

Accessibility T3 Barcamp

Who is presenting this session?

Kai :~)!

Software Developer since 2016

Has successfully survived the software development era
before AI, thanks to Stack Overflow.







Disclaimer of this session

This presentation is intended for informational purposes to provide an overview and initial impression of web accessibility. The content is based on my current research and knowledge. However, accessibility standards and legal requirements are subject to change. Please verify all information independently before implementation, as this session does not constitute professional or legal advice.

Global Disability Reality

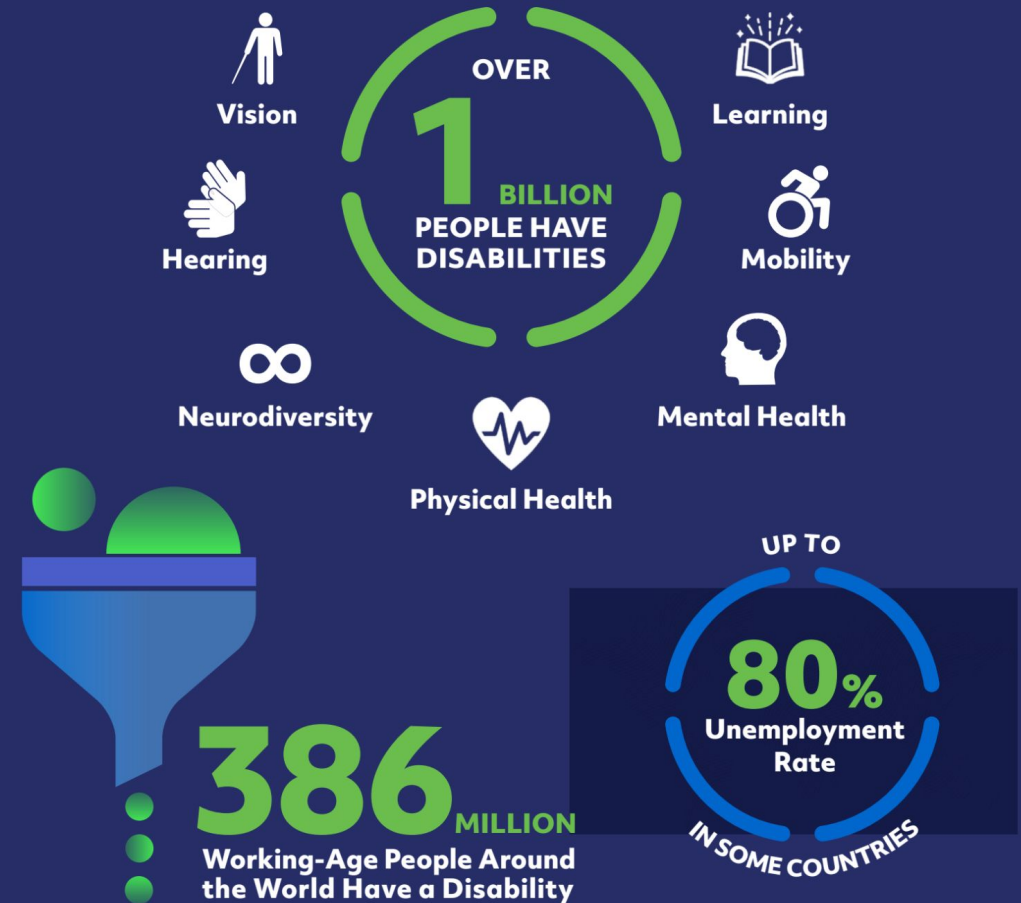
1.3B People worldwide

-  Visual impairments
-  Hearing loss
-  Motor difficulties
-  Cognitive challenges

Source: WHO Global Report on Health Equity (2022)

DISABILITY FACTS: GLOBAL EMPLOYMENT

Disability:IN
Your business partner
for disability inclusion.



IN THE U.S.

People with Disabilities

LABOR FORCE PARTICIPATION

19.6%

UNEMPLOYMENT RATE

12.0%

People Without

66.4%

6.6%

For more resources, visit: DisabilityIN.org/Resources

Data from the World Health Organization, International Labour Organization, and U.S. Dept. of Labor, Office of Disability Employment Policy.

The Spectrum of Barriers



Permanent

One arm, blindness, or deafness. Long-term conditions requiring consistent assistive technology support.



Temporary

An arm injury, cataract surgery, or ear infection. Short-term limitations affecting daily interaction.



Situational

Holding a baby, driving a car, or bright sunlight. Environmental factors limiting user capability.

Types of barriers

- Blindness
- Low vision
- Visual impairment
- Intellectual disability
- Deafness
- Hearing impairment
- Motor impairment
- Other

Permanent

Vorübergehend

Situativ

Berühren



Einarmig



Arm verletzt



Elternteil
von Kleinkind

Sehen



Blind



Grauer Star



Abgelenkt
beim Fahren

Hören



Hörbehindert



Ohrinfekt



Laute Umgebung
(Bartender)

Sprechen



Non-Verbal



Heiser



Starker Akzent

The Spectrum of Barriers

How Users Experience the Web



Screen Readers

NVDA, JAWS, VoiceOver (macOS/iOS), TalkBack (Android). Translates code into speech.



