2023 Google Season of Docs Case Study

Improve p5.js Reference and Example Docs

Organization or Project: p5.js

Organization Description: p5.js is a JavaScript library for creative coding, with a focus on making coding accessible and inclusive for artists, designers, educators, beginners, and anyone else.

Authors: Qianqian Ye, Nick McIntyre

Problem Statement

What problem were you trying to solve with new or improved documentation?

The p5.js library has become increasingly popular among beginners who are learning to code in JavaScript. However, the current reference and example documentation, while comprehensive, can be overwhelming and difficult to navigate for users who are new to programming or p5.js. Challenges the documentation faces include: 1) unclear organization of information architecture, 2) assumptions of prior knowledge of programming concepts, and 3) inconsistencies in writing style.

Improving the reference and example documentation is crucial for p5.js and its community. The proposed project will ensure that the library's documentation is well-organized, beginner-friendly, up-to-date, and consistent by implementing a clear structure based on community feedback, using simple language, avoiding technical jargon, and providing beginner-friendly examples and explanations. By improving the documentation, users of all levels will be able to learn p5.js more easily, reducing frustration and improving their confidence in using the library. Ultimately, this will help promote the growth and development of the p5.js community, which is essential for the project's long-term sustainability.

Proposal Abstract

A brief summary of your original organization proposal. Link to the proposal page on your project site, if possible.

You can access our original proposal at this link.

- The "Improve p5.js Reference and Example Docs" project aims to:

- Conduct an audit of the existing reference (https://p5js.org/reference/) and example (https://p5js.org/examples/) documentation for organization, simplicity, and consistency.
- Create a friction log for the current documentation based on specific use cases, such as learning p5.js as a beginner and teaching p5.js to beginners.
- Gather feedback from documentation testers (including volunteers in the project) and the wider p5.js community.
- Use the friction log and community feedback as a guide to update the structure, descriptions, and examples on reference and example pages to ensure that they are accessible, beginner-friendly, and up-to-date with the latest software version.
- Share project findings with the p5.js community through a live presentation or blog post.

Work that is out-of-scope for this project:

- This project will not add new references.

Project Description

Creating the proposal

How did you come up with your Season of Docs proposal? What process did your organization use to decide on an idea? How did you solicit and incorporate feedback?

The p5.js project created the proposal by looking at our existing documentation through the lens of access. In what ways could our documentation afford greater access to contributing and working on the p5.js project? The p5.js project lead Qianqian Ye and our technical writer candidate Nick McIntyre worked collaboratively to answer this question. We collected feedback from students and educators about the existing Reference and Example page on p5.js website, with a focus on making the documentation more accessible. We framed the proposal into two key focus areas: 1) readability of writing and 2) clarity of code examples.

Budget

Include a short section on your budget. How did you estimate the work? Were there any unexpected expenses? Did you end up spending less than the grant award? Did you have other funds outside of Season of Docs that you were able to use?

Our project stayed within our $15,000 grant budget. We paid the technical writer $12,000 total, based on an estimated 5-10 hours of work per week throughout the project. We paid two community testers $750 each for their feedback on edits made by the technical writer. We also paid one mentor $1,000 and one advisor $500 for their support. In the future, we would like to increase stipends for all contributors to more closely match the time and effort spent on projects.

The Processing Foundation did use internal funds (from donations and other grant awards) to cover administrative costs for the p5.js project lead throughout the Season of Docs project.

Budget Item	**Amount**	**Running Total**	**Notes**
Technical Writer - p5.js Contributor Docs Organization & Development	$12,000	$12,000	The technical writer Nick McIntyre will audit, update, test, and publish p5.js reference and example pages.
Mentor Stipend	$1,000	$13,000	Dave Pagurek
Advisor Stipend	$500	$13,500	Jaleesa Trapp
Community Testers Honorarium	$1,500	$15,000	Darren Kessner and Layla Quiñones ($750 x 2)
TOTAL		$15,000	

Participants

Who worked on this project (use usernames if requested by participants)? How did you find and hire your technical writer? How did you find other volunteers or paid participants? What roles did they have? Did anyone drop out? What did you learn about recruiting, communication, and project management?

