

# Expanding what's possible on the Web: Project Fugu



*Web  
Engines  
Hackfest*



Thomas Steiner  
@tomayac

# Have you been there?



This PNG should  
really be an SVG!



<https://raw.githubusercontent.com/lgalia/webengineshackfest/master/2022/img/web-engines-hackfest-logo.png>

# The usual solution

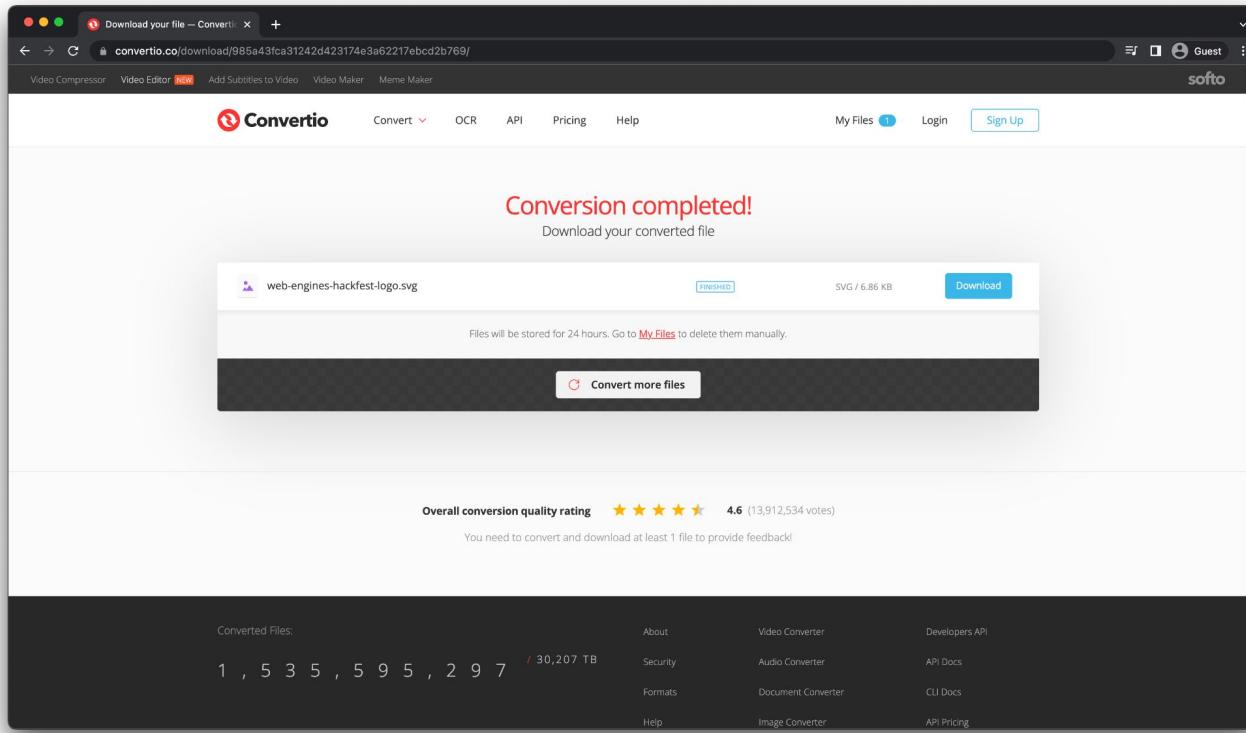
A screenshot of a Google search results page for the query "convert png to svg". The search bar at the top shows the query. Below the search bar, there are navigation links for All, Videos, Images, Shopping, News, and More. The main content area displays several search results:

- https://convertio.co › png-svg**  
PNG to SVG (Online & Free) - Convertio  
Best way to convert your PNG to SVG file in seconds. 100% free, secure and easy to use!  
Convertio — advanced online tool that solving any problems with any ...  
★★★★★ Rating: 4,3 · 214,944 votes
- https://image.online-convert.com › convert-to-svg**  
Online SVG image converter  
If you convert from raster images like PNG or JPG, this SVG converter will convert your shapes and objects to black and white vector graphics that are ...  
★★★★★ Rating: 4,5 · 81,105 votes · Free · Business/Productivity
- https://www.freecsv.com › png-to-svg**  
PNG to SVG - FreeConvert.com  
This PNG to SVG converter lets you create perfect SVG (Scalable Vector Graphics) from PNG images. You can also, adjust smoothness or the number of colors.
- https://www.pngtosvg.com**  
JPG, PNG to SVG Converter - Image to High Quality Color ...  
Are you looking for an SVG Converter that would convert your png and jpg files to SVG? Our online image Vectorizer tool easily converts your images to SVGs.

