

Making the postmodern web

About

- Presenter: Daniel F. Dickinson
 - Currently works on his two websites built from scratch
 - A small number of other past web projects
 - More of a software/firmware developer, DevOps specialist, constant thinker, tinkerer, and would-be writer (non-fiction)
- MPL MakerPlace
 - ‘This is the place’ to explore technology in Midland
 - 3D Printing, laser cutting, video and audio creation, old school physical making, and the digital world (like today’s topic).

Technology / organization

- Technology
 - HTML, CSS, JavaScript
 - Images/videos
 - Accessibility
 - Security
- Who is doing the work
 - Team or individual
 - Technical skill and/or time
 - Budget
- SEO (Search Engine Optimization)
 - Metadata / microformats
 - Structured data
 - Page load speed
 - Security
 - Lighthouse
 - Time & content
 - Cheating tends to backfire

Law, art, and social

- Legal (topics to consider)
 - Accessibility
 - Privacy
 - Intellectual property
 - Expression vs defamation, libel, etc
 - What is your 'reach' (international law)
 - “You have freedom of expression until you are heard”
- Audience
 - For yourself, audience, or customer?
- Aesthetics
 - Graphics design
- Social
 - Values
 - Taste
 - Expectations

Technology

The web is polyglot

- While the main part of most websites is based on the core language of the web (HTML), websites are built using multiple computer languages and technologies.
- This presentation will briefly touch on three of these languages[1] (HTML, CSS, JavaScript) and related microformats (Open Graph, Schema.org, microformats.org).
- Resources for learning more will be provided.
- Remember to start small and simple, and/or tweaking existing.
- Another important aspect of building for the web is ‘search engine optimization’ (aka SEO) (which is about how to be found).

HTML example

- View it live
- *Italicized teal text is optional.*
- Optional text helps the browser properly interpret the 'encoding' and language / locale of the content.
- Only `<body>` text appears in the browser pane
- `<head>` `<title>` is the tab title

```
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4.     <meta charset="utf-8" />
5.     <title>There's no one here but us
   mice...</title>
6. </head>
7. <body>
8.     <h1>There's no one here but us
   mice...</h1>
9. </body>
10.</html>
```

Semantic HTML

- HTML 5 introduced the notion of ‘semantic’ markup which is intended to make it easier for machines (including screen readers) to ‘understand’ your site
- This includes putting the main part of the site inside a `<main>` element, and within that, any article part inside an `<article>` element, additional information in an `<aside>`, and other sections in `<section>` elements.
- See definition, HTML overview, and relation to accessibility.

CSS example

- Usually an HTML page uses an element to pull in the CSS.
- Can also include as a `<style>` element directly in the HTML
- The mice page with CSS
- Take a look at the page source to see how it's done.
- CSS overview

```
h1 {  
    border-bottom: 5px ridge;  
    border-top: 5px ridge;  
    margin-top: 3rem;  
    padding: 2rem;  
}
```

JavaScript example

- Usually an HTML page uses an element to pull in the CSS.
- Can also include as a `<script>` element directly in the HTML
- Mice page with JavaScript
- Use the source, Luke
- JavaScript overview
- Keep JS to the necessary

```
function noonehere() {  
    var h1element =  
    document.getElementById("noon  
ehere")  
  
    h1element.innerHTML =  
    "There's no one here."  
  
}
```

```
<h1 id="noonehere"  
onclick="noonehere()">
```

Accessibility aka a11y

- Vision: contrast, colour-impairment, font size and spacing
- Vision with screen readers / braille 'display'
- Can your reader understand you with less than a PhD?
- Images: 'alt' text and universal design
- Navigation: Not breaking the back button and more (especially for less tech savvy viewers)
- Videos: subtitles / DV (descriptive video)
- PDF: use a text-based not image-based PDF
- Test it and include testing from those barriers

Security

- Always use HTTPS; if your provider doesn't offer at least a free "Let's Encrypt" certificate, move your hosting
- DNSSEC: This is harder; I'm still working finding hosting that supports this as I wish (especially without additional fees).
- CSP (Cross-site Scripting Policy): Not so much a factor for me, yet, as I don't have any JavaScript on my site, but it is important to avoid XSS (Cross site scripting vulnerabilities), among others.
- If you're using a CMS: strong passwords and Multi-factor Authentication are a must (seriously).
- **Keep your hosting up to date!**

Who is doing the work

Team vs. individual

- Usually this means a smaller number of individuals handle the technology, while the 'content' is written / designed by others
- If the writers / designers need to be able to add content directly, then you need a CMS (Content Management System).
- If the content goes through the technical person/team then you may not need a web-facing CMS as the technical person/team would be responsible for getting the content published
- Of course if your writers / designers can access to publish then, if they get phished or compromised, so can an attacker.

