

Accessibility in Design Systems: Easy Wins and Lessons Learnt

Table of Content:

Intro	03
A11Y-101	06
Easy Wins	10
Lessons Learnt	18
Outro	34
Resources	39



Intro

Riccardo Erra Senior Frontend Engineer

<https://riccardoerra.com>



04/40













Management system for hotels and properties as well as their guests

05/40

A11Y-101

Accessibility is the practice of making your websites usable by as many people as possible

Over 1 billion
people currently
experience some
form of disability

	Permanent	Temporary	Situational
Touch	 One arm	 Arm injury	 New parent
See	 Blind	 Cataract	 Distracted driver
Hear	 Deaf	 Ear infection	 Bartender
Speak	 Non-verbal	 Laryngitis	 Heavy accent

08/40

Businesses lose approximately
£2B/m by ignoring the needs of
disabled people

09/40

Easy Wins

10/40

Design systems present a unique opportunity to build accessibility and inclusion into your component libraries from the get go

Start with your colours

Foreground Color

#000000

Lightness

↔

Background Color

#FFFFFF

Lightness

Contrast Ratio

21:1

[permalink](#)

Normal Text

WCAG AA: **Pass**

WCAG AAA: **Pass**

The five boxing wizards jump quickly.

Large Text

WCAG AA: **Pass**

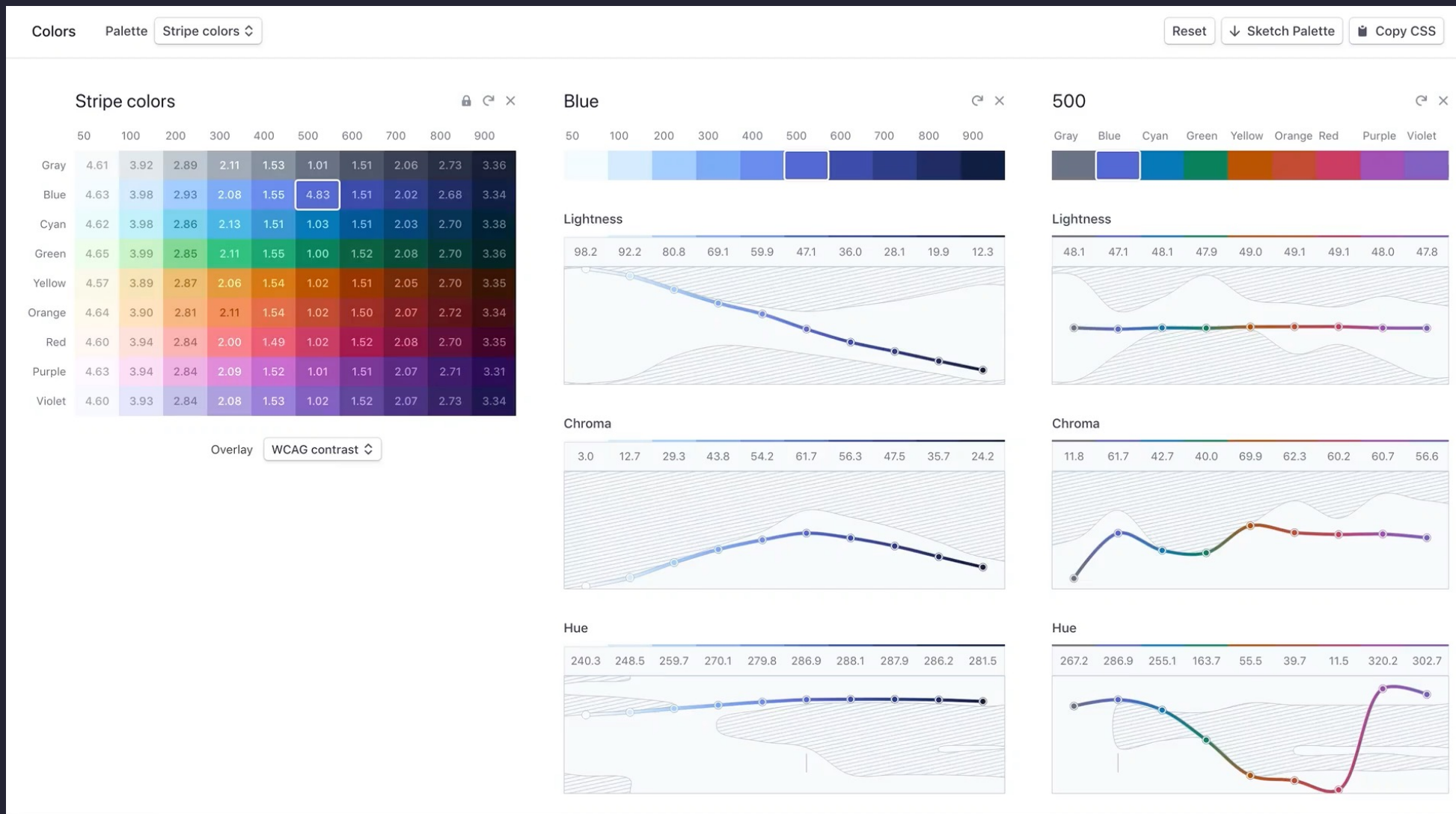
WCAG AAA: **Pass**

The five boxing wizards jump quickly.

Graphical Objects and User Interface Components

WCAG AA: **Pass**

✓
Text Input



Web content can be made accessible just by making sure the correct Hypertext Markup Language elements are used

```
<div id="saveChanges" tabindex="0" role="button" aria-pressed="false">Save</div>
```

```
<button type="button" id="saveChanges">Save</button>
```

Ensure that all content can be
accessed with the keyboard
alone

Screen readers enable the independent use of digital devices by those who otherwise would not be able

