Plans for this blog in 2022

Rather than write about whatever catches my eye, I want to be a little more disciplined about what I write. These are some of the things I plan to do.

If there's something that particularly catches my eyes, I may write about it and either include it in the regular Monday and Wednesday post or put it in the schedule for a Friday Special... no guarantees either way.

Course Builder

This idea looks for a way to programmatically build courses packages to use either in an LMSs or as static content.

It may be a good idea to base this on the static site generator code and then also build codelabs where appropriate.

This may also be something I want to do in a language other than Javascript.

Editor (again)

I started working on an editor based in Monaco, but it was not working well or I couldn't figure out how to make it work.

Decide on format

Is it worth adding the complexity of an Electron app or should I just use a locally hosted web app or PWA?

Using a different codebasse

Rather than using Monaco, I decided to use TinyMCE as the basis and build from there to experiment with creating an editor that I could use.

Package as an Electron app

It'll be interesting to see if it can be packaged as an electron app.

Web Components again

Web Components have been around for a while. From Polymer to Lit and everything in between has changed.

Every so often I've documented the status of web components and where they are in terms of usability, performance and accessibility.

For this year, I want to work with two different Web Component libraries:

- Lit
- Lightning Components

I also want to continue working on the evolution of web components. Things like slots, shadow parts, using ESMoodules to import components and how do new things like constructable styles works in web components. I've written about some of these (modules and constructable styles) but I want to bring them together with web components.

As for a practical application, I want to build patterns based on web components and integrate them into the pattern library I'm working on.

Finishing up writing-related projects

There are three projects that still need some love to see them to completion:

- My starter kit
- The standalone HTML converter
- My static site generator.

The starter kit and the generator need to add Markdown-it plugins to the workflow so I can do things like definition lists, figures instead of images and do it without having to type HTML.

The Static site generator also needs additional templates to be able to generate a static site.

Finally, the static site generator needs a template to generate a page from a list of file names.

Implement a PostCSS workflow

One of the things I want to do is to develop and implement a PostCSS based workflow and move away from SASS once and for all. The work has already started but there are a lot of details to still work out.

Languages other than Javascript in the context of WebAssembly

Awesome WebAssembly Languages gives you a list of languages that can be compiled to WebAssembly bytecode and Awesome Wasm gives you a list of tools and libraries that I make use of WebAssembly or produce WebAssembly byytecode.

> Specific languages I want to work with.

- Rust and wasm-pack
- C/C++
- Go

There may be others.

Explore functional programming

I've always been curious about functional programming and how it can be used to solve problems, particularly in the web development space.

> Languages I would like to take a closer look at:

- Common Lisp
- Clojure
- Racket
- Javascript
    - lodash/fp – An instance of Lodash with its methods wrapped to produce immutable, auto-curried, iteratee-first, data-last methods
    - functional.js – A lightweight functional JavaScript library that facilitates currying and point-free / tacit

Reevaluate Gutenberg

As much as I hate to admit it, Gutenberg has moved faster and in better directions thatn I expected them to. It may be time to revisit whether it's a good idea to use it or whether I should stay in my current pre-Gutenberg theme.

There are two areas I want to pay particular attention to:

- Have the components improved? A few years ago I filed an issue because I couldn't insert a list inside a block quote. Have issues like this been resolved?
- Has the barrier to entry been lowered or removed? As a complete beginner, React it's not an easy thing to learn or master. Has this changed?

Build a block-based theme

Since now you can build a theme entire from blocks, using Javascript and a minimal amount of PHP it may be time to build a theme from blocks to see if it's worth it.

The idea is that the theme will mirror my existing `rivendellweb` theme (code on Github, live on The Publishing Project) at a minimum.

Build templates for gutenberg components and Gutenberg related-code

I have a lot of examples of Gutenberg blogs and other WordPress elements. It may be a good idea to consolidate them into one package and write about them as they are published.

Perhaps this would be another good concept for a monorepo.

Streams are your friends

From what I read, streams are a nice way to work with files in Node and Javascript. I want to research what they are and how I can leverage them in my work.

The hardest part (for me) is to find projects where it makes sense to use them.

Finish design pattern library

I'm working on finishing up a pattern library. It's holding on the PostCSS workflow I'm working on separately.

As soon as I have a working PostCSS workflow, I'll be able to start building components based on the rivendellweb theme as well as other independent ideas I've been thinking about.

More on Web Workers

Web Workers are a good way to offload heavy work to a separate thread. I want to further explore how they would work on different projects and use libraries like Comlink to make them easier to work with.

WordPress Vue Components

I have a set of Vue components that render parts of a WordPress site. I want to separate the components into individual single-file components as I migrate them to Vue 3.

There are some components that are sill work in progress like pagination. I want to finish those and move them to a separate repo.

React and Vue

I don't like React or Frameworks in general, since I'm also exploring Gutenberg and considering moving forward with it, I'm going to take React more seriously than I have in the past.

I have existing code in Vue 2 that I would like to migrate to Vue 3 and see how much better it works.

AI and Machine Learning

It's been a while since I've been hearing about AI and Machine Learning. The challenge, as with many other things is how to incorporate them into my projects.

Some thing that I would like to see if they are possible:

- Handwritten text recognition: Can I get a computer to recognize my handwritten notes and turn them into text?
- OCR for scanned documents: Can I get a computer to recognize text from a scanned document?
- Image recognition: Can an ML model recongize images? How much training does it require and how expensive does it get?

Related questions:

- Will the libraries work as web assembly?
- how much will it cost if I decide to go through Google Clour or any other cloud provider?
- Will any of these projects run on a web browser?