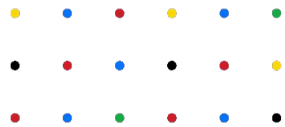




How To Accessibility If You're Mostly Back-End

Creating an inclusive tech environment for colleagues of all kinds



Hello!



Hilary Stohs-Krause

Madison, Wisconsin, USA

- Senior software engineer at Red Canary
- Has an adorable ragdoll cat, Gus
- Dresses up for the Renaissance Faire

Overview

1. Intro
2. What I mean by “accessibility”
3. Specifics: The code
4. Specifics: General practices
5. Wrap-up

1. INTRO

Why do back-end devs need to worry about accessibility?





BUT ALSO INTERNAL USERS!

16%

People globally with
some kind of disability²⁸

20%

People who have
dyslexia²

8%

Men who are colorblind³

There's a strong chance this includes one or more of your coworkers.



Accessibility improvements benefit everyone

- Alexa⁵
- Audiobooks⁶
- Electric toothbrushes⁷

2. DEFINING “ACCESSIBILITY”

Going beyond captions and screenreaders



WHAT IS DISABILITY?

“

A disability is any condition of the body or mind that makes it more difficult for the person with the condition to do certain activities and interact with the world around them.”

U.S. Centers for Disease Control⁸

What states fall under “accessibility”?



- Long-term disability
 - EX: Blindness, ADHD, paralysis
- Temporary disability
 - EX: Broken arm, long covid, whiplash

There are also experiences or conditions that can make things more difficult

Think about coworkers who ...

- Have a different first language than the rest of the team
- Are early career programmers
- Have a baby at home
- Use a different tech stack in their current role than at previous jobs

MORE THAN JUST THE UI

“

We make products accessible, but not the processes by which they are built. Our development practices themselves are inaccessible.”

Julia Ferraioli, software engineer at AWS with limited vision⁹



— IF THERE'S ONE THING YOU TAKE AWAY:

Avoid Abbreviations / Acronyms

- Decreases need for context or interpretation²⁰
- Prevents confusion and misunderstanding
- “Can you bug TSE to connect with the CSM about AWS and CB?”

3. THE CODE

Readability, Naming and Linters



THE CODE

Readability

- Balance between meaning and length⁹
 - Verbose: Easier to understand
 - Shorter: Easier to skim

- Pronounceable words
 - Better for screenreaders
 - Makes pairing discussion easier

```
1
2  failed = ((current_failure_count + 1) >= MAX_FAILURE_COUNT)
3  old = endpoint_sync.created_at < MAX_RUNTIME_MINUTES.ago
4
5  if failed || old
6    | # do something
7  end
8
```

TOO SHORT

```
1
2  has_reached_max_failures_before_finishing = ((current_failure_count + 1) >= MAX_FAILURE_COUNT)
3  has_been_running_longer_than_acceptable = endpoint_sync.created_at < MAX_RUNTIME_MINUTES.ago
4
5  if has_reached_max_failures_before_finishing || has_been_running_longer_than_acceptable
6    | # do something
7  end
8
```

TOO LONG

```
1
2  reached_max_fails = ((current_failure_count + 1) >= MAX_FAILURE_COUNT)
3  running_too_long = endpoint_sync.created_at < MAX_RUNTIME_MINUTES.ago
4
5  if reached_max_fails || running_too_long
6    | # do something
7  end
8
```

JUST RIGHT

THE CODE

Naming

- “Let the code be unambiguous and self-documenting.”¹⁰
- Use words for placeholders in loops / iterations (vs. letters)
- Choose existing method approaches that read like sentences

```
1  
2  cb = message.gsub("\n", "")  
3
```

OK

```
1  
2  carbon_black = message.delete("\n")  
3
```

BETTER

THE CODE

Linters



- Code is consistent and predictable
- Rules are written down for everyone to reference
- For example, Red Canary uses:
 - Rubocop
 - HAML-lint
 - ESLint
 - Hadolint
 - SCSS-Lint

THE CODE

General

- Line length
 - Between 75 to 100 characters
- Indentation
 - All about balance¹²

```
1  
2 = f.input :cat_collection, as: :check_boxes,  
3 | | | | | | | | | | collection: @cat_collection,  
4 | | | | | | | | | | label: false  
5
```

