New in Chrome 64

By Pete LePage
Pete is a Developer Advocate

- Support for `ResizeObservers`, will notify you when an element's content rectangle has changed its size.
- Modules can now access to host specific metadata with import.meta.
- The pop-up blocker gets strong.
- `window.alert()` no longer changes focus.

And there's plenty more!

I'm Pete LePage. Let's dive in and see what's new for developers in Chrome 64!

Note: Want the full list of changes? Check out the Chromium source repository change list.

ResizeObserver

Tracking when an element's size changes can be a bit of a pain. Most likely, you'll attach a listener to the document's `resize` event, then call `getBoundingClientRect` or `getComputedStyle`. But, both of those can cause layout thrashing.

And what if the browser window didn't change size, but a new element was added to the document? Or you added `display: none` to an element? Both of those can change the size of other elements within the page.

`ResizeObserver` notifies you whenever an element's size changes, and provides the new height and width of the element, reducing the risk of layout thrashing.

Like other Observers, using it is pretty simple, create a `ResizeObserver` object and pass a callback to the constructor. The callback will be given an array of `ResizeOberverEntries` – one entry per observed element – which contain the new dimensions for the element.

```
const ro = new ResizeObserver( entries => {
  for (const entry of entries) {
    const cr = entry.contentRect;
    console.log('Element:', entry.target);
    console.log(`Element size: ${cr.width}px × ${cr.height}px`);
    console.log(`Element padding: ${cr.top}px ; ${cr.left}px`);
  }
});

// Observe one or multiple elements
ro.observe(someElement);
```

Check out `ResizeObserver`: It's like `document.onresize` for Elements for more details and real world examples.

Improved Pop-up Blocker

I hate tab-unders. You know them, it's when a page opens a pop-up to some destination AND navigates the page. Usually one of them is an ad or something that you didn't want.

Starting in Chrome 64, these type of navigations will be blocked, and Chrome will show some native UI to the user - allowing them to follow the redirect if they want.

`import.meta`

When writing JavaScript modules, you often want access to host-specific metadata about the current module. Chrome 64 now supports the `import.meta` property within modules and exposes the URL for the module as `import.meta.url`.

This is really helpful when you want to resolve resources relative to the module file as opposed to the current HTML document.

And more!

These are just a few of the changes in Chrome 64 for developers, of course, there's plenty more.

- Chrome now supports named captures and Unicode property escapes in regular expressions.
- The default `preload` value for `<audio>` and `<video>` elements is now `metadata`. This brings Chrome in line with other browsers and helps to reduce bandwidth and resource usage by only loading the metadata and not the media itself.
- You can now use `Request.prototype.cache` to view the cache mode of a `Request` and determine whether a request is a reload request.
- Using the Focus Management API, you can now focus an element without scrolling to it with the `preventScroll` attribute.

`window.alert()`

Oh, and one more! While this isn't really a 'developer feature', it makes me happy. `window.alert()` no longer brings a background tab to the foreground! Instead, the alert will be shown when the user switches to back to that tab.

No more random tab switching because something fired a `window.alert` on me. I'm looking at you old Google Calendar.

Be sure to subscribe to our YouTube channel, and you'll get an email notification whenever we launch a new video, or add our RSS feed to your feed reader.

I'm Pete LePage, and as soon as Chrome 65 is released, I'll be right here to tell you – what's new in Chrome!

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.