After receiving the grant from Google, we hired Nick McIntyre as the technical writer. Nick has been an active contributor to p5.js codebase, p5.js website, and various p5.js-related educational initiatives. For example, he served as a mentor for the Dynamic Learning and p5.teach.js projects through Google Summer of Code.

We decided to have a mentor Dave Pagurek and an advisor Jaleesa Trapp to help the writer plan and organize the project. These collaborators offered feedback drawn from experiences that complemented the writer's. We also recruited two community testers, Darren Kessner and Layla Quiñones, who have decades of combined experience teaching creative coding to young people and supporting K–12 computer science teachers. The community testers provided the writer with feedback on his edits to the p5.js Reference.

Timeline

Give a short overview of the timeline of your project (indicate estimated end date or intermediate milestones if project is ongoing).

Our technical writer Nick McIntyre kicked off the project in May, and we had monthly check-ins with the writer to discuss progress, challenges, and questions. We used the first month for the technical writer to define his own timelines and work deliverables within the general headings provided in our project description. After that, Nick operated on his own timelines with support from his team.

Task	Completed By
Technical writer hired	May 3, 2023
First meeting with mentor and advisor	May 16, 2023
Create a documentation style guide	Jun 15, 2023
Edit reference docs	Jun 17 – Nov 13, 2023
Recruit community testers	Aug 30, 2023
This case study created	Nov 14, 2023
Case study submitted	Nov 20, 2023

Results

What was created, updated, or otherwise changed? Include links to published documentation if available. Were there any deliverables in the proposal that did not get created? List those as well.

Defining the p5.js Documentation Style Guide

The initial design of the friction log included a field for suggested revisions. We realized that such revisions would benefit from clear guidelines, so the writer created the p5.js Documentation Style Guide. The guide remixed style suggestions from several open source

projects including the Ruby on Rails API Documentation Style Guide, the WordPress Documentation Style Guide, and the Airbnb JavaScript Style Guide. The writer incorporated feedback from several core contributors throughout the process.

Editing the Reference

During research, the writer identified the Hemingway Editor as a useful tool for ensuring readability. The editor highlights words and sentences that are difficult to read. It also displays the minimum education level needed to read a document. We opted to drop the friction log in favor of a streamlined workflow:

- Copy existing docs into Hemingway
- Simplify and update the docs based on the style guide
- Incorporate feedback through GitHub

Editing proved to be more labor-intensive than originally thought. The writer explored existing code examples, which often required reading the p5.js source code. The process was illuminating. Making prose and code examples more beginner-friendly stretched the writer as a communicator and programmer in a good way.

The project mentor, community testers, contributors, and stewards provided helpful feedback throughout the editing process. They spotted bugs and typos, suggested ways to simplify, and generally kept the writer attuned to the needs of beginners.

Laying Groundwork

We realized early in the project that we needed to put significant effort into the Reference. We focused on a subset of the API most commonly used by beginners along with priority areas such as accessibility. Examples and the full Reference would simply have to wait. After discussing with the mentor, advisor, and maintainers of p5.js, we collectively decided to drop the Example page from our work scope of Season of Docs 2023 to ensure the quality of Reference updates. A team of contributors is working on Examples outside of Season of Docs 2023.

Relevant Issues and PRs

We addressed 8 issues and merged 20+ pull requests during Season of Docs 2023.

