11 Web presentation formats and networked resources

This chapter is an overview of platforms for presenting digital scholarship online, discussing various approaches such as archives/repositories, themed-based projects, exhibits, and non-linear critical editions. These projects can be presented in static formats, which deliver consistent and stable content to all viewers, or dynamic formats, which generate tailored presentations in response to user input or live data. Picking a platform requires making a list of the design specifications and functions your project requires and then finding the tool that will support these needs. In addition, this chapter covers web-based collaboration, prototyping, sharing code, and participation. The standards for sharing digital assets have become well-established with large-scale organized frameworks like Linked Open Data (LoD), IIIF, and CIDOC-CRM, but different teams and institutions will have to adjust for their various resources, technical capacities, and budgets.

11a Web presentation formats

Bonus content: choosing a web presentation format

- Do you have lots of material you want to make accessible in a collection or archive?
    - Then you will need a *collections* management system to hold the digital files and metadata that describes them. ArchivesSpace is designed by professionals for this purpose.
- Do you want to create an exhibit or narrative interpretation of your materials?
    - Then you will need a *content* management system like Omeka's free version.
- Do you want to create a theme-based project with many kinds of materials, media, formats, and a range of experiences in static and dynamic formats?
    - Then you will need a more robust system for managing material, perhaps even a custom-built platform or at least a platform that allows customization. Try Manifold.
- Do you want to publish articles or write substantive posts on a regular basis?
    - Then you might want to use a blogging platform like WordPress or Substack.
- Do you want to work collaboratively?
    - Then you might want to use Google Docs for texts or MiroBoard for a graphic approach.
- Do you want to publish a critical edition of a significant literary or historical work?
    - Then you might want to work with LeafWriter or another editing platform.
- Do you want to create a branching or multiple-path narrative?
    - Then you might try Scalar.

As you consider a platform choice, make a list of features you want in your project and check to see if they are supported by the platforms you are exploring. Here is a basic list:

- Embedded media (images, audio, video, 3D, XR, or AR)
- Linear and/or branching narratives
- Standard metadata
- Shared or linked data
- Live data feed

- Collaboration among team members
- User contributed material
- Active coding environment

Bonus content: ethical aspects of choosing platforms

Finally, reflect on the ethical considerations you will bring to your project and its design around these areas:

- Accessibility: Will the platform support ADA-compliant presentation? If not, then how will you manage to make the materials available?
- Sustainability: Does the platform have a broad enough user base to last? Is there a user community to assist with troubleshooting? Are there special features that might become obsolete too quickly? Does the platform have a record of being "back-ward compliant" so that if they issue a new version your work will still be usable?
- Intellectual property issues: If your materials are under copyright, do you have permission to use them? Be sure you know the rules of Fair Use. Also consider the licensing terms under Creative Commons' Licenses (https://creativecommons.org/).

Bonus content: file naming and organization

Any web-based project you undertake will depend on digital assets that are organized into folders and files, given names that are both human and machine readable, and able to be retrieved easily.

File naming and organization should be as simple and logical as possible and support the storage and retrieval of files by both human and machine. If your file structure is too simple, it can mean each folder contains lots of material. If it is too complicated, the pathways for retrieval will become too complex. File and folder names should be specific enough not to be confused with each other or with files from other projects. Generic folder names can cause confusion if projects are aggregated or joined (e.g. "documents" or "images" vs. "Wedding_documents" or "Travel_images").

How-to example

Look carefully at the differences between the first and second file names. Note the differences and consider the ways they reflect a logical organization. Also note the machine-readable features of the second set (underscores, no spaces, no apostrophes or dashes, numbering conventions).

Sample of folder names that will cause difficulties:

Documents: Dr. Washington's Notebooks—First group

Notebook page 1
Notebook page 2
Notebook page 3

Documents: Dr. Washington's Notebooks—Second batch

Notebook page 1
Notebook page 2
Notebook page 3

Images of Documents: Dr. Washington's Notebooks—photographs

Photo 1
Photo 2

Sample of correctly formatted file folder names with pathways:

Docs/1_Dr_Washington_Notebooks_Group_01

01_Notebook_01
01_Notebook_02
01_Notebook_03

Docs/2_Dr_Washginton_Notebooks_Group_02

02_Notebook_01
02_Notebook_02
02_Notebook_03

Docs/Images/1_Dr_Washington_Photographs

01_Notebook_photo_01
01_Notebook_photo_02
02_Notebook_photo_01

Numbering the files sequentially makes them easy to sort. Underscores can be read by the browser. The file pathways show the nested structure of the files. These will be included in the HTML so the correct file can be found by the browser. No special characters, including periods, are used in the filenames. When numbering, remember to include enough places to include all of your values (00 for up to 99 items, 000 for up to 999 items, etc.) and keep the zeros in place to assure numerical order.

