Diamond Dust Writers Process Documentation

Table of Contents

---

1. On creating a baseline

Before I began coding this project, my client, the Diamond Dust Writers (DDW), and I developed our project plan. We discussed why they were interested in a website and what features were needed. We also discussed the audience that would visit the site. Having completed the draft of the static site, I can say that the product is far from meeting all of the goals we identified in our project plan.

This is not because the site is incomplete (if a website can ever really be complete), but because my client and I decided that an iterative approach would be more practical than assuming that our project plan captured every need. Since my client has never had a website before, there are certain to be "things we can't live without that we never knew we needed". The result of an iterative approach would be a site that could more easily evolve with the project plan. The current static site is, therefore, a baseline that meets all of my clients needs; the design is intended to be basic, but flexible.

2. On research, Redish, Williamson, and HTML5

I began my project by researching sites with a similar purpose to my client's. I noted what features were most important to my client, as well as those that common to the genre of site I was designing. I found Redish's discussion about conversation helpful during this phase. Looking at site features from Redish's perspective gave me a better understanding of how they should be implemented. For example, many sites post details of upcoming events. This is information that many visitors go to a site specifically to find. Yet, many sites bury this information two or three mouse clicks deep, creating layers of work for the visitor. A more conversational approach would be to display the information upfront, but a way that is unobtrusive.

After matching my list of features with my client's goals for the site, I started to look at source code to see if I could understand how different features were built. I downloaded several free templates and explored their code in the Atom editor. I also used the Developer Tools in Chrome and Firefox to understand the relationship between different elements and their styles.

During this phase I started looking broadly at how I would solve some of the challenges that I already knew I would run into. For example, it is important to my client that visitors be able to content them through the site. However, the group does not have a phone number, address, or email that I could list as contact information. A contact form would be an effective solation, yet a typical email contact form relies on PHP-- something not available on a GitHub-hosted site.

Looking at source for existing sites and thinking about some of these technical challenges gave me enough information to sketch out the structure of my client's site. I wanted to combine features that I saw consistently in my site's genre, those of a blog and a portfolio. I wanted to look to be minimal with a balance of white space and text. I also wanted the site to be light-weight so that it could be easily moved or rewritten for another platform that offered more functionality than static html. To do this, I designed

the page as a single html document with in-page anchor links. This allows a visitor to navigate the site using the navigation menu at the top or to simply scroll down to view additional content. All of the site content (except images) is contained within a single file.

As I began actually coding on my client's site, I realized just how much has changed since the last time I created a website from scratch. I really appreciated the Lynda tutorial by Williamson when he talks about the differences between HTML4 and HTML5. Using tags to organize information instead of just using them to display elements (as we did throughout the 1990's) makes html so much more versatile. I was once one of those that Williamson refers to who used heading tags based on the size of the resulting text, rather than on the structure of the document!

3. On tools and technical challenges

While doing the work of coding, I found I did not refer back to Redish or Williamson. The one exception was when I went back to the Lynda.com exercises to see how he managed the spacing on his pages. I found that my work had more flow if I opened a new browser tab and "Google'd" the problem I ran into as it happened. Most of my questions were readily answered. I used several tutorials on the W3C Schools (https://www.w3schools.com/) website (https://www.w3schools.com). Deconstructing the example code snippets led to a better understanding. I also appreciated that their examples contained inline styles, which let me see how a change in style affected the element without switching between two separate files.

In a few cases, I found neither the W3C Schools nor the Mozilla Developer Network provided a helpful example. One of these was the use of the <Section> element. Neither site provided context to how it was used in comparison to similar elements, such as <Article>. I eventually came to an understanding by reading several blog posts. I would not call any of the blogs academically creditable, but in combination, they provided the context I was missing. (Although Williamson discusses this in his tutorial, he talks through it rather quickly.)

One site element that took more time than I anticipated was the banner image. The image is a photograph of my client, the Diamond Dust Writers. I used Infinite Painter on my Android tablet to manipulate the original photograph. I am familiar with the application because I use it for my own digital art. However, I found that I needed to use another application to reduce both the file size and the canvas size of the image. Even after uploading the image and displaying it on the site, I had to make additional adjustments to the size and scale using CSS so that the image was displayed in correct relationship to the page size and the navigation bar. Since I knew that Chrome, Firefox, and Opera would each interpret the site differently, I tested each adjustment in all three browsers. Minor adjustment like this one were a repetition of trial and error.

The photograph I used for the banner image also provided the color palette used for the site. The website Canva (https://www.canva.com/) (https://www.canva.com/) allows you to upload an image to use as the base color swatch. I wanted a minimal design that focused attention on text (my clients are writers), so the limited color palette provided by Canva worked well for my client.

I experimented with features before deciding that they were not right for the site or the client. In particular, I wanted a way to display information about upcoming events in such a way that the visitor did not have to click a link in order to find. I trialed variants of a floating text box that sat over or at the edge of the main body of text. This was done using the CSS attributes of *position* and *z-index*. After several attempts, I decided that this was distracting to the reader and it was left out.

3.1. Client goals vs. technical challenges

In some cases, technical challenges were presented by the goals we had identified for the site. I've summarized two of these below:

3.1.1. Contact form

Goal: My client wants the site to facilitate conversation between the group and its audience. Initially, this is to be accomplished by letting visitors contact the group through the website.

Challenge: This is typically done through a contact form that stores information (name, email address, message) in a database or sends it to an email address. HTML-CSS-JavaScript does not provide this functionality; it is typically provided via PHP. The developer must upload the script to a server running PHP. GitHub-hosted sites cannot run PHP scripts unless an outside server hosts the script.

Solution: Google Forms allows anyone with a Google account to create forms and surveys, which can be embedded in a website using a code snippet. For my client's site, I created a simple contact form, which I embedded under the Contact section of the site. Contact form submissions are stored in a Google spreadsheet, which can be shared with multiple people so that any group member can respond.

Limitations: The embedded form allow only limited styling. The font does not appear to be changeable. Site styles do not apply to embedded objects.

3.1.2. Book sales

Goal: My client wants to sell chapbooks and other material from their website.

Challenge: Online storefronts typically use page templates the rely on PHP to retrieve produce information from a hosted database. GitHub provides neither PHP servers nor database functionality.

Solution: PayPal and other online merchant services allow a user to create a database of products which are stored on their (PayPal, etc.'s) servers. Users can embed these products in their webpage using code snippets. [1]

Limitations: PayPal objects offer limited customization. Styling is done via the PayPal website rather than through the client's site. Therefore, the client site styles do not apply to the object. In addition, the developer must be logged into the user's PayPal account in order to make style changes-- which could result in breaches of privacy and security.

4. ToDo

The following items are still open on my todo list:

- ☑ Move banner image to an "images" folder.
- ☐ Add Google Analytics code
- ☐ [1] Rewrite body content get rid of "wall of text" users see when they come to site.
- ☐ [2] Add inline text box to Readings section to display info about the next event.
- ☐ [3] Add descriptive info about each chapbook for sale.
- ☐ Add bottom site navigation and [home] links to sections.
- ☐ Clean up styles.css

---

1. At this time, the group does not have a PayPal account. Our secondary solution was to refer questions about the availability and cost of chapbooks to the contact form.