Web Accessibility FOR NEWBIES

Karina Chow
July 15, 2021

Questions we will answer today!

- What does it mean to be “accessible”?
- What is an “ADA Lawsuit”?
- Why should we care about accessibility?
- How can we evaluate the accessibility of our websites?
- What can we do to make them more accessible?
- Where else can I look to find more information?

A little about me…

- I’m a Bay Area Native! I was born and raised in Mountain View
- I’m German-Chinese American 🇩🇪🇨🇳🇺🇸
- I’m a 3rd generation “tech worker”
- I went to Carnegie Mellon University and got a B.S. in Computer Science with an additional major in Human-Computer Interaction
- I’ve worked as a frontend/software engineer at companies such as Adobe, Yammer (now Microsoft), Honor, and Patreon
- I’ve had greenish hair for 11 years and my socials say “Screw the rules, I have green hair”
- I’m currently self-employed, and yes, my company name is KACHOW!

OK Karina. That’s nice.

But why are you talking to us about accessibility today?

Because it’s important!

OK, but why are you talking to us about accessibility today?

I am NOT an expert.

What does it mean to be “accessible” anyway?

To be accessible is to be

POUR

Wait what?

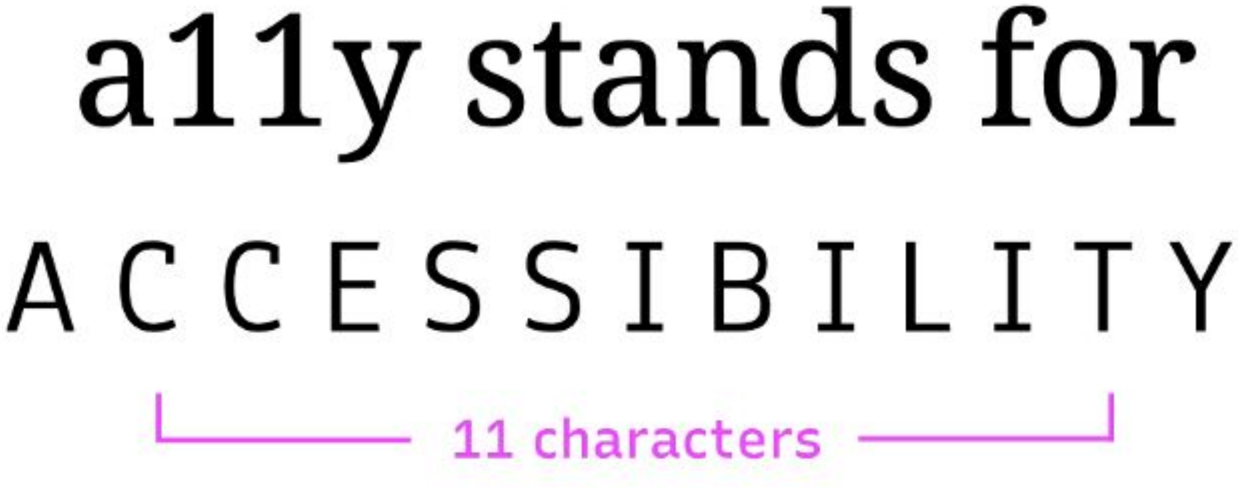
Accessible places and products should be

Perceivable

Operable

Understandable

Robust



a11y stands for

ACCESSIBILITY

11 characters

Source: The A11Y Project

“a11y”

“a11y”
That’s annoyingly cute.

Right?

OK, but **why** should we care about “a11y”?

Why?

It's the law.

The **Americans with Disabilities Act** (ADA) Title III prohibits discrimination on the basis of disability

