

valtech.

# Web accessibility for devs

André Elmoznino Laufer  
Software Developer

# Agenda

- 1 Introduction
- 2 Common problems and their fixes
- 3 Tools



---

# Section 1

# Introduction



Accessible design is good design - it benefits people who don't have disabilities as well as people who do.

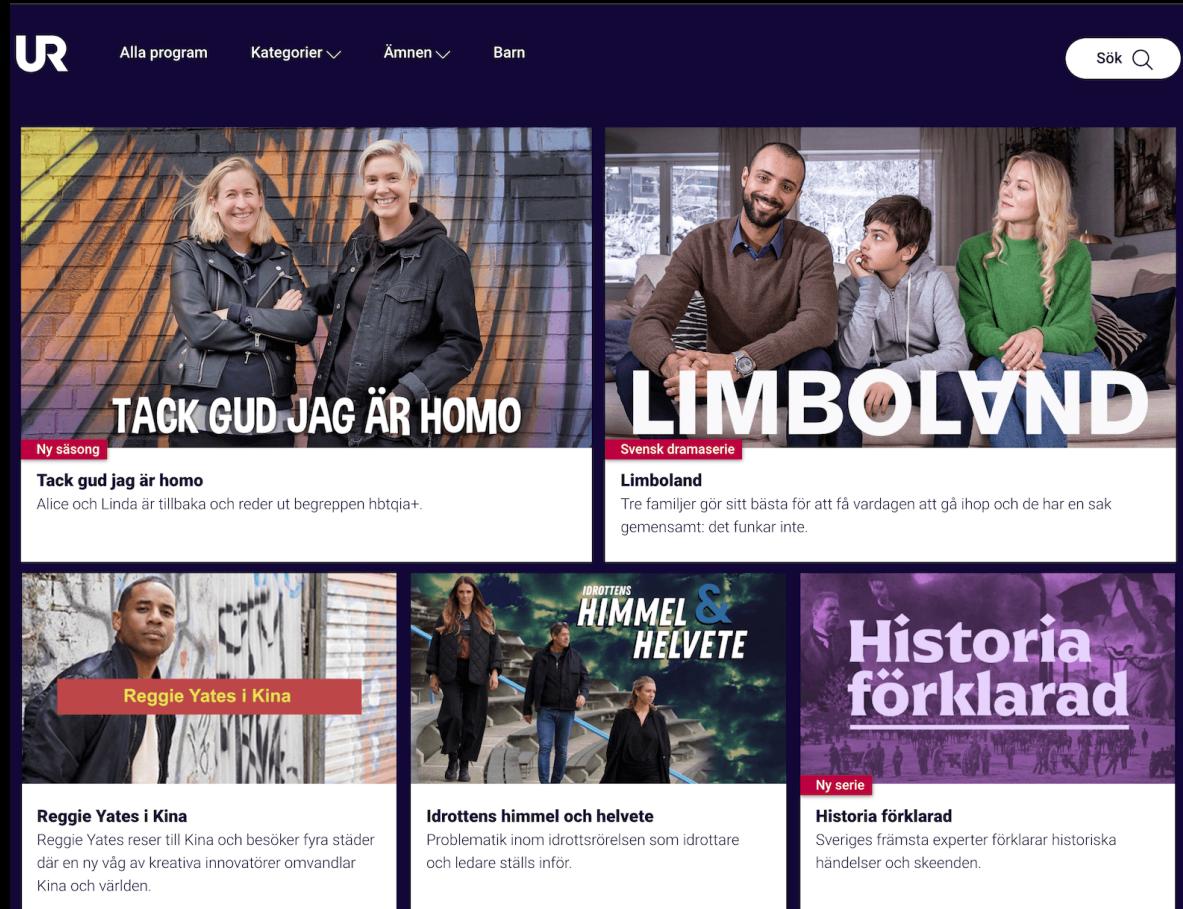
Accessibility is all about removing barriers and providing the benefits of technology for everyone.

Steve Ballmer  
Microsoft ex-CEO



# Who am I

- / Frontend Developer (React, React Native, Vue, Angular)
- / At Valtech since Feb 2019
- / Clients include Sveriges Utbildningsradio (UR), Goodyear Tyres, Invidem