Keyboard Only

Used by motor-impaired users or screen reader users. Relies on logical tab order and focus states.



Braille Displays

Hardware that renders text as tactile dots. Crucial for deaf-blind users.

Braille



BFSG: The Legal Deadline

The **Barrierefreiheitsstärkungsgesetz (BFSG)** implements the European Accessibility Act (EAA) in Germany.

June 28, 2025: Enforcement starts

Fines up to **€100,000**

Authorities can stop service provision

BFSG: The Legal Deadline

- **Origin:** Its the German implementation of the **European Accessibility Act (EAA)**.
- **The Shift:** It marks a major shift from covering only public authorities (BITV) to holding the **private economy** accountable.
- **Key Deadline:** Products and services placed on the market after **June 28, 2025**, must be accessible.
- **Purpose:** To ensure a harmonized EU market for accessible products

BFSG: The Legal Deadline

Key Sectors:

- **E-Commerce:** Online shops (entire journey: search, payment, shipping).
- **Services:** Banking, Telecommunications, E-Books.
- **Transport:** Websites and apps for bus, train, and air travel (tickets, schedules).

Technical Standard: The law references the European norm **EN 301 549**, which aligns almost 1:1 with **WCAG 2.1 Level AA**.

Micro-Enterprise Exemption: Companies with fewer than 10 employees and less than €2 million annual turnover are currently exempt from service requirements.

W3C

The **World Wide Web Consortium** is the global benchmark shaping the Web's long-term health.

- **Mission:** "Web for everyone."
- **WAI:** Web Accessibility Initiative (since 1997).
- **Output:** Recommendations (REC) like HTML, CSS, and WCAG.

<https://www.w3.org/>



WAI - Web Accessibility Initiative

Skip to Content | Change Text Size or Colors | All Translations

W3C Web Accessibility Initiative WAI | Strategies, standards, resources to make the Web accessible to people with disabilities | Get Involved | About W3C WAI | Search

Accessibility Fundamentals | Planning & Policies | Design & Develop | Test & Evaluate | Teach & Advocate | Standards/Guidelines

Making the Web Accessible

Strategies, standards, and supporting resources to help you make the Web more accessible to people with disabilities.

W3C
The World Wide Web Consortium (W3C) develops international standards for the Web: HTML, CSS, and many more.

WAI
The W3C Web Accessibility Initiative (WAI) develops standards and support materials to help you understand and implement accessibility.

You
You can use W3C WAI resources to make your websites, applications, and other digital creations more accessible and usable to everyone.

News

Current Work
(updated monthly)
[See What We're Working On - Accessibility Activities and Publications](#)

For Review: WCAG 3 Working Draft - March 2026
(2026-Mar-03)
W3C Accessibility Guidelines (WCAG) 3.0 Working Draft is updated. The starting place for background, timeline, and up-to-date information is the [WCAG 3 Introduction](#). For a list of changes in the March 2026 draft, see [Status: In-progress drafts](#).

For Review: W3C Accessibility Guidelines Evaluation Methodology (WCAG-EM) 2.0 - First Draft Note
(2026-Feb-05)
W3C Accessibility Guidelines Evaluation Methodology (WCAG-EM) 2.0 is available as a first Draft Note. WCAG-EM describes a methodology with a step-by-step process to evaluate how well digital products

See what we have for you:

Get Resources for...

- [Getting Started](#)
- [Content Writers](#)
- [Designers](#)
- [Developers](#)
- [Evaluators, Testers](#)
- [Managers](#)
- [Policy Makers](#)
- [Trainers, Educators](#)
- [Web Users, People with Disabilities, Advocates](#)
- [Everyone \(All WAI Resources\)](#)
- [Other Languages](#)

Training Course:
Digital Accessibility Foundations

The WAI is the key component of the W3C. The WAI was established in 1997 and is a standing working group dedicated exclusively to web accessibility

<https://www.w3.org/WAI/>

WCAG - Web Content Accessibility Guidelines

Home > Standards/Guidelines > Web Content - WCAG 2

Standards/Guidelines

Web Content - WCAG 2

- How to Meet WCAG 2 (Quick Reference)
- At a Glance
- The Documents
- Applying to Non-Web ICT
- New in 2.2
- New in 2.1
- Translations
- Commenting
- Conformance Logos
- FAQ
- WCAG 3 Draft
- Authoring Tools - ATAG
- User Agents - UAAG
- WAI-ARIA
- Evaluation - ACT & EARL

WCAG 2 Overview

Summary

This page introduces the Web Content Accessibility Guidelines (WCAG) international standard, including WCAG 2.0, WCAG 2.1, and WCAG 2.2. WCAG documents explain how to make web content more accessible to people with disabilities.

A different page [introduces WCAG 3](#).

WCAG is not an introduction to accessibility. For introductions, see [Accessibility Fundamentals Overview](#).