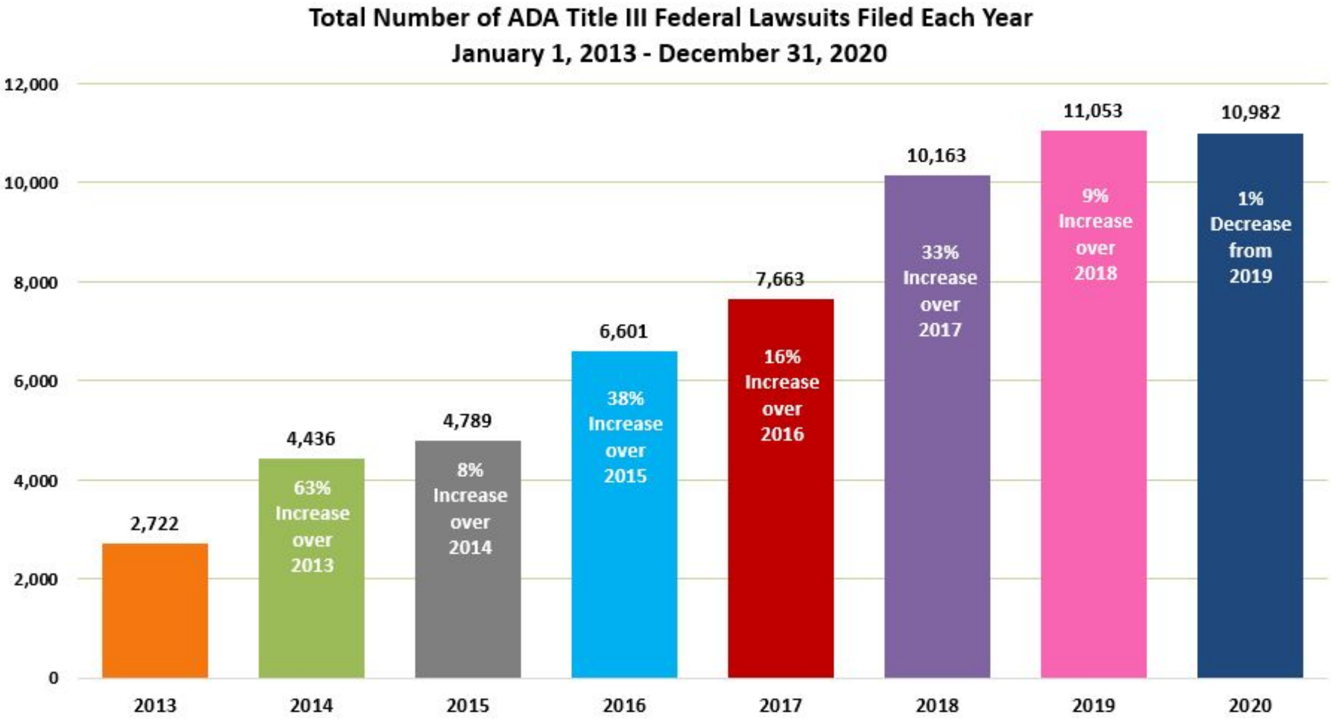
> *"[it] prohibits discrimination on the basis of disabilities in places of public accommodations, commercial facilities, and private entities that offer certain examination and courses related to educational and occupational certification"*

Other countries have similar laws

Why?

It's the law.

Turns out, you can get sued for breaking the law!



Source: Seyfarth

KACHOW! LLC

Why?

It improves SEO and UX

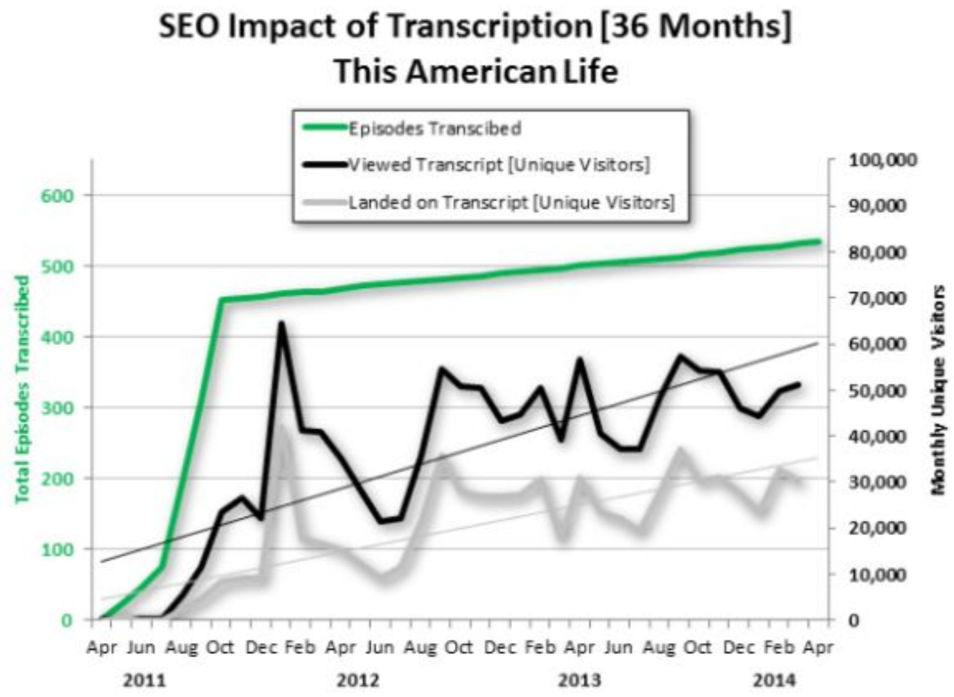
A lot of the work needed for improving a11y actually improves SEO and UX.