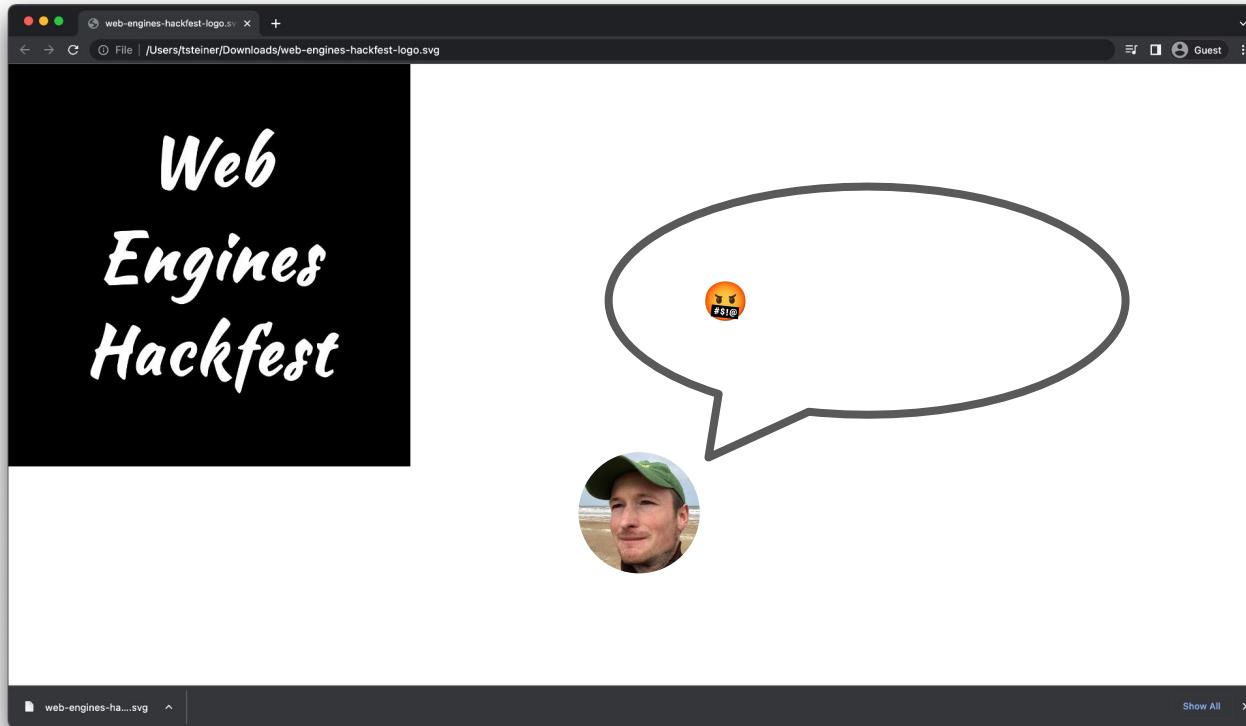
Below the search results, there is a "People also ask" section with two collapsed dropdowns:

- Can you turn a PNG into a SVG?
- How do I convert PNG to SVG for free?

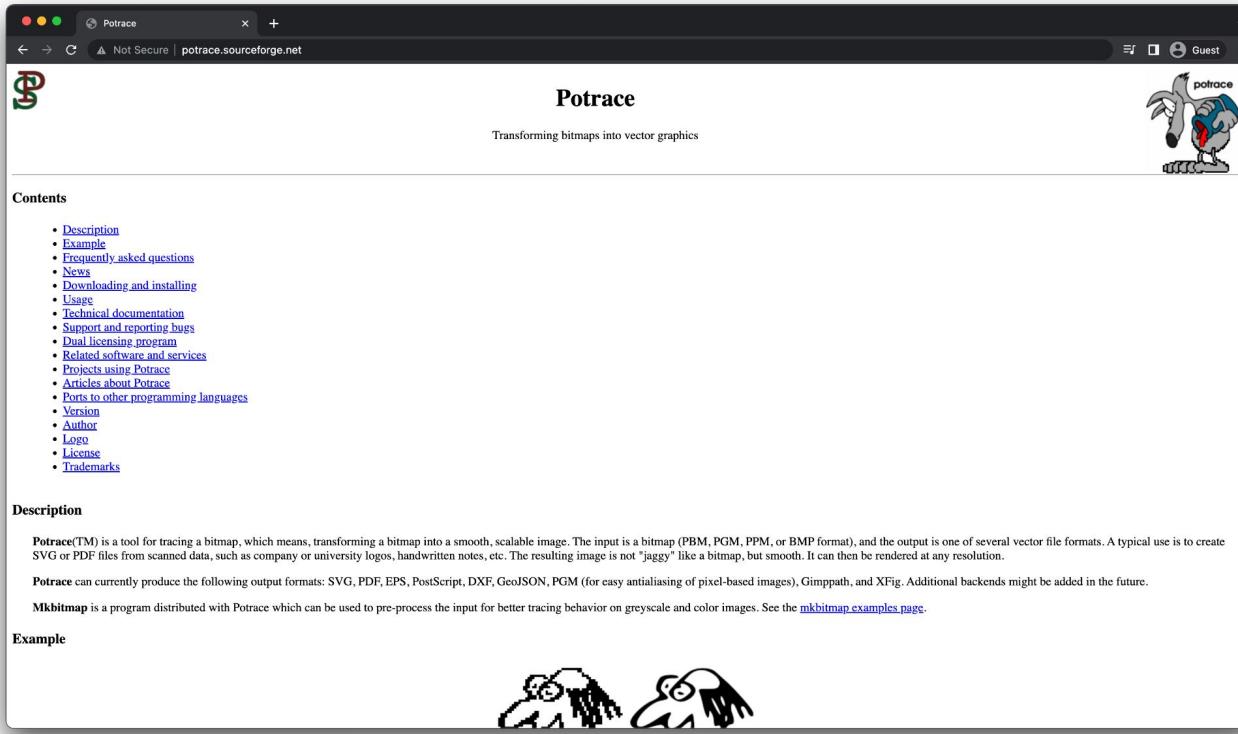
# The usual solution (continued)



