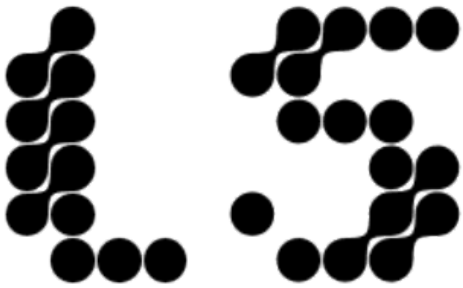


L5 Usability Report

Group 6: HCDE 517 Winter 2026

Kathryn Rambo, Swathi Sridhar, Manish Varrier, Auli Badoni

March 18, 2026

Table of Contents

Executive Summary

The purpose of this study was to evaluate the usability of L5's documentation for experienced creative coders, or anyone with 1+ years of creative coding experience who is highly familiar with p5.js. The client anticipated the majority of L5 users would be p5.js users who are eager to try a new library, hence the focus on this demographic. The study was administered virtually through three scenario-based prompts, wherein each participant was asked to think aloud while using the L5 website documentation to complete tasks such as creating 2D shapes, adding interaction, and saving the artwork.

This study addressed this main overarching research question and the following sub questions:

★ **What aspects of L5 Lua's reference pages are most frustrating to experienced creative coders in resolving errors and progressing independently during real coding workflows? How does this affect users' likelihood of adopting L5 long-term?**

1. To what extent does the user effectively utilize L5 Lua processing documentation to diagnose and find a solution to their error independently?
2. How often do users misinterpret aspects of documentation?
3. To what extent is the information architecture and search functionality effective in supporting users in utilizing reference pages in their project workflows?
4. What are major pain points that lead to external support use such as Github, Browser, etc?
5. What potential factors come into play that affect long term adoption of L5 for experienced creative coders?

The key findings and recommendations are listed below and presented based on priority level and connection to the aforementioned sub-research questions:

#	Finding	Q1	Q2	Q3	Q4	Q5	Recommendation	Priority
F1	p5.js mental models shaped interactivity expectations		X		X		Runnable code blocks within the reference pages	Major Project
F2	Setup friction compounded into most common overall task error				X		Clear examples to run code from the command line and/or anchored jump link at the top of the p5.js users page.	Major Project

F3	p5.js transfer worked but bridge content missing	X			X		Table that compares p5.js/processing to L5, and highlights basic syntax differences.	Quick Win
F4	Sidebar Information architecture mixed reference with other content			X			Reorganize sidebar content into clear functional buckets for better navigation.	Major Project
F5	Shape parameter documentation insufficient for some users	X	X				Small inline diagram showing parameter (x, y, w, h) controls spatially for better comprehensibility.	Quick Win
F6	Search used by 3/6 only			X			Increase visual accessibility and contrast of the search bar for better discoverability.	Fill In
F7	Low-bandwidth design principle raised adoption likelihood					X	Increase visibility of the low-bandwidth design principle through a visual component.	Quick Win

Overview

L5 Description

L5 is a fun, fast, cross-platform, and lightweight implementation of the Processing API in Lua. It is a free and open source coding library to make interactive artwork on the computer, aimed at artists, designers, and anyone that wants a flexible way to prototype art, games, toys, and other software experiments in code.

It is designed to work cross-platform, including on desktop, phone, and tablet. Beyond running fast on modern machines, L5 is optimized for older and lower-powered devices, minimizing resource usage to keep creative coding accessible to everyone. This helps with its goal of building resilient, long-lasting software projects. L5 is built in Lua, a robust but lightweight, long-running, lightning-fast, extensible language.

L5 brings the familiar Processing creative coding environment to Lua, offering some of the best aspects of both Processing and p5.js with some twists of its own. But you don't need to know Processing already to get started with L5. L5 is built on top of the Love2D framework, and offers near-instant loading times and excellent performance while maintaining the intuitive API that makes Processing accessible to artists and designers. L5 is not an official implementation of Processing or the Processing Foundation. It is a community-created project.

Study Goals

The goal of this study is to evaluate how effectively experienced creative coders can utilize the L5 reference documentation during real coding tasks. Specifically, the study examines whether users with prior experience in creative coding environments can independently navigate the documentation, resolve technical issues, and transfer their existing knowledge when learning and using the L5 framework.

Primary Goals

1. The first objective/goal of this study is to assess participants ability to diagnose and resolve errors independently. Creative coding often involves iterative experimentation, where errors and unexpected behaviors occur frequently. Therefore, an important indicator of effective documentation is whether users can identify problems and correct them using the provided resources without external assistance. This study observes how participants approach debugging tasks whether the documentation provides sufficient guidance for problem resolution.

2. The second objective is to evaluate how the participants navigate the documentation during real coding workflows. Rather than assessing documentation comprehension in isolation, the study focuses on how users interact with the reference materials while actively coding. This includes examining how participants search for information, interpret documentation examples, and integrate retrieved information into their ongoing work. Observing these behaviors provides insight into usability and accessibility of the documentation structure.

3. The third objective is to investigate how participants apply prior knowledge from processing and/or p5.js when learning L5. Since L5 is designed for users with backgrounds in creative coding frameworks, it is important to understand whether existing conceptual knowledge supports the learning process. The study examines whether participants rely on familiar concepts, patterns or syntax from these environments and how effectively they transfer knowledge to the L5 context.

Secondary Goals

In addition to the primary objectives, the study aims to identify usability barriers and potential frustrations that may influence whether experienced creative coders choose to adopt L5 in the long term. Even if users are technically capable of completing tasks, issues such as unclear documentation structure, insufficient examples, or difficulty locating information may negatively affect their overall experience. By documenting these barriers, the study seeks to highlight areas where improvements to the documentation could enhance usability and encourage broader adoption of the L5 framework.

Participants Profile

A total of six participants took part in this study. All participants had prior experience in creative coding, specifically with processing and p5.js, which are closely related frameworks to L5. Recruiting participants with this background was important because the study aimed to evaluate how experienced creative coders interact with and learn from the L5 documentation.

Five out of six participants reported more than two years of experience using both processing and p5.js. This level of experience suggests that participants were already familiar with common creative coding concepts such as drawing functions, animation loops, and interactive programming workflows. One of the participants reported between one and two years of experience with both frameworks, representing a slightly less experienced but still knowledgeable creative coding user. Overall, the participant pool consisted primarily of individuals, with intermediate to advanced familiarity with creative coding environments.

Participants were also distributed across two geographical locations. Four participants were based in New York, while two participants were based in Seattle.

Methodology

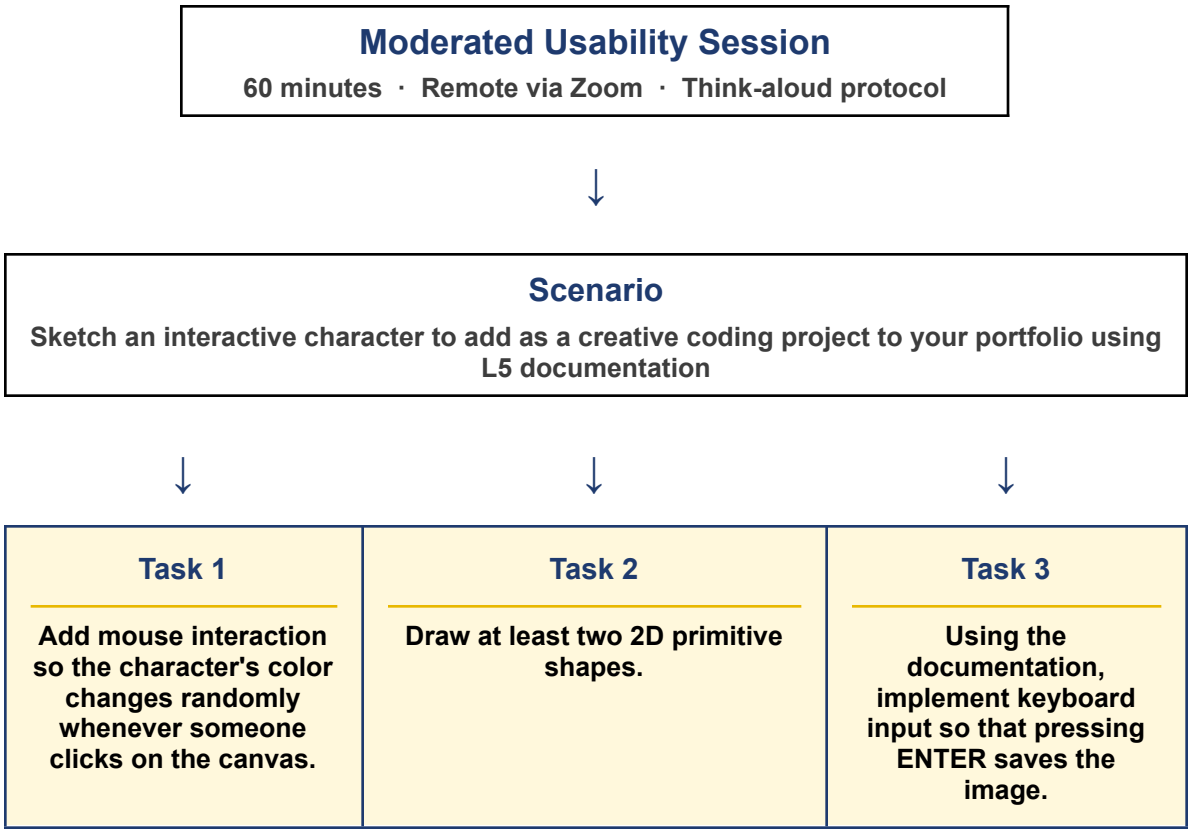
Test Environment

The first phase of our methodology focused on establishing the test environment. Participants received a pre-session email containing preparation requirements and Zoom instructions to facilitate the logistics of the usability testing session.