Examples: Site maps, video and audio transcriptions, those generated Yelp pages

Why?

It improves SEO and UX

Transcribing "This American Life" podcasts improved SEO, and also increased the number of ESL visitors because it was easier to translate.



SEO Impact of Transcription [36 Months]
This American Life

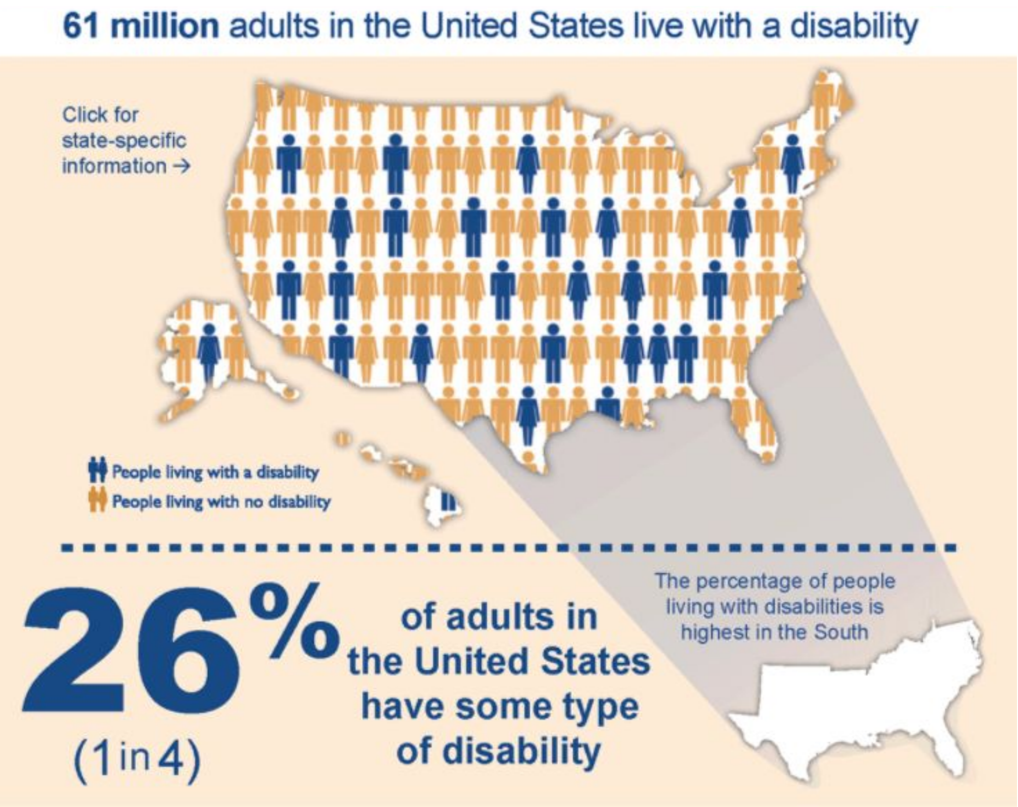
Source: 3PlayMedia

Why?

It expands your user base.

The data shows that the global market of people with disabilities is over 1 billion people with a spending power of more than $6 trillion.

Source: The CDC



61 million adults in the United States live with a disability

Click for state-specific information →

26% (1 in 4) of adults in the United States have some type of disability

The percentage of people living with disabilities is highest in the South

People living with a disability
People living with no disability

Why?

It expands your user base.

Don't be like Beyoncé!
Invest in users early!

