Web Accessibility

Accessibility is not
the goal.
It's a tool for building
an inclusive society.

Laura Wissiak

- Hope Tech Plus Austria
- RnD for assistive tech, web accessibility testing & consultation services
- Focus Points: Low Vision
- How to reach me:
  - Linkedin
  - E-Mail Laura@HopeTech.Vision

Laura Wissiak

- Accessibility First Project
- Newsletter: A11y News
- VÖSI Special Interest Group A11y in ICT
- Volunteer at Access Austria
- Trashcan blogger

We can’t talk about accessibility without talking about disabilities

Side quest 1: How to talk about Disability

Disability is not a bad word!

Person-first vs Identity-first Language

- person -first: “person with a disability”, “... has low vision”, “... uses a wheelchair”
- Identity-first: “disabled person”, “blind person”, “ADHDer”, “an autistic person”
- Note: identity-first doesn’t really apply to mobility aids

Overriding rule: Always respect individuals’ preferences!

How does AT interact with Websites?

How does assistive technology interact with a website?

Assistive Technology (AT) doesn’t have eyes. It looks at the code to know what’s on the screen and can be accessed.

Proper code tells AT:

- How the website is structured: header, main content, footer, navigation, heading-structure
- what content it displays
- Which elements can be interacted with: buttons, links, input fields…

Disclaimer:

This only works if the Website has been build properly.

Not only with perceivable elements in mind, but also with the information that hardware devices or software programs need to operate it.

The Accessibility Fundamentals: The what is what

Abbreviations to know

WCAG: Web Content Accessibility Guidelines

W3C: Web 3 Consortium

AT: assistive technology

SR: screen reader

a11y: accessibility

Alt-text: alternative text

EAA: European Accessibility Act

ADA: Americans with Disabilities Act

UNCRPD: United Nations Convention on the Rights of Persons with Disabilities

WCAG - Web Content Accessibility Guidelines

https://www.w3.org/TR/WCAG22/

- Published by the W3C - Web 3 Consortium
- Put together by various working groups that continually review and re-evaluate them
- EAA is informed by them

What's the EAA?

European Accessibility Act applies to the following products placed on the market after 28 June 2025:

- computers and operating systems
- ATMs, ticketing and check-in machines
- smartphones
- TV equipment related to digital television services
- telephony services and related equipment
- access to audiovisual media services such as television broadcast and related consumer equipment
- services related to air, bus, rail and waterborne passenger transport
- banking services
- e-books
- e-commerce

What's the EAA?

European Accessibility Act applies to the following products **placed on the market after 28 June 2025**:

- ➢ Meaning: it's already in effect but consequences will happen from June 28, 2025

European equivalent of the ADA - Americans with Disabilities Act

https://ec.europa.eu/social/main.jsp?catId=1202

Side Quest 2: The ADA & human rights

Americans with Disabilities Act of 1990

Disability rights are civil rights. From voting to parking, the ADA is a law that protects people with disabilities in many areas of public life.

https://www.ada.gov/

Convention On The Rights Of Persons With Disabilities (CRPD)

TL;DR: access is a human right

https://social.desa.un.org/issues/disability/crpd/convention-on-the-rights-of-persons-with-disabilities-crpd

AT - Assistive Technology

- Hardware: white cane, wheelchairs, powerchairs, hearing aids…
- Interface hardware: adaptive mouse, buttons/switches, eye tracking devices…
- Software: screen readers, speech-to-text software…

SR Screen Readers

Most popular screen readers:

- NDVA (because it's free) https://www.nvaccess.org/
- JAWS (paid)
- iOS: Voice Over
- Android: Talk Back

For testing it's fine to use built-in SR + browser-based

SR on mobile can also have a "touch and explore" feature. Try it yourself!

Side quest 3: How to install & use a screen reader

NVDA Download: https://www.nvaccess.org/download/

How to change language settings on NVDA

Screen Reader Basics: NVDA -- A11ycasts #09 (Chrome for Developers)

How does a screen reader read a screen?