Bonus content: critical editions, a special case

One specialized type of publication is a critical or scholarly edition of a work. This is generally focused on the text of a single author (Shakespeare, Molière), even on a single text such as Beowulf, Piers Plowman, or the Septuagint (Greek version of the Bible). The purpose of a scholarly edition is to establish the most authentic and accurate version possible of a particular text, often by tracking all of the variations in extant "witnesses" (documents that record a version of the text). Critical commentary and annotation often accompany this work to make allusions, references, or obscure material more legible to readers. Because many corpora (collections of works) have long histories of revision, computational tools can be very useful in studying their composition or reception. The digital humanities community has developed specialized tools for the editing work and analysis. LeafWriter (https://www.leaf-vre.org/docs/documentation/leaf-commons/leaf-writer-documentation-basic) was designed specifically to assist in critical editing but relies on knowledge of TEI and XML. CEDAR (https://voices.uchicago.edu/cedar/) is made for performing computational analysis of edited texts.

Bonus content: non-linear site design and motion/animation

If you want to create a site that has multiple narratives or branching storylines, these platforms contain useful features developed specifically for this purpose. Keep in mind that export functions from these programs are more limited and that navigating the non-linear formats may be confusing to some users.

- Scalar (https://scalar.me/anvc/scalar/)
- Manifold (https://manifoldapp.org/features)

Hyperlinking creates multiple pathways through a project in ways now so familiar we don't consider them. A site design should be developed with the recognition that someone might enter the site and move around at will, not necessarily following a prescribed path. In fact, many sites—particularly repositories or collections—do not have a path. Those that do create an overarching narrative for the viewer. This illusion of continuity might conceal even more of the back-end operations.

Explicitly non-linear arguments require careful design. A platform that was specifically designed to create narratives with multiple, non-linear or not-necessarily linear navigation are Scalar and Twine (slightly more game-like). Developed from within a community that wanted to embed media, particularly time-based media, within scholarly work, Scalar has built-in capacities for forking-path design that are very well suited to creating non-linear arguments. As in all online projects, the conceptual approach required is modular—what are the logical pieces and chunks, and how might they fit together in a variety of ways? Scalar supports many scholarly—and creative—projects and has a strong commitment to argument. The innovative design offers alternatives to a single linear organization or pathway and encourages branching and networked structures.

Example: scalar showcase

Scalar is an experimental publishing platform meant to provide multiple points of entry to a project and various pathways through it. It extends work that was done in an earlier project, Vectors, where every interface was custom designed to suit the projects. Look through the Vectors archive and think about narration, navigation, and orientation conventions in these projects.

- http://scalar.usc.edu/
- http://vectors.usc.edu/issues/index.php
- https://scalar.me/anvc/scalar/features/
- https://scalar.me/anvc/showcase/dna-seven-interactive-essays-on-nonlinear-storytelling/

Bonus content: specialized functions—mapping, timelines, visualization tools

Many humanities projects make use of visualization tools for mapping, timelines, or data presentation. The digital humanities community has created several platforms designed specifically to serve their needs.

- Mapping applications use geospatial data to display information on a base map. These range from relatively easy to use off the shelf applications to extremely technical programs. Leaflet (https://leafletjs.com/) and StoryMapJS (https://storymap.knightlab.com/) are both designed for non-technical users and have tutorials on their sites to show how to add pins, pop ups, and combine maps with stories.
- Timelines provide a useful scheme not only for presenting information but also for providing access to documents and materials. TimelineJS (https://timeline.knightlab.com/) and TikiToki (https://www.tiki-toki.com/) use standard timelines as their framework and have tutorials on site to support links for embedding comments and documents into a presentation.
- Visualization tools cover a wide range of functions from creating networks in Gephi (https://gephi.org/) and Cytoscape (https://cytoscape.org/) to graphing data in charts, graphs, and diagrams using any of the many formats in Tableau (https://www.tableau.com/).

Many of these tools were discussed in detail earlier. (See Chapter 7 Information Visualization and Chapter 8 Mapping and GIS).

Exercise 11.1: Explore ArchivesSpace

Go to https://archivesspace.org/ and open the “About” drop-down menu. Click on “Who’s Using ArchiveSpace?” (https://archivesspace.org/about/whos-using-archivesspace). What kinds of institutions and what kinds of collections are best handled on this platform? How global is the range of users? Go to the blue menu bar above this and click on “Watch Demo” to see if you understand the terminology (e.g. “Top Containers” and “Assessment Attributes”). Now go to “How it Works” and click through the topics. How can you determine if this platform will work for your project?

