

# Smooth Scrolling in Chrome 49



By Paul Lewis

Paul is a Design and Perf Advocate

If there's one thing that people really want from scrolling, it's for it to be smooth. Historically Chrome has had smooth scrolling in some places, like – say – when users scroll with their trackpads, or fling a page on mobile. But if the user has a mouse plugged in then they'd get a more jittery “stepped” scrolling behavior, which is way less aesthetically pleasing. That's all about to change in Chrome 49.

The solution to the stepped native, input-driven scroll behavior for many developers has been to use libraries, the goal of which being to remap it to something smoother and nicer on the eyes. Users also do this, too, through extensions. There are downsides to both libraries and extensions that change scrolling, though:

- **An uncanny valley feel.** This manifests itself in two ways: firstly, one site may have a smooth scroll behavior, but another may not, so the user can end up feeling disoriented by the inconsistency. Secondly, the library's smoothness physics won't necessarily match those of the platform's. So while the motion may be smooth it can feel wrong or uncanny.
- **Increased propensity for main thread contention and jank.** As with any JavaScript added to the page, there will be an increased CPU load. That's not necessarily a disaster, depending on what else the page is doing, but if there is some long-running work on the main thread, and scrolling has been coupled to the main thread, the net result can be stuttering scrolls and jank.
- **More maintenance for developers, more code for users to download.** Having a library to do smooth scrolling is going to be something that has to be kept up-to-date and maintained, and it will add to the overall page weight of the site.

These drawbacks are often also true of many libraries that deal with scroll behaviors, whether that's parallax effects, or other scroll-coupled animations. They all too often trigger jank, get in the way of accessibility, and generally damage the user experience. Scrolling is a core interaction of the web, and altering it with libraries should be done with great care.

In Chrome 49, the default scroll behavior will be changing Windows, Linux, and Chrome OS. The old, stepped scrolling behavior is going away, and scrolling will be smooth by default! No changes to your code are necessary, except maybe removing any smooth scrolling libraries if you've used them.

## More scrolling goodies

There are other scroll-related goodies in the works that are also worth mentioning. Many of us want scroll-coupled effects, like parallaxing, smooth scrolling to a document fragment (like [example.com/#somesection](http://example.com/#somesection)). As I mentioned earlier, the approaches that are used today can often be detrimental to both developers and users. There are two platform standards that are being worked on that could help: Compositor Worklets and the `scroll-behavior` CSS property.

## Houdini

Compositor Worklets are part of [Houdini](#), and are yet to be fully spec'd out and implemented. That said, [as the patches land](#), they will allow you to write JavaScript that's run as part of the compositor's pipeline, which in general means that scroll-coupled effects like parallaxing will be kept perfectly in sync with the current scroll position. Given the way that scrolling is handled today, where scroll events are only periodically sent to the main thread (and can be blocked by other main thread work), this would represent a huge leap forward. If you're interested in Compositor Worklets, or any of the other exciting new features that Houdini brings, look over the [Intro to Houdini post by Surma](#), the [Houdini specs](#), and contribute your thoughts to the [Houdini mailing list](#)!

## scroll-behavior

When it comes to fragment-based scrolling, the [scroll-behavior CSS property](#) is something else that could help. If you want to try it out you'll be pleased to know it's shipped in Firefox already, and you can enable it in Chrome Canary using the **"Enable experimental Web Platform features"** flag. If you set – say – the `<body>` element to `scroll-behavior: smooth`, all scrolls that are triggered either by fragment changes or by `window.scrollTo` will be animated smoothly! That's way better than having to use and maintain code from a library

that tries to do the same thing. With something as fundamental as scrolling, it's really important to avoid breaking user expectation, so while these features are in flux it's still worth adopting a Progressive Enhancement approach, and removing any libraries that attempt to polyfill these behaviors.

## Go forth and scroll

As of Chrome 49, scrolling is getting smoother. But that's not all: there are more potential improvements that may land, through Houdini and CSS properties like `smooth-scroll`. Give Chrome 49 a try, let us know what you think, and, most of all, **let the browser do the scrolling where you can!**

---

*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.*