

# danburzo/percollate

By danburzo

Source: <https://github.com/danburzo/percollate>

Percollate is a command-line tool to turn web pages into beautifully formatted PDFs. See [How it works](#) → [#how-it-works](#).

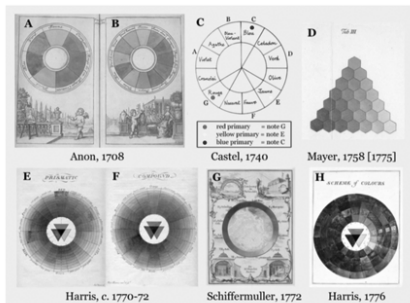


Figure 7.2.1. Eighteenth century hue systems (click to enlarge → <http://www.huevaluechroma.com/misc/7.2.1.1a.jpg>). A, B, Hand-painted colour circles from the 1708 edition of the *Traite de la peinture en miniature*. C, Louis-Bertrand Castel, 1740, *L'Optique des couleurs*, as illustrated by Kemp, 1990. D, Tobias Mayer, 1758, *De affinitate colorum commentatio*, as illustrated by Georg Christoph Lichtenberg in 1775. E, F, Moses Harris, c. 1770-72, *The natural system of colours*. G, Ignaz Schiffermüller, 1772, *Versuch eines Farbensystems*. H, Moses Harris, 1776, *Exposition of English insects* (1782 edn). For details of these publications see Kuehni and Schwaninger (2009) → [http://www.colourchocolate.com/view/01\\_5033a9e9e7a99797919196814914001a5e9e0f4f82935189641](http://www.colourchocolate.com/view/01_5033a9e9e7a99797919196814914001a5e9e0f4f82935189641).

After Newton introduced the circular dimension of hue in his *Opticks* of 1704, it was only a small step to arrange the three hues of the seventeenth century artists' linear scale → <http://www.huevaluechroma.com/071.nhp#agulloni> around a circle. This step was first taken in an anonymous chapter on pastel painting added to the 1708 Hague edition of the anonymous *Traite de la Peinture en Miniature*, in a pair of hand-coloured circles showing seven and twelve equal-sized divisions respectively (Fig. 7.2.1A, B). In

the text the anonymous author hesitates over whether there are really three "primitive" colours or four (yellow, blue and two reds - "fire red" or vermilion, and crimson), but interestingly the primary status of "red" ultimately survives the fact that it had to be mixed here from two pigments. The seven-hue circle shows the four pure pigments, plus three mixtures respectively of yellow, blue and the reds, while the twelve-hue version adds further mixtures, to place yellow, the mixed "primary" red and blue evenly spaced around the circle. In both diagrams the clockwise sequence of hues follows the order of the seventeenth century linear scale: (yellow-red-blue), and so is reversed compared to Newton's spectral or (red-yellow-blue).

This twelve-hue circle is the earliest example of the so-called "art colour wheel", an arrangement of regularly spaced hue divisions structured around the three historical primaries. While it incorporated Newton's discovery that hues form a closed loop, the "artists' colour wheel" was otherwise an implicit rejection of the assumption that Newton's circle and rules of additive mixing (in which all spectral hues "primary") also applied to paints. Despite this early example, published circular systems are few in number until the first decades of the nineteenth century, and in this early period triangular systems such as those of Ma (1757) and Sowerby (1809) are just as prominent. (These triangular systems also arrange hues in a closed loop, but designate colours by their proportions of their yellow, red, and blue "components", rather than by hue as such). Nineteenth century "colour wheels" are very diverse geometrically, and incorporate variously subdivided triangles, hexagon and 6- to 24-pointed stars, either in combination with a circle, or alone (Figs 7.2.2, 7.2.3, 7.2.4).

Several of the systems shown here follow Moses Harris (c. 1770-; in consisting of two or even three diagrams, in order to display hue categories specifically for low-chroma colours, designated as tertiary colours by Field (1817). These low-chroma yellows, reds and blues were respectively known as *olive*, *brown* and *slate* (following Harris) or *citri russet* and *olive* (following Field). This use of the word "tertiary" persi

2

3


*Example spread from the generated PDF of a chapter in Dimensions of Colour* → <http://www.huevaluechroma.com/072.php>; rendered here in black & white for a smaller image file size.

## Table of Contents

- [Installation](#) → [#installation](#)

- [Usage](#) → [#usage](#)
  - [Available commands](#) → [#available-commands](#)
  - [Available options](#) → [#available-options](#)
- [Examples](#) → [#examples](#)
  - [Basic PDF Generation](#) → [#basic-pdf-generation](#)
  - [Custom page size / margins](#) → [#custom-page-size-margins](#)
  - [Using a custom HTML template](#) → [#using-a-custom-html-template](#)
  - [Using a custom CSS stylesheet](#) → [#using-a-custom-css-stylesheet](#)
  - [Customizing the page header / footer](#) → [#customizing-the-page-header-footer](#)
- [How it works](#) → [#how-it-works](#)
- [Troubleshooting](#) → [#troubleshooting](#)
- [Contributing](#) → [#contributing](#)
- [See also](#) → [#see-also](#)

## Installation

 percollate needs Node.js version 8.6.0 or later, as it uses new(ish) JavaScript syntax. If you get *SyntaxError: Unexpected token* errors, check your Node version with `node --version`.