How-to example

ArchivesSpace is used across a wide range of institutions but has the most adoption in what they call “very small level” institutions. The platform does serve users across the globe, though most adopters are in the United States and Europe. To know if this platform is a good fit for your project, consider factors such as the complexity of your collections, the need for metadata management, and the level of user interaction required. ArchivesSpace’s open-source nature and community support make it a robust choice for many archival projects, especially where there is not a need for detailed customization.

Exercise 11.2: Try Omeka

Go to: https://www.omeka.net/. Read through the documentation on the homepage. What kinds of projects make sense for what you are trying to accomplish? Does this feel easier than ArchivesSpace? Do you see any built-in limits? Click on the Showcase tab (https://info.omeka.net/showcase/) tab and look at the range of projects, designs, and presentations. Look at the pricing and hosting options. Create an account and test the free version. Do the metadata fields work for describing your materials?

How-to example

While Omeka shares many of the cataloging features of ArchivesSpace (demo (https://archivesspace.org/application/demo)), Omeka (Get Started Documentation (https://info.omeka.net/build-a-website/)) has many more media-rich options for presenting archival recordings in the public-facing interface. Omeka offers users three layers for organizing their content. At the Item level, you can create individual records and catalogue media using the built-in Dublin Core schema. Users can then choose to organize those items into Collections. Finally, at the Exhibit layer, users can design a digital presentation of collections and items, alongside text and other media, to create a narrative and/or curated structure for the material. Themes are available to further customize the front-end appearance, and plugins can be used to offer additional functionality, like map or timeline displays for item content with location or time information. There are also other versions of Omeka that are more tailored to institutional deployment, as opposed to on an individual level. You can learn about them at: https://omeka.org/.

Miriam Posner has created a number of Omeka resources that are useful for getting started:

- Omeka Vocab Worksheet (http://miriamposner.com/dh101f14/wp-content/uploads/2014/10/OmekaVocabWorksheet.pdf)
- Up-and-Running-with-Omeka (http://miriamposner.com/dh101f14/wp-content/uploads/2014/10/Up-and-Running-with-Omeka.pdf)
- Omeka_Exhibit_Handout (http://miriamposner.com/dh101f14/wp-content/uploads/2014/10/Omeka_Exhibit_Handout.pdf)
- Installing Themes (http://omeka.org/codex/Managing_Themes_2.0)
- Installing Plugins(https://docs.google.com/document/d/1gN4NYMEfDI3cvVZMCNgOViTwX2FTND93V1CNZPNuqPw/edit)

Exercise 11.3: WordPress or Substack

Go to: https://wordpress.com/ and also https://substack.com/get-started. Take a look at sample projects and assess their design and tone. Which feels more appropriate for your work? Try out the Get Started information on WordPress (https://wordpress.com/support/category/get-started/) information on WordPress and "How to start a Substack" tutorial. Which feels easier?

How-to example

While it started off as a blogging platform, WordPress has become and remained one of the leading website development platforms. As such, it has a commercialized approach, with a graduated fee structure for hosting (which does include free options) and a plethora of documentation to describe its many features and options for web-building. Substack is more narrowly tailored to individuals looking to publish a regular newsletter with media-rich content. Substack is free to authors. Users access the material for free unless the author decides to charge a subscription fee, of which Substack will then take a percentage.

Exercise 11.4: Look at Drupal, Vue.js, and React

Go to these in turn and try to understand their differences: Drupal (https://new.drupal.org/home), Vue.js (https://vuejs.org/), and React (https://react.dev/). Each is a bit more technically challenging than the last. What would be involved in learning to use these? Is the difficulty in learning owing to the unfamiliarity of concepts? Are the tutorials adequate? Read up on their current adoption and use. How clear is their documentation compared with Omeka or Substack?

How-to example

Drupal (https://new.drupal.org/home) is a very robust content management system that can allow for glandular detailing of both functionality and appearance for web-building. Vue.js (https://vuejs.org/) and React (https://react.dev/) are for building customized web interfaces and require an understanding of programming for development and deployment. Learning to build with Drupal requires web developers to become familiar with the various components of the CMS and how they can be customized and configured to meet specific needs (user guide). Vue.js documentation (https://vuejs.org/guide/introduction.html) and React Quick Start (https://react.dev/learn) offer interactive learning documentation to share how programming in their frameworks can allow for users to build interactive web applications. But for users to actually get started, they will need to know how to use an editor, like VS Code, be comfortable with command line, install various other packages and compilers, and have an understanding of how to deploy to the Web. These are tools to D.I.Y., as opposed to Omeka and Substack, which are designed to be used out of the box.