Lessons Learnt

Learn about standards, guidelines and requirements

19/40



BETA [Contact the Service Manual team](#) if you have feedback, questions or suggestions.

[Service manual](#) > [Accessibility and assisted digital](#)

Accessibility and assisted digital

[Give feedback about this page](#)

Understanding WCAG 2.1

From: [Accessibility community](#)

Page contents:

- [Meeting government accessibility requirements](#)
- [WCAG 2.1 design principles](#)
- [Applying WCAG 2.1 guidelines](#)
- [Further reading](#)
- [Related guides](#)



This is an overview of WCAG 2.1 level AA. It does not replace the [WCAG 2.1 guidelines](#), which provide a full explanation of all principles and requirements.

The Web Content Accessibility Guidelines (known as WCAG) are an internationally recognised set of recommendations for improving web accessibility.

They explain how to make digital services, websites and apps accessible to everyone, including users with impairments to their:

- vision - like severely sight impaired (blind), sight impaired (partially sighted) or colour blind people
- hearing - like people who are deaf or hard of hearing
- mobility - like those who find it difficult to use a mouse or keyboard
- thinking and understanding - like people with dyslexia, autism or learning difficulties

Meeting government accessibility requirements

IBM Accessibility

Equal Access Toolkit

Requirements

Product Accessibility (ACR/VPAT)

Toolkit license

Submit a GitHub issue

IBM Accessibility Central

#accessibility-at-ibm

Requirements

Release notes

Technology ⓘ

Standards ⓘ

Level to achieve ⓘ

☒ Web

☐ Non-web software

☐ Web documents

☐ Non-web documents

☐ Authoring tools

☒ WCAG 2.1

☐ US 508

☐ EN 301 549

☐ Level 1

☒ Level 2

☐ Level 3

34 of 96 requirements selected | [expand all](#)