Quick links to resources:

- [How to Meet WCAG 2 \(Quick Reference\)](#)
- [WCAG 2.2 Standard, What's New in WCAG 2.2](#)
- [WCAG 2.1 Standard](#)

Page Contents

- [Introduction](#)
- [Who WCAG is for](#)
- [What is in WCAG 2](#)
- [WCAG 2.0, 2.1, 2.2](#)
- [Translations](#)
- [ISO/IEC 40500, EAA, EN 301 549](#)
- [Who develops WCAG](#)
- [More about WCAG](#)

For the content.

<https://www.w3.org/WAI/standards-guidelines/wcag/>

WCAG Conformance Levels

Level A: Essential Accessibility

Minimum level of accessibility

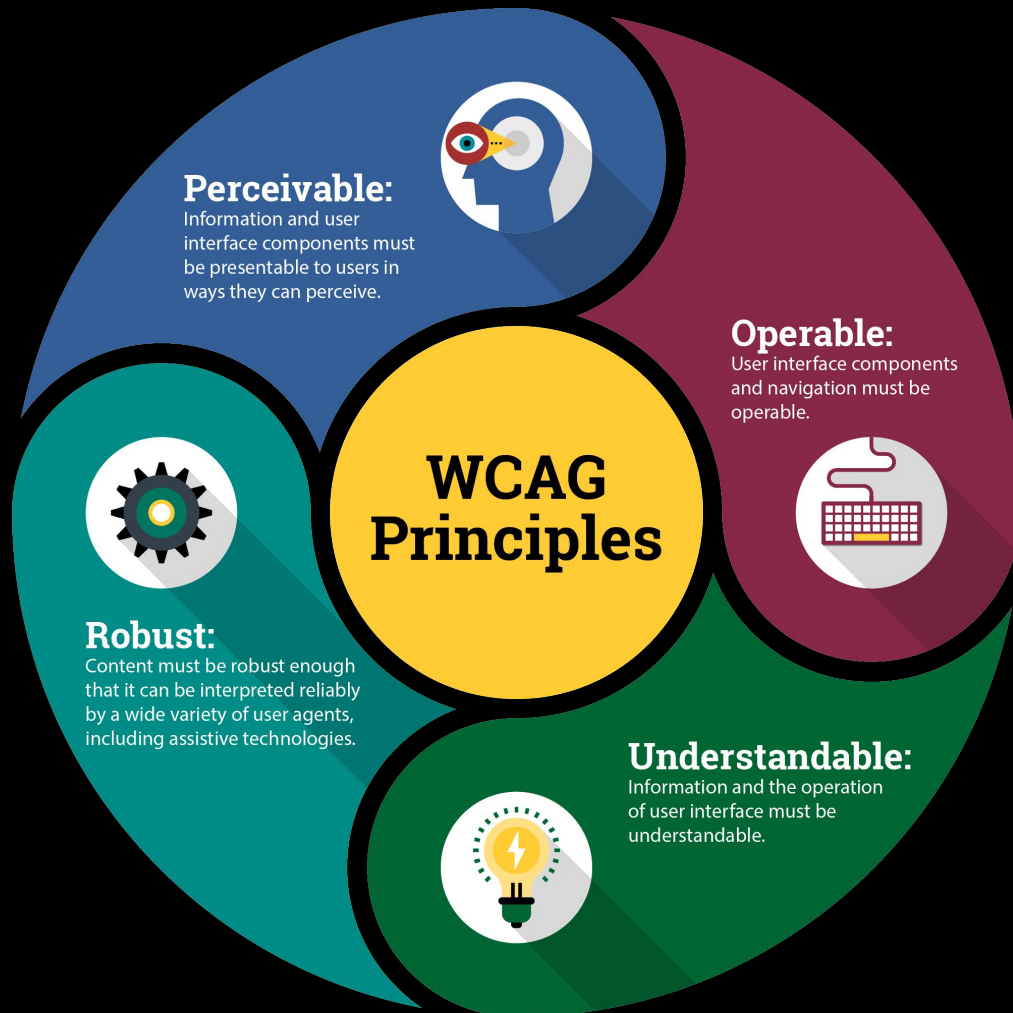
Level AA: The Global Standard

This level is the internationally recognized benchmark and the legal requirement for most frameworks

Level AAA: Maximum Inclusivity

Highest level of accessibility

WCAG Principles



WCAG 2.1 / 2.2

Web Content Accessibility Guidelines (WCAG) are based on the **POUR** acronym:

Perceivable: Users can see/hear it

Operable: Users can navigate/use it

Understandable: Content is clear

Robust: Works with current/future AT

<https://www.w3.org/WAI/standards-guidelines/wcag/>

ATAG - Authoring Tool Accessibility Guidelines

The screenshot shows the 'Authoring Tool Accessibility Guidelines (ATAG) Overview' page. On the left is a navigation sidebar with a 'Standards/Guidelines' section containing links for 'Web Content - WCAG 2', 'WCAG 3 Draft', and 'Authoring Tools - ATAG'. Under 'Authoring Tools - ATAG', there are links for 'At a Glance', 'For LMS', 'For No-Code Tools', 'For Social Media Platforms', 'User Agents - UAAG', 'WAI-ARIA', 'Evaluation - ACT & EARL', 'WAI-Adapt', 'Pronunciation', 'Standards Harmonization is Essential', 'W3C Process for Developing Standards', 'Referencing and Linking to Standards', 'Cognitive Accessibility at W3C', and 'Mobile Accessibility at W3C'. The main content area has a 'Summary' section stating the page introduces the ATAG web standard. Below this is a 'Quick links to resources:' section with a bulleted list of links: 'ATAG at a Glance', 'ATAG 2.0 Standard', 'Implementing ATAG 2.0', and 'ATAG Report Tool'. A 'Page Contents' section follows with a bulleted list of links: 'Authoring Tools and ATAG', 'Who ATAG is for', 'What is in ATAG', 'ATAG resources', 'Translations', and 'Who developed ATAG'. At the bottom, there is a section titled 'Authoring Tools and ATAG' with a paragraph explaining that authoring tools are software and services used by web developers, designers, and writers to produce web content, with examples listed under the 'Who ATAG is for' link.

Standards/Guidelines

- Web Content - WCAG 2
- WCAG 3 Draft
- Authoring Tools - ATAG**
 - At a Glance
 - For LMS
 - For No-Code Tools
 - For Social Media Platforms
- User Agents - UAAG
- WAI-ARIA
- Evaluation - ACT & EARL
- WAI-Adapt
- Pronunciation
- Standards Harmonization is Essential
- W3C Process for Developing Standards
- Referencing and Linking to Standards
- Cognitive Accessibility at W3C
- Mobile Accessibility at W3C

Authoring Tool Accessibility Guidelines (ATAG) Overview

Summary

This page introduces the Authoring Tool Accessibility Guidelines (ATAG) web standard.

Quick links to resources:

- [ATAG at a Glance](#)
- [ATAG 2.0 Standard](#)
- [Implementing ATAG 2.0](#)
- [ATAG Report Tool](#)

Page Contents

- [Authoring Tools and ATAG](#)
- [Who ATAG is for](#)
- [What is in ATAG](#)
- [ATAG resources](#)
- [Translations](#)
- [Who developed ATAG](#)

Authoring Tools and ATAG

Authoring tools are software and services that “authors” (web developers, designers, writers, etc.) use to produce web content (static web pages, dynamic web applications, etc.). Examples of authoring tools are listed below under [“Who ATAG is for”](#).

For editors and CMS platforms such as TYPO3.

<https://www.w3.org/WAI/standards-guidelines/atag/>

UAAG - User Agent Accessibility Guidelines

The screenshot shows the W3C website's navigation menu on the left, with 'Authoring Tools - ATAG' selected. The main content area is titled 'Authoring Tool Accessibility Guidelines (ATAG) Overview'. It includes a 'Summary' section with a paragraph and a 'Quick links to resources:' section with four bullet points. Below that is a 'Page Contents' section with five bullet points. At the bottom, there is a section titled 'Authoring Tools and ATAG' with a paragraph of text.

Standards/Guidelines

- Web Content - WCAG 2
- WCAG 3 Draft
- Authoring Tools - ATAG**
 - At a Glance
 - For LMS
 - For No-Code Tools
 - For Social Media Platforms
- User Agents - UAAG
- WAI-ARIA
- Evaluation - ACT & EARL
- WAI-Adapt
- Pronunciation
- Standards Harmonization is Essential
- W3C Process for Developing Standards
- Referencing and Linking to Standards
- Cognitive Accessibility at W3C
- Mobile Accessibility at W3C

Authoring Tool Accessibility Guidelines (ATAG) Overview

Summary

This page introduces the Authoring Tool Accessibility Guidelines (ATAG) web standard.

Quick links to resources:

- [ATAG at a Glance](#)
- [ATAG 2.0 Standard](#)
- [Implementing ATAG 2.0](#)
- [ATAG Report Tool](#)

Page Contents

- [Authoring Tools and ATAG](#)
- [Who ATAG is for](#)
- [What is in ATAG](#)
- [ATAG resources](#)
- [Translations](#)
- [Who developed ATAG](#)

Authoring Tools and ATAG

Authoring tools are software and services that “authors” (web developers, designers, writers, etc.) use to produce web content (static web pages, dynamic web applications, etc.). Examples of authoring tools are listed below under “[Who ATAG is for](#)”.

For browsers and media players.

<https://www.w3.org/WAI/standards-guidelines/uaag/>

Do we really need this?

Yes!:D

Without the W3C, we wouldn't have an interoperable web today (“Browser Wars”).

Fundamental to standards, and legislation also relies on this standards.