Key-Elements:

- Web Page Structure
- Heading Levels
- Descriptive Links
- Correct Fonts
- Alt-Text, Image Descriptions

Accessibility API

Accessibility API

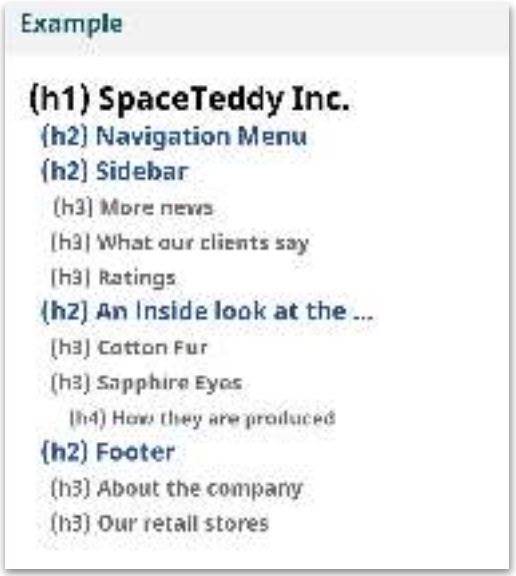
- Screen readers gain access to important information about the objects and events available to the user on your website or in your application through the **Accessibility API**.
- Be aware that every browser (-bases SR) has its own quirks! That's why DOM is important too to provide all information.
- all APIs have the same goal: to make the operating system and the applications that run on it accessible to everyone.

https://dev.to/yuridevat/whats-the-accessibility-api-5agn

https://www.marcozehe.de/why-accessibility-apis-matter/

Page Structure

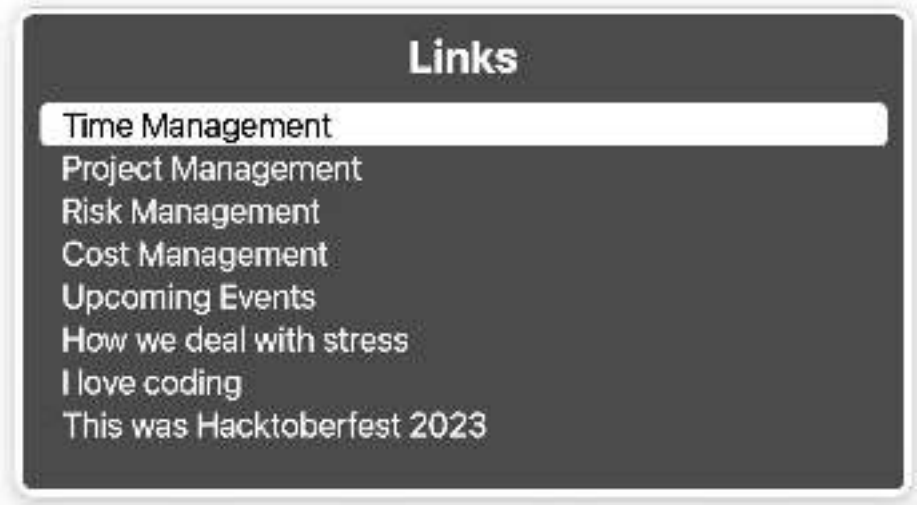
```
<body>
  <header>
    <h1>Space Teddy Inc.</h1>
  </header>
  <nav aria-labelledby="mainnavheader">
    <h2 id="mainnavheader">Navigation Menu</h2>
    …
  </nav>
  <main>
    <article>
      <h2>An inside look at the new Space Teddy 6</h2>
      <nav aria-labelledby="tocheader">
        <h3 id="tocheader">Table of Contents</h3>
        …
      </nav>
      <p>…</p>
```



https://www.w3.org/WAI/tutorials/page-structure/example/

https://www.w3.org/WAI/tutorials/page-structure/headings/

Descriptive Links



- Screen readers have scanning capabilities, where only the links on a webpage are read out.
- Make your links distinguishable from one another
- Tell users where a link will take them

alt-text

```
<img src="img_girl.jpg" alt="Girl in a jacket" width="500" height="600">
```

Guidelines for the alt text:

- The text should describe the image if the image contains information
- The text should explain where the link goes for alt attribute on <a> tag.
- Use alt="" if the image is only for decoration, don’t leave the alt-text attribute out!