Exercise 11.5: Preparing your materials

Now that you have looked at a number of platforms, this would be a good time to assess what the materials are that you want to digitize, make accessible, or publish. What are the assets you have digitized and how are they organized, described, and named? How will they be used and how much flexibility do you want in your site? Should scholars be able to access the entire set of assets, or only see them in the interpretative frameworks you have created?

Exercise 11.6: Export functions

Look at some of the tools and platforms you use regularly for word processing, image viewing, presentation authoring, or audio editing. What export formats do they each have? How about a spreadsheet program? How much interoperability is there among data types?

How-to example

Certain file types were designed with interoperability in mind or are platform agnostic. For spreadsheets,.CSV files are transferable across all spreadsheet applications or applications that read tabular data. If you are building with any tool, it is always prudent to first consider if you will be able to extract your data back out, and what format it will be provided in. XML or JSON will keep your data's structure and allow for it to be fairly easily re-structure, if need be. File formats for media data, such as video, audio, or 3D files, can be more complicated. Popular file formats change over time, and platforms may automatically compress your data when you ingest it. This means you may want to keep copies of your original files in separate storage.

Recommended readings

Edwards, Dustin, and Bridget Gelms. 2018. "The Rhetoric of Platforms." *Present Tense: A Journal of Rhetoric in Society* 6 (3). https://www.presenttensejournal.org/editorial/vol-6-3-special-issue-on-the-rhetoric-of-platforms/.

Jacobson, Daniel. 2009. "COPE: Create Once, Publish Everywhere." *Daniel Jacobson Blog*. https://danieljacobson.net/blog/11/comment-page-1.

Wikle, Olivia M., and Evan Peter Williamson. January 26, 2024. "Exploring Static Web in the Digital Humanities Classroom: The Learn-Static Initiative." *IDEAH* 4 (2). https://ideah.pubpub.org/pub/mf7o4gk5.

11b Tools for web-based collaboration, prototyping, sharing code, and participation

Bonus content: Git's history

Git, on which GitHub was built, was invented by Linus Torvalds, a Finnish software engineer with a sense of humor. He picked the term Git because the word means "a disagreeable old codger" but he suggested it could mean any number of things, depending "on your mood." To be specific, Torvalds listed several possibilities: that it was a "random three-letter combination that is pronounceable, and not actually used by any common UNIX command," or that it meant "Stupid. Contemptible and despicable." Or it could mean "global information tracker" since when it works "Angels sing, and a light suddenly fills the room." His final definition, for when it breaks, was less polite. Torvalds was the inventor of Linux, an open-source version of the ubiquitous operating system Unix. His aspirations in creating GitHub as a version control system came directly from his experience in software development and understanding the challenges in its workflow. Any project that has a programming component to it can benefit from GitHub's basic functions and learning how to read files on the repository is a useful skill.

Bonus content: revisiting standards

The purpose of standards for formats, metadata, services, and other features is to make materials more accessible to more communities over a longer period of time. Many people will never think about infrastructure. In many cases, the infrastructure that supports access to digital assets and services only becomes apparent when it breaks. A little bit of knowledge goes a long way toward ensuring that resources remain usable. Nothing substitutes for professional expertise and projects should be designed from the outset with advice on best practices.

Bonus content: linked open data standards and the concept of interoperability

By this point, you may feel lost in acronyms and references to standards with which you are not familiar. But some principles discussed multiple times here are useful to keep in mind, among which the most important is that standards only work when they are applied and applied consistently and across a substantial portion of data. (Refer to the Dublin Core examples in Chapter 4.) This requirement has been a sticking point since the application of data standards to existing data within cultural repositories is, like so many digital processes, very labor intensive, time consuming—in a word, expensive. Whether LoD will succeed depends on the extent to which it is adopted and whether alternatives arise that can be automated to do some of the same work of integrating information that was created in a wide range of formats and structures.

Bonus content: image viewer IIIF, International Image Interoperability Framework

IIIF, the International Image Interoperability Framework, is important because it is integrated into the operation of so many institutions worldwide, and because it serves as an exemplary model of the "framework" concept mentioned in earlier sections. **[Exercise 11.9: Evaluating image viewers and trying Mirador]**

Bonus content: intellectual property issues and accessibility

Any project involving intellectual materials has to be designed with respect for intellectual property. [See Chapter 13 for IP discussion and Fair Use guidelines.]

The same standards for practice apply in creating web presentation platforms as in any other aspect of digital humanities work. Be sure that you build in text-to-voice, alternative descriptions of visual materials, and address color-blindness as a part of design decisions.