Helpful Links:)

Example:

<https://www.w3.org/WAI/ARIA/apg/>



The screenshot shows the homepage of the ARIA Authoring Practices Guide (APG). The page has a blue header with the title "ARIA Authoring Practices Guide (APG)" and the subtitle "How to build accessibility semantics into web patterns and widgets". The W3C Web Accessibility Initiative logo is in the top right. A navigation menu includes "APG Home", "Patterns", "Practices", "Index", and "About". The main content area features a large blue box with the heading "ARIA Authoring Practices Guide (APG) Home" and a paragraph explaining the guide's purpose. A "View Patterns" button is located below the text. To the right is an illustration of a laptop displaying a person icon. Below this is a section titled "APG RESOURCES" with the subtitle "Building blocks that help you make the web accessible". Underneath, there is a section for "Design Patterns and Examples" with a paragraph and a "Learn More" button. The bottom right corner contains an illustration of various web UI components like buttons, forms, and lists.

ARIA Authoring Practices Guide (APG) | How to build accessibility semantics into web patterns and widgets

W3C Web Accessibility Initiative WAI

APG Home Patterns Practices Index About

ARIA Authoring Practices Guide (APG) Home

Learn to use the accessibility semantics defined by the Accessible Rich Internet Application (ARIA) specification to create accessible web experiences. This guide describes how to apply accessibility semantics to common design patterns and widgets. It provides design patterns and functional examples complemented by in-depth guidance for fundamental practices.

[View Patterns](#)

APG RESOURCES

Building blocks that help you make the web accessible

Design Patterns and Examples

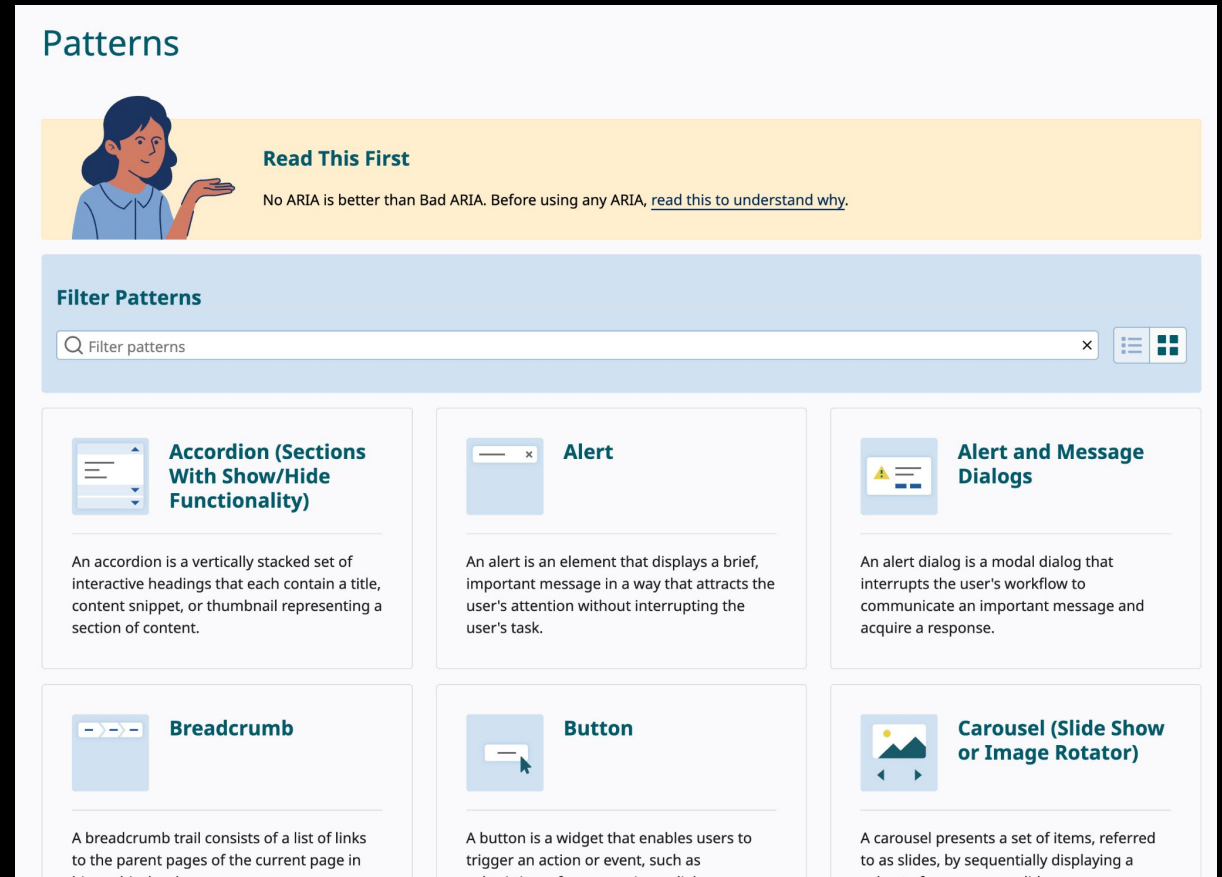
Learn how to make accessible web components and widgets with ARIA roles, states and properties and by implementing keyboard support. One or more ways of implementing each pattern is demonstrated with a functional example.

