dh+lib

where the digital humanities and librarianship meet

About Features dh+lib Review Resources Calendar

2020 Special Issue What is Static Web and What's it Doing in the Digital Humanities Classroom?

« Previous Next »

What is Static Web and What's it Doing in the Digital Humanities Classroom?

By Olivia Wikle, Evan Williamson and Devin Becker

22 Jun 2020 | 2020 Special Issue

Almost a decade ago, Matthew Kirschenbaum and Micah Vandegrift presented compelling and well-argued ideas about where the locus of digital humanities, or, more broadly, digital humanists should be within the academic context. The intervening years have demonstrated the unique capacity of DH to thrive in a variety of departments, centers, and libraries with specialties that range from making things to theoretical discourse and encompassing everything in between. As the community of DH practitioners has grown, so too has the popularity of several entry-level DH tools. In the classroom context, popular platforms like Omeka and Scalar play an important role in removing barriers and facilitating a relatively easy entry into web authorship for those without coding skills. New static web-based approaches, however, have emerged as important additions to the DH pedagogical toolbox. These approaches and the tools that facilitate them, such as Ed, Wax, and CollectionBuilder, continue to implement the critical thinking, curation, and storytelling literacies that DH tools teach, while also expanding students' technological literacies into more fundamental areas of computing.

The expanded literacies that these tools encourage include basic knowledge of file systems, web servers, and data management, concepts that students pursuing a humanities-centered education may not typically encounter. Broadening the pedagogical scope to include these concepts provides an opportunity for those teaching DH focused classes and workshops to avoid focusing solely on what John Russell and Merinda Kaye Hensley have termed the "buttonology" of a platform, i.e., teaching specifics of an interface without introducing students to basic technical concepts and methodologies that make the system work. As Dennis Tenen argues, focusing on these broader concepts when introducing a platform makes students less likely to misconstrue the tool itself as a methodology. Such explanations, in turn, help them to avoid the tendency to interpret a project's output as the end goal without trying to understand the hidden algorithms and data manipulation that produces that output.

> [W]e should be mindful of how the tools we use fit the contexts in which we teach, and, importantly, how we can use them to encourage both types of learning.

Teaching these fundamental digital skills does not entail a sacrifice: we should not have to give up teaching critical thinking skills in order to incorporate more fundamental computational concepts—part of the uniqueness of DH is its capacity to encourage new ways of thinking via innovative modes of knowledge production. Rather, we should be mindful of how the tools we use fit the contexts in which we teach, and, importantly, how we can use them to encourage both types of learning. Static web approaches, and static site generators, in general, can be used to make explicit the relationship of content as data, which is of both technological *and* critical value to humanities students who are often asked to engage with the question, "What is humanities data?"

Static web tools designed for the DH classroom facilitate teaching fundamental digital literacies because they ask that students use them without the familiarity of a GUI interface. By encouraging students to engage in this exploration in a supported environment, educators can help students learn how to approach digital content with a critical mindset and a nuanced understanding of the systems that control the technology we use, thereby empowering them with a more informed approach to the digital systems that permeate most aspects of their lives outside the classroom. In a DH context, static web tools have the capacity to reveal rather than hide the computational workings that drive them, promoting hands-on classroom engagement that increases literacies of the web, data, and digital objects.

Static Tools in DH Contexts

In the last decade, dynamic web applications, including content management systems (CMS) such as WordPress and Drupal, have dominated the web landscape as DH platform choices, which often include features such as user authentication, live comments, and endless personalized streams where pages are dynamically rendered on the fly. The functionality that these systems afford, however, comes at significant infrastructure cost, requiring robust server-side processing, databases, and complex software stacks (and the IT expertise necessary to maintain them) to deliver content to users. The details of this complexity are hidden from content creators who interact with the platform only via a web-based administrative interface, positioning learners as software users rather than software authors. This approach fundamentally limits the technical concepts that can be taught, and, as Paige C. Morgan argues, constrains the types of research questions that DH practitioners can ask.

> In a DH context, static web tools have the capacity to reveal rather than hide the computational workings that drive them, promoting hands-on classroom engagement that increases literacies of the web, data, and digital objects.

As an alternative to these complex systems, minimalistic approaches powered by modern static web generators have experienced a recent boom. Static site generators are tools that transform a structured folder of files containing content, templates, configuration options, and data to build out a complete website composed of "static" HTML, CSS, and JS files. These generated files can then be copied onto a minimal web server, which requires no database or server-side processing and will deliver the files unchanged to your browser.

In contrast to the dynamically generated pages of CMS platforms, static websites provide several benefits, including:

- faster performance
- lower bandwidth usage
- minimal hosting requirements
- fewer security vulnerabilities
- simple version control

This simplicity also means that static sites are easier to preserve and more sustainable than dynamic sites, as the basic files on the server, even if left unmaintained, will still deliver the website years later, despite the fact that their look may become dated. This is especially important for DH projects given the lack of long term support most DH centers and practitioners can provide for projects. Projects built on CMS platforms, in contrast, are more at risk of becoming malware zombies, a reality that led Quinn Dombrowski to recently caution the DH community not to "leap into buying a pony."

In DH, the use of modern static web tools to build projects is often referred to as minimal computing, which is both a computing practice enacted "under a set of significant constraints," as well as a critical movement that seeks "balance between gains and costs in related areas that include social justice issues and de-manufacturing and reuse." As Alex Gil defines it, the essence of minimal computing is that it attempts to address what a project really needs, using sustainable tools and methods. In practice, minimal computing often entails stripping away unnecessary overhead in order to mitigate reliance on databases and middleware, as well as to relieve significant requirements for processing power and storage.