Exercise 11.7: Compare MiroBoard, InVision, and Tapestry-tool

Look at Miro (https://miro.com/app/dashboard/) and InVision (https://appsource.microsoft.com/en-us/product/office/WA200003579). These are both useful for collaboration, but what are the differences? What kinds of conversations would you imagine each could support? What kinds of projects do they suggest? What does collaboration mean in each instance? Now, look at Tapestry Tool (https://www.home.tapestry-tool.com/) which is made for pedagogical collaboration and describes itself as an online learning platform.

How-to example

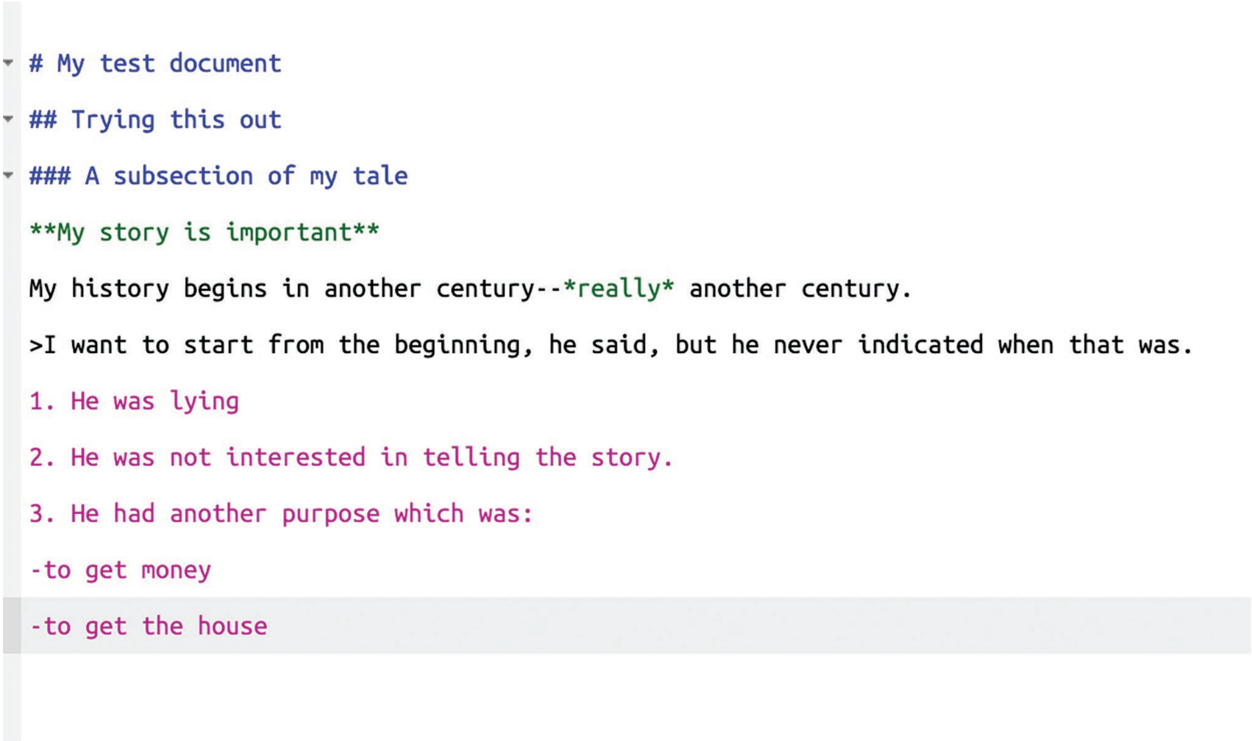
Miro (https://miro.com/app/dashboard/) and InVision (https://appsource.microsoft.com/en-us/product/office/WA200003579) are digital whiteboard tools that allow users to ideate on an infinite canvas. InVision is tailored more toward product design, whereas Miro's features allow it to be used at additional stages of production and have integrated project management tools. The Tapestry Tool (https://www.home.tapestry-tool.com/) also makes use of an infinite digital canvas, but the focus is on the connections that users can build between the information shared on the canvas.

Exercise 11.8: Write a sample Markdown document

Go to http://markdown.org, review the Basic Syntax page (https://www.markdownguide.org/basic-syntax/), and create a file using the cheat sheet. Every markdown file has to go through a processor before it will display properly in HTML. Is this easier or harder than working directly in an HTML editing environment?

