New In Chrome 56

By [Pete LePage](#)

Pete is a Developer Advocate

- Users can now select Bluetooth Low Energy devices to connect to web sites that use the [Web Bluetooth API](#).
- `position: sticky` is back - making it easy to create elements that scroll normally until sticking to the top of the viewport.
- And [HTML5 by Default](#) is enabled for all users.

I'm Pete LePage, let's dive in and see what's new for developers in Chrome 56.

Web Bluetooth API

Until now, users who have bluetooth devices were forced to install native apps to communicate with them. With Chrome 56, users can select nearby Bluetooth Low Energy devices to provide to web sites that use the [Web Bluetooth API](#).

Similar to selecting files to share with a web site, this is more secure than installing a native application that can read data from anything. Users are in full control of when and which device is shared with a specific web site.

The Web Bluetooth API uses the [GATT protocol](#), which enables apps to connect to devices such as light bulbs, toys, heart-rate monitors, LED displays and more with just a [few lines of JavaScript](#). Web Bluetooth can also be combined with physical web beacons to make discovering nearby devices even easier.

Francois has a great article on Updates, be sure to check out some of neat demos to go along with it. And be sure to check out the Web Bluetooth Community.

CSS `position: sticky;`

Previously, building content headers that scrolled normally until sticking to the top of the viewport required listening to scroll events and switching an element's position from relative to fixed at a specified threshold. It was difficult to synchronize, and often results in small visual jumps.

Chrome now supports CSS `position: sticky;`, a new way to position elements.

An element that is position sticky, starts relative; but becomes fixed, after the element reaches a certain scroll position.

Simply set `position: sticky`, and set a threshold for it to become sticky.

```
h3 {
  /* Element will be 'fixed' when it ... */
  position: sticky;
  /* ... is 10px from the top of the viewport */
  top: 10px;
}
```

Paul Kinlan has an Updates post about it.

HTML5 By Default

Last August, we announced that we'd be moving to HTML5 By Default to offer a safer, more power-efficient experience. This change disables Adobe Flash Player unless there's a user indication that they want Flash content on specific sites, and eventually all websites will require the user's permission to run Flash.

In Chrome 56, HTML5 By Default has been enabled for all users, which means they will be prompted to run Flash on sites they've never visited.

More details about how and when users will be prompted, and recommendations on how to test your Flash sites.

And more

And of course, there's plenty more.

- [WebVR](#) is available as an Origin Trial.
- The [WebGL 2.0 API](#) is now available.
- And the Payment Request API has a [variety of new features](#).

If you want to stay up to date with Chrome and know what's coming, be sure to [subscribe](#), follow [@ChromiumDev](#) on Twitter and be sure to check out the [videos from the Chrome Dev Summit](#) for a deeper dive into some of the awesome things the Chrome team is working on.

I'm Pete LePage, and as soon as Chrome 57 is released, I'll be right here to tell you – what's new in Chrome!

Subscribe to Chrome Developers on YouTube

Subscribe to our [YouTube channel](#) or our [RSS feed](#)

⌄   New in Chrome Easter Egg

If you read this far, you deserve to see the blooper reel from [New in Chrome 52](#)! I felt like every time I opened my mouth, a truck would drive by, a helicopter would fly over, a car would honk it's horn.

Oh, and a big thanks to Andrew for lending me his shirt! I had a bit of a wardrobe malfunction.

Last updated July 2, 2018.