# The usual solution (continued)



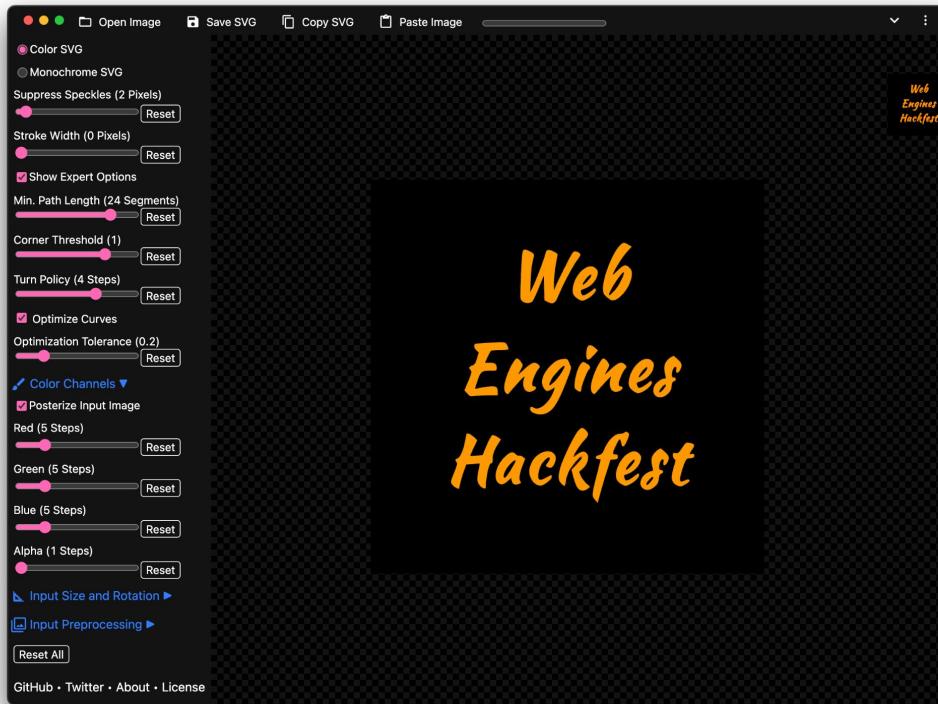
# There's got to be a better way



The screenshot shows a web browser window displaying the official Potrace website at [potrace.sourceforge.net](http://potrace.sourceforge.net). The page has a dark header with the title "Potrace" and a "Not Secure" warning. Below the header is a logo featuring a stylized green and red bird-like creature. The main content area contains the following sections:

- Contents**: A sidebar menu with links to various documentation and support pages.
- Description**: A detailed explanation of what Potrace does, mentioning it's a tool for tracing bitmaps into smooth, scalable vector graphics.
- Potrace**: The main heading for the project.
- Transforming bitmaps into vector graphics**: A subtitle describing the software's function.
- Logo**: An illustration of a cartoonish animal (resembling a donkey or mule) wearing a blue vest with the word "potrace" on it.
- Image**: A small thumbnail image showing the result of a tracing process on a black background.
- Example**: A section showing two versions of a cartoon character's head: a low-resolution, jagged bitmap on the left and a smooth, vectorized version on the right.

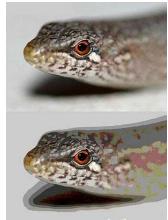
# Introducing SVGcode (svgco.de)



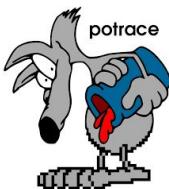
# What SVGcode does differently



Offline-enabled PWA, runs entirely in the client. No server to trust.



Posterizes images and runs potrace per color, then reassembles the fragments for the resulting image.



Exposes all of potrace's options, including expert options.