It’s the right thing to do!

More about folks with disabilities

Categories of disability types

Visual

Blindness, color blindness, low vision, glaucoma

Hearing

Deafness, hard-of-hearing

Motor

Limited fine-motor control

Cognitive

Difficult focusing, learning disabilities

Seizures

Different ways of interacting with a product

- **Screen Readers**
  - Assistive technology that reads what's on the screen, most commonly used by those with blindness or low vision.
  - Try out VoiceOver if you have OSX. Start it with Command + F5
- **Keyboard**
  - Some folks do not use a mouse and only use the keyboard
- **Browser Zoom** (ideally support up to 200%)
- **Custom styling**
- **Other assistive technology**
  - Refreshable braille displays
  - Magnification software

Let’s develop some empathy

How would you feel if you experienced a website like this?

Bad navigation example

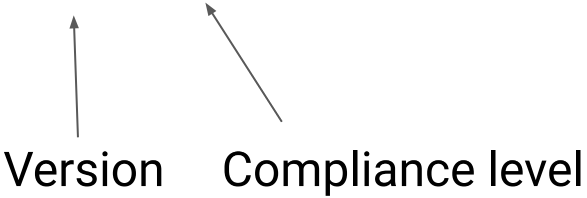
Good navigation example

How do we evaluate accessibility?

Web Content Accessibility Guidelines

WCAG is a set of requirements published by W3C.

You might see
“We need to be **`WCAG 2.1 AA`** compliant”



Version

Compliance level

WCAG Compliance is like HIPAA Compliance

A team can’t “solve” accessibility at a company; rather, they must make a commitment to continually address it.

Just like how HIPAA compliance is an ever-changing standard as the healthcare industry and the internet expands, WCAG compliance, too, is always evolving and isn’t confined to a small set-and-forget list of TODOs.

WCAG Compliance

For example, for time-based media, here are some examples of the differences between compliance levels:

- A — “Minimal Compliance” — Captions for **pre-recorded** video/audio, alternatives provided for pre-recorded video/audio (e.g. transcriptions)
- AA — “Acceptable Compliance” — Everything in A *and* captions for **live** video/audio, additional narrations provided during pre-recorded video
- AAA — “Optimal Compliance” — Everything in AA *and* sign language interpretation for video, more detailed narrations provided during and in pauses for pre-recorded video, live transcriptions

WCAG Compliance

Compliance documents exist for each version

But no matter which version, the four main guiding principles of WCAG are: **perceivable, operable, understandable, and robust (POUR)**

If your eyes just glazed over,
that’s ok.

Just remember that you want

WCAG 2.1 AA Compliance

Let’s go back to that POUR thing, shall we?

P. O. U. R.

Perceivable - information needs to be available to at least one sense (sight, hearing, touch)

Operable - a user must be able to perform all interface actions (i.e., with a keyboard)

Understandable - a user should understand the language on a page and how to operate it

Robust - different user agents, including assistive technologies, content should be able to access all content

Some questions to ask while auditing accessibility

- Can I use the keyboard to do everything a mouse can do?
- Is the screen reader giving me all relevant info? (Role, content/name, state)
- What happens when we set the browser zoom to 200%?

Testing tools

WAVE browser extension/website

WAVE browser plugin

- This extension will audit your page for you with helpful error and warning messages
- Remember: automated audits can only find ~30% of accessibility issues. The work isn't done when WAVE returns 0 errors.

A11y Linters & SDE Plugins

There are now linters and plugins that help catch and automate accessibility fixes as part of your normal linting process

- There are Eslint linters for React projects
- React Axe spits a11y errors it finds into the dev console while you're running in development

But note, like WAVE, these catch only ~30% of accessibility issues. Certain issues, like an element not having a focus state, might not be found.

Screen readers

If you have time, do a quick pass through with a screen reader and your keyboard before opening a PR.

Never used a screen reader? Practice on https://screenreader-practice.glitch.me/#

Screen readers

- VoiceOver on OSX
- JAWS on Windows
- Linux Screen Reader (LSR) on Linux
- In-browser ChromeVox on Google Chrome

So…How can I start prioritizing accessibility?

Great question.

Great question.
There’s no silver bullet way.