[Learn More](#)

Helpful Links:)

Example:

<https://www.w3.org/WAI/ARIA/apg/patterns/>



The screenshot shows the 'Patterns' page from the WAI ARIA APG. At the top, there is a 'Read This First' section with an illustration of a woman and the text: 'No ARIA is better than Bad ARIA. Before using any ARIA, [read this to understand why.](#)' Below this is a 'Filter Patterns' search bar with the placeholder text 'Filter patterns'. The main content area is a grid of six pattern cards, each with an icon, a title, and a description:

- Accordion (Sections With Show/Hide Functionality)**: An accordion is a vertically stacked set of interactive headings that each contain a title, content snippet, or thumbnail representing a section of content.
- Alert**: An alert is an element that displays a brief, important message in a way that attracts the user's attention without interrupting the user's task.
- Alert and Message Dialogs**: An alert dialog is a modal dialog that interrupts the user's workflow to communicate an important message and acquire a response.
- Breadcrumb**: A breadcrumb trail consists of a list of links to the parent pages of the current page in hierarchical order.
- Button**: A button is a widget that enables users to trigger an action or event, such as submitting a form, opening a dialog, or navigating to another page.
- Carousel (Slide Show or Image Rotator)**: A carousel presents a set of items, referred to as slides, by sequentially displaying a subset of one or more slides.

The Tech Stack

Accessibility is not a plugin - its the core architecture of your code.

ARIA: Enhances rich interaction

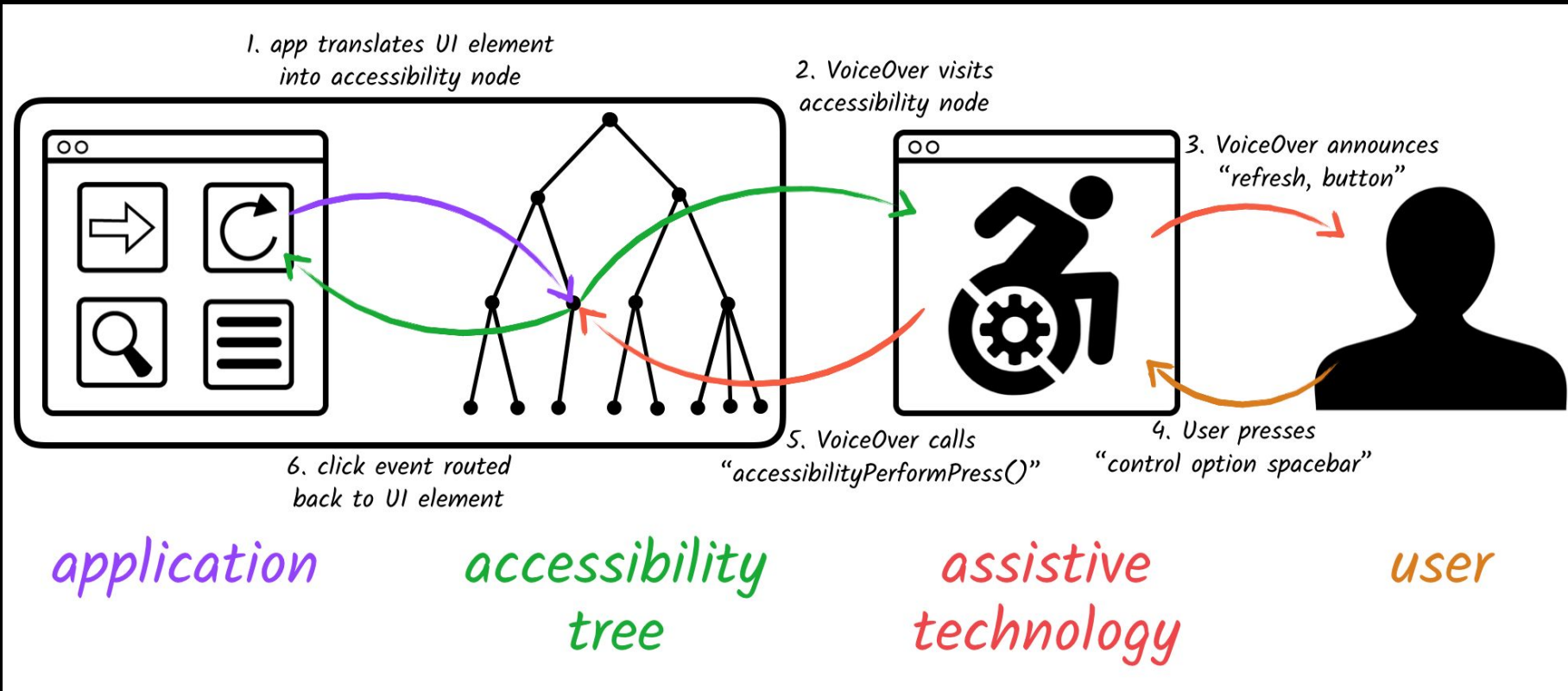
JS: Extends behavior/logic

CSS: Visual presentation

HTML: Structural foundation



The Accessibility Tree (AOM)



The Accessibility Tree (AOM)

The browser transforms HTML/DOM into a specialized tree for Assistive Technology (AT).