# Why a11y matters

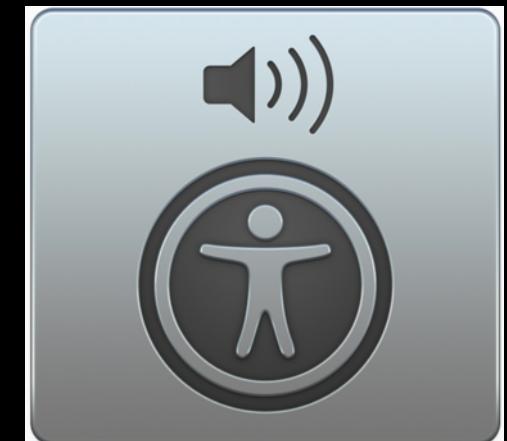
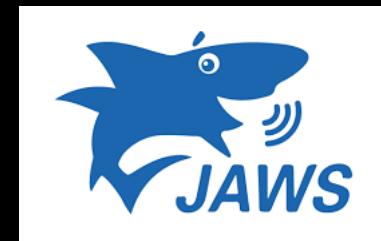
- / All disabilities aren't visible
- / Can be temporary: injury, out in the sun etc.
- / Don't you want everyone to be able to use that awesome app or website you're building?
- / It could happen to you or someone you care about too
- / Requirement by law for public sector (including level AA of [WCAG 2.1](#))

Area	People in Sweden with impairment
Movement & physical coordination	1.300.000
Reading & writing (incl. dyslexia)	1.920.000
Comprehension ("svagbegåvade")	1.340.000
Sight	297.000

Source: <https://www.webbriktlinjer.se/tillganglighet/statistik/>

# Assistive technology

- / Braille
- / Screen readers: JAWS (popular with IE11), NVDA, VoiceOver (Mac built-in)
- / Magnifying software
- / Contrast enhancers
- / Alternate mice or keyboard solutions
- / More: <https://www.w3.org/WAI/people-use-web/tools-techniques/>



# Examples of WCAG requirements

- / 1.1: Provide **text alternatives**, e.g. alt-tags for images, aria-label on buttons
- / 1.3.4: Content isn't restricted to a specific **orientation** (landscape/portrait) unless essential
- / 1.4.1: **Color** is not used as the only visual means of conveying information or indicating an action
- / 1.4.3: Text and images have a **contrast ratio** of at least 4.5:1 (3:1 for large text)
- / 1.4.4: **Text can be resized** up to 200% without loss of content or functionality
- / 1.4.12: **Text spacing** - No loss of content/functionality when increasing text spacing (e.g. line height, letter spacing, word spacing)

# Examples of WCAG requirements (continued)

- / **2.1: Keyboard accessible** - Make all functionality available from a keyboard
- / **2.1.2: No keyboard trap** (not being stuck when tabbing to certain elements)
- / **2.1.4:** Don't interfere with screen readers' **character key shortcuts** (user can remap or turn off)
- / **2.2.2:** User can **pause/stop/hide** automatically playing content (e.g. carousels, video)

Read more: <https://www.w3.org/WAI/WCAG21/quickref>

---

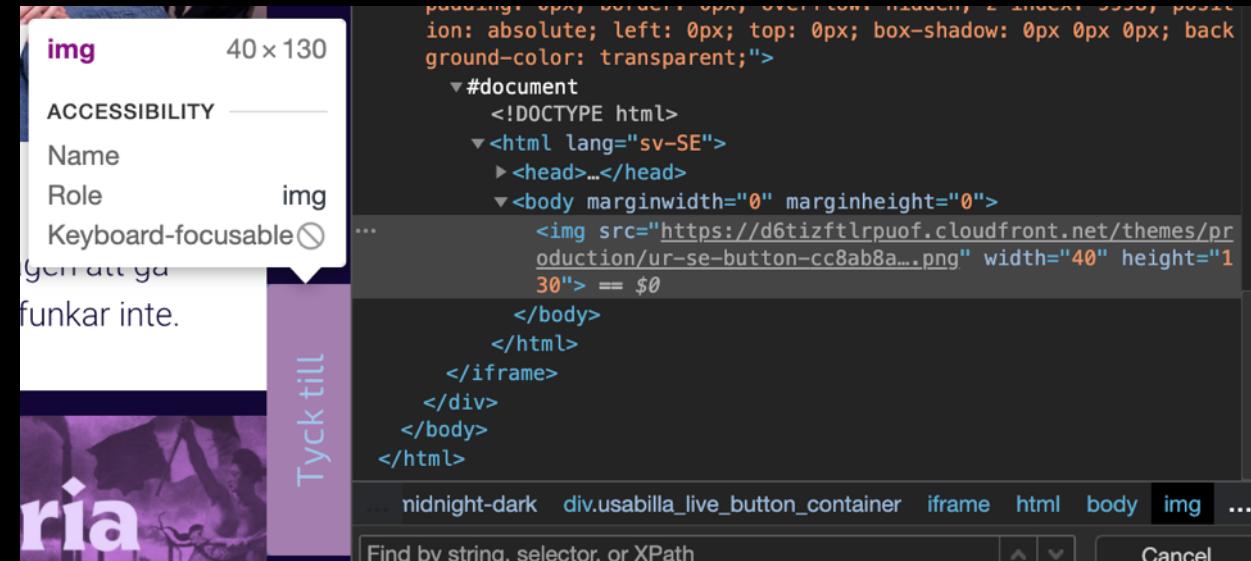
## Section 2

# Common problems and their fixes



# Issue: Alt text for images

- / “If you can't see, all types of visual information, such as images, are completely useless unless a digital text alternative is provided so that screen readers can convert that text into either sound or braille.”
- / If purely decorative, just set alt="”, it makes screen readers ignore the image (otherwise the full src is read out!)
- / More info:  
<https://dequeuniversity.com/rules/axe/4.0/image-alt>