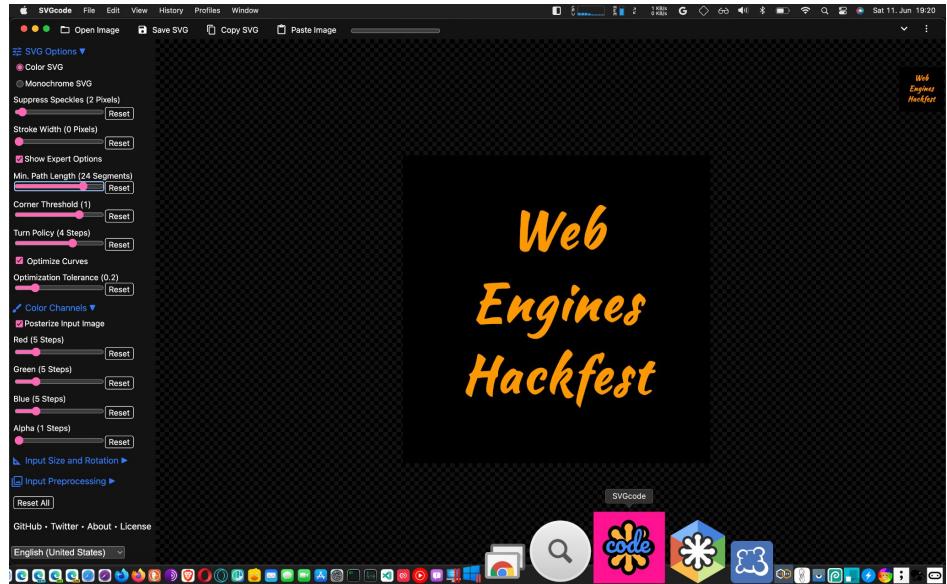


Optimizes SVGs via svgo.

# Installable, offline-enabled PWA

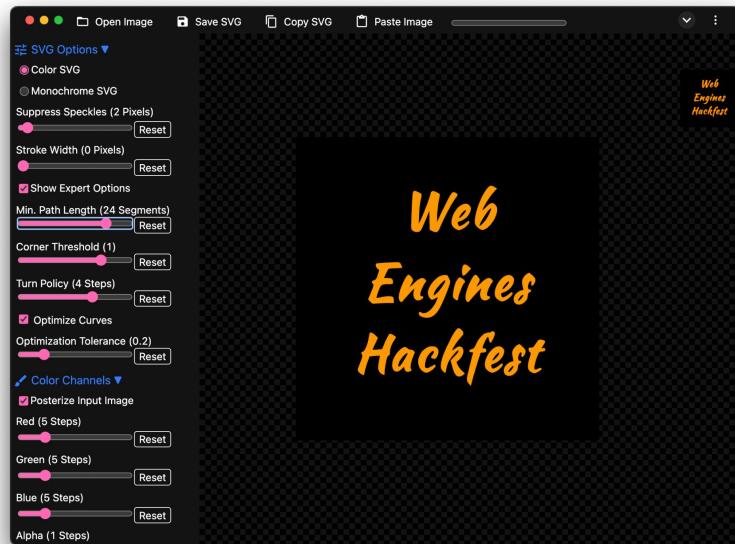
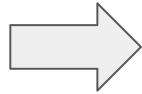
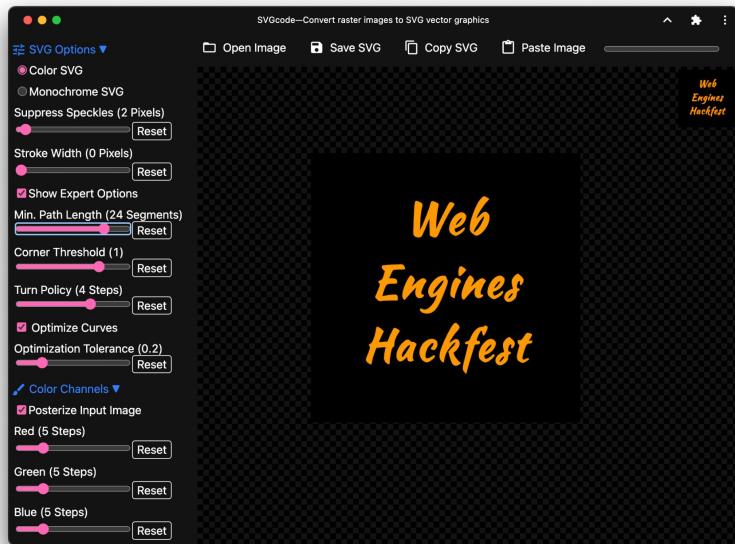
SVGcode is an installable Progressive Web App and therefore fully offline enabled.

The app is based on the [Vanilla JS template](#) for [Vite.js](#) and uses the popular [Vite plugin PWA](#), which creates a service worker that uses [Workbox.js](#) under the hood.