The session was a moderated, 60-minute remote study conducted via Zoom, during which the participant shared their screen. Prior to recording, the moderator obtained verbal consent from the participant. The session was captured as an audiovisual recording and subsequently transcribed using Otter.AI.

Test Kit

The test kit consisted of a flyer, pre-session screener, a moderated session script, and a post-session survey. The moderated session included one scenario supported by three tasks. The scenario framed the session around building an interactive character project intended for the participant's creative coding portfolio. The format is visualized below.



Metrics Collected

To evaluate participant performance and interaction with the L5 documentation, we collected the following metrics:

1. **Time on Task**

   Time taken to complete each task was recorded as a measure of efficiency. This metric helped identify how easily participants could locate relevant information and implement required functionality. Longer times often indicated challenges in navigation, comprehension, or problem-solving.

2. **Number of Times Using Reference Pages**

   We tracked how often participants accessed official reference pages within the documentation. This provided insight into reliance on structured documentation and whether these pages effectively supported task completion. It also gave us an insight into the value the reference page provides.

3. **Number of Times Using the Search Function**

   The frequency of search usage was recorded to assess the discoverability of information. High reliance on search suggested either that users preferred direct lookup or that navigation through menus and categories was insufficient or unclear. As an

4. **Number of Conceptual Errors**

   We documented instances where participants demonstrated misunderstandings of concepts or misapplied information from the documentation. These errors excluded simple syntax mistakes and instead focused on deeper issues of comprehension and clarity.

5. **Qualitative Observations and Quotes**

   Participant verbalizations from the think-aloud protocol and session transcripts were collected to capture user reasoning, frustrations, and expectations. These qualitative insights provided context to observed behaviors and helped explain underlying usability issues.

6. **Post-Session Survey Responses**

   Participants completed a post-session survey which included Likert-scale questions and one open-ended question. This captured subjective measures such as perceived ease of use, ease of completion of each task, satisfaction, future adoptability, and overall impressions of the documentation.

Justification behind Methodology

We chose to run this study as a moderated remote usability test because it was easier to recruit enough participants within the given timeframe. Using Zoom allowed participants to join from their own setup, which made the sessions more convenient and closer to how they normally use documentation while coding. The moderator was able to guide the session, ask follow up questions, and make sure participants stayed on track without interrupting their natural workflow too much. We also used a scenario based approach so that the tasks felt realistic. By asking

participants to build a small creative coding project for their portfolio, we encouraged them to interact with the documentation in a way that reflects how they would actually use it outside of a study.

The three tasks were designed to cover core features used by creative coders, such as interaction, drawing shapes, and handling input. At the same time, the tasks required participants to apply some of their own logic and coding knowledge instead of just copying from the documentation. This helped us understand how well they could interpret and use the information provided. We also used a think aloud method so participants could explain what they were thinking as they worked, which helped us identify confusion and decision making processes. Finally, we combined measurable data like time on task and errors with qualitative feedback from participants and surveys. This gave us a more complete understanding of both what issues occurred and why they happened.

Data Analysis

We had originally planned to triangulate between time on task, post survey data and qualitative coding. However, after data collection, we decided not to focus on time on task and instead targeted other quantitative metrics, along with post survey data and qualitative coding. Time on task did not adequately reflect participants' coding workflows because debugging, scanning documentation, and other coding methods varied drastically across participants. Hence, other metrics such as number of times using reference pages, number of conceptual errors, and number of times using search function were examined instead.

Raw participant quotes from the Zoom transcripts were qualitatively coded on the FigJam platform. They were initially categorized based on participant number, and eventually divided into 10+ themes, out of which documentation clarity, prior knowledge transfer, experience with L5, and recommendation for L5 emerged as most significant, thus laying the foundation for our findings.

Findings

The following section presents seven findings derived from our data analysis of qualitative coding and collected metrics. Since L5 is a free, open-source library built for broad accessibility, these findings will be evaluated with L5's core design principles considered. Each finding was prioritized by weighing ease of implementation against potential impact on user usability within those constraints, and together they serve as the foundation for the recommendations that follow.

Overview of Findings

Findings derived from observations throughout the moderated session.

Metric	Task 1 — Shapes	Task 2 — Mouse interaction	Task 3 — Save image
Task prompt	Draw 2D primitive shapes	Random color on mouse click	ENTER key saves image
Avg no. of times reference pages visited	**1.5**	**4.5**	**2.0***
Total errors	**2**	**9**	**5 + DNF**
Total searches	**0**	**4**	**5**
Completion	**6/6 completed**	**6/6 completed**	**5/6 (P5 DNF)**

Participant	Code Editor/IDE	No. of times used in-built search feature	No. of times used ctrl+F/cmd+F	What they searched for (exact keyword)
P1	**Text edit**	**0**	**4**	
P2	**Visual Studio (Not VS Code)**	**0**	**0**	**NA**
P3	**VS Code**	**0**	**0**	**NA**
P4	**VS Code**	**3**	**0**	**random, keyboard, save**
P5	**Brackets**	**6**	**0**	**local, save, print, save, false, keypressed**
P6	**Sublime text**	**4**	**0**	**variables, random, key, save**

Findings Breakdown

Finding 1	Users affected	Priority
Existing p5.js mental models shaped expectations for documentation interactivity and editable examples.	**4 / 6**	Would be Nice

Participant Quotations

> *"Felt interactivity was missing from documentation. All of the examples were kind of going on their own, I wish it would respond to your inputs like on the p5 library, like you can put your mouse on this canvas and it'll have an effect on it." -Participant 2*

> *"The examples on the reference page felt dry without intractability. For example, I wish the keyPressed actually checked for what keys you press on the website and updates the little canvas as you type instead of a pre-defined gif." -Post Screener Data*

> *“I wish it would have, like on the p5 library, it responds to your inputs, like you can put your mouse on this canvas and it'll have an effect on it. This is just going on its own, like I would have liked it to, like, if I typed J within the website, it would show up j here as like part of one of the examples.” -Participant 6*

Finding 2	Users affected	Priority
Running L5 projects introduced friction across drag-and-drop and terminal-based workflows	**6 / 6**	Major Project

Participant Quotations

> *"[drag and drop] I wish that it didn't work that way. I wish that there was an easier way to run it. But I think that's more of a problem with my IDE, which is Visual Studio." -Participant 2*

> *"[unsuccessful terminal run] Well, there was another way to do this, right? I'm pretty sure y'all talked about it in the email, right? Actually, dragging that into the Love 2D."*
> *-Participant 3*

Finding 3	Users affected	Priority
Documentation clarity was consistently appreciated, but participants still sought explicit translation between L5 conventions and familiar p5.js/Processing patterns.	**5 / 6**	Quick Win

Participant signal	Type	Quote / observation
Prior knowledge helped	**Positive**	'I was thinking how I did it in p5 and then what functions I used... converted my p5 knowledge to L5.'
Assumption mismatch	**Friction**	'I initially assumed the enter key would be enter but it makes sense that it's return.'
Scope confusion	**Friction**	'Oh, my variable doesn't exist up here, so we're gonna make global variables.'
End declaration unfamiliar	**Friction**	'The end declaration after a function…that's new to me.'
Asked for bridge content	**Gap**	'Maybe a quick cheat sheet… if you're coming from processing, the most salient differences.' P2
Used ChatGPT as bridge	**External**	'I'll use ChatGPT to read the tutorials and give it my goal.' - P1

Finding 4	Users affected	Priority
Sidebar Information architecture mixed reference with other content	**5 / 6**	Major project

Participant Quotations

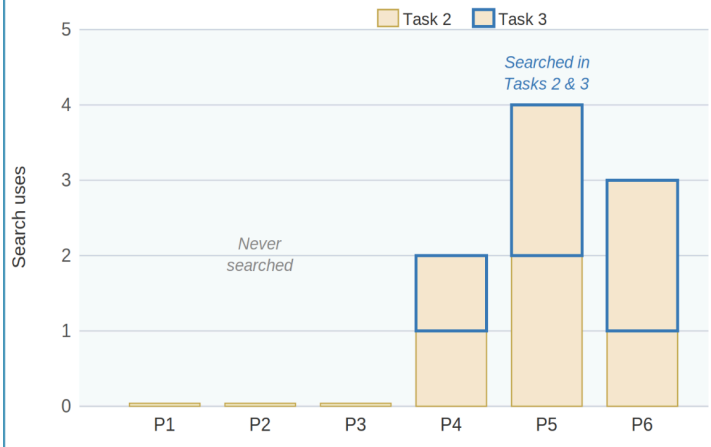
> *"but having reference grouped in under all this information that isn't about coding, reference took me a second to like find right? And I think in some ways, it's still here, right? If I'm in the reference docs, I still see links to download. I still see links to getting started. That's a little strange. I don't necessarily know how to recommend remedying that, but maybe that's kind of a user interface flow issue." -Participant 2*

