in this war.

- [10-year-old Firefox bug with 20 duplicates and no sign of discussion slowing down](#)
- [A Hacker News thread](#)
- [How to prevent keystroke grabbing/hijacking?](#)
- [How do I prevent pages I visit from overriding selected Firefox shortcut keys?](#)
- [How to prevent sites from hijacking keyboard shortcuts](#)

On one side is truth and justice, on the other, the misguided people who think their sites need a keyboard shortcut “feature.” This [comment on Super User](#) sums it up:

It's so absurd to me, because the only way anyone will discover this “feature” is by trying to use [the slash key] for its usual purpose! The only people who are affected by this are the ones who are frustrated by it!

Custom shortcuts are bad for accessibility, bad for power users, unlikely to be useful to most users, and difficult to implement correctly.

1.2.1 Custom shortcuts break user expectations

Not only do custom shortcuts interfere with what power users expect of their browser, but they are even worse for users who *can't* physically use a mouse. This includes motor impairments and, to some extent blindness.

“Wait,” you might think, “if my users can't use a mouse, won't my shortcuts help them?”

No!

² Video from the [UK National Archives](#)

Your users won't know what your particular shortcuts are and you have a good chance of breaking shortcuts that they *need* to navigate your site. Screen readers, for example, provide many shortcuts you don't know, but their users require.

1.2.2 Custom shortcuts are hard to implement

Oh sure, it seems simple enough, but wait until you really get into the details.

Have you thought about [international keyboard layouts](#)? If “dead key” doesn't mean anything to you, you don't know enough to implement keyboard shortcuts reliably.

Do you know how all the keyboard event parameters differ? Here are some of them:

- `keyCode`
- `charCode`
- `code`
- `keyIdentifier`
- `key`
- `which`

1.2.3 When you should make custom shortcuts

Despite the above, there are times when custom shortcuts are appropriate, so Keyboard Savior Xtreme makes *exceptions*.

- Custom modals like popups should always implement `escape` to close
- Rich-text editors and similar should implement shortcuts analogous to the well-known shortcuts in their native counterparts
- Games

In almost every other case, custom keystroke handlers are a mistake.

1.3 Related add-ons and scripts

1.3.1 Fixers for specific hijacks

- [Google stop focus Greasmonkey script](#)
- [Free Control Firefox add-on](#)
- [An answer on Super User](#)
- [Slash Preserver Firefox add-on](#)
- [Save Slash Greasmonkey script](#)

1.3.2 More general

- [Disable website keyboard hooks Greasmonkey script](#)
- [Vimperator Firefox add-on](#)

1.4 About

Keyboard Savior Xtreme is a project by [Josiah Ulfers](#).

Source code and original content on this site is [unlicensed](#).