# Window Controls Overlay API

To maximize the available screen real estate, SVGcode uses [Window Controls Overlay](#) customization by moving its main menu up into the titlebar area.



```
const WINDOW_CONTROLS_OVERLAY = 'window-controls-overlay';

const onGeometryChange = () => {
  const visible = navigator.windowControlsOverlay.visible;
  menu.classList.toggle(WINDOW_CONTROLS_OVERLAY, visible);
  main.classList.toggle(WINDOW_CONTROLS_OVERLAY, visible);
};

navigator.windowControlsOverlay.addEventListener(
  'geometrychange',
  debounce(async () => {
    onGeometryChange();
  }, 250),
);
```

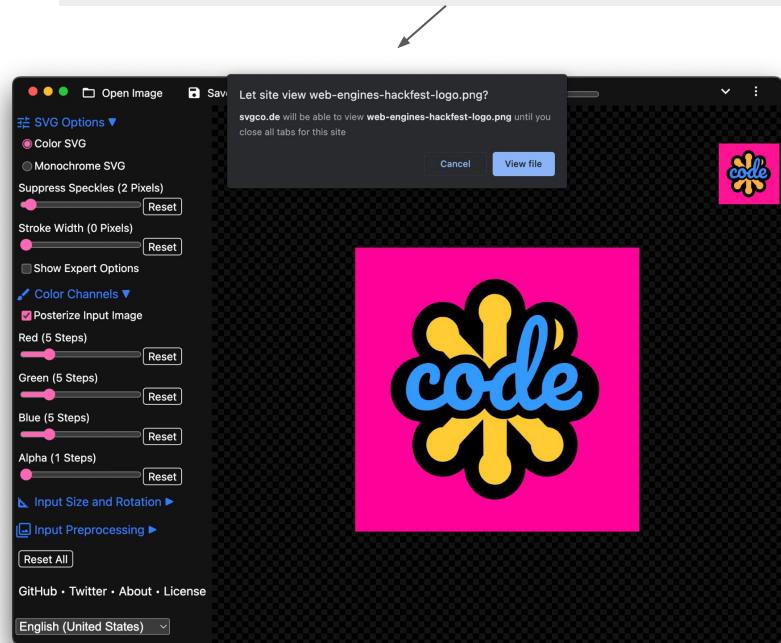
# File System Access API

SVGcode uses the File System Access API via the [browser-fs-access](#) ponyfill.

On supporting browsers, it can store a serialized `FileSystemFileHandle` to let the user continue where they left off after a reload.

This also works with the Drag and Drop API.

Dialog asking if you want to continue with `web-engines-hackfest-logo.png` where you left off.

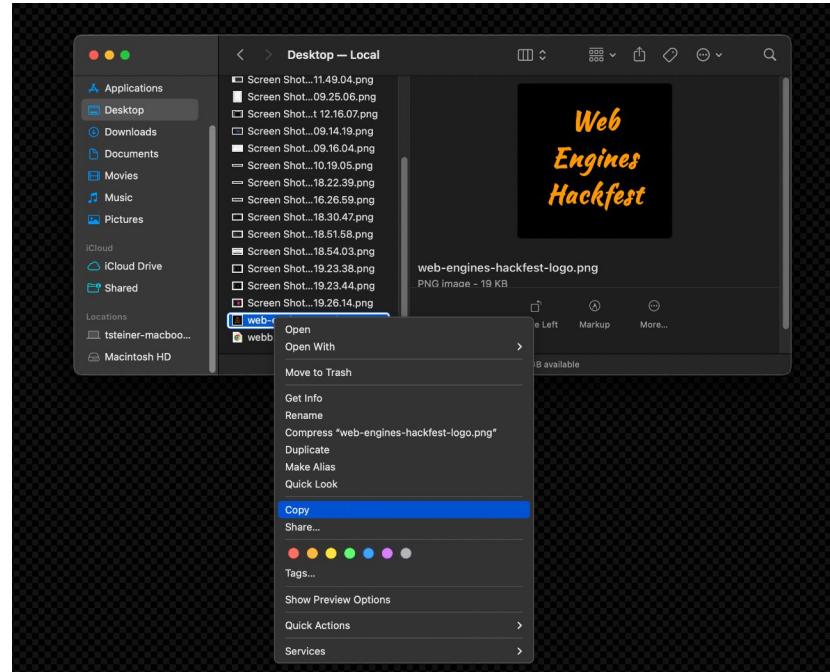


```
document.addEventListener('drop', async (event) => {
  event.preventDefault();
  const item = event.dataTransfer.items[0];
  if (item.kind === 'file') {
    inputImage.addEventListener('load', () => {
      URL.revokeObjectURL(blobURL);
    }, {once: true});
    const handle = await item.getAsFileSystemHandle();
    if (handle.kind !== 'file') return;
    const file = await handle.getFile();
    const blobURL = URL.createObjectURL(file);
    inputImage.src = blobURL;
    await set(FILE_HANDLE, handle);
  }
});
```