# Issue: Low contrast ratio

- / “Text that is too close in luminance (brightness) to the background can be hard to read.”
- / 1/12 men have color vision deficiencies
- / Tip: try switching on/off dark mode
- / Detected by tools, but also when you inspect element in Chrome
- / More info and a color contrast analysis tool:  
<https://dequeuniversity.com/rules/axe/3.5/color-contrast>

Hacker News new | past | comments | ask |

- ▲ 51% of 4M Docker images have critical vulnerabilities  
113 points by AnnieNma 4 hours ago | hide | 60 comments
- ▲ GWSL: Run graphical Linux apps in WSL (github.com/magnio/gwsl)  
98 points by magnio 3 hours ago | hide | 54 comments
- ▲ More information on the plant disturbance at  
24 points by ericdanielski 54 minutes ago | hide | 3 comments
- ▲ **span#score\_25455279.score** 43.63 × 11  
Color #828282  
Font 9.33333px Verdana, Geneva, sans-serif
- ▲ ACCESSIBILITY  
Contrast Aa 3.54 !  
Name  
Role generic  
Keyboard-focusable ✘

Hacker News new | past | comments | ask | show | jobs | submit

- ▲ Goodreads plans to retire API access, disables existing API keys (joealcorn.co.uk)  
217 points by buttscicles 3 hours ago | hide | 191 comments
- ▲ Unlimited is a like Ponzi Scheme (staysaasy.com)  
55 points by chesterarthur 2 hours ago | hide | 28 comments
- ▲ If-then-else had to be invented (github.com/ericfischer)  
40 points by fanf2 2 hours ago | hide | 25 comments