1.1.1 Non-text Content

1.2.2 Captions (Prerecorded)

1.2.4 Captions (Live)

1.3.1 Info and Relationships

1.3.3 Sensory Characteristics

1.3.4 Orientation

1.4.1 Use of Color

1.4.2 Audio Control

1.4.3 Contrast (Minimum)

1.4.5 Images of Text

1.4.10 Reflow

1.4.11 Non-text Contrast


<

Never stop improving your testing skills

TOTAL ISSUES

34

AUTOMATIC ISSUES 34

.....  **NEEDS REVIEW** 21

GUIDED ISSUES 0

Critical 8 Serious 5

Moderate 0 Minor 0

Best Practices: OFF

 EXPORT ISSUES

ALL ISSUES: 34

 Guided Tests

Elements must only use allowed ARIA attributes

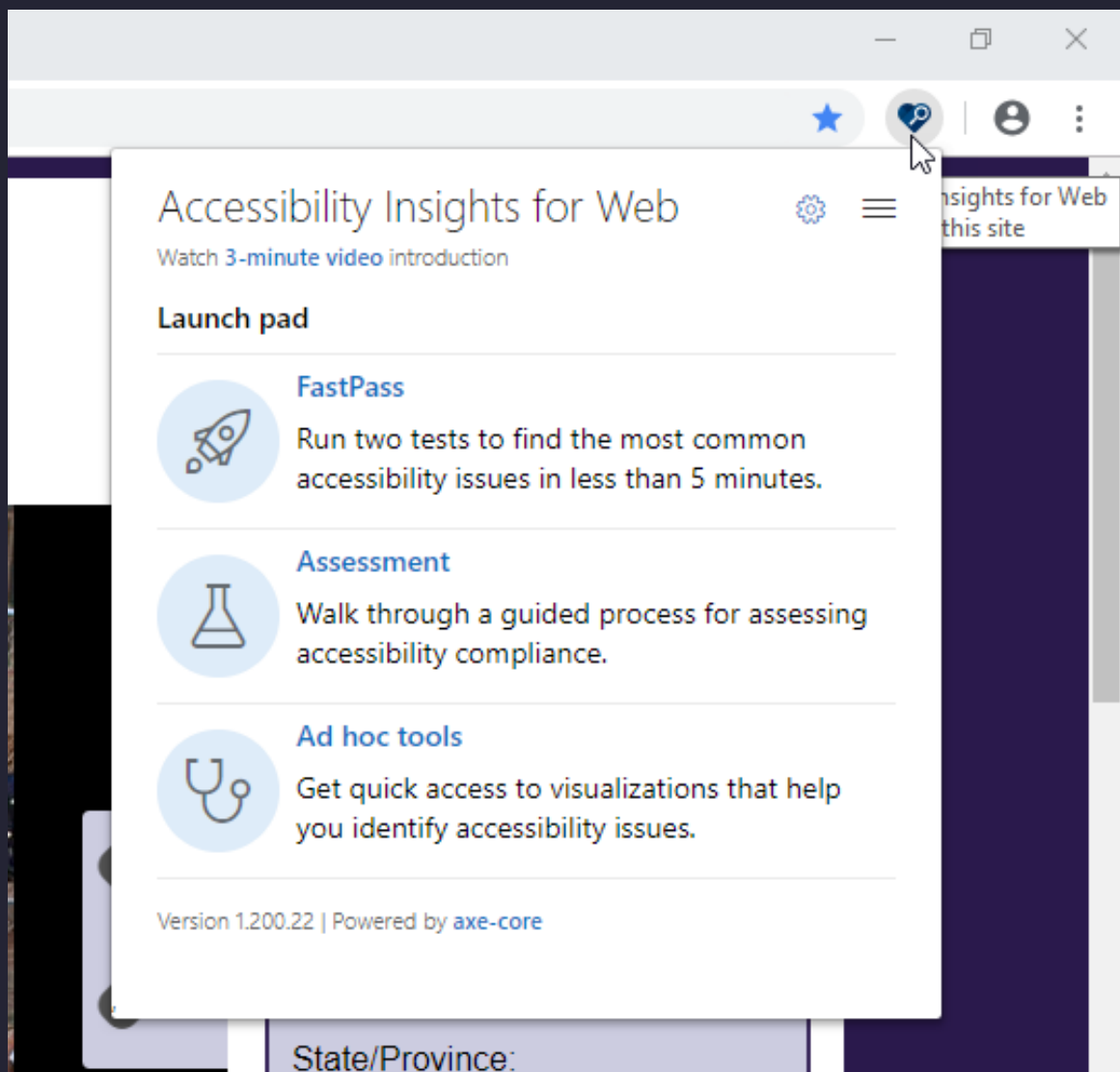
1

Elements must have sufficient color contrast

25

Images must have alternate text

8



The image shows a browser window with the 'Accessibility Insights for Web' extension installed. The extension's icon, a blue circle with a white magnifying glass, is in the top right of the browser's address bar. A mouse cursor is clicking on this icon, which has opened a white dropdown menu. The menu has a title 'Accessibility Insights for Web' and a subtitle 'Watch 3-minute video introduction'. Below this is a section titled 'Launch pad' containing three items: 'FastPass' (with a rocket icon), 'Assessment' (with a flask icon), and 'Ad hoc tools' (with a stethoscope icon). At the bottom of the menu, it says 'Version 1.200.22 | Powered by axe-core'. In the background, a portion of a website is visible, showing a form with a label 'State/Province:'.

