Feature-Policy

In this lesson, we'll discuss the Feature-Policy header.

In July 2018, security researcher Scott Helme published a very interesting blog post detailing a new security header in the making, `Feature-Policy`.

Currently supported by very few browsers, only Chrome and Safari at the time of writing, this header lets us define whether a specific browser feature is enabled within the current page. With a syntax very similar to CSP, we should have no issue understanding what a feature policy such as the following one means.

```
Feature-Policy: vibrate 'self'; push *; camera 'none'
```

If we still have doubts about how this policy impacts the browser APIs available to the page, we can simply dissect it:

- `vibrate 'self'`: this will allow the current page to use the vibration API and any nested browsing contexts (iframes) on the same origin.
- `push *`: the current page and any iframe can use the push notification API.
- `camera 'none'`: access to the camera API is denied to the current page and any nested context (iframes).

The feature policy might have a short history, but it doesn’t hurt to get a head start. If your website allows users to take a selfie or record audio, it would be quite beneficial to use a policy that restricts other contexts from accessing the API through your page.

In the next lesson, we’ll discuss the header `X-Content-Type-Options`.