# Issue: Using image for text

## Reasons for using text:

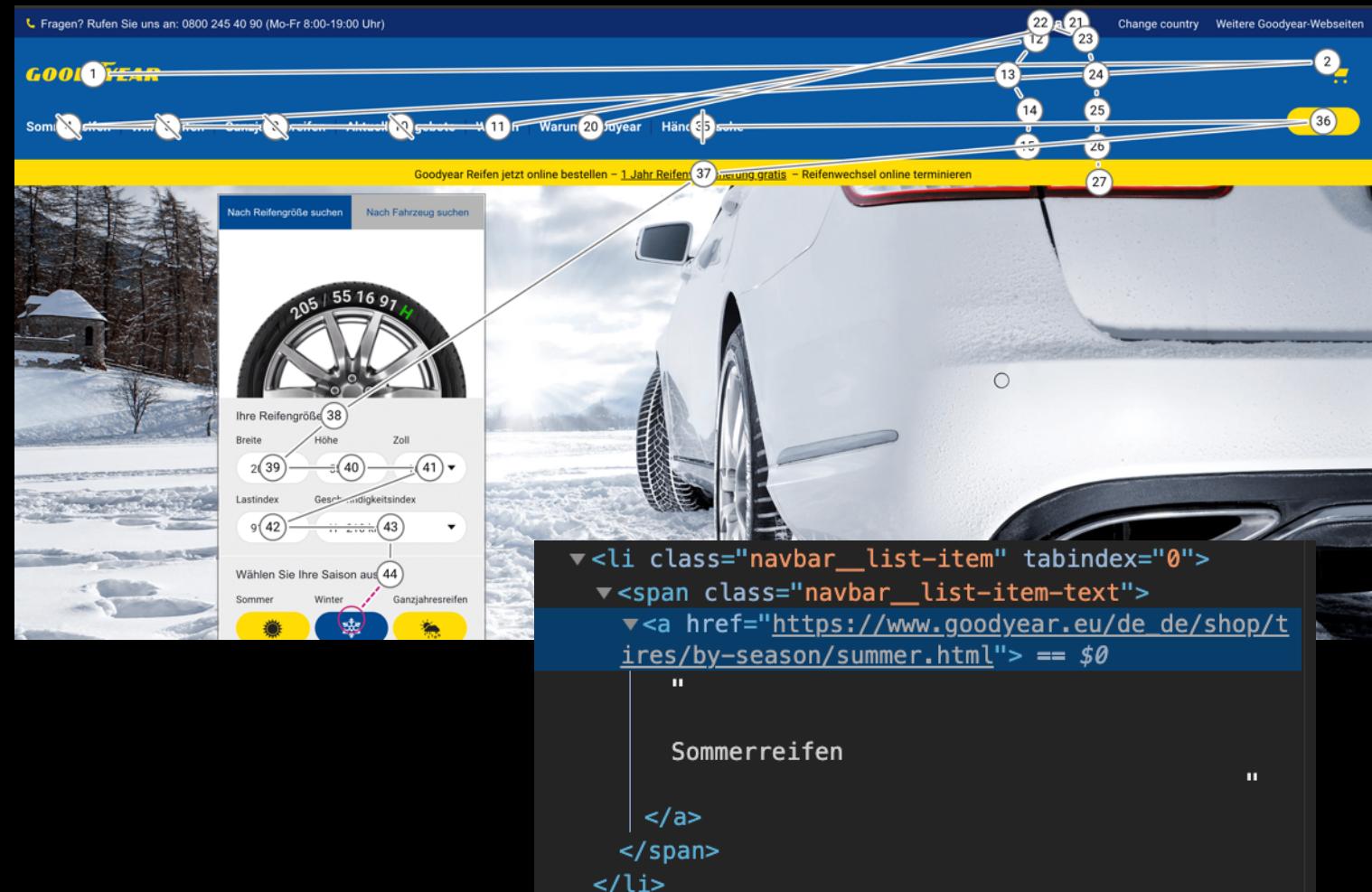
- / Text needs no alt tags
  - / Natively accessible for screen readers
  - / Remains crisp when zooming in



```
  
</div>  
<div class="a-section"></div>  
</div>  
<div>  
<div>  
<script>...</script>  
</div>  
  
ss="a-carousel-card" role="listitem" aria-setsize="5" aria-posinset="4" aria-hidden="true" style="visibility: visible; width: 1000px; margin: 0px; "></li>  
ss="a-carousel-card" role="listitem" aria-setsize="5" aria-posinset="5" aria-hidden="true" style="visibility: visible; width: 1000px; margin: 0px; "></li>  
  
ss="a-carousel-col a-carousel-right celwidget" style="visibility: visible; z-index: 10;" cel_widget_id="gw-decel" data-cel-widget="gw-desktop-hero-right-cel">...</div>  
  
nd aok-hidden"></span>  
  
  
ss="a-section a-spacing-none aok-relative">...</div>  
script>...</script>  
preload id="a-popover-gw-asin-popover">...</div>  
k-btn class="a-button a-button-base">...</span>  
  
sGateway">...</div>  
  
</div>  
-otf -->  
load -->
```

# Issue: Focus &TabIndex

- / Provide a visual indicator of the element in focus (don't just remove outline because it's "ugly")
- / Avoid tabIndex on things that don't need it (e.g. on an `<li>` with an `<a>`, see image)
- / Don't set tabIndex above 0 (gives unexpected tab order, all tabIndex items are tabbed to before any non-tabIndex items)
- / Add `focusable="false"` on SVG:s if supporting IE11
- / More info:  
<https://dequeuniversity.com/rules/axe/4.0/tabinde>



# Issue: Semantic HTML

- / Don't place text straight into `<div>`. It should be within `<p>` or `<h*>` tags. Headers should group content.
- / Use landmarks like `<header>`, `<nav>`, `<main>` and `<footer>`
- / Use button when user takes action (e.g. “read more” links) - user can tab to it, gets visible focus
- / Advice editors to use correct header level
- / Landmarks convey structure and lets users jump there.
- / Read more: <https://web.dev/use-landmarks/>

```
<header>
  <p>Put product name and logo here</p>
</header>

<nav>
  <ul>
    <li>Put navigation here</li>
  </ul>
</nav>

<main>
  <p>Put main content here</p>
</main>

<footer>
  <p>Put copyright info, supplemental links, etc. here</p>
</footer>
```

# Issue: Not using aria live regions for dynamic content

- / Use **aria-live** ("polite" waits, "assertive" interrupts) and **aria-controls** to inform screen readers that content has changed.
- / Usage includes: error messages for forms, if a form submit fails, other dynamically loaded info
- / Read more: [https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA\\_Live\\_Regions](https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Live_Regions)

Planet information

Planet: Mars

## Mars

Mars is the fourth planet from the Sun and the second-smallest planet in the Solar System after Mercury. In English, Mars carries a name of the Roman god of war, and is often referred to as the "Red Planet".

Information courtesy [Wikipedia](#)

 [Open in CodePen](#)  [Open in JSFiddle](#)

2 Mars Mars is the fourth planet from the Sun and the second-smallest planet in the Solar System after Mercury. In English, Mars carries a name of the Ro...