Finding 5	Users affected	Priority
Shape parameter documentation insufficient for some users. [Second most common error amongst participants]	**4 / 6**	Quick win

Participant response	Sentiment	Quote
Parameters helpful via examples	**Positive**	'W and H, rather than reading the terms out… just looking at an example, super helpful.' P5
Parameters helpful from prior knowledge	**Positive**	'x, y, width and height… generally familiar coming from Processing.' P5
Parameters unclear. Had to guess	**Friction**	'I had to guess which parameter was width and which was height.'
Parameters found late	**Friction**	'I didn't see the parameters at first but then I scrolled down and found it.'
Sub-category labels too vague	**Friction**	'2D primitives.. I wish there was a blurb of what these are under it. I have to open it to find out.'

Finding 6	Users affected	Priority
The search function was utilized by only 50% of participants. The primary use was in debugging navigation flows.	**3 / 6**	Fill in

Finding 6 — Search used by 3/6 participants only; all 9 queries succeeded RQ3



All 9 queries returned a result
"Random" → random()
"Keyboard" → keyboard input
"floor" → floor()
"Local" → local()
"Save" → save()
"Print" → printtoscreen()
"Random" → random()
"Key" → key()
"Save" → save()

Finding 7	Users affected	Priority
Knowledge of L5's low-bandwidth design intent raised long-term adoption likelihood	**2 / 6**	Quick Win

Participant Quotations

> *"If it can run on something like an Arduino, due to its lightweightness, that would be really awesome. That would 100% allow me to answer the question. I think I would like to use L5 frequently." -Participant 5*

> *"I think this is very easy. And I think because it's so lightweight, which is like, I mean, the best thing about Lua, and why I was so excited to see that you all were doing this study is that it's always going to be lightweight like this, and it has so many applications. It's very exciting to see." -Participant 3*

Long Term Adoption Factors

Signal type	Quote	Implication for adoption
Positive Physical computing	'If it can run on something like an Arduino, due to its lightweightness, that would be really awesome. That would 100% allow	Direct adoption intent tied to low-bandwidth device capability

	me to answer the question. I think I would like to use L5 frequently.' P5	
Positive Non-technical accessibility	'I helped my mom with a lot of her computing... it is really nice that you're being so sensitive to each of the platform steps for getting someone going.' P1	Tutorial style and cross-platform sensitivity perceived as unique differentiator
Barrier — Browser instant run	“allure and sort of, like the benefits for people who are just starting out with creative coding is just the fact that you can open p5 JS in browser, like, make a sketch in browser, run it in browser” P6	Browser-based run is a hard expectation; L5's setup friction is a direct adoption barrier

Recommendations and Next Steps

Impact vs. Effort Prioritization Matrix for Recommendations

	LOW EFFORT	HIGH EFFORT
HIGH IMPACT	QUICK WINS • Shape Parameter visualization • Increase saliency of Syntax differences table in Lua Tips section • Increase visibility of low bandwidth design principle	MAJOR PROJECTS • Visual Examples accompanying “running from the command line” instructions • Restructure left sidebar for scannability.

LOW IMPACT		
	FILL-INS • Increase Search bar saliency • Include embedded anchor link in programmer from p5js and processing	WOULD BE NICE • Interactive / runnable code blocks in reference pages

Quadrant Key

Quick Wins	Major Projects	Fill-ins	Would be Nice
Fix immediately	Plan and prioritize	Do when time allows	Deprioritize

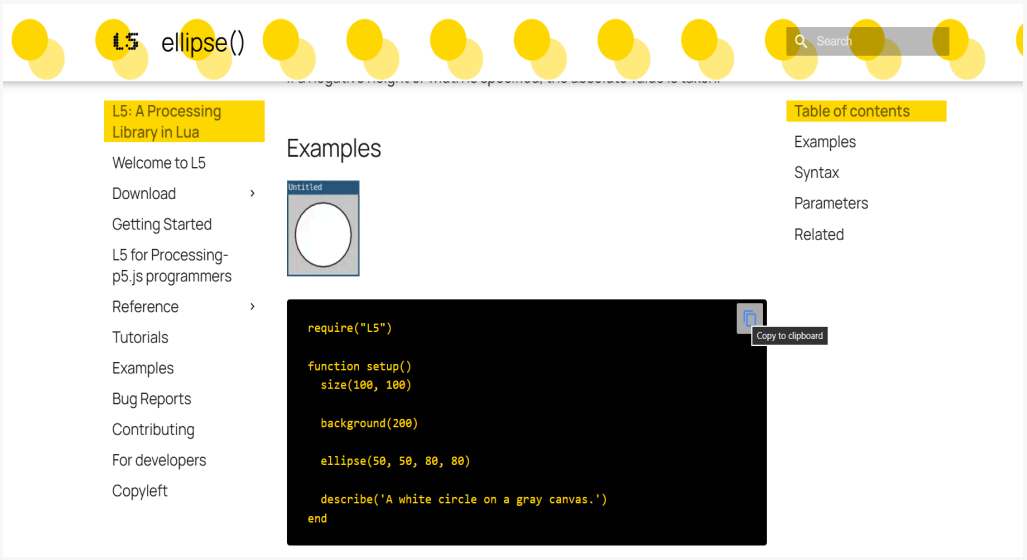
Recommendations Breakdown

1 **Add Interactivity to Reference Page**
Would be Nice

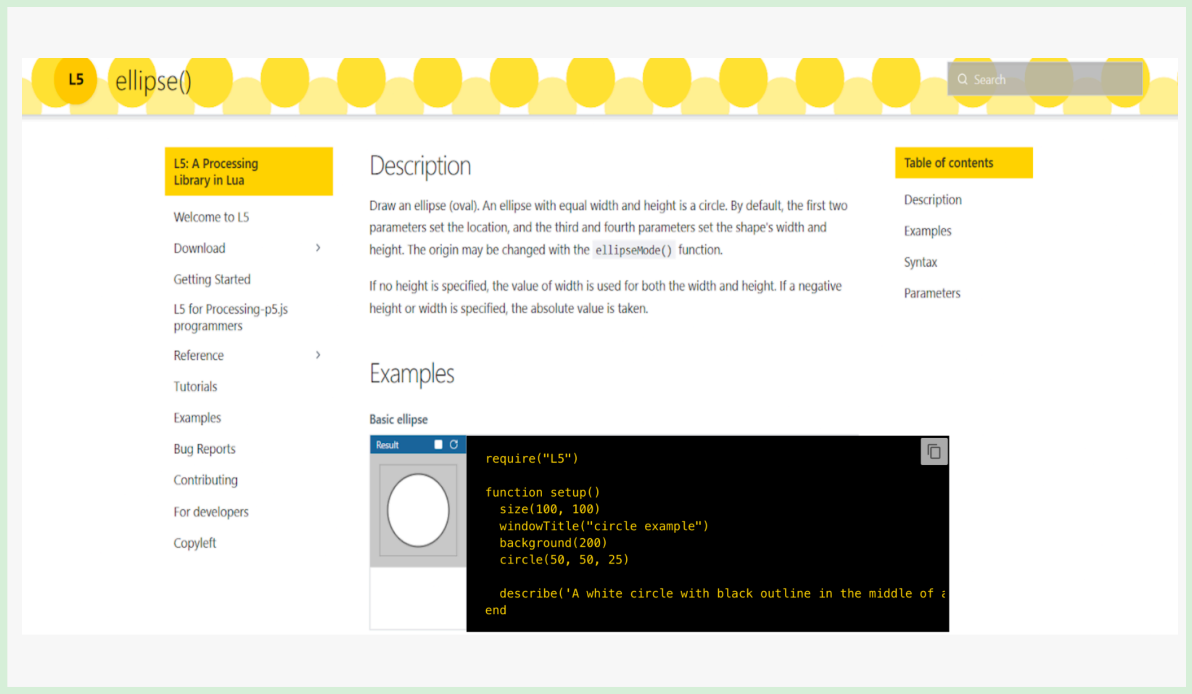
Finding
Participants' mental models, rooted in p5.js, drove demand for interactive, editable components.

Recommendation
Runnable Code Blocks within each reference page.