DOM Tree: The developer's view

Accessibility Tree: The AT's view

AT reads the **Name**, **Role**, **State**, and **Value** of elements from this tree via platform APIs.

```
▼ RootWebArea "W3C" focusable: true url: https://www.w3.org/
  ▼ Ignored
    ▼ Ignored
      ▼ link "Skip to content" focusable: true url: https://www.w3.org/#main
        ▶ StaticText "Skip to content"
      ▼ Ignored
        ▼ Ignored
          ▼ banner ""
            status "" atomic: true
            ▶ navigation "Language options"
            ▶ navigation "Main"
          ▶ main ""
          ▼ complementary ""
            ▶ heading "Latest news"
            ▶ paragraph ""
            ▶ heading "W3C Advisory Committee Elects Advisory Board"
            ▶ paragraph ""
            ▶ paragraph ""
            ▶ heading "Age-restrictions on the web and user privacy and safety"
            ▶ paragraph ""
            ▶ paragraph ""
            ▶ heading "W3C Establishes Representative Office in Shenzhen, China"
            ▶ paragraph ""
            ▶ paragraph ""
            ▶ heading "W3C establishes Representative Office in Shenzhen, reinforcing commitment to open web standards in China"
            ▶ paragraph ""
            ▶ paragraph ""
          ▼ contentinfo ""
            ▶ list ""
```

01. HTML

The Semantics of *Machine Readability*

HTML General

How many different HTML elements do you think the current specification includes?:)

A) approx. 50

B) approx. 80

C) over 100

D) over 200

HTML General

The current HTML Living Standard (WHATWG - <https://whatwg.org/>) currently defines approximately 110 to 115 active elements.

100+ Semantic Elements

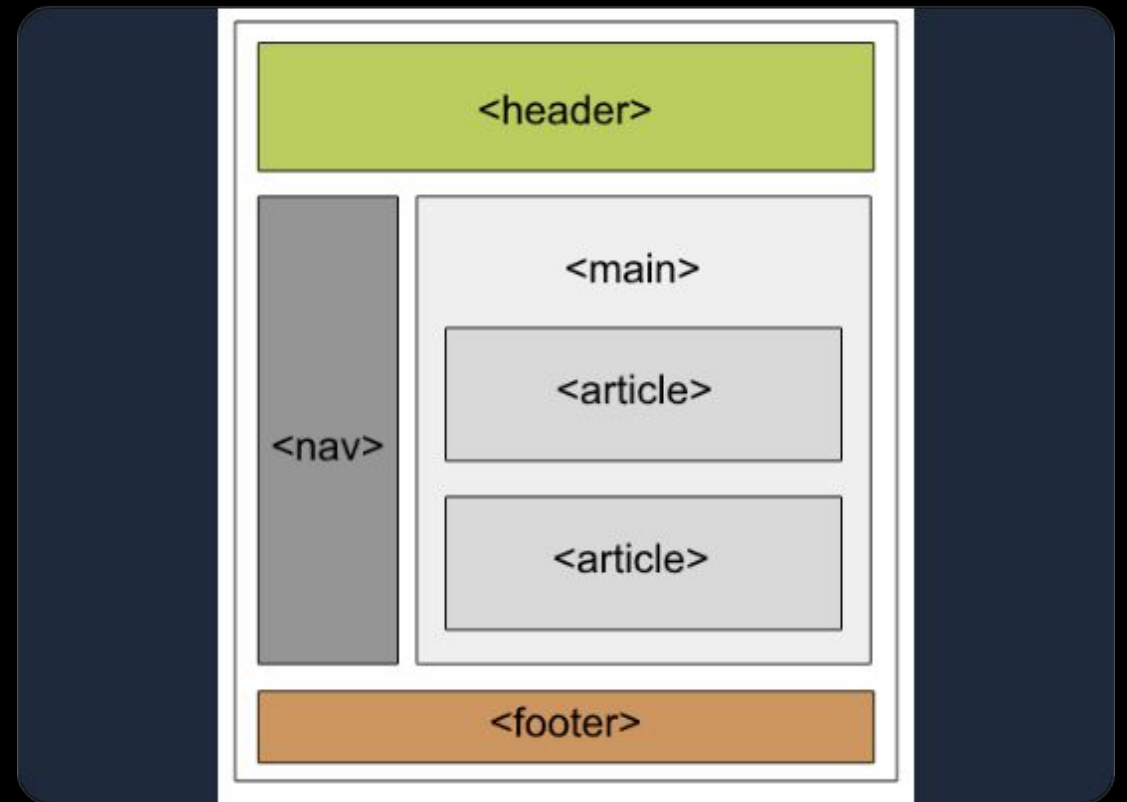
Modern HTML5 provides elements for almost every structural need. Using `<div>` and `` should be your **last resort**.

`<header>`, `<footer>`, `<main>`

`<nav>`, `<aside>`, `<article>`

`<figure>`, `<figcaption>`

`<abbr>`, `<time>`, `<cite>`



100+ Semantic Elements

- Use native HTML elements whenever possible
- Always use the most semantically appropriate HTML element
- Use the semantically neutral containers `<div>` (block element) and `` (inline element) only as a last resort
- Never use HTML elements for layout or styling purposes (use CSS!)
 - “Heading, because it’s large and bold”
 - Layout tables
 - Line breaks / empty paragraphs for spacing
 - Multi-column layout using spaces

Raw vs. Semantic Content

Non-Semantic

```
<div class="h1">Headline</div>  
<div class="p">This is text.</div>  
<div class="list">...</div>
```

AT only sees "Text". No structure, no importance, no lists.