Gil has been particularly active in developing the concept and enacting the practice of minimal computing. With his collaborator Marri Nyrop, he has developed two "minicomp" tools, Ed and Wax, that serve as excellent examples of the promise of this approach. Both make use of the static web generator Jekyll, as well as GitHub Pages' capacity to host websites from GitHub repositories, acting as templates that facilitate users' entry into the static web within a DH framework. When used in a pedagogical context, as Gil, Nyrop, and others have done in workshops across the country, these projects open up possibilities for students to learn transferable fundamentals of web development and data management that are just as meaningful as the final output itself. In a similar vein, our own project for creating digital collections, CollectionBuilder-GH, is specifically designed to teach both the critical and technical literacies involved in producing digital libraries.

A Scaffolded Approach to DH Literacies

CollectionBuilder is an open source template for creating digital collection and exhibit websites that are driven by metadata and hosted on GitHub Pages. To generate a digital collection, participants:

- create metadata in a spreadsheet
- organize a folder of digital objects
- set up a repository on GitHub
- configure their site's basic settings
- explore their collection website hosted on GitHub pages
- iteratively customize and debug to learn more

The steps to build the collection expand on one another, producing a scaffolded framework that begins with a firm foundation in quality metadata creation and encourages the exploration of new concepts as the collection is developed. The technical and critical skills that emerge from this process encourage the development of interwoven data and web literacies, centered around the collection's metadata as represented within a comma-separated values file (CSV).

By creating well-formed metadata in a spreadsheet, students learn fundamental data (and library!) literacies related to controlled vocabularies, unique identifiers, table-based data representations, and collaborative data cleaning and analysis. As they use these concepts to distill digital archival objects into data in the form of records and fields on a spreadsheet, students also confront the difficulty inherent in curating and representing archival materials online in a way that conveys their original forms and context, making explicit the interpretative biases that necessarily go into this descriptive work. This lesson is further driven home when they see their changes published on the web, which inevitably surfaces anomalies, breakages, and misrepresentations tied to issues in the metadata that they return to the spreadsheet to fix. The iterative nature of this process encourages students to learn the importance of well-structured data and attention to detail, while also helping to demystify "data" in general and complicate the claims often made for its objectivity.

> CollectionBuilder's scaffolded nature not only encourages these literacies but also makes the tool flexible enough to be staged for a variety of learning environments

The data literacies students develop in this process are intertwined with several web literacies as well. Students using CollectionBuilder edit and revise their data in a GitHub repository, using Git-based version control. Doing so, they must navigate their repository's directory structure and conceptualize the ways these separate files work together to produce the site. In the process of committing these edits and observing the changes they make, students learn valuable coding, computing, and collaboration concepts that are inherent to version control practices and foundational to modern web development practices. Version control also allows students a safety net to break the code itself, as they can be taught to revert the repository to a former status. This enables them to safely make edits to Markdown, HTML, and CSS files and observe how these edits make a visible impact on the collection site, altering anything from the site's About page to the algorithms producing the visualizations.

CollectionBuilder's scaffolded nature not only encourages these literacies but also makes the tool flexible enough to be staged for a variety of learning environments to focus student engagement in different aspects of the digital collection process. For example, a class of undergraduate History majors at the University of Idaho used CollectionBuilder to create a digital collection using archival materials they curated and digitized themselves. This learning experience prioritized engagement with traditional archival research methods while expanding students' critical understanding of digital repositories and their technical skills. In another scenario, a University of Idaho graduate student created a sophisticated digital collection to complement and expand her dissertation during a summer learning fellowship. In this case, CollectionBuilder provided a new way to think about research data and communicate results. In both of these examples, students integrated data and web literacies with their disciplinary knowledge, employing technical methods that informed and enabled further humanistic inquiry.

Conclusion

Overall, the pedagogical approach we use with CollectionBuilder scaffolds users' learning of open data and web fundamentals via a sequence of tasks that begin with and build off of the simple act of creating a spreadsheet. As this and similar tools (such as Wax and Ed) demonstrate, incorporating static web tools and methodologies into our DH pedagogical practices has the capacity to expand the literacy concepts we teach and to empower students to more critically engage with the digital systems pervasive throughout society.

This work is licensed under a Creative Commons Attribution 4.0 International License

About the Authors

Olivia Wikle is the Digital Initiatives Librarian at the University of Idaho, where she coordinates the digitization of the University's archival material and builds digital collections that disseminate historical resources. She also works closely with humanities faculty to create digital scholarship projects and teach digital literacy skills to students.

Evan Peter Williamson is the Digital Infrastructure Librarian at the University of Idaho Library, working with Data & Digital Services to bring cool projects, enlightening workshops, and innovative services to life. Despite a background in Art History, Classical Studies, and Archives, his recent focus has been on data-driven, minimal infrastructure web development, currently embodied in the CollectionBuilder project.

Devin Becker is the Director of the Center for Digital Inquiry and Learning (CDIL) and the Head of Data & Digital Services at the University of Idaho Library. Becker is also a writer. His most recent (static!) web project, CTRL+Shift, provides visualizations and analyses of interviews he conducted with prominent poets across the country.

More Posts

« Previous Next »

Contact Us Contribute Submission Guidelines Rights + Permissions

about dh+lib | ISSN 2380-1255 (online)