BEFORE



AFTER



2 Provide Examples in Terminal Documentation

Fill In

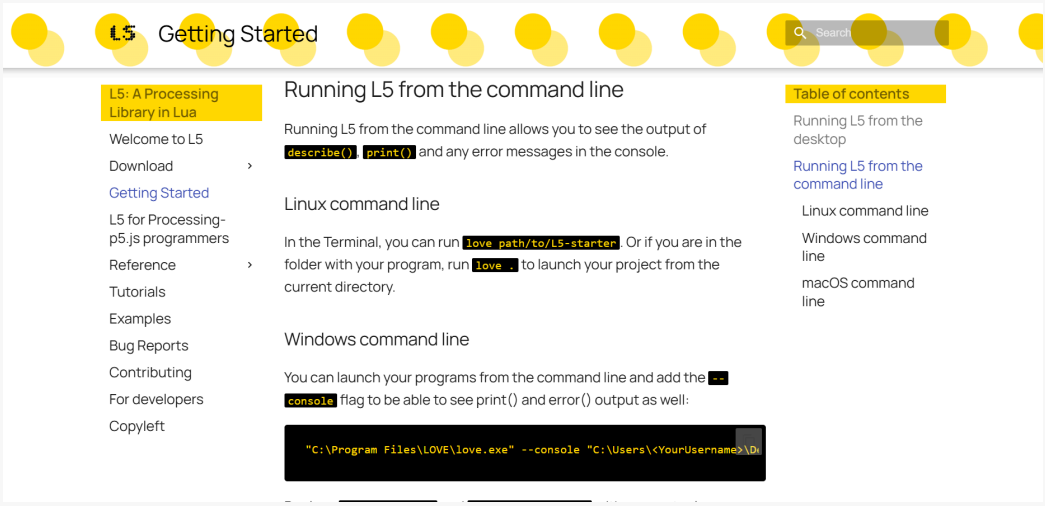
Issue observed

6 out of 6 participants experienced friction with the drag and drop/IDE run within their creative coding workflows. However, 3/6 Participants expressed feeling uncomfortable with using the terminal.

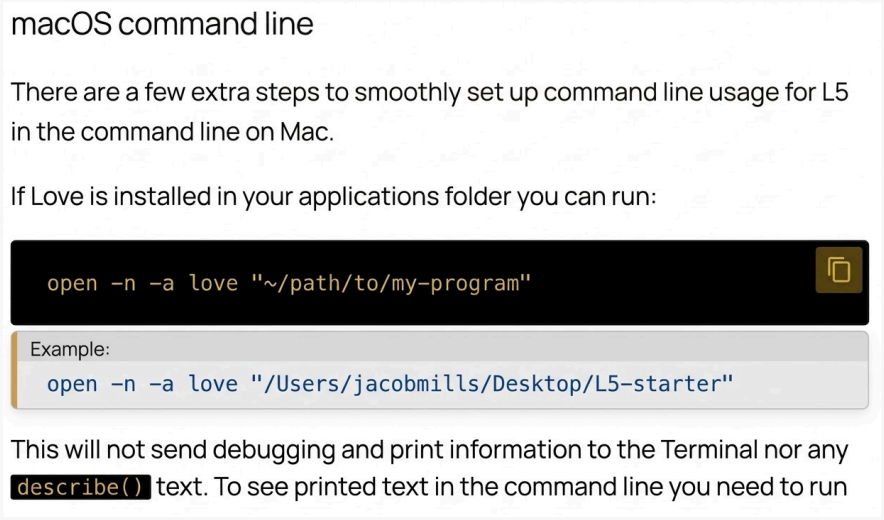
Recommendation

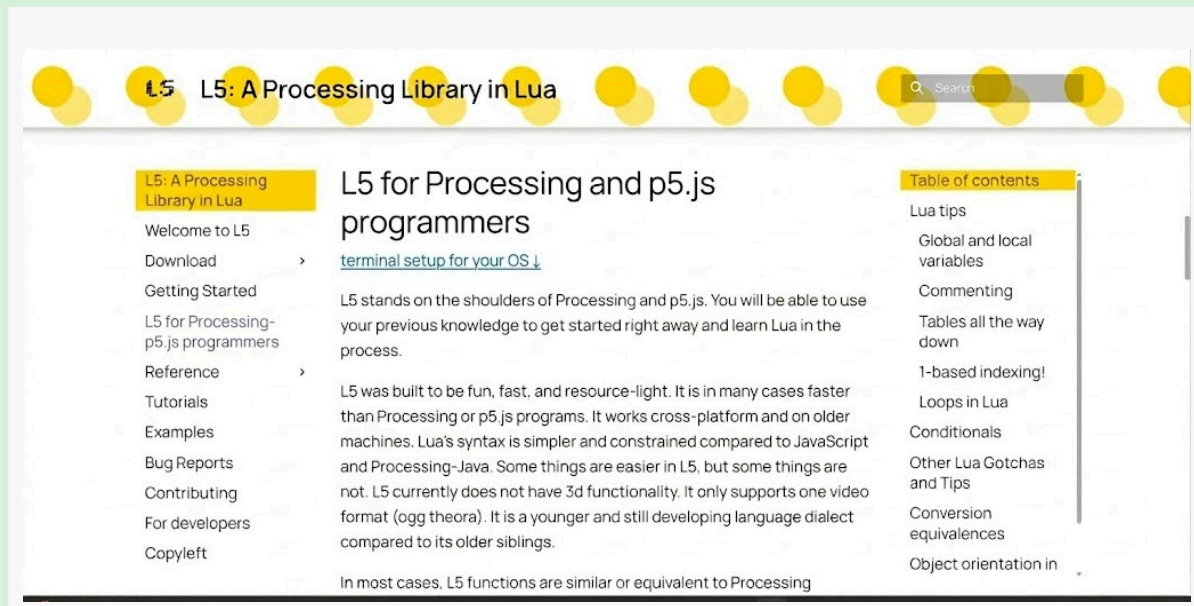
Have an example of running from the command line and/or include an anchored jump link at the top of the L5 Page Processing and p5.js programmers to increase saliency of the option

BEFORE



AFTER





3 Syntax Differences Summary Table in Lua Tips

Quick win

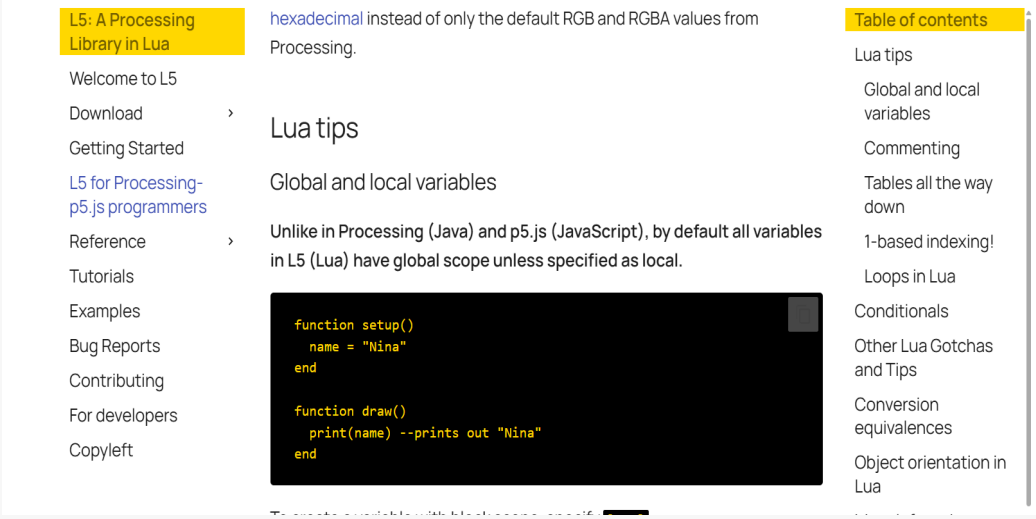
Finding

p5.js transfer worked but the saliency of bridging content was missing

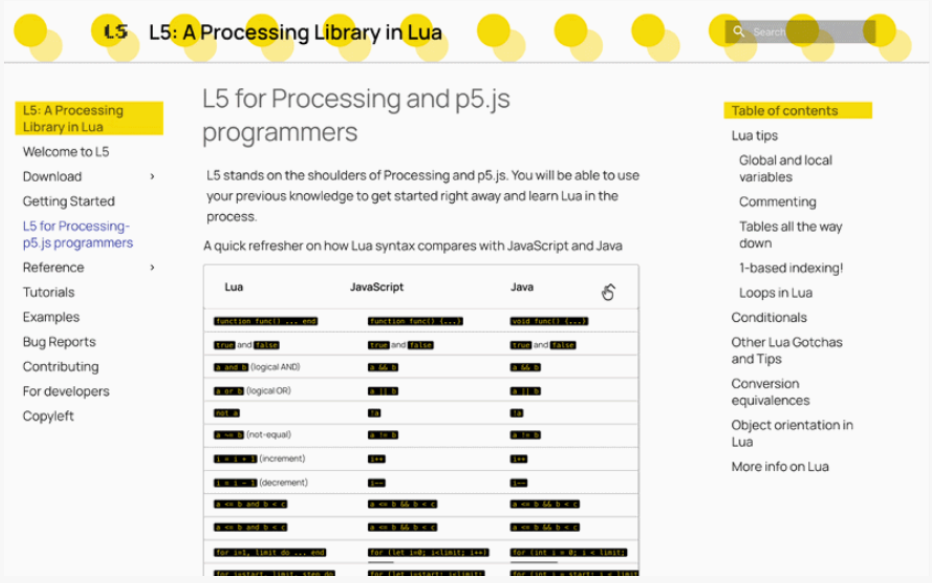
Recommendation

Adding a table that clearly compares p5.js/processing to L5, and highlights basic syntax differences for better clarity and scannability.