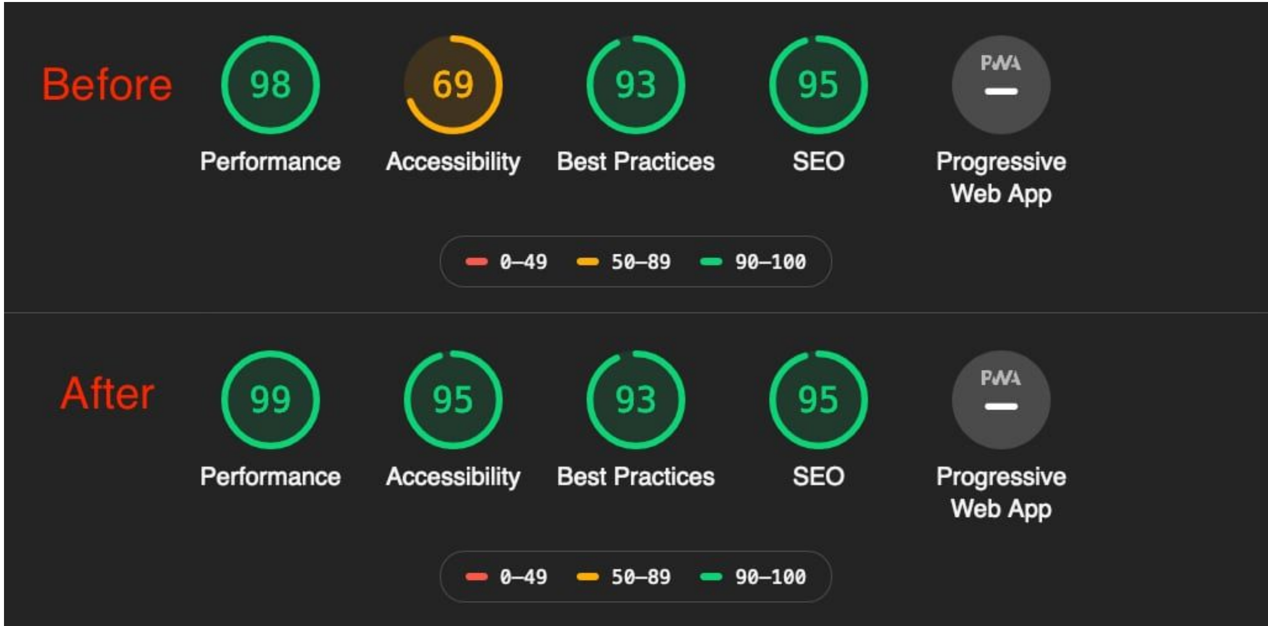
But here’s some ideas.

Create milestones

Treat accessibility like any other measurable metric, like a test coverage or performance goal.

Set a quantifiable goal

You could select a specific WCAG compliance as a goal, or a Lighthouse accessibility score



