sips: Scriptable image processing system

I wanted to convert some `.webp` images to `.png` on my Mac. I asked ChatGPT:

> On MacOS use CLI to convert webp images to PNG

And it told me about `sips`:

```
sips -s format png image.webp --out image.png
```

Or to run it against all PNGs in a folder:

```
for file in *.webp; do sips -s format png "$file" --out "${file%.*}.png"; done
```

I had never heard of `sips` before - but apparently it's been a default command on macOS for a very long time.

It stands for "scriptable image processing system". `man sips` starts like this:

```
NAME
     sips – scriptable image processing system.

SYNOPSIS
     sips [image-functions] imagefile ...
     sips [profile-functions] profile ...

DESCRIPTION
     This tool is used to query or modify raster image files and ColorSync ICC
     profiles.  Its functionality can also be used through the "Image Events"
     AppleScript suite.  It also supports executing JavaScript to either
     modify or generate images.
```

It can run JavaScript!

I asked ChatGPT for an example, but it halucinated something that didn't actually work.

After some searching I found [manicmaniac/sips-js-api](#) which is the only place online I could see that documented how to use the `sips --js` option. Here's their example - save this in `smile.js`:

```
const canvas = new Canvas(150, 150)
canvas.beginPath()
canvas.arc(75, 75, 50, 0, Math.PI * 2, true)
canvas.moveTo(110, 75)
canvas.arc(75, 75, 35, 0, Math.PI, false)
canvas.moveTo(65, 65)
canvas.arc(60, 65, 5, 0, Math.PI * 2, true)
canvas.moveTo(95, 65)
canvas.arc(90, 65, 5, 0, Math.PI * 2, true)
canvas.stroke()
const output = new Output(canvas, sips.outputPath)
output.addToQueue()
```

Then run:

```
sips -j smile.js -o smile.png
```

This produces an alpha-transparent PNG of a smiling face.

The only other relevant documentation I could find was this section of the `man sips` page:

```
JavaScript
     HTML Canvas objects can be created and used to create a 2D drawing context.  The commands
for drawing into the context are well documented elsewhere.  This
     section will describe the sips global object and other interesting classes.

     Global variable (sips) properties
     images
           Valid images passed as arguments converted into an array of Image objects
     arguments
           Arguments passed into the program as an array of strings
     size  Recommended size for output. Setting the crop or resample flags will set this value.
     longestEdge
           If specified, the value of the -Z/--resampleHeightWidthMax option. [default: 0]
     outputPath
           Output directory [default: current directory]

     Image Object
     name  Name of image
     size  Size of image (pixels)
     properties
           Image properties
     getProperty(name)
           Return the image property for name, if any.
     sizeToFitLongestEdge(length)
           Return the size that will contain the image with the longest edge set to length.
Maintains aspect ratio.

     Output Object
     new Output(context, name[, type])
           Output the context to disk with name and optional type (extension or UTI).
     addToQueue()
           Adds the output to the queue to be written to disk.

     Functions
     print(str)
           Output to standard output. Equivalent to console.log(str).
```

Related

electron Signing and notarizing an Electron app for distribution using GitHub Actions - 2021-09-08

javascript JavaScript date objects - 2022-01-16

purpleair Calculating the AQI based on the Purple Air API for a sensor - 2021-08-31

awslambda Deploying Python web apps as AWS Lambda functions - 2022-09-18

aws Running OCR against a PDF file with AWS Textract - 2022-06-28

Created 2023-02-18T10:28:00-08:00 · Edit