BEFORE



AFTER



4 Sidebar Information architecture mixed reference with other content

Major Project

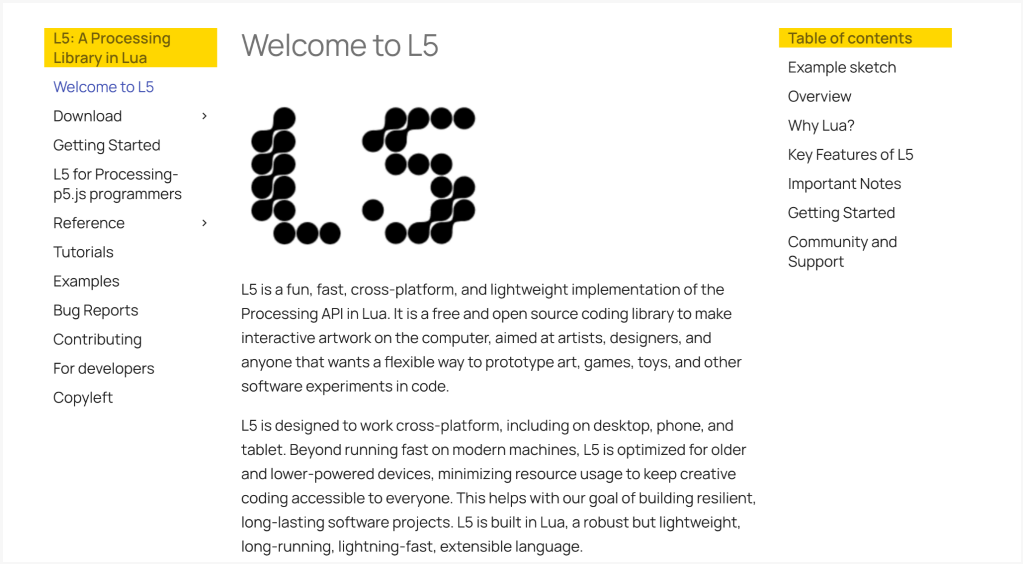
Finding

Sidebar Information architecture mixed reference with other content

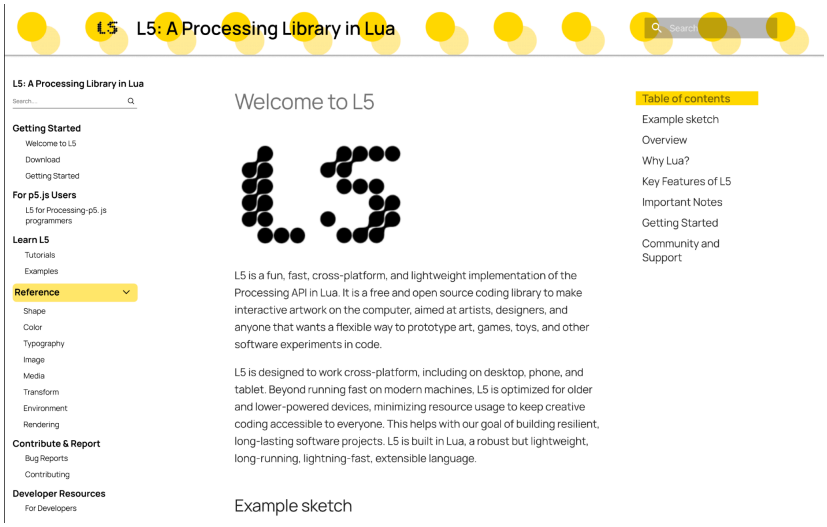
Recommendation

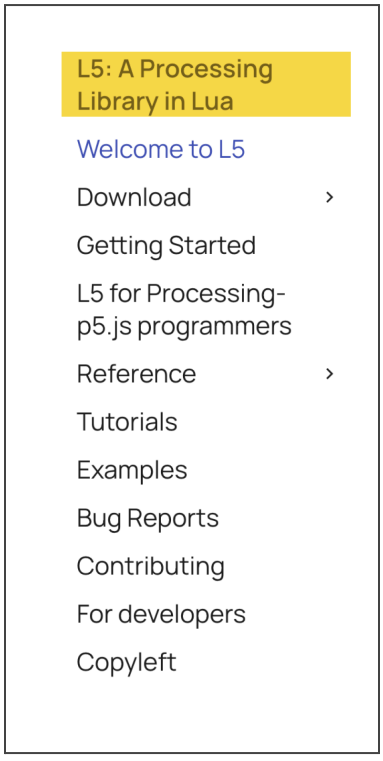
Reorganize sidebar content into clear functional buckets such as Getting Started, Learning Resources, Reference, Contribution, and Developer Resources so users can more easily distinguish documentation's purpose and navigate directly to relevant sections.

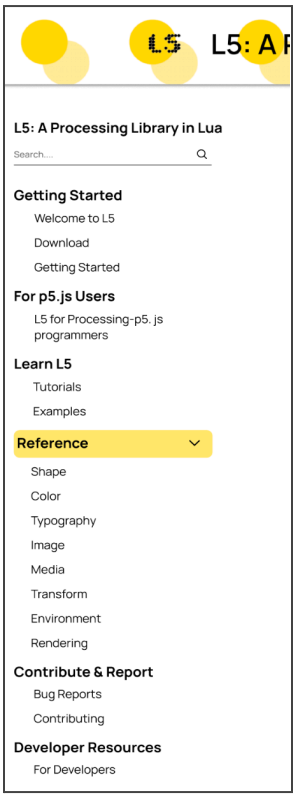
BEFORE



AFTER







Old Navigation Bar (left) vs New Navigation Bar (right)

5 Improve shape parameter documentation clarity

Quick win

Finding

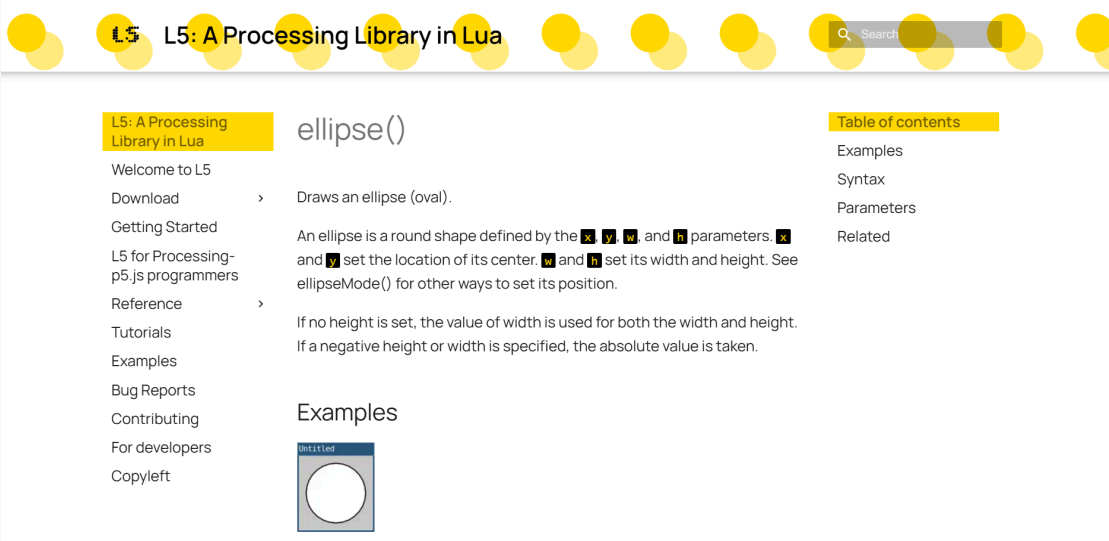
Shape parameter documentation insufficient for some users