Before

98 Performance

69 Accessibility

93 Best Practices

95 SEO

PWA — Progressive Web App

▬ 0–49 ▬ 50–89 ▬ 90–100

After

99 Performance

95 Accessibility

93 Best Practices

95 SEO

PWA — Progressive Web App

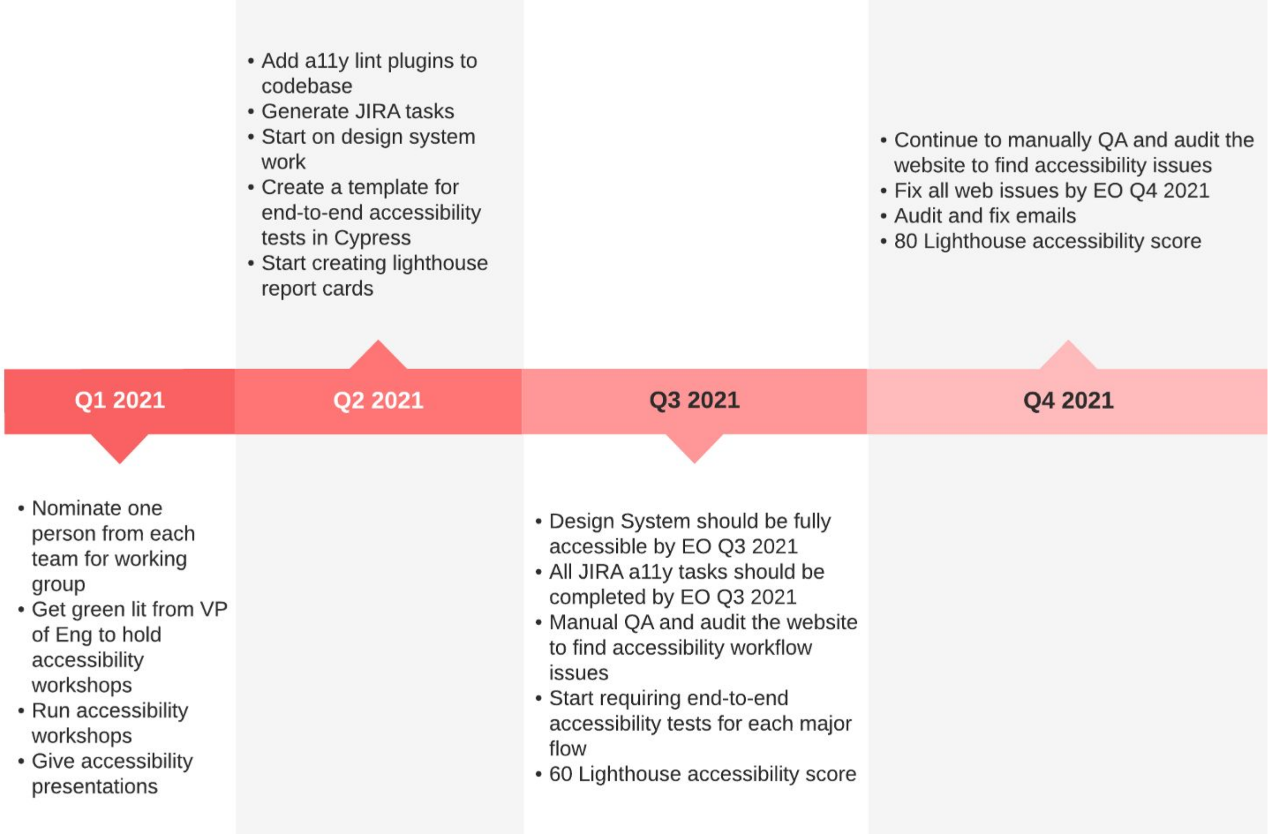
▬ 0–49 ▬ 50–89 ▬ 90–100

Create milestones

We had to make a roadmap as part of one of our settlements…it actually really helped whip us into shape.

It could be a good idea to do this BEFORE you get slammed with a lawsuit!

Make an a11y roadmap



Q1 2021

- Nominate one person from each team for working group
- Get green lit from VP of Eng to hold accessibility workshops
- Run accessibility workshops
- Give accessibility presentations

Q2 2021

- Add a11y lint plugins to codebase
- Generate JIRA tasks
- Start on design system work
- Create a template for end-to-end accessibility tests in Cypress
- Start creating lighthouse report cards

Q3 2021

- Design System should be fully accessible by EO Q3 2021
- All JIRA a11y tasks should be completed by EO Q3 2021
- Manual QA and audit the website to find accessibility workflow issues
- Start requiring end-to-end accessibility tests for each major flow
- 60 Lighthouse accessibility score

Q4 2021

- Continue to manually QA and audit the website to find accessibility issues
- Fix all web issues by EO Q4 2021
- Audit and fix emails
- 80 Lighthouse accessibility score

Audit everything

A11y issues are bugs.

That's it. End of story.

Add a11y linters to the codebase

Linters only catch 20-25% of all accessibility issues, but they're still a start!