W3C: HTML <img> alt Attribute

mdn WebDocs: HTMLImageElement: alt property

Alt-text vs image description?

Alt-text:

- Always in the alt-text attribute
- Intended for screen readers, but also shown if the image can’t be loaded
- Doesn’t disturb the design

Image description:

- Can be in the image details (or post in a CMS)
- Not every person with low vision uses a SR
- Creates awareness about the need for alternative for non-text content.
- Can look out of place or unexpected
- Doesn’t replace the alt-text tag. Always include the alt-text attribute for any image!

What is better?

Side Quest 4: What to write as alt-text?

```
<img src="img_girl.jpg" alt="Girl in a jacket" width="500" height="600">
```

Rule of thumb:

1. Go in order of importance (subject, action, objects, surroundings)
2. No need to repeat yourself, if the information in the picture can be found in the text surrounding it.
3. Be concise: users can't interact with the text, so keep it short (~125 characters, some SR stop reading after that)

Alt-text decision tree

Exercise: Describe this image

Exercise: Describe this image

What I wrote as alt-text:

“White woman with curly brown hair and glasses standing in front of a green and white wall with the zero Project logo that looks like a sprouting plant with 3 leaves. She is wearing a lilac blouse with black pants and has her arms crossed in front of her chest.”

Include distinctive criteria, in my case: white woman, hair color, glasses, outfit…

Exercise: How would you describe yourself?

What is a distinctive criteria depends on the situation:

- In a Zoom call, stick to what can be seen on the screen
- In Person: consider adding your age and height

TL;DR:
A11y considerations have to make sense in context.

Accessible Fonts (FYI, brand issue)

avoid serifs, or combine them with bigger font size and weight

Inclusive sans

𝖆𝖛𝖔𝖎𝖉 𝙩𝙝𝙚𝙨𝙚 𝕥𝕪𝕡𝕖𝕤 ♡𝒻□ᴏɴᴛ□, 🅃🄷🄴🅈 ⓐⓡⓔ ɠıɓɓɾıʂɦ ƭo screen/rēคɗērŞ

The POUR-Principles of Web A11y:

1. **P**erceivable: The user can perceive all website elements with
2. **O**perable: the webpage can be operated with a variety of devices
3. **U**nderstandable: The content is easy to understand and interact with
4. **R**obust: Operation allows for mistakes on the client side and can compensate for them.

Perceivable: Color Contrast

The most common web a11y issue.

The standard minimum requirements:

- For text smaller than 24px (=18pt font): 7:1
- For large text bigger than 24px or, if combined with bold, bigger than 19px (=14pt font): 4.5:1
- UI Elements: 3:1

[WebAIM Color Contrast Checker]

[Link Contrast Checker]