- Add a style guide for writing and code in the p5.js docs
    - Update doc guides to use let for declarations
    - Add a style guide for documentation. Addresses #6718
    - Modify inline code style for reference
- Edit DOM docs to align with style guide
    - Edit docs for p5.Element
    - Revise docs for dom functions

    - Edit docs for dom objects
- Edit environment docs to align with style guide
    - Edit docs for accessibility
    - Edit docs for environment
- Edit typography docs to align with style guide
    - Edit docs for typography load and display
    - Edit docs for p5.Font
    - Edit docs for typography attributes
- Edit image docs to align with style guide
    - Edit docs for pixels functions
    - Edit docs for image
    - Edit docs for loading & displaying images
    - Update docs for p5.Image
- Edit math docs to align with style guide
    - Edit docs for math functions
    - Edit docs for p5.Vector
- Edit docs for 2D primitives and attributes
    - Edit docs for 2D primitives. Resolves #6217
- Edit docs for color
    - Edit color creating & reading docs. Addresses #6219
    - (Advised) Edit color setting docs. Addresses #6219

Metrics

What metrics did you choose to measure the success of the project? Were you able to collect those metrics? Did the metrics correlate well or poorly with the outcomes you wanted for the project? Did your metrics change since your proposal?

Our proposal included metrics based on beginner-friendliness, GitHub contributions, and website visits. We can summarize our initial progress toward the first two:

- Readability, accuracy, and simplicity improved significantly for approximately 270 elements in the API reference. That accounts for about 47% of the total reference. As an illustrative example, the level of education needed to read the reference for the noise() function dropped from Grade 10 to Grade 6.
- Contributors (including the technical writer Nick McIntyre) opened 25 pull requests related to Reference documentation between June and November 2023. That's a 6X increase over the prior six-month period.

One of our proposed metrics, a decrease in documentation issues, is likely to increase rather than decrease. We expect contributions to documentation to accelerate as we continue editing and translating. The number of (good) issues opened will accelerate in kind.

Our other proposed metric, visits to the p5.js Reference and Example pages, will likely increase as planned. The analysis is complicated by a website redesign that's currently underway. The redesign is substantial and emphasizes accessibility, so the specific metrics may need adjustment.

Analysis

What went well? What was unexpected? What hurdles or setbacks did you face? Do you consider your project successful? Why or why not? (If it's too early to tell, explain when you expect to be able to judge the success of your project.)

The project was successful for multiple reasons. We made significant improvements to nearly half of the p5.js reference and laid the groundwork for a team of writers currently working on documentation. Along the way, we engaged a broad cross-section of the community and strengthened bonds among core contributors.

We did face a few setbacks and false starts. Some ideas for gathering feedback, such as surveys for K–12 teachers and their students, didn't get past the prototype stage. We also realized the effort required to revise all Reference and Example pages exceeded initial estimates. The writer also realized a bit late that the community testers needed an easier way to preview edits across multiple pull requests. Even so, the lessons learned were invaluable.

Summary

In 2-4 paragraphs, summarize your project experience. Highlight what you learned, and what you would choose to do differently in the future. What advice would you give to other projects trying to solve a similar problem with documentation?

Our project served as a testbed for new ways to increase access using documentation. We hoped to edit more docs, but we're happy with the significant progress made during the season. p5.js simplifies complex tools for beginners. That simplicity requires focused, intentional work. We have three big takeaways that may help similar projects:

- Surround your writer(s) with good people who care. Our writer's work was only possible with the help of thoughtful collaborators with deep expertise.
- Use data from a structured test run to estimate effort and adjust timelines. Our project experience directly informed planning for follow-up efforts.
- Look around. There's a lot to learn from others' experiences. The p5.js docs have many diverse sources of inspiration.

Appendix

If you have other materials you'd like to link to (for example, if you created a contract for working with your technical writer that you'd like to share, or templates for your documentation project, or other open documentation resources, you can list and link them here). The Appendix is also a good place to list links to any documentation tools or resources you used, or a place to add thanks or acknowledgments that might not fit into the sections above.

Created

p5.js Documentation Style Guide

Referenced

Overview: Write the Docs

General: Red Hat Supplementary Style Guide and Ruby on Rails API Documentation Guidelines

Accessibility: MDN What is accessibility? and WordPress Accessibility

Inclusivity: WordPress Inclusivity

Code Style: Airbnb JavaScript Style Guide