# Async Clipboard API

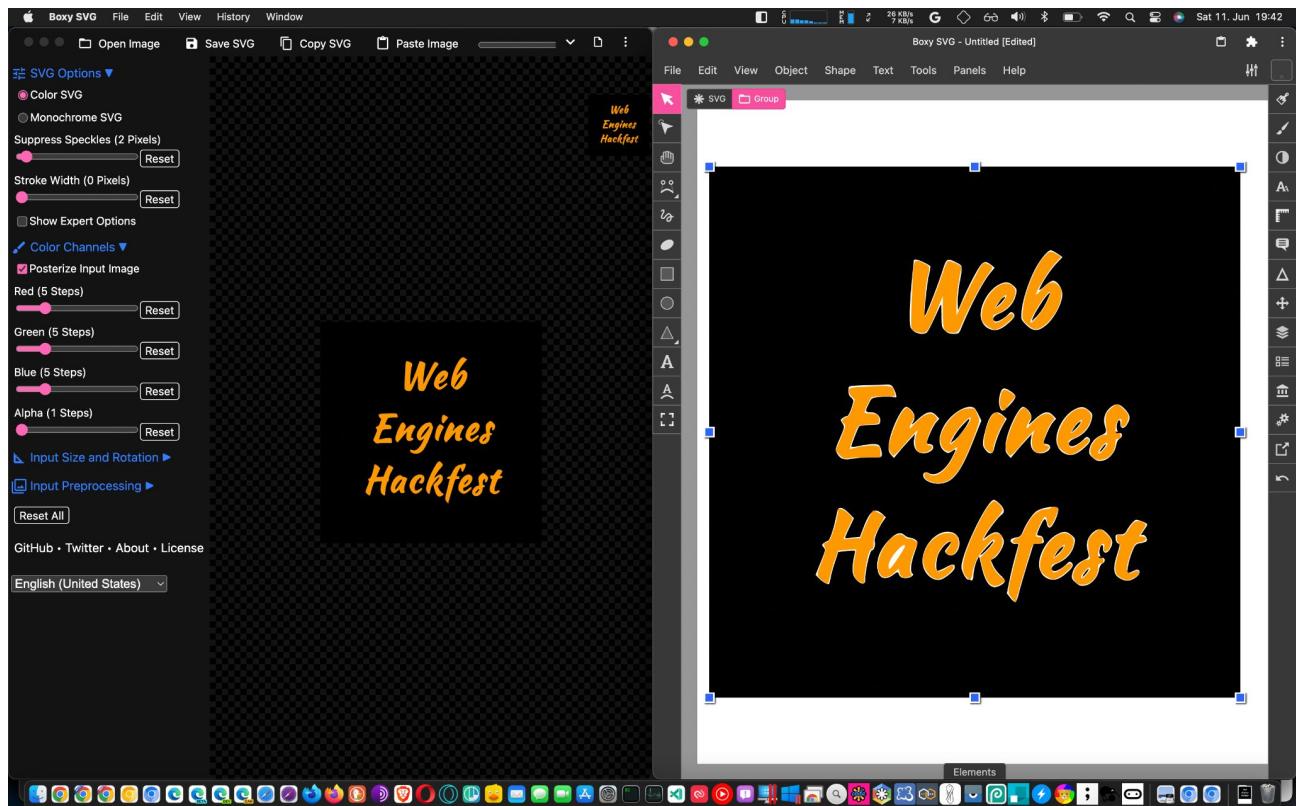
SVGcode is also fully integrated with the operating system's clipboard via the Async Clipboard API.

You can paste images from the operating system's file explorer into the app either by clicking the **Paste Image** button or by pressing **command/control + v** on your keyboard.



# Async Clipboard API (continued)

The Async Clipboard API has recently gained the ability to deal with SVG images as well, so you can also copy an SVG image and paste it into another application for further processing.