Recommendation

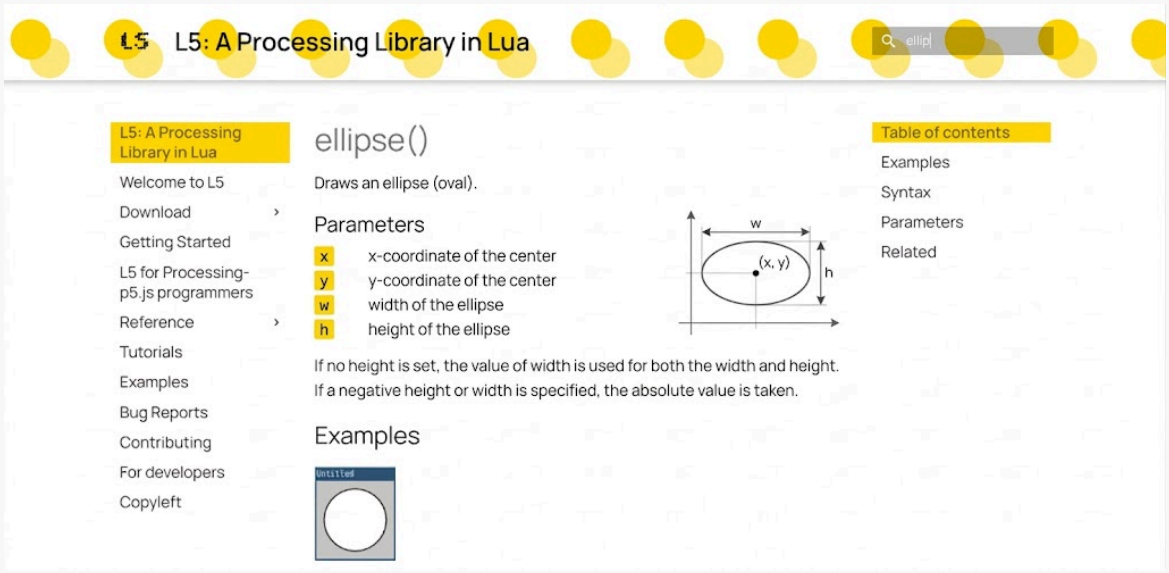
Add a small inline diagram showing what each parameter (x, y, w, h) controls spatially for better comprehensibility and scannability. Move parameters above the fold on reference entries.

*Participants who found them helpful only did so after scrolling past examples.

BEFORE



AFTER



6 Increase Search Bar Saliency

Fill In

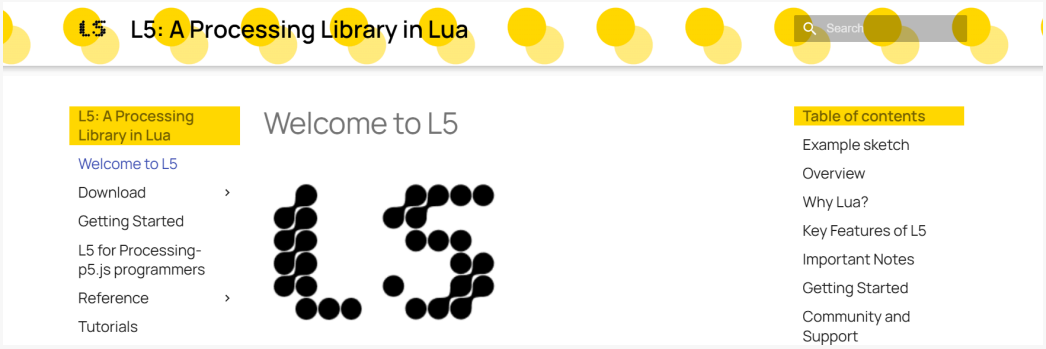
Finding

The search function was utilized by only 50% of participants. The primary use was in debugging navigation flows

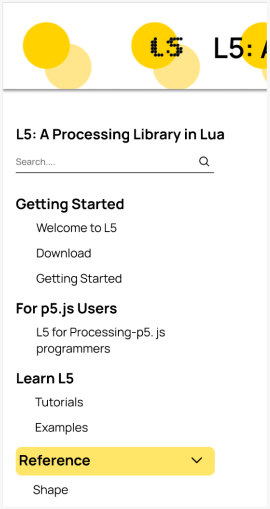
Recommendation

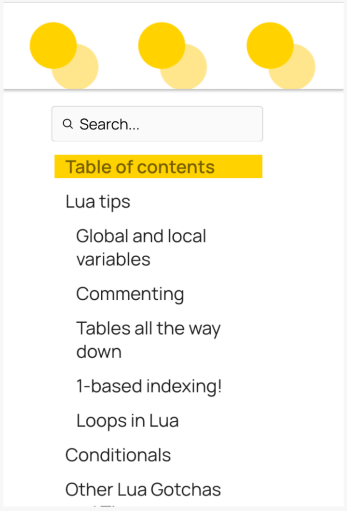
Increase visual accessibility and contrast of the search bar, helping it standout which in turn improves its discoverability

BEFORE



AFTER





Integrate search more directly with sidebar navigation so users can move between category browsing and direct lookup without shifting attention across interface regions.

7 Emphasize Low Bandwidth Design Principle

Quick win

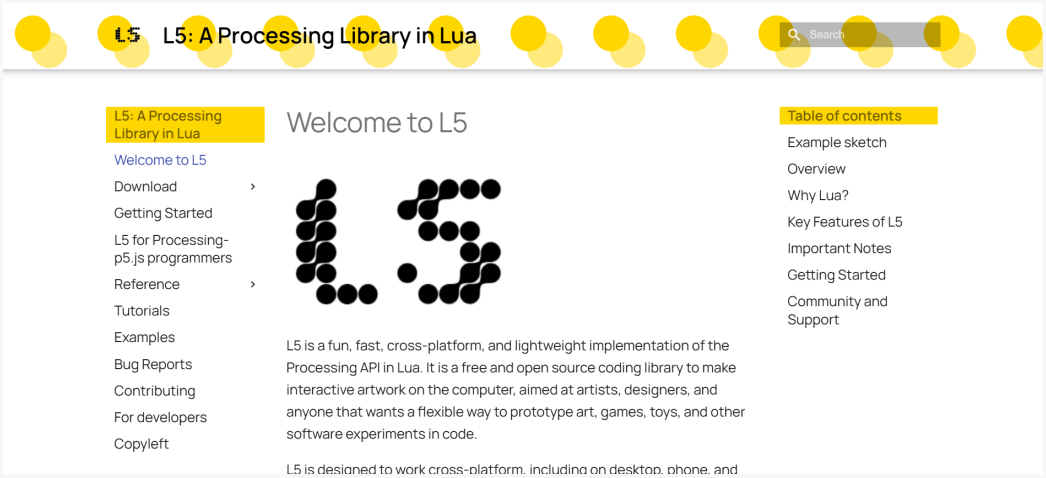
Issue observed

Low-bandwidth intent raised adoption likelihood

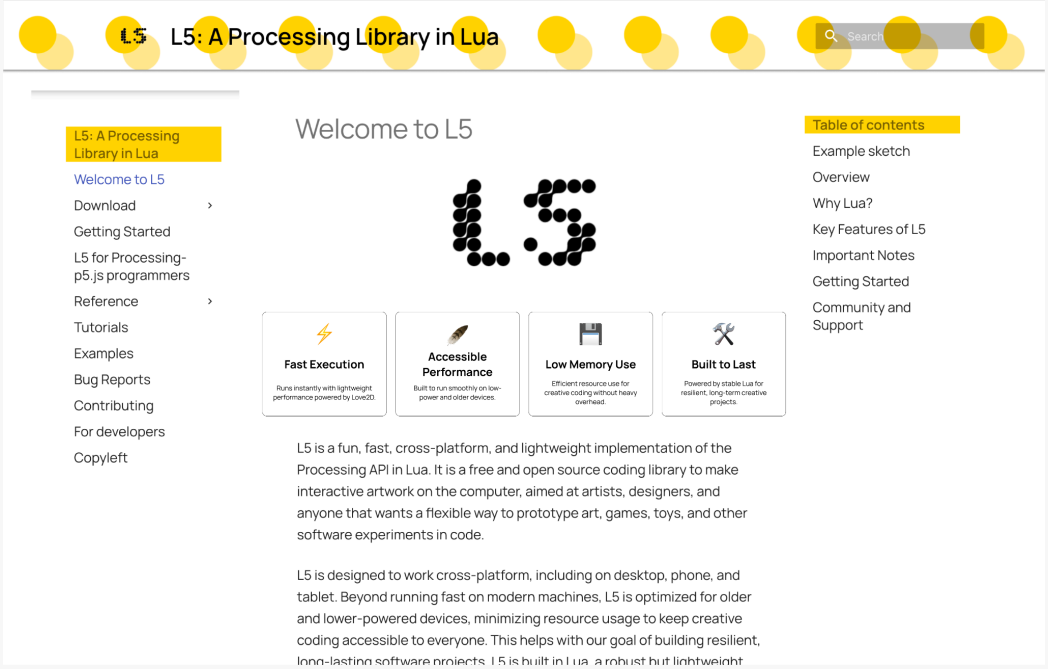
Recommendation

Increase visibility of the low-bandwidth design principle through a visual component