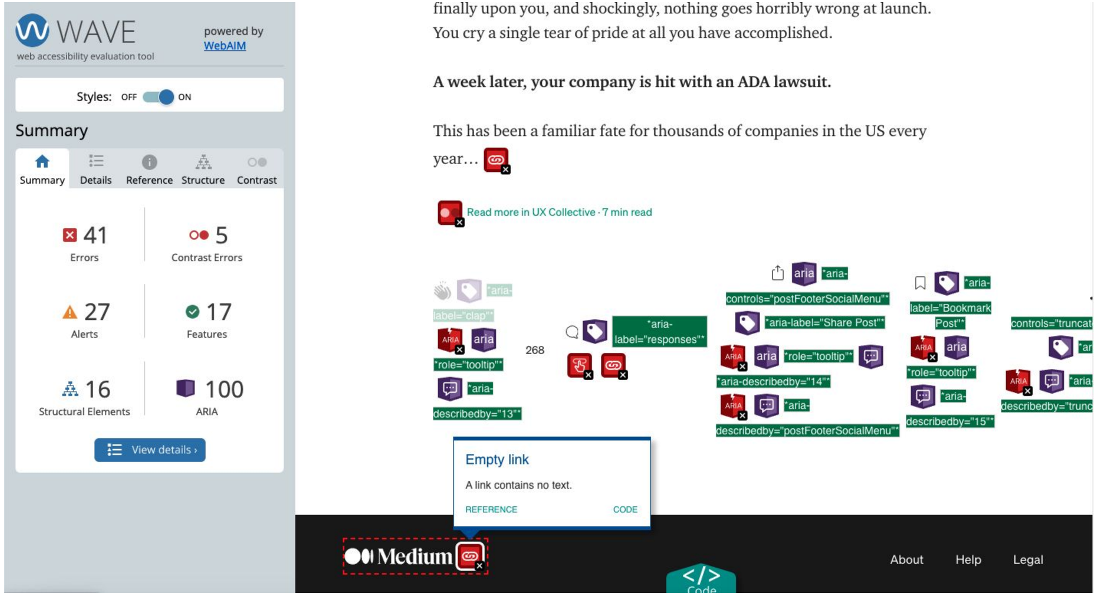
- Add an a11y linter to your codebase
- Run it initially and create TODO tasks from every single issue raised
- Prioritize a couple tasks every week
- Enable the linter in pre-commit hooks such that no one can create new code that creates a11y issues

Audit everything

Use a11y browser plugins

Browser plugins, like WAVE or ANDI, can create overlays over your website and show you accessibility issues.

WAVE even gives you a report card and a score as to how you're doing.



WAVE
web accessibility evaluation tool
powered by WebAIM

Styles: OFF ON

Summary

41 Errors

5 Contrast Errors

27 Alerts

17 Features

16 Structural Elements

100 ARIA

finally upon you, and shockingly, nothing goes horribly wrong at launch. You cry a single tear of pride at all you have accomplished.

A week later, your company is hit with an ADA lawsuit.

This has been a familiar fate for thousands of companies in the US every year…

Read more in UX Collective · 7 min read

Empty link

A link contains no text.

KACHOW! LLC

Audit everything

You will have to manually audit. There's no avoiding it!

Manually audit

Look at every big flow in your website and ensure it's keyboard accessible.

- You don't have to know how to use a screen reader to make your website accessible, just learn how to use keyboard-only
- Essential flows, especially ones that involve paying or receiving money or opting in or out of emails, should be completely keyboard-accessible
- Modals, or image lightboxes or carousels, should be usable via keyboard

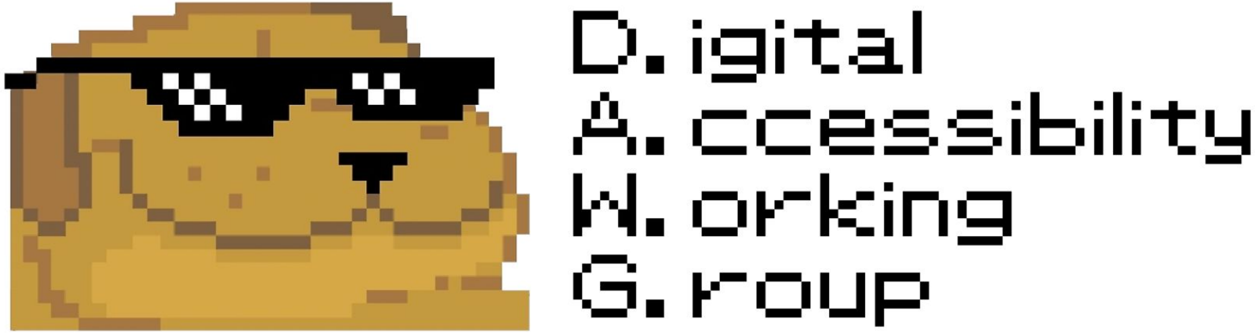
Learn together

Create a culture of learning and an inclusive environment for people to learn together.