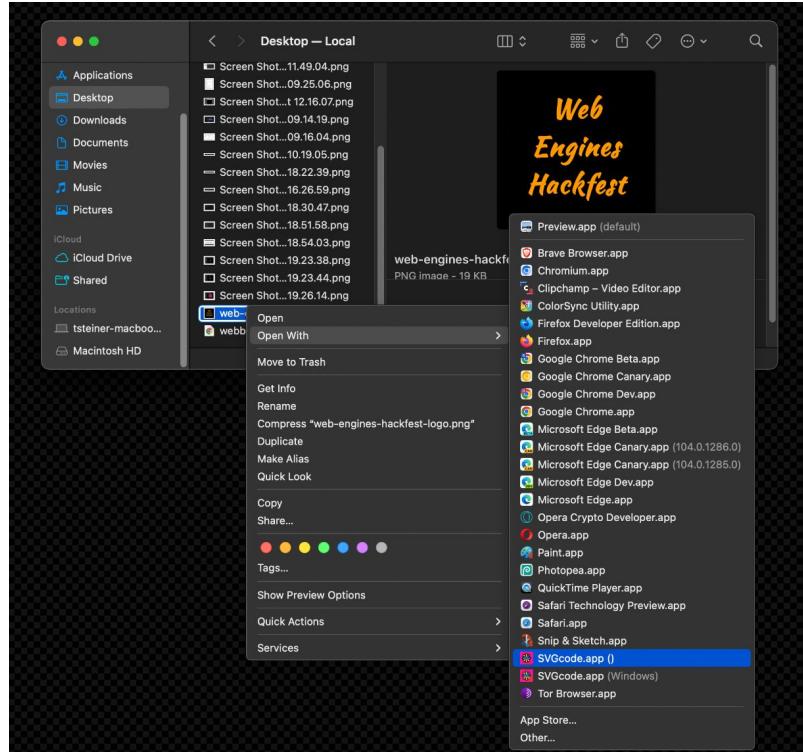


```
copyButton.addEventListener('click', async () => {
  let svg = svgOutput.innerHTML;
  showToast(i18n.t('optimizingSVG'), Infinity);
  svg = await optimizeSVG(svg);
  const textBlob = new Blob([svg], {type: 'text/plain'});
  const svgBlob = new Blob([svg], {type: 'image/svg+xml'});
  navigator.clipboard.write([
    new ClipboardItem({
      [svgBlob.type]: svgBlob,
      [textBlob.type]: textBlob,
    }),
  ]);
  showToast(i18n.t('copiedSVG'));
});
```

# File Handling API

As an installed PWA, SVGcode can become a file handler, or even the default file handler, for image files.

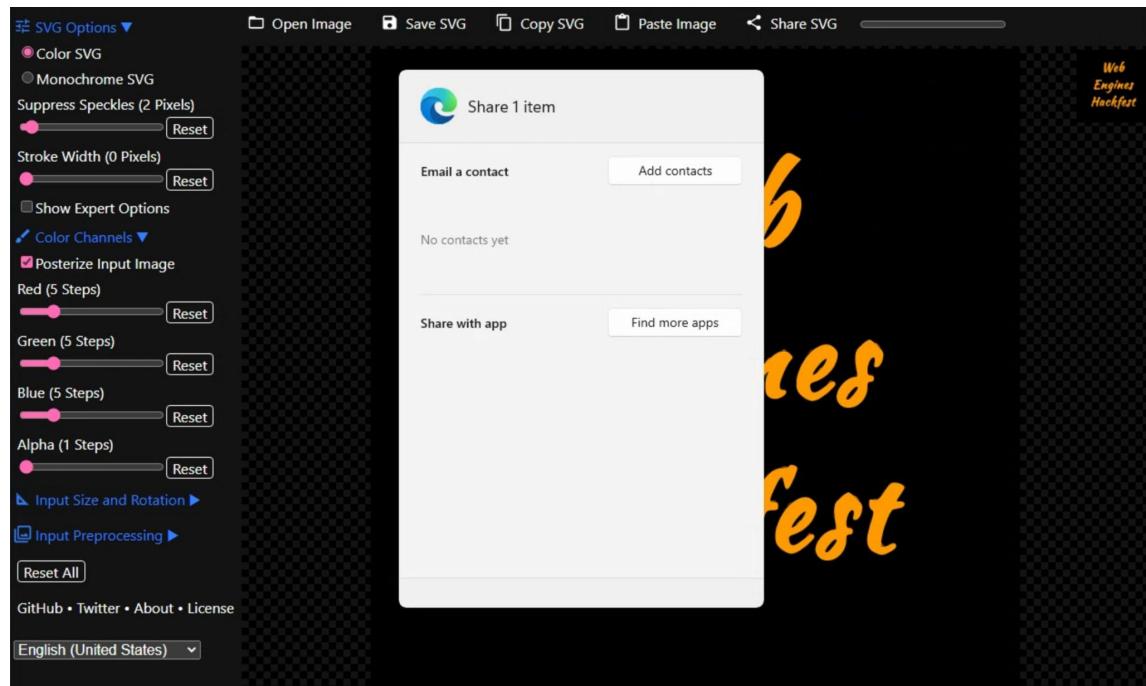
This means image files can be opened in SVGcode right from the operating system's file explorer.



```
window.launchQueue.setConsumer(async (launchParams) => {
  if (!launchParams.files.length) return;
  for (const handle of launchParams.files) {
    const file = await handle.getFile();
    if (file.type.startsWith('image/')) {
      const blobURL = URL.createObjectURL(file);
      inputImage.addEventListener('load', () => {
        URL.revokeObjectURL(blobURL);
      }, {once: true});
      inputImage.src = blobURL;
      await set(FILE_HANDLE, handle);
      return;
    }
  });
});
```