# Issue: Not testing manually

- / Tools can be wrong, you need to verify manually
- / Example: possible to tab to links, but not visit them

Login to Invidem Pilot  
Select your identification method



## — Section 3

# Tools

valtech.



# axe

- / Extension for Chrome, Firefox & Edge
- / Run anytime (on any site) to find a11y issues
- / Provides instructions (with links to more details) for how to fix your a11y errors
- / Good to use while developing, or before committing/opening a PR
- / Download: <https://www.deque.com/axe/browser-extensions/>

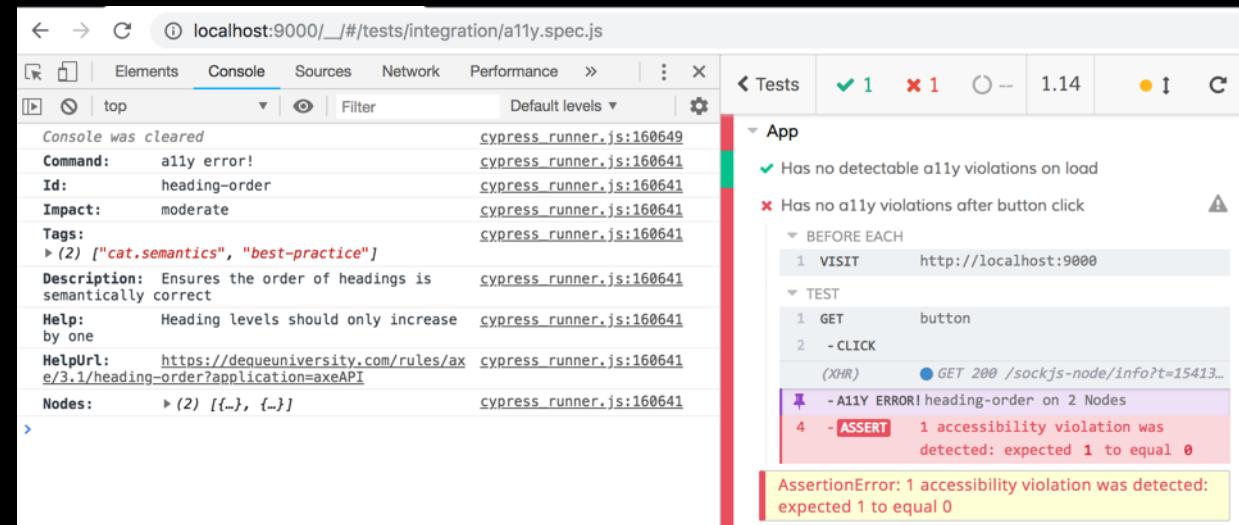
The screenshot shows the axe browser extension interface. At the top, it displays the axe logo and version v4.6.1 (axe-core 4.0.2). Below that, it says "What you're analyzing" and shows the URL <https://localhost:5000/#/>. There are buttons for "Sign in" and "Save results". A summary bar indicates "Automatic issues found 3" with a dropdown arrow, and buttons for "Save results" and "Run again".

Issue Description	Count
Document must have one main landmark	1
All page content must be contained by landmarks	1
Elements must have sufficient color contrast	1

On the right side, there's a sidebar with sections for "Issue description" (Document must have one main landmark), "Element location" (html), "Element source" (<html lang="en">), and "To solve this violation, you need to...". It also includes a "Fix the following:" section with the message "Document does not have a main landmark".