OK

```
1  
2 = f.input :cat_collection,  
3 | | | | | as: :check_boxes,  
4 | | | | | collection: @cat_collection,  
5 | | | | | label: false  
6
```

BETTER

4. GENERAL PRACTICES

Documentation



Documentation

Traditional accessibility considerations

- Don't use images of text, code samples or terminal output: use the actual text.
- Use semantic HTML, whether it's via markdown, WYSIWYG or straight code
- Include captions or transcripts for video documentation

Documentation

Keep it simple

Simple writing helps a wide variety of people, including those:

- With cognitive considerations
- Whose primary language isn't the same as the team's
- Who are unfamiliar with the concepts you're communicating
- Who are stressed or multitasking and have limited attention spans

Documentation

Keep it simple



- Avoid ampersands²¹
- Keep sentences short (15 to 20 words)²⁴
- Test reading level of your text / get AI suggestions for simplifying it:
 - Hemingway²² (English only; free)
 - LanguageTool²⁹ (30 languages; paywall)
 - ParaphraseTool³⁰ (100+ languages; free tier)


Documentation

Additional tips and tricks

- Link text should be informative, not “Click here”²⁰
- Write dates / times in unambiguous ways
 - May 10th: Is it “05/10/2024” or “10/05/2024”?
- Avoid using socially-charged terms for technical concepts²¹
 - “allowlist” vs. “whitelist”
 - “main” vs. “master”
 - “core feature” vs. “native feature”

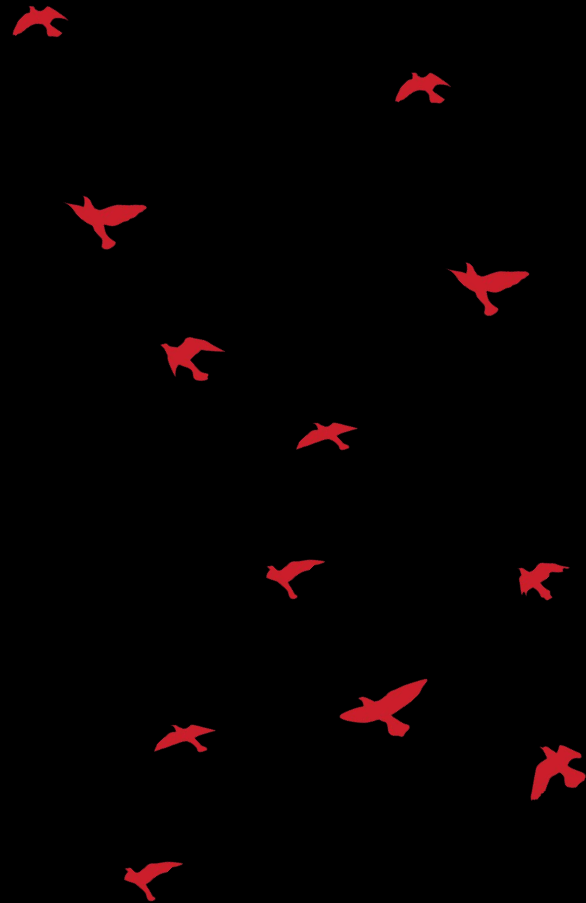
Documentation

Overall communication

- Be careful of assuming cultural context
- Watch for gender assumptions (or exclusion)
- Add pronouns and name pronunciation to profiles 
- Avoid phrases like “It should be easy.”

4. GENERAL PRACTICES

Engineering Policies



Engineering Policies

Empowering people to work effectively

- Flexible work schedules
- Choosing vendors and tools that prioritize accessibility
- Providing education and training on how to support colleagues with disabilities
- Offering folks a variety of tools to get the job done

FLEXIBILITY

“

Tools like k9s can make using Kubernetes more accessible for individuals with dyslexia or color blindness than using kubectl ... Allowing developers to customize their setup to their preferences is a crucial aspect of enhancing accessibility.”