BEFORE



AFTER


L5: A Processing Library in Lua
Search
L5: A Processing Library in Lua
Welcome to L5
Download
Getting Started
L5 for Processing-p5.js programmers
Reference
Tutorials
Examples
Bug Reports
Contributing
For developers
Copyleft
Welcome to L5
Fast Execution
Runs instantly with lightweight performance powered by Love2D.
Accessible Performance
Built to run smoothly on low-power and older devices.
Low Memory Use
Efficient resource use for creative coding without heavy overhead.
Built to Last
Powered by stable Lua for resilient, long-term creative projects.
L5 is a fun, fast, cross-platform, and lightweight implementation of the Processing API in Lua. It is a free and open source coding library to make interactive artwork on the computer, aimed at artists, designers, and anyone that wants a flexible way to prototype art, games, toys, and other software experiments in code.
L5 is designed to work cross-platform, including on desktop, phone, and tablet. Beyond running fast on modern machines, L5 is optimized for older and lower-powered devices, minimizing resource usage to keep creative coding accessible to everyone. This helps with our goal of building resilient, long-lasting software projects. L5 is built in Lua, a robust but lightweight
Table of contents
Example sketch
Overview
Why Lua?
Key Features of L5
Important Notes
Getting Started
Community and Support

Findings Summary

RESEARCH QUESTION

What aspects of L5's reference pages are **most frustrating** to experienced creative coders in resolving errors and progressing independently **during real coding workflows**, and how does this affect their likelihood of **adopting L5 long-term?**

RQ1: To what extent do users utilize L5 docs to diagnose and resolve errors independently?	RQ2: How often do users misinterpret aspects of documentation?	RQ3: To what extent is the IA and search effective in supporting users in their workflows?	RQ4: What are major pain points that lead to external support use?	RQ5 What potential factors come into play that affect long term adoption of L5 for experienced creative coders?
F3 **p5.js transfer worked, but bridging content was hard to find** Users leveraged p5.js knowledge to attempt independent resolution, but absent explici translations blocked them from completing it unaided.	F1 **p5.js mental models shaped interactivity expectations** Prior p5.js expectations led users to view static examples as incomplete or mistake them as initially broken.	F4 **Sidebar IA mixed reference with non-reference content** The blended sidebar structure undermined navigation. Users struggled to locate reference pages during active coding workflows.	F2 **Setup friction compounded into the most common task error overall** Drag-and-drop drove one users to use external help (ChatGPT) and most often asked question to the moderator	F7 **Low-bandwidth intent raised adoption likelihood** Users who became aware of the low bandwidth design principle of L5 expressed they would adopt long term knowing the library was internationally designed for this purpose
F5 **Shape parameter docs insufficient for some users** Ambiguous parameter labels prevented independent diagnosis. Users could not confirm their understanding without guessing or seeking outside help.	F5 **Shape parameter docs insufficient for some users** Unclear parameter naming caused users to misinterpret which value mapped to width vs. height, resulting in errors.	F6 **Search used by only 3 of 6 participants** Low search adoption reveals that the IA did not naturally guide users to utilize search for information	F3 **p5.js transfer worked, but bridging content was hard to find** When L5-to-p5.js translations were absent, users fell back on external resources to bridge the gap themselves.	
		F6a **Search primarily used reactively in debugging**		

Conclusion

This study found that while L5 documentation was generally understandable for experienced creative coders, several usability challenges affected early task completion and confidence. The most prominent issues related to environment setup, syntax translation between familiar languages and Lua, and navigation within documentation sections. Participants frequently relied on prior knowledge from p5.js or Processing, indicating that existing mental models strongly shaped how they interpreted and used L5 resources. Small structural changes to documentation and onboarding materials could therefore significantly improve initial usability and reduce friction.

Study Reflection

A key reflection from this study is that usability was influenced not only by the documentation itself, but also by participants' technical contexts, including their development environments, device setups, and prior programming backgrounds. Although participants represented a versatile group of creative coders, most were unfamiliar with Lua, which introduced additional interpretation effort during tasks. This highlighted how documentation for creative coding tools must account for variability in both technical familiarity and setup conditions.

Future Research / Next Steps

Future research could examine how L5 supports users across a broader range of technical experience, particularly creative coders with limited prior exposure to programming languages beyond JavaScript-based environments. Further studies could also explore the long-term effect of interactive examples, onboarding resources, and clearer setup guidance on continued adoption. Evaluating documentation use over repeated sessions rather than first-use tasks may provide deeper insight into how users build confidence and integrate L5 into their creative workflows.

Appendix

Contents

Appendix A

Recruitment Flyer

Appendix B

Screener

Study description

Title: L5 Lua Study Screener

We are a team of students at the University of Washington, conducting research on the documentation in the L5 creative processing library. Through our research, we hope to better understand the usability of L5 documentation, and more specifically, how experienced creative coders are navigating it. Participation will include one 45–60 minute moderated test administered either in-person or virtual, depending on your location and availability.

If selected, we will reach out to you via email. Questions? Contact Manish Varrier (manishrv@uw.edu).

Section 1 — Background

1. Email

○ _____

2. Where are you currently located?

○ _____

Section 2 — Creative coding experience

1. How long have you been doing creative coding?

○ Less than 3 months
○ 3–6 months
○ 6 months–1 year
○ 1–2 years
○ 2+ years

2. Which creative coding tools/libraries have you used? (Select all that apply)

☐ p5.js
☐ Processing (Java mode)
☐ OpenFrameworks

☐ Three.js
☐ Unity
☐ Other: _____
☐ I have not used any creative coding tools

3. How long have you used p5.js or Processing?
○ Less than 3 months
○ 3–6 months
○ 6 months–1 year
○ 1–2 years
○ 2+ years
○ I have not used these tools

Section 3 — Technical comfort

1. How comfortable are you reading and debugging code?

1 — Not comfortable at all	2	3	4	5 — Very comfortable
○	○	○	○	○

2. How often do you use documentation or reference pages while coding?

Never	Rarely	Sometimes	Often	Always
○	○	○	○	○

Section 4 — L5 & Lua experience

1. Have you ever used the L5 Lua creative coding library?
○ No
○ Yes, briefly experimented
○ Yes, used for multiple projects
○ Yes, I am highly experienced

2. How familiar are you with the Lua programming language?
○ Not familiar at all
○ Slightly familiar
○ Moderately familiar

- ○ Very familiar
- ○ Expert

Section 5 — Logistics

1. What operating system do you primarily use?

- ○ macOS
- ○ Windows
- ○ Linux
- ○ Other: _____

2. Are you comfortable installing software and troubleshooting minor setup issues independently?

- ○ Yes
- ○ No

3. Are you available for a 45–60 min moderated usability session?

- ○ Yes
- ○ No

4. Are you comfortable walking through your experience during the interview (sharing screen or describing steps)?

- ○ Yes
- ○ No

5. When are you generally available for a 45–60 min interview? (Check all that apply)

Time zone: _____

Days: ☐ Monday ☐ Tuesday ☐ Wednesday ☐ Thursday ☐ Friday ☐ Saturday ☐ Sunday

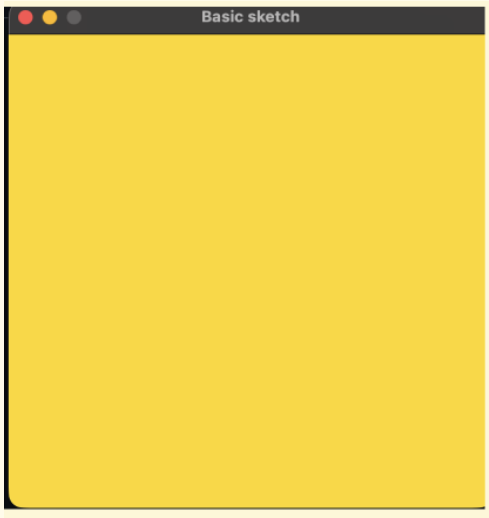
Times: ☐ Morning (9am–12pm) ☐ Afternoon (12pm–5pm) ☐ Evening (5pm–9pm)

Appendix C

Pre-Test Requirements — Email Template

Hello!
Thank you for filling out our screening questionnaire. You've been selected to participate in our usability study for **L5 (Lua)**.
Before we meet, please complete the following:

- Ensure you can join a **Google Meet** call and **share your screen**
- Install **Love2D** and **L5** on your device - **Download Instructions**
- Confirm you can run the **starter program** (output shown below). You can download the starter program using this **link** or you can find it in the download instructions linked above. You can also find instructions on how to run the code **here.**



If you run into any trouble during setup, feel free to reach out; we're happy to help.

Thank you,
Auli, Swathi, Kathryn, Manish

Appendix D

Usability Testing Script & Guide

Opening

Hello, thank you so much for joining today. We really appreciate you taking time out of your day to participate. How are you doing today?

My name is (Primary moderator) and this is (secondary moderator/notetaker). We are going to walk through three scenario-based tasks with you today. You will be exploring parts of a website related to L5 Library. The goal of our research is to evaluate whether the L5 documentation is intuitive and helpful for experienced creative coders.

If selected unfamiliar with the library in screener, give participant short rundown on L5:

L5 is a cross-platform, lightweight implementation of the Processing API in Lua. It is a free and open source coding library to make interactive artwork on the computer, aimed at artists, designers, and anyone that wants a flexible way to prototype art, games, toys, and other software experiments in code.