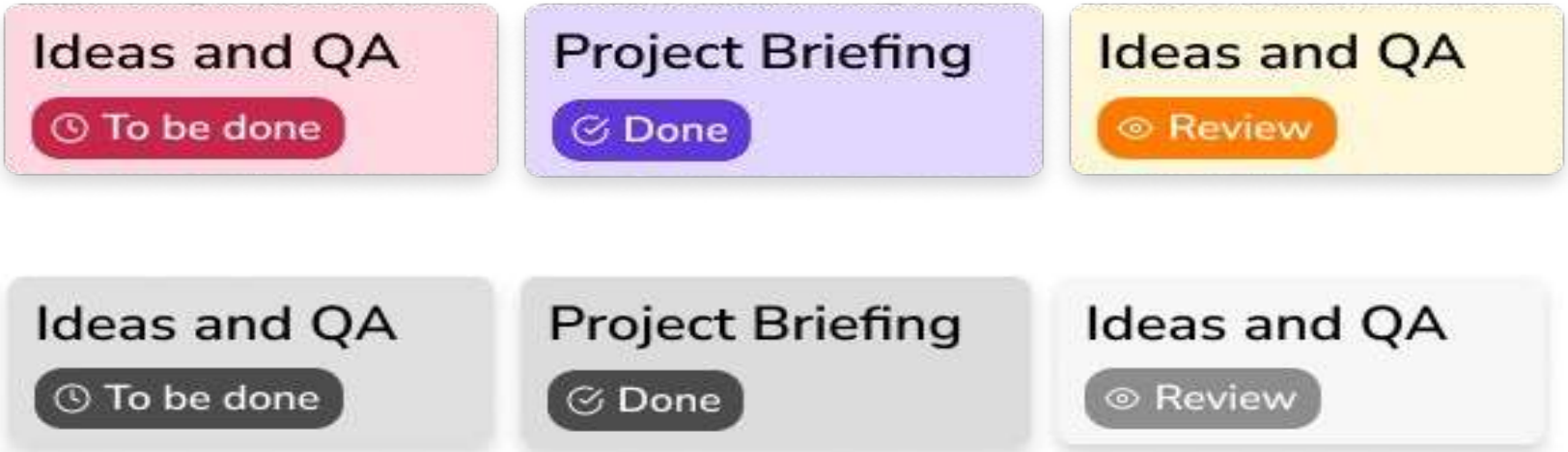
Color Independence

Don’t rely on color alone to convey meaning!

https://www.w3.org/WAI/WCAG22/Understanding/use-of-color.html

Color Independence

What you can do instead:



Operabel: Keyboard Navigation

All functionality must be usable with the keyboard.

Users can access and move between links, buttons, forms, and other controls using the **Tab key** and other keystrokes.

Websites should not require a mouse. E.g. pop-up calendars should also let users type in a date.

Keyboard Compatibility

Operabel: Keyboard Navigation

If all functionality can be achieved using the keyboard, it can also be accomplished by

- keyboard users
- speech input (which creates keyboard input, think speech-to-text)
- Mouse, duh
- and by a wide variety of assistive technologies that create simulated keystrokes as their output

You can test these functionalities out yourself by navigating any website using the tab and space keys.

Developing a keyboard Interface

Side Quest 5: Let's try out keyboard navigation

Try to navigate these websites without a mouse

- https://www.diepresse.com/
- https://www.servustv.com/
- https://www.derstandard.at/
- https://orf.at/
- https://www.linguee.com/
- https://www.timeanddate.com/
- https://www.viennaairport.com/en/passengers
- https://wien.arbeiterkammer.at/index.html
- https://www.wien.gv.at/

How is the UX?

Why is keyboard-only navigation so important?

Most AT relies on it. If it only somewhat works, it is frustrating to navigate a website, especially if this is your only way to access it.

Keyboard-only users can encounter keyboard traps, e.g. a fly-out menu or search bar that won’t close.

Even without traps, navigating can be bothersome.

Understandable: Use of Language (FYI)

“Easy to read” refers to the presentation of text in an accessible, easy to understand format. It is often useful for people with learning and cognitive disabilities, affecting how they process information.

Guidelines for Easy Language:

- Use bigger print and fonts that are comfortable to read
- short sentences: ideally no more than 6 words
- no foreign loanwords: e.g instead of “zenith” use “highest point”
- only use common words, no specific jargon: e.g. don’t use the acronym “a11y”

𝖆𝖛𝖔𝖎𝖉 𝓽𝓱𝓮𝓼𝓮 𝖙𝖞𝖕𝖊𝖘 ♡𝒻 □ᴏɴᴛ□, 🅃🄷🄴🅈 ⓐⓡⓔ ɠıɓɓɾıʂɦ ʈo s̳c̳r̳e̳e̳n̳ rēɲɗērȘ

Robust: Error Handling

If I input the wrong data (e.g. wrong format for the date) the Web Interface helps me to locate and correct the error.

The error message should help the user to locate and correct the mistake.

Example: Login Interface from AsTeRICS grid

Responsive Design

Responsive web design is about creating web pages that look good on **all** devices! A responsive web design will automatically adjust for different screen sizes and viewports (desktop PCs, laptops, tablets, smartphones, screens of any size).