Accessibility Insights for Web

Watch [3-minute video](#) introduction

Launch pad

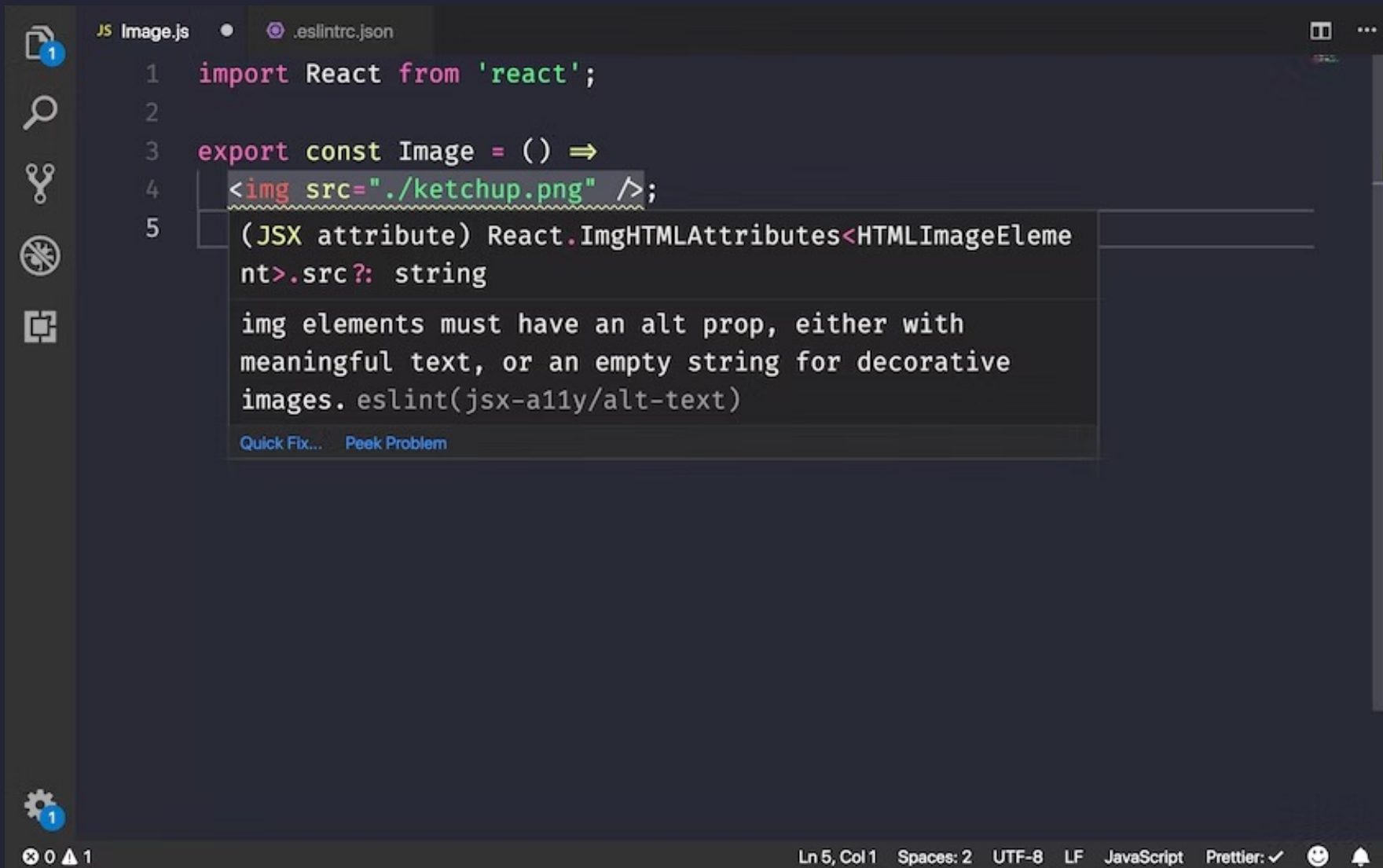
- FastPass**
Run two tests to find the most common accessibility issues in less than 5 minutes.
- Assessment**
Walk through a guided process for assessing accessibility compliance.
- Ad hoc tools**
Get quick access to visualizations that help you identify accessibility issues.

