CSS Filter Effects Landing in WebKit

By Eric Bidelman

Engineer @ Google working on web tooling: Headless Chrome, Puppeteer, Lighthouse

Background

Filter effects have been around for awhile but were designed to work in SVG. They're fantastically powerful at applying effects like color intensity, warping, or blurring to an image before it's composited and rendered into the document.

Well,...way back in 2009 Mozilla said SVG wasn't enough! They ended up taking filters one step further and allowed them on HTML content in Firefox 3.5. Check out Paul Irish's timeless demo of SVG filters on a playing `<video>`. Again, only works in Firefox but still the bees knees.

Today

Flash forward to the end of 2011 and Adobe (plus others) have been hard at work bringing this amazing technology to CSS. Specifically, I'm referring to CSS Filter Effects 1.0, which WebKit has started to implement.
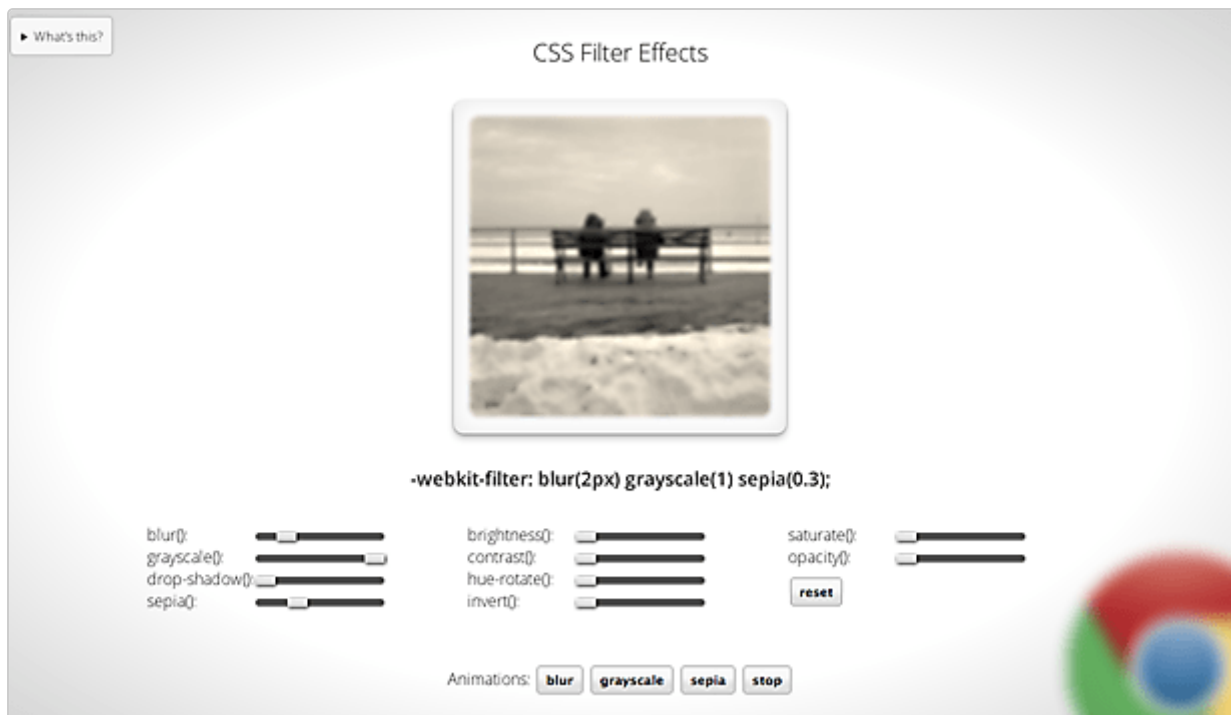
The magic happens with a new prefixed `filter' property in a style rule:

```
/* gray all images by 50% and blur by 10px */
img {
  -webkit-filter: grayscale(0.5) blur(10px);
}
```

Enabling filters directly in CSS means that nearly any DOM element can take advantage of them! Images, `<video>`, `<canvas>`,... you name it. You can even see what the web looks like without glasses.

Demo

[Try it out!](#) (works in Chrome Canary and WebKit nightlies)

Future

The spec also defines [CSS shaders](#), which will eventually bring OpenGL shader technology to CSS. That's very VERY exciting! However, there are security considerations any time you open up the GPU of a system. For this reason, WebKit only has CSS filter functions implemented for now.

Support

Chrome 18.0.976.0 (currently canary), Webkit nightly

In Webkit nightlies, filters can be applied to hardware accelerated content (e.g. `img { -webkit-transform: translateZ(0); }`). In Chrome, filters on accelerated content are still a work in progress (use the `--enable-accelerated-filters` flag). This includes `<video>`, which is accelerated by default.

Last updated July 2, 2018.