Every person has a personalized set up to ideally accommodate their user behavior. So don’t design for the “standard” set up, make sure to allow for flexibility!

“All users, regardless of a disability, must have equal access to the information.” (Accessibility Principle 1: Perceivable, W3C)

We would never ship unsecured code to a login interface.
So why is it okay to ship inaccessible code to production?

Adrián Bolonio

Senior Accessibility Software Engineer at Miro (amazing person)

Actually write clean code

Depending on your content, you can avoid up to 80% of web a11y issues

You will already avoid most technical issues by using the proper tags for elements, such as:

- Buttons `<button type="button">Button Text</button>`
- Links `<a href="url">Link Text</a>`
- Forms with fieldsets

Preserve website structure by nesting headings by their rank (or level): The most important heading has the rank 1 (<h1>), only 1 h1 per page, the least important heading rank 6 (<h6>). Don’t skip levels in between!

Example: Form Input (HTML)

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label>
    <input type="text" id="fname" name="fname"><br><br>
    <label for="lname">Last name:</label>
    <input type="text" id="lname" name="lname"><br><br>
    <label for="email">Email:</label>
    <input type="email" id="email" name="email"><br><br>
    <label for="birthday">Birthday:</label>
    <input type="date" id="birthday" name="birthday"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

The Fieldset Element

HTML & ARIA tags

ARIA = Accessible Rich Internet Applications

- Use semantic HTML elements whenever possible, as semantic HTML has far better support for assistive technology.
- Some people who rely on AT are reluctant to upgrade their software, out of fear of losing the ability to interact with their computer and browser.

Think of ARIA tags as prosthesis for code: They can be a great thing, but don't use them if you can use semantic HTML instead!

```
<body>
  <header>
    <h1>Space Teddy Inc.</h1>
  </header>
  <nav aria-labelledby="mainnavheader">
    <h2 id="mainnavheader">Navigation Menu</h2>
    …
  </nav>
  <main>
    <article>
      <h2>An inside look at the new Space Teddy 6</h2>
      <nav aria-labelledby="tocheader">
        <h3 id="tocheader">Table of Contents</h3>
        …
      </nav>
      <p>…</p>
        <p>Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
      <p>…</p>
      <ul>
        <li>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</li>
        <li>Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</li>
        <li>Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.</li>
      </ul>
    …
```

Unordered Lists

Don’t override built-in a11y features!

 creates <li>... which are default rendered with bullet points

- These can be overruled with styling

BUT: Don’t use list style: none; for this!

List style: none; is not recognized as list by some SR (e.g. Safari), even if you use semantic HTML with it.

Use list style-type: ” ”; instead, to preserve the unordered list structure

Image Description: meme with rapper Drake having a preference for the latter code snippet, instead of the first one.

(From Sara Soueidan)

The Number 1 takeaway form today:

Be aware that disabilities are a spectrum

Disability is a highly individual experience!

E.g. The same type of blindness can present differently for every person.

Everyone has a different set up for their access needs. Involve the PwD community as much as possible in your development process to understand the bandwidth of access requirements!

Keep in mind: understanding and empathizing with PwDs is still different from the lived experience of having a disability 24/7.

So much to learn! Where do I start?

Content Creators

- @NatashaCaudill: fully color-blind
- @MollyBurkeOfficial: blind speaker & model with a guide dog
- @Mr.Blindlife: legally blind influencer
- @SightlessKombat: blind game tester & streamer on twitch
- @DrJasmineSimmons: Author & Speaker with Usher Syndrome (deaf-blindness)
- @BlindishLatina: deaf-blind Disability Advocate
- @MatthewAndPaul couple with a guide dog

Linkedin

- Accessible EU Centre & https://accessible-eu-centre.ec.europa.eu/
- Digital A11y: local a11y professionals from DACH region
- GDI Hub: Global Disability Innovation Hub https://www.disabilityinnovation.com/
    - Disability Innovation Lectures on YouTube: www.youtube.com/@GlobalDisabilityInnovationHub/
- Haben Girma
- Raul Krauthausen
- Catarina Rivera
- Accessibility First

