

webtreemap

New 2017-Oct-16: master is now webtreemap v2, a complete rewrite with bug fixes, more features, and a different (simpler) API. If you're looking for the old webtreemap, see the [v1](#) branch.

A simple treemap implementation using web technologies (DOM nodes, CSS styling and transitions) rather than a big canvas/svg/plugin. It's usable as a library as part of a larger web app, but it also includes a command-line app that dumps a self-contained HTML file that displays a map.

Play with a [demo](#).

Usage

Web

The data format is a tree of `Node`, where each node is an object in the shape described at the top of [tree.ts](#).

```
<script src='webtreemap.js'></script>
<script>
// Container must have its own width/height.
const container = document.getElementById('myContainer');
// See typings for full API definition.
webtreemap.render(container, data, options);
```

Command line

```
$ webtreemap -o output_file < my_data
```

Command line data format is space-separated lines of "size path", where size is a number and path is a '/'-delimited path. This is exactly the output produced by `du`, so this works:

```
$ du -ab some_path | webtreemap -o out.html
```

But note that there's nothing file-system-specific about the data format -- it just uses slash as a nesting delimiter.

Development

Web piece

Use `npm run dev` to bring up file watchers that keep the demo JS bundle up to date. Then load `demo/demo.html` in a browser. The file generated by `npm run dev` is also used by the command line app.

Command line app

Use `tsc -w` to keep the npm-compatible JS up to date, then run e.g.:

```
$ du -ab node_modules/ | node build/cli.js --title 'node_modules usage' -o demo.html
```