Web Animations Resources

By Sam Thorogood (https://developers.google.com/web/resources/contributors/samthorogood)
Evangelises Chrome and the mobile web in the Developer Relations team at Google.

The Web Animations API provides powerful primitives to describe imperative animations from JavaScript - but what does that mean? Find out about the resources available to you, including Google's demos (https://web-animations.github.io/web-animations-demos) and codelabs (https://github.com/web-animations/web-animations-codelabs).

Background

At its core, the API provides the `Element.animate()` method. Let's see an example, which animates the background color from red to green-

```
var player = document.body.animate(
    [{'background': 'red'}, {'background': 'green'}], 1000);
```

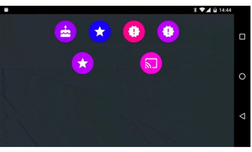
This method is supported in all modern browsers, with a great polyfill fallback (more on that later). Native support for this method - and its player object - became widely available as part of Chrome 39 (https://developers.google.com/web/updates/2014/12/web-animation-playback). It's also natively available (http://caniuse.com/#feat=web-animation) in Opera, and is under active development (https://birtles.github.io/areweanimatedyet/) for Firefox. This is a powerful primitive that deserves a place in your toolbox.

> Note! For a primer on the different approaches to animation on the web, including Web Animations, check out Modern Animation Fundamentals on YouTube (https://www.youtube.com/watch?v=WaNoqBApBNI) and Animations on Web Fundamentals (https://developers.google.com/web/fundamentals/design-and-ux/animations/).

Codelabs

A growing number of codelabs are available for the Web Animations API (https://github.com/web-animations/web-animations-codelabs). These are self-paced guides that demonstrate different concepts in the API. In most of these codelabs, you'll take static content and enhance it with animation effects.

These codelabs (https://github.com/web-animations/web-animations-codelabs), and related links or resources, are the absolute best place to start if you're looking to understand the new primitives available to you in Web Animations. For an idea of what you might build, check out this Android-inspired reveal effect-



So if you're just getting started, then look no further!

Demos

If you're looking for inspiration, be sure to check out the Material-inspired Web Animations Demos (https://web-animations.github.io/web-animations-demos), with source hosted on GitHub (https://github.com/web-animations/web-animations-demos). These demonstrate a variety of amazing effects and you can view each demo's source code inline.

The demos include a colorful spinning galaxy (https://web-animations.github.io/web-animations-demos/#galaxy), rotating Earth (https://web-animations.github.io/web-animations-demos/#globe), or even just a variety of effects (https://web-animations.github.io/web-animations-demos/#animate_css) on a plain old element.

Polyfill

To ensure great support across all modern browsers, you can use a polyfill library. The Web Animations API has a polyfill available right now (https://github.com/web-animations/web-animations-js) that brings it to all modern browsers, including Internet Explorer, Firefox, and Safari.

If you're feeling adventurous, you can use the web-animations-next polyfill, which also includes features that are yet to be finalized - such as the composable `GroupEffect` and `SequenceEffect` constructors. For a comparison between the two polyfills, please see the homepage (https://github.com/web-animations/web-animations-js#different-build-targets).

To use either polyfill in your code, you have a few options.

1. Use a CDN, such as cdnjs (https://cdnjs.com/libraries/web-animations), jsDelivr (https://www.jsdelivr.com/projects/web-animations), or target a specific release via rawgit.com (https://rawgit.com)
2. Install via NPM or Bower

```
$ npm install web-animations-js
$ bower install web-animations-js
```

In all cases, you can simply include the polyfill in a script tag before any other code-

```
<script src="https://cdn.jsdelivr.net/web-animations/latest/web-animations.min.js"></script>
<script>
  document.body.animate([
    {'background': 'red'},
    {'background': 'green'}
  ], 1000);
</script>
```

Other resources

If you'd like to read a more technical introduction, please check out the W3C spec (https://w3c.github.io/web-animations/).

Dan Wilson has also written a great set of posts on Web Animations (http://danielcwilson.com/tags/web-animations-api/), including on how to use it alongside the new CSS `motion-path` property (http://danielcwilson.com/blog/2015/09/animations-part-5/). For some samples using `motion-path`, check out Eric Willigers' doc (https://docs.google.com/document/d/15nn0tc9meyahzSBAauYtlUpGFsuHaieZt403k1v9B90/edit).