Version 1.200.22 | Powered by [axe-core](#)

State/Province:

Quick disclaimer

Take advantage of the available tools



```
JS Image.js • .eslintrc.json
1 import React from 'react';
2
3 export const Image = () =>
4   ;
5   (JSX attribute) React.ImgHTMLAttributes<HTMLImageEleme
     nt>.src?: string
     img elements must have an alt prop, either with
     meaningful text, or an empty string for decorative
     images. eslint(jsx-a11y/alt-text)
     Quick Fix... Peek Problem
```

Ln 5, Col 1 Spaces: 2 UTF-8 LF JavaScript Prettier: ✓

Storybook

Find components

EXAMPLE

Introduction

Button

Primary

Header

Logged In

Logged Out

Page

Logged In

Logged Out

Canvas

Docs

Button

Controls (4)

Actions

Accessibility

1 Violations

3 Passes

0 Incomplete

Highlight results

> Elements must have sufficient color contrast

Tests completed

Use the force aka provide the right data or you'll get yelled at by your computer

```
import PropTypes from "prop-types";

const Image = ({ src, alt, className }) => (
  <img src={src} alt={alt} className={className} />
);

Image.propTypes = {
  src: PropTypes.string.isRequired,
  alt: PropTypes.string.isRequired,
  className: PropTypes.string,
};

export default Image;
```

```
interface ImageProps {  
  src: string;  
  alt: string;  
  className?: string;  
}  
  
const Image = ({ src, alt, className }: ImageProps) =>  
  <img src={src} alt={alt} className={className} />;  
  
export default Image;
```

Accessible component \neq Accessible page

Don't forget to commission a proper professional audit

Outro

The work is never done but a bit of effort is better than no effort at all

Your most powerful weapon is
empathy

36/40

Thank you!

37/40

We are hiring!
We are looking for a
Senior Product
Manager in the
Design System team



Resources

- Slides: <https://github.com/riccardoerra/accessibility-design-systems/blob/main/accessibility-in-design-systems.pdf>
- Accessibility definition: <https://developer.mozilla.org/en-US/docs/Web/Accessibility>
- Disabilities illustration: <https://devblogs.microsoft.com/xamarin/the-journey-to-accessible-apps>
- Financial statistic: <https://wearepurple.org.uk/the-purple-pound-infographic>
- Design systems and accessibility: <https://xd.adobe.com/ideas/principles/design-systems/accessibility-in-design-systems>
- Contrast checker: <https://webaim.org/resources/contrastchecker>
- Stripe colour system: <https://stripe.com/blog/accessible-color-systems>
- HTML and accessibility: <https://developer.mozilla.org/en-US/docs/Learn/Accessibility/HTML>
- ARIA button example: https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles/button_role
- Keyboard accessibility: <https://webaim.org/techniques/keyboard>
- Screen readers accessibility: <https://blog.hubspot.com/website/screen-reader-accessibility>

- GOV.UK understanding WCAG: <https://www.gov.uk/service-manual/helping-people-to-use-your-service/understanding-wcag>
- IBM accessibility requirements: <https://www.ibm.com/able/requirements/requirements>
- GOV.UK accessibility statement: <https://design-system.service.gov.uk/accessibility>
- Deque axe DevTools extension: <https://www.deque.com/axe/devtools/chrome-browser-extension>
- Accessibility Insights: <https://accessibilityinsights.io/>
- Accessibility auditing with React: <https://web.dev/accessibility-auditing-react>
- Accessibility tests with Storybook: <https://storybook.js.org/docs/react/writing-tests/accessibility-testing>
- Enforcing accessibility best practices: <https://bradfrost.com/blog/post/enforcing-accessibility-best-practices-with-component-proptypes>
- Senior Product Manager role: <https://grnh.se/914aa322teu>