Jason Kalili, LMU Loyola Law School²⁵


5. WRAP-UP

A place for everyone



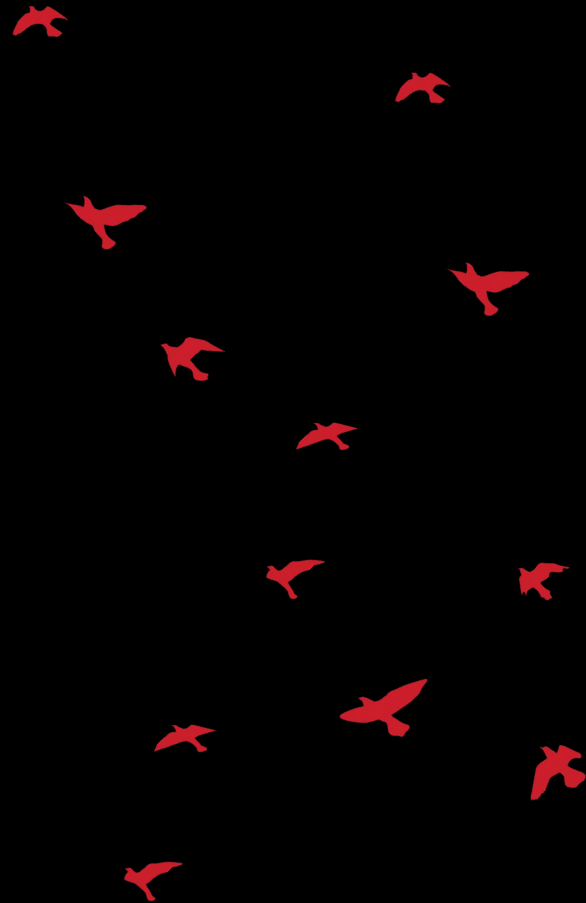
Wrap-Up

Takeaways

- Most of the work for accessibility comes from small, individual efforts
- Changes to make products or processes easier to use benefit *everyone!*
- Even if you rarely touch the UI, you can have an impact on accessibility
- AVOID ABBREVIATIONS 

5. WRAP-UP

Thanks!



Citations



1. ["Disability Impacts All of Us" \(CDC\)](#)
2. ["Debunking the Myths about Dyslexia" \(University of Michigan\)](#)
3. ["Color Blindness" \(National Eye Institute\)](#)
4. ["8 facts about Americans with disabilities" \(Pew Research Center\)](#)
5. ["Who Made America?" \(PBS\)](#)
6. ["A short history of the audiobook, 20 years after the first portable digital audio device" \(PBS News Hour\)](#)
7. ["Safety and Design Aspects of Powered Toothbrush—A Narrative Review" \(National Library of Medicine\)](#)
8. ["Disability and Health Overview" \(CDC\)](#)
9. ["Writing Accessible Go" \(Julia Ferraioli\)](#)
10. [WordPress Handbook](#)
11. ["Accessibility on the backend" \(Samuel Levy\)](#)
12. ["Python code style for blind programmers" \(Quang Wu\)](#)
13. [RuboCop documentation](#)
14. ["How to design APIs for Accessibility" \(Endjin\)](#)
15. [Swagger UI Github repo](#)

Citations (cont.)

16. ["The Case for WET Tests" \(Thoughtbot\)](#)
17. ["Atlassian Accessibility" \(Atlassian\)](#)
18. ["2022 Developer Survey" \(Stack Overflow\)](#)
19. ["Improving The Accessibility Of Your Markdown" \(Smashing Magazine\)](#)
20. ["WCAG 2.1 Understanding Docs" \(W3C\)](#)
21. ["Write accessible documentation" \(Google developer documentation style guide\)](#)
22. [Hemingway Editor](#)
23. ["Use simple words and phrases" \(PlainLanguage.gov\)](#)
24. ["Readability: Why are long sentences over 20 words?" \(Siteimprove\)](#)
25. ["Universal Back-End Design" \(Loyola Law School\)](#)
26. ["2022 Top Companies for Women Technologists" \(AnitaB.org\)](#)
27. ["The WebAIM Million" \(WebAIM\)](#)
28. [Disability fact sheet \(World Health Organization\)](#)
29. [Language Tool](#)
30. [Paraphrase Tool](#)

Questions?



hilary.stohs-krause@redcanary.com

Hilary Stohs-Krause