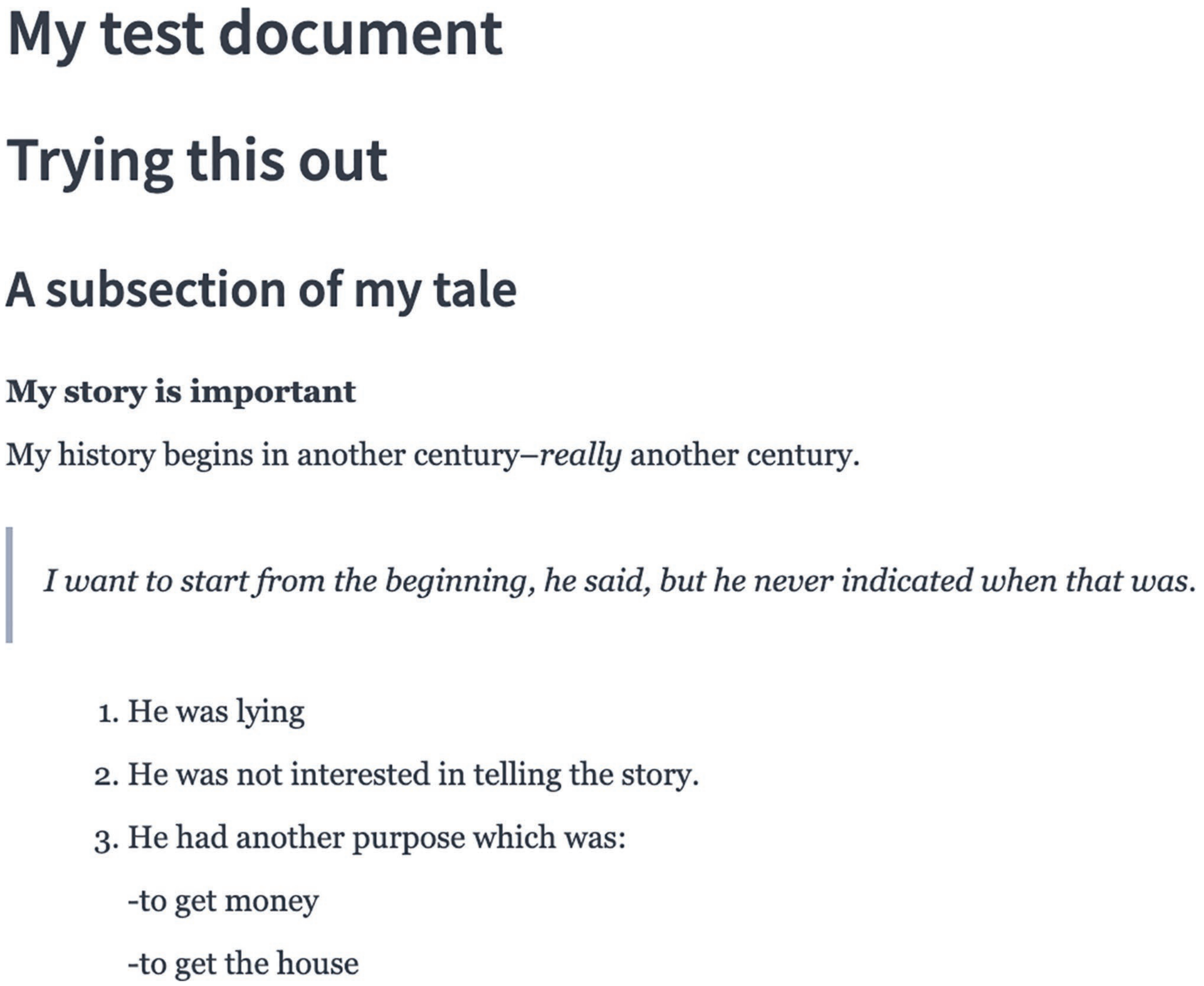
How-to example

Go to: https://www.markdownguide.org/getting-started/. Be aware that in addition to writing your document, you will need to send it through a processor to get it to display properly (Figures 11.1 and 11.2).



```
# My test document

## Trying this out

### A subsection of my tale

**My story is important**

My history begins in another century--*really* another century.

>I want to start from the beginning, he said, but he never indicated when that was.

1. He was lying

2. He was not interested in telling the story.

3. He had another purpose which was:

-to get money

-to get the house
```

Figure 11.1 An example of a markdown file with syntax. Note the minimal nature of the code. (Johanna Drucker).



My test document

Trying this out

A subsection of my tale

My story is important

My history begins in another century–*really* another century.

> *I want to start from the beginning, he said, but he never indicated when that was.*

1. He was lying
2. He was not interested in telling the story.
3. He had another purpose which was:

   -to get money

   -to get the house

Figure 11.2 Formatted markdown sample in a browser. The same sample is displayed in a browser. Markdown is essentially a formatting tool. (Johanna Drucker).