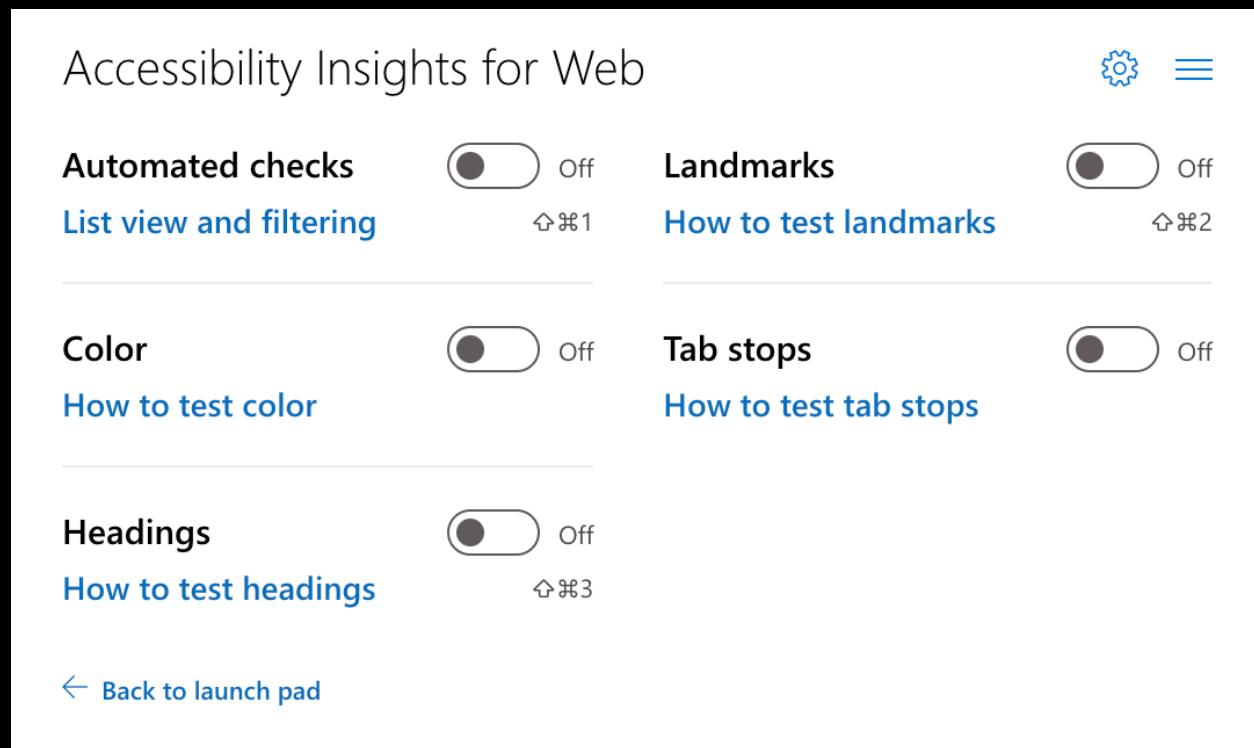
# cypress-axe

- / Plugin for Cypress that runs a11y checks at any given moment, and fails tests if any a11y violation
- / Run when first visiting a page, after opening a modal, or expanding menus etc.
- / Tech agnostic, doesn't care about your framework
- / Provides instructions (with links to more details) for how to fix your a11y errors
- / Download and setup instructions:  
<https://www.npmjs.com/package/cypress-axe>



# Accessibility Insights for Web

- / Extension for Chrome, Edge & Android
- / Similar to axe for finding issues, with additional “Ad hoc” tools:
  - / Color –see your page in grayscale
  - / Headings – highlights headings with their H-level
  - / Landmarks – highlights landmarks
  - / Tab stops – visualizes tab stops, showing order
- / Download:  
<https://accessibilityinsights.io/downloads/>



The screenshot shows the settings interface for the Accessibility Insights for Web extension. At the top right are three icons: a gear for settings, a three-line menu, and a refresh button. Below is a title bar with the text "Accessibility Insights for Web". The main area contains several sections with toggle switches and associated documentation:

- Automated checks**: Off. Sub-sections: [List view and filtering](#) (with a keyboard shortcut ⌘1), [Landmarks](#), and [How to test landmarks](#) (with a keyboard shortcut ⌘2).
- Color**: Off. Sub-sections: [How to test color](#).
- Headings**: Off. Sub-sections: [How to test headings](#) (with a keyboard shortcut ⌘3).

At the bottom left is a blue link "[← Back to launch pad](#)".

# Google Lighthouse

- / Find it in Chrome's Dev tools => Lighthouse tab => select Accessibility
- / Can be run automatically in CI and fail your builds if score drops below certain level with <https://github.com/GoogleChrome/lighthouse-ci>

The screenshot shows the Google Lighthouse accessibility audit results. At the top, a large green circle displays the score '100'. Below the score, the section title 'Accessibility' is shown. A descriptive text explains that these checks highlight opportunities to improve the accessibility of a web app, noting that only a subset of issues can be automatically detected and manual testing is encouraged. An expandable section titled 'Additional items to manually check (10)' provides a list of items that an automated tool cannot cover, with a link to a guide on conducting an accessibility review. The main audit results are grouped under 'Passed audits (23)', which is also expandable. The list of passed audits includes various ARIA-related items such as matching roles, valid values, and sufficient contrast ratios.

100

## Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Only a subset of accessibility issues can be automatically detected so manual testing is also encouraged.