A11y Specialists to follow

- Adrian Roselli https://adrianroselli.com/
- Leonie Watson (blind a11y tester) https://tink.uk/
- Makoto Ueki (Infoaxia, Inc.) https://weba11y.jp/news/
- Manuel Matuzovic (focused on HTML and CSS) https://matuzo.at/
- Scott O'Hara https://www.scottohara.me/
- Sara Soueidan https://www.sarasoueidan.com/
- Julia Undeutsch (Dev & UX) https://www.juliaundeutsch.com/
- Johannes Lehner (UX) linkedin.com/in/jbeargraphics/
- Adrián Bolonio (a11y software engineer) https://www.adrianbolonio.com/

Newsletters

- WCAG of the Day: 1 success criteria per day
- Disability News Digest: articles about everything a11y & disability & inclusion
- Disability Thinking: opinion piece on 3 articles every weekday
- Wheelchair Travel: accessible travel blog
- Rolling with it: blog on living with a disability
- Andererseits: inclusive Journalism (in German)
- A11y News: monthly a11y event updates & free resources

Resources

- https://www.a11yproject.com/resources/
- https://makeitfable.com/glossary/
- NVDA screen reader (free to download)
- https://www.who.int/teams/noncommunicable-diseases/sensory-functions-disability-and-rehabilitation/global-report-on-health-equity-for-persons-with-disabilities
- https://www.w3.org/WAI/fundamentals/accessibility-principles/#perceivable
- https://webaim.org/projects/million/
- https://inclusive.microsoft.design/
- https://m3.material.io/
- https://webaim.org/resources/contrastchecker/

Browser Extensions

- Colorless A11Y - Color blindness empathy test Color blindness empathy test
- WAVE https://chromewebstore.google.com/detail/wave-evaluation-tool/jbbplnpkjmmeebjpijfedlgcdilocofh
- axe DevTools Automatic Accessibility Testing Tool Browser Extension https://chromewebstore.google.com/detail/axe-devtools-web-accessib/lhdoppojpmngadmnindnejefpokejbdd?hl=en-US&utm_source=ext_sidebar
- Visual ARIA: https://chrome.google.com/webstore/detail/visual-aria/lhbmajchkkmakajkjenkchhnhbadmhmk/related
- Google Lighthouse https://developer.chrome.com/docs/lighthouse/accessibility/scoring
- Accessibility Resources to get you started https://medium.com/@b_atish/ui-development-accessibility-resources-6c27d0b680fb
- DOM accessibility tree https://developer.chrome.com/blog/full-accessibility-tree
- https://developer.chrome.com/docs/devtools/accessibility/reference
- CoBliss Color Blindness Simulator: https://www.color-blindness.com/coblis-color-blindness-simulator/

Advanced reading

- How Designing for Users With Disabilities Enhances UX for Everyone: https://builtin.com/articles/designing-users-disabilities-ux
- WCAG Figma Plugin: https://www.figma.com/community/plugin/1373362852131056921/wcag-plugin
- Handling Hover, Focus, and Other States: https://tailwindcss.com/docs/hover-focus-and-other-states
- Digital accessibility: Employees matter too: https://www.hsbc.com/news-and-views/views/hsbc-views/digital-accessibility-employees-matter-too
- GitHub offers details on internal accessibility programs; amplifies voices of disabled devs: https://cxotoday.com/press-release/github-offers-details-on-internal-accessibility-programs-amplifies-voices-of-disabled-devs/
- The lesson when designing for accessibility: Keep learning! https://www.adnews.com.au/opinion/the-lesson-when-designing-for-accessibility-keep-learning
- The WCAG 3 Working Draft update is ready for your review: https://www.w3.org/blog/2024/the-wcag-3-working-draft-update-is-ready-for-your-review/
- Disability Dongles: https://stimpunks.org/2021/08/27/disability-dongles-designing-for-the-individual-not-the-collective/

Tools & Links mentioned:

https://weba11y.carrd.co/