Exercise 11.9: Evaluating image viewers and trying Mirador

Examine the IIIF App Demos (https://iiif.io/apps-demos/) available on their website, as well as the documentation shared on the README file (https://github.com/IIIF/awesome-iiif#image-viewers) of this IIIF GitHub repository. Look at the list of viewers available for IIIF and see what you learn reading their descriptions. Build your vocabulary for understanding open-source and proprietary formats. Note that some are built in JavaScript and HTML5, others make use of JSON-formatted data (the suffix.js is frequently present). Learn to evaluate feature sets for these applications in relation to your project needs: comparison, annotation, sequencing, high-resolution image displays, and so on. Then look at Mirador (https://projectmirador.org/).

How-to examples

The following video tutorials introduce a variety of IIIF resources and methods:

- Teaching and Research with Digital Collections (Part 1) (https://youtu.be/SuPGRmryCqI?si=IEi7QuoTdIvDUb5o)
- Teaching and Research with Digital Collections (Part 2) (https://youtu.be/_xOnkUfSpM8?si=AcwPMKVHkol6w0_7)
- Teaching and Research with Digital Collections (Part 3) (https://youtu.be/N1ZuugRm04E?si=CajF3K4u-2sn1hxh)
- Annotate Digital Imagery: Theory, Applied Practice, and Tools for Research(https://youtu.be/igOv6kYhw5I?si=v9n7P59P_OAN6fUT)

Exercise 11.10: Exploring GitHub

Go to Github Docs (https://guides.github.com/introduction/git-handbook/#basic-git). Look through GitHub's pricing structure and think about how it embodies their ethical values. Be sure to compare several categories. Perhaps, keep in mind the extent to which many proprietary and open-source projects alike are built on infrastructure developed with government funding for military and other research—such as GPS and Wi-Fi. These histories are not apparent in our daily use. Take a look through the materials on the Collections Github page (https://github.com/collections). Look, for instance, at the Pixel Art Tools repositories (https://github.com/collections/pixel-art-tools) and think about working with these. GitHub is designed for ease of use and has a very low threshold of entry.

Bonus exercise: crowdsourcing

Review this blogpost (https://www.adamenfroy.com/crowdsourcing-platform) by Adam Enfroy on the seven Best Crowdsourcing programs (2024) to get an informed sense of the overview of this activity. How does Enfroy assess these platforms and how would you choose a platform based on this advice?

Now look at FromThePage (https://fromthepage.com/landing), which is a platform built specifically to assist in crowdsourced transcription projects. What kinds of projects have been successful using this approach? Why do you think so? What project might you design to use FromThePage?

How-to example

Enfroy assesses the crowdsourcing platforms based on what they allow users to do most effectively with the platform. He also includes a list of features and pricing as an additional part of his consideration. In any project, you will need to make decisions based on your own needs, time, and resources.

Scroll down to "Case Studies Using FromThePage" (https://fromthepage.com/public_libraries) to read how this IIIF-based crowdsourcing tool is assisting libraries in transcribing a variety of special collections documents. It is ideal for projects that are looking to transcribe, translate, or add metadata to documents collaboratively.

Bonus exercise: reverse engineering projects

Look at these projects and see if you can reverse engineer what was involved in making them. Is there a project that has features (design, functionality, organization, scale) that you would like to include in your own work?

- Open Restitution (https://openrestitution.africa/)
- African Digital Heritage (https://africandigitalheritage.org/)
- Buddha Nexus (https://buddhanexus.net/)
- The Long Emergency (https://www.longemergency.demx.in/)
- Europeana (http://www.europeana.eu/en)
- Nines (https://nines.org/)
- Pelagios (https://pelagios.org/about-us/)
- Renaissance Knowledge Network (https://rekn.org)
- Smithsonian Libraries Online Exhibitions (https://library.si.edu/exhibitions/online)
- Stanford Spatial History Project(https://web.stanford.edu/group/spatialhistory/static/)

Recommended readings

A Digital Infrastructure for Humanities (Report on the activities of DARIAH). "European Commission." https://projects.research-and-innovation.ec.europa.eu/en/projects/success-stories/all/digital-infrastructure-humanities-research.
https://cordis.europa.eu/project/id/675570.
https://has.dariah.eu/.

Braunstein, Laura. 2017. "Open Stacks: Making DH Labor Visible." *Dh+lib.* https://dhandlib.org/open-stacks-making-dh-labor-visible/.

Elliott, Dorinda. 2023. "Mapping the Future of Digital Humanities in Asia." *Fairbank Center for Chinese Studies*, April 30, 2023. https://fairbank.fas.harvard.edu/research/blog/mapping-the-future-of-digital-humanities-in-asia/.