Introducing the a11y working group

Create a group that

- Researches together
- Is highly democratic and welcoming
- Holds office hours and is available
- Has dedicated time every week to do this important work

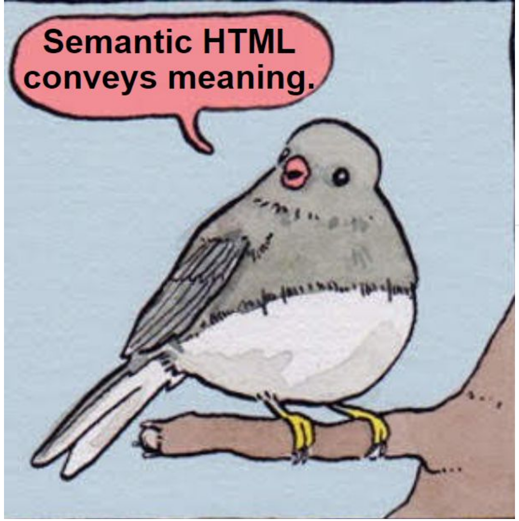


I know what you’re thinking.

But like…how do I actually fix things?

Here’s a novel concept.

Use HTML



Use HTML

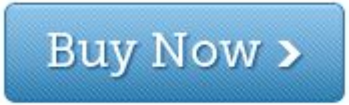
(correctly)

Semantic HTML conveys meaning.
This help
<div>
<div>
<div>
<div>
<div>
<div>
<div>
<div class="button">
KAPWING
FALSEKNEES.COM © 2016

Standard HTML elements come with a lot of great built-in functionality.

Default to these whenever possible.

Using HTML properly



Let’s say we inspect this custom button and find that it’s a `<div />` with an `onclick` and some fun CSS. To make it accessible, we’d need to add:

- `tabindex=0` to add it to the tab order.
- A key event handler so that “Enter” does the same thing as `onclick`
- `role=”button”` so that the screen reader knows its a button
- Additional styling for the focus state so the keyboard users can keep track of their location on the page

Now if we just change the `<div />` to a `<button />` we’d get all of that for free!

Semantic HTML features

- Standard HTML elements come with a lot of great built-in functionality. Default to these whenever possible.
  - role
  - Keyboard support
  - Visual focus indicators
- Use aria- props
- The element order in the DOM should make sense. Don't rely on CSS to order pieces of your page.
- Other noteworthy props are label[for] and tabindex

Semantic HTML features (cont’d)

- Add `alt` text to images
  - This attribute is required
  - If the image is purely decorative it should have an `alt=””`
- Headers `h1` to `h6` should be in order on the page.
  e.g. `h3` should only be after an `h2`.
- Be mindful of nesting.
  e.g. a `<button />` should not be inside another `<button />`

A11y seems pretty cool. Where can I learn more?

Resources

Informational

- WebAIM
  A great source of a11y information
- WAI-ARIA Authoring Practices 1.1
  Everything you need to know to make common widgets and UI patterns accessible.
- Web Content Accessibility Guidelines 2.1
  The standards themselves. Dense, but useful.
- A11ycasts
  Videos by the Google Chrome Dev team
- The A11y Project
  Education around making accessible designs
- Mozilla's a11y documentation
  Easy-to-comprehend a11y documentation

Resources

Tooling

- Screen Reader practice
- ANDI
  An automated accessibility testing tool to run on your webpages
- WebAIM’s Color Contrast checker
  A great tool to see if your text is readable
- Deque Systems
  A highly esteemed group that makes lots of testing and auditing tools and even offers consulting services
- Lighthouse
  See performance, SEO, a11y, and other scores on the same dashboard
- AccessMonitor
  A plugin that tests a11y on your WordPress site

Resources

If you would like to read the content you were just presented in far more detail, check out these 4 articles this presentation was based on!

Also, a totally shameless plug.

Karina's a11y articles

- Web Accessibility for Newbies
- How to Convince Company Leadership to Care About Accessibility
- 5 Ideas on How to Incorporate Web Accessibility into your Dev Workflow
- The Developer's Guide to Web Accessibility Auditing and Tooling

My Contact Info

@karomancer

Twitter, Medium, LinkedIn, Dribble

karinachowtime@gmail.com

https://www.karinachowtime.com