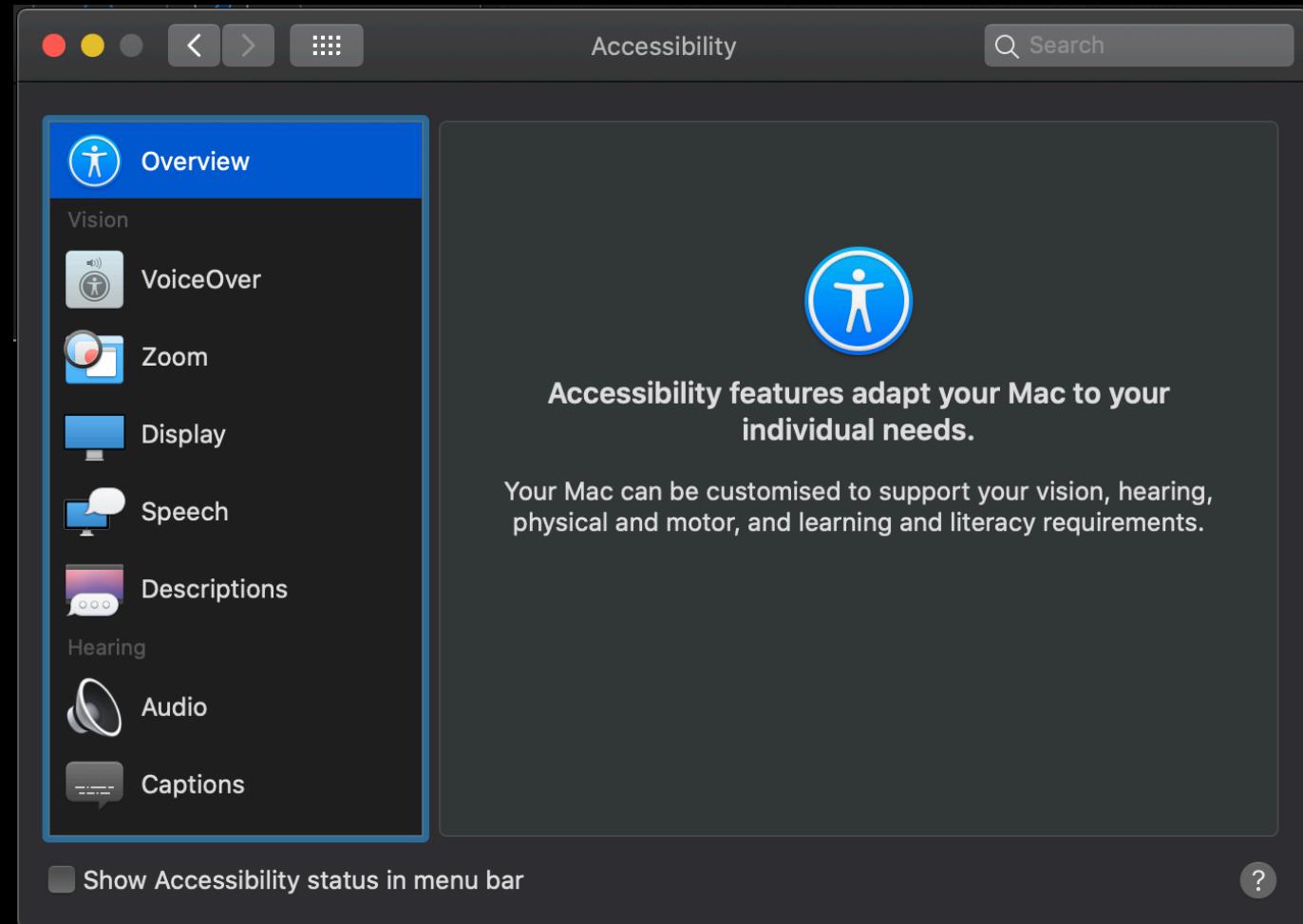
Additional items to manually check (10) — These items address areas which an automated testing tool cannot cover. [Learn more in our guide on conducting an accessibility review.](#)

Passed audits (23)

- [aria-\*] attributes match their roles
- [aria-hidden="true"] is not present on the document <body>
- [aria-hidden="true"] elements do not contain focusable descendants
- [role]s have all required [aria-\*] attributes
- Elements with an ARIA [role] that require children to contain a specific [role] have all required children.
- [role]s are contained by their required parent element
- [role] values are valid
- [aria-\*] attributes have valid values
- [aria-\*] attributes are valid and not misspelled
- Buttons have an accessible name
- The page contains a heading, skip link, or landmark region
- Background and foreground colors have a sufficient contrast ratio
- Document has a <title> element
- <frame> or <iframe> elements have a title
- Heading elements appear in a sequentially-descending order
- <html> element has a [lang] attribute
- <html> element has a valid value for its [lang] attribute
- Image elements have [alt] attributes
- Links have a discernible name
- Lists contain only <li> elements and script supporting elements (<script> and <template>).
- List items (<li>) are contained within <ul> or <ol> parent elements

# Mac's built-in tools

- / Visit **System Preferences => Accessibility**
- / VoiceOver (screen reader)
- / Zoom
- / Color inversion, reduced motion, increased contrast
- / Cursor size
- / and much more