Maya Dodd, and Nidhi Kalra, eds. 2020. *Exploring Digital Humanities in India: Pedagogies, Practices, and Institutional Possibilities*. New Delhi: Routledge.

Gasparyan, Armen Yuri, Marien Yessirkepov, Alexander A. Voronov, Anna M. Koroleva, and George D. Kitas. 2019. "Comprehensive Approach to Open Access Publishing: Platforms and Tools." *Journal of Korean Medical Science* 34 (27). www.ncbi.nlm.nih.gov/pmc/articles/PMC6624413/.

Li, Wenqi, Zhenyi Tang, Pengyi Zhang, and Jun Wang. 2025. "Collaborative Data Behaviors in Digital Humanities Research Teams." In *Proceedings of the 24th ACM/IEEE Joint Conference on Digital Libraries*, 18:1–18:5. New York: Association for Computing Machinery. https://doi.org/10.1145/3677389.3702537.

Nelson, Chris Hoffman. January 13, 2014. "What Is GitHub, and What Do Geeks Use It For?" *How-To Geek.* Accessed April 3, 2025. https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/.

Bibliography

Blaney, Jonathan. 2020. “Introduction to the Principles of Linked Open Data.” *The Programming Historian*. https://programminghistorian.org/en/lessons/intro-to-linked-data.

Burnett, Jeffrey. 2021. “Contemplating Crowdsourcing in Digital Cultural Heritage.” *Cultural Heritage Informatics Initiative*, December 10, 2021. https://chi.anthropology.msu.edu/2021/12/contemplating-crowdsourcing-in-digital-cultural-heritage/.

Driscoll, Mike. “Jupyter Notebook: An Introduction.” *Real Python*, n.d. Accessed April 3, 2025. https://realpython.com/jupyter-notebook-introduction/.

Kim, Soon, Hyungwood Choi, Nayon Kim, EunKyung Chung, and Jae Yun Lee. 2018. “Comparative Analysis of Manuscript Management Systems for Scholarly Editing.” *Science Editing*. https://www.escienceediting.org/journal/view.php?number=148.

New York Public Library. “What’s on the Menu?” *New York Public Library*, n.d. Accessed April 3, 2025. https://www.nypl.org/research/support/whats-on-the-menu.

Princeton Geniza Lab. “Crowdsourcing and the Humanities.” *Princeton University*, n.d. Accessed April 3, 2025. https://genizalab.princeton.edu/crowdsourcing-and-the-humanities.

Tomasis, Rebecca. 2025. “Static vs Dynamic Website: Key Differences & Which One to Choose.” *Wix Blog*. Accessed April 3, 2025. https://www.wix.com/blog/static-vs-dynamic-website.

Resources

- This is an excellent NYU library guide (https://guides.nyu.edu/digital-humanities/tools-and-software) to digital humanities tools and is regularly updated.
- About Git (https://guides.github.com/introduction/git-handbook/#basic-git)
- Intro to Git and Github (https://github.com/kirschbombe/learngit)
- Git README (https://github.com/git/git/blob/e83c5163316f89bfbde7d9ab23ca2e25604af290/README)
- Github pricing (https://github.com/pricing)
- IIIF Demos (https://iiif.io/apps-demos)
- National Library of Russia Online Exhibitions (http://expositions.nlr.ru/eng/)
- NYU Tools and software guide (https://guides.nyu.edu/digital-humanities/tools-and-software/visualization)
- HTML5Up (https://html5up.net/)
- Visual Studio (https://visualstudio.microsoft.com/)
- GitLab_Jekyll (https://pages.gitlab.io/jekyll/jekyll/update/2016/03/24/welcome-to-jekyll.html) a shared code resource for site building
- WikiData (https://www.wikidata.org/wiki/Wikidata:Main_Page) – use of creative commons spaces and shared data repositories
- Why IIIF? Benefits & Value for Adopters (https://docs.google.com/document/d/1h9SPg9nlLA3TAdzkYxkyaFFNNf8m4PsbaIExcYwVLyo/edit#heading=h.q7mmne7d3bk2)
- A list of Awesome IIIF Resources (https://github.com/IIIF/awesome-iiif)
- IIIF Image API Playground (http://www.learniiif.org/image-api/playground)
- Digirati’s Introduction to IIIF (https://resources.digirati.com/iiif/an-introduction-to-iiif/)
- IIIF Workshop by Jason Ronallo (http://ronallo.com/iiif-workshop-new/beginnings.html)
- CIDOC-CRM (http://www.cidoc-crm.org/)
- CIDOC-CRM Use Cases (http://www.cidoc-crm.org/useCasesPage)