You can install percollate globally:

```
# using npm
npm install -g percollate

# using yarn
yarn global add percollate
```

To keep the package up-to-date, you can run:

```
# using npm, upgrading is the same command as
installing
```

```
npm install -g percollate
```

```
# yarn has a separate command
```

```
yarn global upgrade --latest percollate
```

## Usage



Run `percollate --help` for a list of available commands.

For a particular command, `percollate <command> --help` lists all available options.

## Available commands

Command	What it does
<code>percollate pdf</code>	Bundles one or more web pages into a PDF
<code>percollate epub</code>	<i>Not implemented yet</i> → <a href="https://github.com/danburzo/percollate/issues/8">https://github.com/danburzo/percollate/issues/8</a>
<code>percollate html</code>	<i>Not implemented yet</i> → <a href="https://github.com/danburzo/percollate/issues/7">https://github.com/danburzo/percollate/issues/7</a>

## Available options

The `pdf`, `epub`, and `html` commands have these options:

Option	What it does
<code>-o, --output</code>	The path of the resulting bundle; when omitted, we derive the output file name from the title of the web page.
<code>--individual</code>	Export each web page as an individual file.
<code>--template</code>	Path to a custom HTML template
<code>--style</code>	Path to a custom CSS
<code>--css</code>	Additional CSS styles you can pass from the command-line to override the default/custom stylesheet styles

## Examples

### Basic PDF generation

To transform a single web page to PDF:

```
percollate pdf --output some.pdf https://example.com
```

To bundle *several* web pages into a single PDF, specify them as separate arguments to the command:

```
percollate pdf --output some.pdf
https://example.com/page1 https://example.com/page2
```

You can use common Unix commands and keep the list of URLs in a newline-delimited text file:

```
cat urls.txt | xargs percollate pdf --output
some.pdf
```

To transform several web pages into individual PDF files at once, use the `--individual` flag:

```
percollate pdf --individual --output some.pdf
https://example.com/page1 https://example.com/page2
```

## Custom page size / margins

The default page size is A5 (portrait). You can use the `--css` option to override it using any supported CSS size → <https://www.w3.org/TR/css3-page/#page-size>:

```
percollate pdf --output some.pdf --css "@page {
size: A3 landscape }" http://example.com
```

Similarly, you can define:

- custom margins: `@page { margin: 0 }`
- the base font size: `html { font-size: 10pt }`

or, for that matter, any other style defined in the default / custom stylesheet.

## Using a custom HTML template

! TODO add example here

## Using a custom CSS stylesheet

! TODO add example here

## Customizing the page header / footer

! TODO add example here

## How it works

1. Fetch the page(s) using `got` → <https://github.com/sindresorhus/got>
2. Enhance → <https://github.com/danburzo/percollate/blob/master/src/enhancements.js> the DOM using `jsdom` → <https://github.com/jsdom/jsdom>
3. Pass the DOM through `mozilla/readability` → <https://github.com/mozilla/readability> to strip unnecessary elements
4. Apply the HTML template → <https://github.com/danburzo/percollate/blob/master/templates/default.html> and the print stylesheet → <https://github.com/danburzo/percollate/blob/master/templates/default.css> to the resulting HTML
5. Use `puppeteer` → <https://github.com/GoogleChrome/puppeteer> to generate a PDF from the page

## Troubleshooting

On some Linux machines you'll need to install a few more Chrome dependencies → <https://github.com/GoogleChrome/puppeteer/blob/master/docs/troubleshooting.md#chrome-headless-doesnt-launch> before `percollate` works correctly. (Thanks to [@ptica](https://github.com/danburzo/percollate/issues/19#issuecomment-428496041) for *sorting it out* → <https://github.com/danburzo/percollate/issues/19#issuecomment-428496041>)

The `percollate pdf` command supports the `--no-sandbox` Puppeteer flag, but make sure you're aware of the implications → <https://github.com/GoogleChrome/puppeteer/blob/master/docs/troubleshooting.md#chrome-headless-fails-due-to-sandbox-issues> before disabling the sandbox.

## Contributing

Contributions of all kinds are welcome! See CONTRIBUTING.md → <https://github.com/danburzo/percollate/blob/master/CONTRIBUTING.md> for details.

## See also

Here are some other projects to check out if you're interested in building books using the browser:

- [weasyprint](https://github.com/Kozea/WeasyPrint) → <https://github.com/Kozea/WeasyPrint> ([website](https://weasyprint.org/) → <https://weasyprint.org/>)
- [bindery.js](https://github.com/evnbr/bindery) → <https://github.com/evnbr/bindery>. ([website](https://evanbrooks.info/bindery/) → <https://evanbrooks.info/bindery/>)
- [HummusJS](https://github.com/galkahana/HummusJS) → <https://github.com/galkahana/HummusJS>
- [Editoria](https://gitlab.coko.foundation/editoria/editoria) → <https://gitlab.coko.foundation/editoria/editoria> ([website](https://editoria.pub/) → <https://editoria.pub/>)
- [pagedjs](https://gitlab.pagedmedia.org/tools/pagedjs) → <https://gitlab.pagedmedia.org/tools/pagedjs> ([article](https://www.pagedmedia.org/pagedjs-sneak-peeks/) → <https://www.pagedmedia.org/pagedjs-sneak-peeks/>)