# react-aria-modal

- / Traps focus in modal, escape closes it, returns to “opener” element after closing
- / Example at <https://urplay.se/sok?query=mat> (click the top-right Filter button)
- / Docs: <https://github.com/davidtheclark/react-aria-modal>

### Filter

Arbetsmaterial finns

Syntolkat

Teckenspråkstolkat

**Utbildningsnivå**

**Språk**

**Titta/Lyssna**

[Rensa alla filter](#) [Visa 798 träffar](#)

# NRK Core Components

- / Includes datepicker, dialog, suggest, scroll (Netflix-style), progress, tabs, toggles
- / VanillaJS or React/Preact
- / Good browser support
- / Docs: <https://static.nrk.no/core-components/latest/index.html>
- / Example at <https://urplay.se/>

```
import CoreScroll from '@nrk/core-scroll/jsx'

<CoreScroll friction={Number}          // Optional. Default 0.8. Controls scroll speed
  ref={(comp) => {}}                  // Optional. Get reference to React component
  forwardRef={(el) => {}}            // Optional. Get reference to underlying DOM custom element
  onScrollChange={Function}           // Optional. Scroll change event handler
  onScrollClick={Function}> // Optional. Scroll click event handler
  {/* elements */}
</CoreScroll>
```

Populärt →

Hitta hem	Djuren på Djuris	Orientexpressen tur och retur	Djuren på Djuris - om känslor	Tack gud ja...
Om vänerna Martina och Jamal, och deras första tid i Sverige.	På den här förskolan dyker både känslor och konflikter upp i vardagen.	Följ med på en resa genom Europa med världens mest mytomspunna tåg.	Hör kompisarna beskriva hur de upplever olika sorters känslor.	Säsong 2. Alic tillbaka och re hbtqia+.

# Reach UI

- / Includes Accordions, Alert Dialogs, Tabs, Tooltips, and **much** more
- / Docs: <https://reach.tech/>
- / Example at:  
<https://urplay.se/serie/218445-retorikmatchen-2020>

The screenshot shows a dark-themed user interface with a top navigation bar containing four tabs: "Avsnitt", "Om programmet", "Pedagogiskt material" (which is highlighted with a red border), and "Liknande". Below the navigation, there are two main content sections. The left section is titled "Pedagogiskt syfte" and contains text about the series being språkstarkande for nyanlända. The right section is titled "Arbetsmaterial" and lists three items: "Lärarhandledning", "Programmanus", and "Arbetsblad", each with a small icon.

```
function Example() {  
  return (  
    <Tabs>  
      <TabList>  
        <Tab>One</Tab>  
        <Tab>Two</Tab>  
        <Tab>Three</Tab>  
      </TabList>  
  
      <TabPanels>  
        <TabPanel>  
          <p>one!</p>  
        </TabPanel>  
        <TabPanel>  
          <p>two!</p>  
        </TabPanel>  
        <TabPanel>  
          <p>three!</p>  
        </TabPanel>  
      </TabPanels>  
    );  
}
```

# Conclusion

- / Remember - everything can't be automated, sometimes you must test it manually
- / Check manually (with axe and Accessibility Insights plugins) while developing
- / Automate with cypress-axe and/or LHCI
- / Tab around manually and verify all interaction can be done with just the keyboard
- / Verify your alt-texts and aria-tags make sense and are in the same language as the app
- / Evaluate 3<sup>rd</sup>-party npm libs before using
- / Look up best-practices
- / One thing at a time - baby steps are better than nothing
- / Ask yourself – can people with disabilities use my fancy new site or feature? Is the UX understandable?

# Useful links

- / Good arguments for making your site accessible:  
<https://www.funka.com/en/design-for-all/information-web-and-it/return-on-inclusion/>
- / WCAG 2.1 checklist (Swedish):  
<https://webbriktlinjer.se/wcag/?checklista>
- / Accessibility in JavaScript Applications, Frontend Masters course:  
<https://frontendmasters.com/courses/javascript-accessibility/>
- / Silktide Chrome plugin (e.g. color blindness, dyslexia): <https://silktide.com/resources/toolbar/>
- / Funkify Chrome plugin (e.g. color blindness, dyslexia, trembling): <https://www.funkify.org/>
- / List of all aria-\* attributes and possible values:  
[https://www.w3.org/WAI/PF/aria-1.1/states\\_and\\_properties#state\\_prop\\_def](https://www.w3.org/WAI/PF/aria-1.1/states_and_properties#state_prop_def)

A man in a red jacket and blue pants is riding a skateboard down a path made of many thin, curved lines that curve upwards and outwards. The lines are colored in a gradient from purple to blue to green. The background is black.

# Thank You

André Elmoznino Laufer, Software Developer  
[andre.elmoznino@valtech.com](mailto:andre.elmoznino@valtech.com)