Available technical skills / time

- If the team is small and main purpose is not building for the web, then designing and maintaining your own theme is probably going to take too much time.
- If contracting out theme design, avoid a 'one-off' relationship as there **will** be times you need updates.
- If you are using an third party theme, and don't have a lot of technical expertise, then consider a *managed* web solution (e.g. where someone else takes care of the updates etc) and just write/design.
- If you have technical skills then a Static Site Generator (SSG) is ideal.
- If you also have time then your own theme can be worthwhile.

More control takes more coding

- There are many no-code / low-code website building options out there—see CIRA's CMS (Website builder) speed dating blog post for examples.
- Ultimately these let you build a site quickly, but in most cases you are limited to the 'building blocks' the provider gives you.
- In the cases where a provider allows you custom 'building blocks', technical skills in HTML, CSS, JavaScript will be important.
- Sites built by a website builder tend to have issues with page load speeds, which can negatively affect your search rankings.

Static site generators (SSG)

- Benefits of static generators from the Hugo website
- There are many choices; see the top starred ones on GitHub.
- In addition to avoiding a database dependency:
 - Speed
 - Security
 - Can review / validate files/sites before you publish
- Can use JavaScript and/or non-static servers for dynamic elements like forms, payments/stores.
 - Jamstack is a popular method of doing this.

CMS based on SSG

- Cloud Cannon
- Netlify CMS
- Tina Cloud
- Quite useful to reduce the required technical skills required to use a static site generator
- Like a managed dynamic CMS generally cost more than managing yourself, but saves you time and effort.

SEO

Search Engine Optimization

Page load speeds

- While not the single most important factor for your search engine ranking, it is a significant factor.
- Slow loading **is** likely the first thing your visitor will complain about on an otherwise interesting / well designed site.
- By complain I mean to themselves; chances of getting feedback other than lost eyeballs due to page load speeds is unlikely.
- Fortunately there are many techniques for improving load speeds, especially for the part 'above the fold' (i.e. what first appears).

Metadata / microformats / structured data

- This is ‘stuff’ that is not displayed on the page, but is designed for search engines, and other automated processing, to ‘understand’ what is on your site.
- It also includes things like ‘Twitter cards’ and ‘Open Graph Protocol’ images and text. These are used by other sites (like Twitter) to create the graphic and text that users on those sites will see when someone includes a link to your site.
- There is also a type of ‘microformat’ that is embedded in the main web page that is designed to help automated processing.

Lighthouse / web.dev

- Testing framework like Google uses as part of their 'scoring' of websites (especially the technical aspects).
- Four topics: Performance, Accessibility, Best Practices, and SEO
- Lighthouse can be used locally from the Chrome or Edge web developer tools (F12).
- web.dev runs against live sites such as this report on my tech site
- Also available for use as an API (e.g. for automated testing before going live).

Content is King

- More than any 'tricks' for SEO content matters
- Also, time, and the snowball effect of getting noticed (and linked to by real sites, and people)
- Also, the web is huge and takes a while for engines to fully index
- Search engines are constantly updating the detection of cheating and penalize those who do
- SEO 'hustlers' will often promote your site in ways hurt you more than they help as being linked to can actually degrade your rating, depending on the quality of the linking sites.

Law, art, and social

Inclusiveness and expression

- Accessibility; the web should be for everyone
 - There is a lot of low hanging fruit, if you are willing to put in a little time
- Freedom of expression vs bigotry et al.
- Freedom of expression vs defamation, libel, etc.
- “You have freedom of expression until you are make a difference”;
 - SLAPP lawsuits
 - Copyright and intellectual property as a silencer
- Safe harbour or “I’m just a host/provider” defence likely to be reduced or eliminated due to social media failure to self-regulate.

Privacy and your 'reach'

- EU is coming down hard on things like Google Analytics
- FTC is making noises about dealing with privacy issues
- Some US states (like California) have also strengthened regulation around privacy
- Depending on who your customer is (for a business), or whether you considered to be subject to a foreign country's jurisdiction is messy. Especially for privacy laws it is better to play it safe.
- Also consider CASL and other anti-spam legislation if you want to market to your visitors (e.g. newsletters, etc)

Intellectual property

- Many countries have agreements such that copyright and trademarks in one country apply to many other countries as well (with the notable exception of countries like China).
- It's generally civil legislation which means that if someone is violating your copyright you have to find out about them, and take legal action yourself.
- However, search engines and web hosting companies usually have a 'notice and takedown' policy (if a site is alleged to be in violation, the site owner is given a chance to respond, depending on response the site may be removed from listing or taken down).

Audience, aesthetics, taste, etc

- There are two aspects to what you can show on a website, aside from the legal aspects
- How the general public will react (content and/or how 'shocking')
- How your 'audience' will react (content and aesthetics)
 - But first you have to identify your audience (or if it is mostly for yourself, and others can like or not as they feel inclined)
 - For example, if your audience prefers minimalism, a flashy and/or emoji-filled site is probably not what you want.
- Graphics design principles generally hold regardless

End notes

- [1] We use the term language very loosely in this presentation (for example CSS is not really a language in pure computer science terms).