# Web Share API

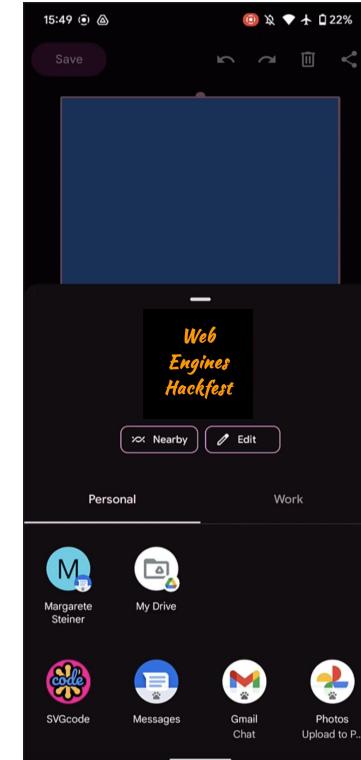
SVGcode can share  
SVGs via the Web Share  
API, for example, if you  
want to send the SVG to  
someone via email.



```
shareSVGButton.addEventListener('click', async () => {
  let svg = svgOutput.innerHTML;
  svg = await optimizeSVG(svg);
  const suggestedFileName = getSuggestedFileName(await get(FILE_HANDLE)) || 'Untitled.svg';
  const file = new File([svg], suggestedFileName, { type: 'image/svg+xml' });
  const data = {
    files: [file],
  };
  if (navigator.canShare(data)) {
    try {
      await navigator.share(data);
    } catch (err) {
      console.error(err.name, err.message);
    }
  }
});
```

# Web Share Target API

In the opposite direction, SVGcode can also serve as a share target, so when you have an image file, you can share it to SVGcode.



```
{
...
"share_target": {
  "action": "https://svgco.de/share-target/",
  "method": "POST",
  "enctype": "multipart/form-data",
  "params": {
    "files": [
      {
        "name": "image",
        "accept": [ "image/jpeg", "image/png", "image/webp", "image/gif" ]
      }
    ]
  }
}
...
}
```

```
self.addEventListener('fetch', (fetchEvent) => {
  if (
    fetchEvent.request.url.endsWith('/share-target/') &&
    fetchEvent.request.method === 'POST'
  ) {
    return fetchEvent.respondWith(
      (async () => {
        const formData = await fetchEvent.request.formData();
        const image = formData.get('image');
        const keys = await caches.keys();
        const mediaCache = await caches.open(
          keys.filter((key) => key.startsWith('media'))[0],
        );
        await mediaCache.put('shared-image', new Response(image));
        return Response.redirect('./?share-target', 303);
      })(),
    );
  }
});
```



# Project Fugu API Showcase

Check out SVGcode and many other amazing apps in the [Project Fugu API Showcase](#).

Filter by Fugu API to see what apps using them exist.

Add not yet listed apps to the showcase.

The screenshot shows a web browser window titled "Project Fugu API showcase - developer.chrome.com". The page displays a grid of Fugu API categories at the top, followed by a search bar and two app cards.

**Boxy SVG**  
Used APIs:  
Add to Home Screen, Async Clipboard, Cache Storage, EyeDropper, File System Access, Local Font Access, Service Worker

**SVGcode**  
Used APIs:  
Add to Home Screen, Async Clipboard (Images), Cache Storage, File Handling, File System Access, Web Share Target (Files), Window Controls Overlay

Below the cards, there is a note about SVGcode and its capabilities.

SVGcode is a Progressive Web App that lets you convert raster images like JPG, PNG, GIF, WebP, AVIF, etc. to vector graphics in SVG format.

# Fugu API Tracker

Keep track of progress the Fugu team makes by regularly looking at the [Fugu API Tracker](#).

See what APIs have shipped, are in origin trial, dev trial, or under consideration.

The screenshot shows the Fugu API Tracker interface. At the top, there are three tabs: STABLE, BETA, and DEV. Below each tab are three browser icons: Chrome, Firefox, and Edge. Under the STABLE tab, it says "Chromium 103" and "Stable in 10 days (Jun 21, 2022)". Under the BETA tab, it says "Chromium 104" and "Stable in 52 days (Aug 2, 2022)". Under the DEV tab, it says "Chromium 105" and "Stable in 80 days (Aug 30, 2022)".

**Shipped #**

Web Bluetooth API	M56	+
WebUSB API	M61	Windows, macOS, iOS, Android
Web Share Target	M71	Android
Web Share API Level 2	M75	Android
Web Share Target Level 2	M76	Android
Async Clipboard: Read and Write Images	M76	Windows, macOS, iOS, Android, Edge
Enter Key Hint	M77	Android
Expand Storage Quota	M78	Windows, macOS, iOS, Android, Edge
Contacts API	M80	Android
Compression codecs	M80	Windows, macOS, iOS, Android, Edge

# Expanding what's possible on the Web!



Thomas Steiner  
@tomayac

*Web  
Engines  
Hackfest*



Slides: [goo.gl/web-engines-hackfest-fugu](https://goo.gl/web-engines-hackfest-fugu)