This should take between 45 to 60 minutes. This is not a test of you or your skills — there are no right or wrong answers. Anything that might feel simple or frustrating is helpful for us to hear. You are welcome to take a break or stop participating at any time.

Do we have your consent to record this session? These recordings will be kept confidential and will be for our team's reference only.

Ask participant permission to record. If virtual, ask them to pull up the L5 webpage, open Lua software, and screen-share. Drop the website link in the chat: https://l5lua.org/

Think-aloud instructions

Alright, I've started recording. As I said earlier, there are three tasks. Throughout the session, I'll ask you to think out loud as we go through each task. When thinking aloud, consider:

— What are you trying to do?

— What do you expect to happen?

— What are you noticing, questioning, or feeling?

We understand this might feel unnatural initially, so we might gently remind you to keep talking or prompt you with a question during the session. Do you have any questions so far?

Here is the scenario for the tasks you will be completing today

Imagine you're creating a small interactive sketch that you plan to use as part of a future creative coding portfolio piece. You have decided to try using the L5 library to experiment with a new environment. Your goal is to build a simple interactive sketch that responds to user input and allows you to export an image of your work.

Task 1 — Creating a character / shapes

Task 1

Using the L5 Lua website, draw at least two 2D primitive shapes. You may use any documentation or reference pages you feel are helpful. Feel free to browse through the website, and remember to verbalize your thoughts as you scroll.

Post-task questions:

1. How easy or difficult was it to find the information you needed?
2. Was anything confusing about the reference pages?
3. Did anything work differently than you expected?
4. How does this compare to your experience using p5.js or Processing?

Task 2 — Adding mouse interaction

Task 2

Now that you have drawn out a character, you want to make it interactive. Using the L5 documentation, implement mouse interaction so that each click changes the character to a random color.

Post-task questions:

1. Was the documentation clear?
2. Did you rely on prior knowledge from p5.js?

Task 3 — Saving the image / character

Task 3

Now that the character is interactive, you'd like to save your artwork. Using the documentation, implement keyboard input so that pressing ENTER saves the image.

Post-task questions:

1. Did the terminology match your expectations? How so?
2. Was there anything you felt was missing from the documentation?

Final reflection

1. Overall, how intuitive did you find the L5 documentation?
2. What were the biggest challenges?
3. What do you think worked well for you?
4. At any point, did you feel unsure about where to look next?
5. How confident would you feel in your ability to debug if you ran into an error?
6. Follow-up: Could you expand on [X]?

Closing

Instruct participant to stop screen share.

Lastly, we'll share a brief 1–2 minute follow-up survey in the chat. This is just to gain some final thoughts on how your experience went today.

That concludes our session. Thank you so much for your time and thoughtful feedback. Your input will directly help improve the L5 documentation.

Let the secondary moderator ask any questions they might have, then invite the participant to ask questions.

Thank you again for your participation. If you have any additional questions or thoughts, please feel free to reach out to any of us. We hope you have a great rest of your day.

Appendix E

Post-Test Survey

Title: L5 Post-Study Survey

Thank you so much for taking the time to participate in our L5 research study! Your feedback is valuable in helping us understand how experienced creative coders navigate L5 workflows.

Q1 — Task difficulty

Please indicate your level of ease/difficulty with the tasks completed today.

	Very Difficult	Difficult	Somewhat Difficult	Neither	Somewhat Easy	Easy	Very Easy
Task 1 — Creating character	○	○	○	○	○	○	○

Task 2 — Adding mouse interaction	○	○	○	○	○	○	○
Task 3 — Saving image/character	○	○	○	○	○	○	○

Q2 — Agreement statements about L5

Please indicate how much you agree or disagree with the following statements about L5.

	Strongly disagree	Disagree	Somewhat disagree	Neither	Somewhat agree	Agree	Strongly agree
I think I would like to use L5 frequently	○	○	○	○	○	○	○
I found L5 unnecessarily complex	○	○	○	○	○	○	○
I thought L5 was easy to use	○	○	○	○	○	○	○
I thought there were too many inconsistencies in L5	○	○	○	○	○	○	○
I found L5 cumbersome to use	○	○	○	○	○	○	○
I felt very confident using L5	○	○	○	○	○	○	○

Q3 — Open response

Is there anything not captured above that you would like to share about your experience?

Appendix F

Data Logging / Note-Taking Template

Note-taking Template

- [ ] **Screen recording Started?**

Participant Name/ID

Operating System

- [ ] **Mac**
- [ ] **Windows**
- [ ] **Linux**

Task 1

- [ ] Pain Point?
- [ ] Utilize Search? How often # ?

# of time Reference page

# of Misinterpretations

Task 2

- [ ] Pain Point
- [ ] Utilize Search? How often # ?

# of time Reference page

# of Misinterpretations

Task 3

- [ ] Pain Point
- [ ] Utilize Search How often # ?

# of time Reference page

# of Misinterpretations

Other Notes

Long Term Adoption

Appendix G

Raw Note Taking

Number of times Reference Pages utilized

Participants	*Task 1*	*Task 2*	*Task 3*

1	*2*	*6*	*1* *Read the source code*
Pages Used	*Square*	*Make a variable,* *Events -> mouse → mouse clicked → random seed* *Variable declaration* *Mouse pressed* *Getting Started*	*Key Pressed*
2	*2*	*3*	*3*
Pages Used	*Square, Ellipse*	*1. Cursor → events → Mouse Button → Mouse Clicked* *2. Color → Fill* *3. Color → Random → Color*	*save()* *Events → keypressed()* *save)(*
3	*1*	*4*	*1*
Pages Used	*Getting started → reference → tutorial → welcome to coding → calling functions*	*Calling function* *Events → mouseclicked()* *Getting Started* *Random ()*	*I/O → save() → keyboard → save → key() → keypressed()*
4	*2*	*3*	*3*
Pages Used	*Shape → Circle* *Rectangle*	*mouseclicked() → mousepressed* *colormode() → cursor() → color()* *random()*	*Keyboard input* *if()* *save()*
5	*1*	*6*	*5 (DNF TASK)*
Pages Used	*Ellipse*	*Events → mouse → mouseclicked()* *color()* *floor()* *Operators* *if()* *Local*	*Reference → keyboard events → key()* *Key() → keypressed()* *save()* *printtoscreen()*
6	*1*	*5*	*2*
Pages Used	*Reference → primitive → square()*	*Color → color()* *Setting → fill() → stroke() → fill()* *mouseclicked()* *Creating variables* *mouseclicked()* *random()*	*key()* *save()*

Number of times search bar was used

Participants	*Task 1*	*Task 2*	*Task 3*

1	*0*	*0*	*0*
2	*0*	*0*	*0*
3	*0*	*0*	*0*
4	*0*	*1*	*1*
How?	*n/a*	*“Random” → fandom()*	*“Keyboard” → keyboard input*
5	*0*	*2*	*2*
How?	*n/a*	*“floor “ —> floor()* *“Local” → local()*	*“Save” → save()* *“Print” → printtoscreen()*
6	*0*	*1*	*2*
How?	*n/a*	*“Random” → random()*	*“Key” → key()* *“Save” → save()*

Number of Errors per task

Participants	*Task 1*	*Task 2*	*Task 3*
1	*0*	*1*	*0*
Error Type/ Debug strategy	*n/a*		*n/a*
2	*1*	*0*	*1*
Error Type/Debug strategy	***Type -** Drag and Drop*	*n/a*	*Pass the string*
3	*0*	*1*	*1*
Error Type/Debug Strategy	*n/a*		
4	*0*	*1*	*0*
Error Type/ Debug Strategy	*n/a*		*n/a*
5	*0*	*3*	*DNF*
Error Type/ Debug Strategy	*n/a*	*Used operators to debug*	
6	*1*	*3*	*2*
Error Type/ Debug Strategy		*Type Global vs local*	*Type End*

Appendix H

Post Survey Raw Data

		Task Difficulty (1–7)			Post-Study Impressions (1–7)						
Participant	Date	T1 Create	T2 Mouse	T3 Save	Use Freq.	Not Complex	Easy to Use	Consistent	Not Cumber.	Confident	Comments / Additional Feedback
P1	2/26/2026	**6**	5	**7**	**6**	4	**7**	1	1	**6**	*https://l5.aesthetic.computer/ — try my tester integration page*
P2	2/27/2026	**7**	**6**	**7**	**7**	2	**6**	1	3	5	*I love this project!!!*
P3	3/1/2026	**7**	**6**	**7**	**7**	2	**6**	1	3	**6**	—
P4	3/2/2026	**6**	5	4	**6**	1	**7**	3	2	**6**	—
P5	3/3/2026	**7**	**6**	2	**7**	1	5	2	4	5	*I think my answers to impressions of L5 would significantly change if I were to use L5 for a while.*
P6	3/3/2026	**7**	**6**	**7**	**6**	1	**7**	1	2	**7**	*The examples on the reference page felt dry without interactability. For example I wish the keyPressed examples could be tried inline.*
Mean		**6.7**	**5.7**	**5.7**	**6.5**	**1.8**	**6.3**	**1.5**	**2.5**	**5.8**	