Semantic

```
<h1>Headline</h1>  
<p>This is text.</p>  
<ul><li>...</li></ul>
```

AT announces "Heading level 1", "List with n items".

Regions & Landmarks

Landmarks allow screen reader users to jump directly to specific areas of the page without tabbing through everything.

HTML5 Element	ARIA Landmark Role	Usage Purpose
<header>	banner	Site-wide top content
<nav>	navigation	Main links container
<main>	main	Primary unique content
<footer>	contentinfo	Copyright, legal info

Interactive Elements

What interactive elements do you know of?

Interactive Elements

```
<span
  role="button"
  tabindex="0"
  id="simulatedButton"
  class="btn-sim"
>
  Klick mich
</span>
```

```
.btn-sim {
  display: inline-block;
  padding: 10px 20px;
  background-color: #eee;
  border: 1px solid #777;
  cursor: pointer;
  user-select: none;
}

.btn-sim:focus {
  outline: 2px solid #818cf8;
  outline-offset: 2px;
}

.btn-sim:active {
  background-color: #ddd;
}
```

```
const btn = document.getElementById('simulatedButton');

function handleAction() {
  alert('Aktion ausgeführt');
}

btn.addEventListener('click', handleAction);

btn.addEventListener('keydown', function(event) {
  if (event.key === 'Enter' || event.key === ' ') {
    event.preventDefault();
    handleAction();
  }
});
```

Interactive Elements

```
<button type="button" onclick="handleAction()">  
  Klick mich  
</button>
```

Native HTML vs. Simulation

The Trap

Simulating a button with `role="button"` requires:

- Adding `role="button"`

- Adding `tabindex="0"`

- Handling Click events

- Handling Enter & Space keys

Total Waste of Time.



Native Power

Native elements have focus, keyboard events, and semantics **built-in**. Don't re-invent the wheel.

Interactive Elements

- Essential, because assistive technologies simulate keyboards
- All interactive elements are relevant
- Native, interactive elements can be operated using a keyboard
- The document order is decisive for navigation

<a>

<button>

<details>

<embed>

<iframe>

<input>

<label>

<select>

<textarea>

Links vs. Buttons



Navigation. Moves the user to a new resource or a different part of the same page.



Action. Triggers a function (e.g., submitting a form, toggling a menu, opening a modal).

Links vs. Buttons: Choosing the right one!

Link:

Navigates the user to a new resource, a different page, or a fragment on the same page.

```
<a href="https://example.com">Visit  
Website</a>  
<a>Future Link</a>
```

Button Action:

- Sends data from the current form
- Reverts all fields in a form to their initial values.
- Triggers a client-side function or script on the current page (e.g., toggling a menu, opening a modal).

```
<button type="submit">Send Form</button>  
<button type="reset">Clear Data</button>  
<button type="button">Open  
Modal</button>
```

Keyboard: Keys

- **Tab key:** Move from one interactive element to another (Move backward using the Shift key)
- **Enter key:** Activate links and buttons
- **Spacebar:** Activate form elements and buttons
- **Arrow keys:** Navigate within groups (e.g., radio buttons and drop-down lists)
- **Esc key:** Close drop-down lists and dialog boxes, Cancel operations

Keyboard: Interaction

- Focus
- Activate / Operate
- Skip

Managing Focus (TabIndex)

`tabindex="0"`

Adds a naturally non-focusable element into the tab order. Use sparingly.

`tabindex="-1"`

Removes from tab order, but remains focusable via JS (`.focus()`). Essential for modal focus traps.

`tabindex > 0`

Anti-Pattern.

Breaks natural document order. Avoid at all costs.

Images & Visual Graphics

The Alt Text Decision

Informative: Describe the content briefly.

Decorative: Use `alt=""` or `aria-hidden="true"`.

SVGs: Must be hidden if decorative, or labeled

Decision Tree for alt:

<https://www.w3.org/WAI/tutorials/images/decision-tree/>

Described

```

```

```
<svg role="img" aria-label="Home">...</svg>
```

Decorative

```

```

```
<svg aria-hidden="true">...</svg>
```

Accessible Forms

- HTML5 `<input type="...">` elements
- Form elements always require a label—preferably visible
- Logical order of fields is essential
- Grouping of related fields, including labels -> radio or checkboxes
- Visual presentation should match the expected input

Accessible Forms

- Each form element requires a label that is as visible as possible, ideally placed before the element and referenced by its ID
- The text can be hidden from view if necessary, but must remain accessible to assistive technologies!

```
<label for="firstname">Vorname</label>
```

```
<input type="text" id="firstname">
```

```
<label for="firstname"
```

```
class="visually-hidden">Vorname</label>
```

```
<input type="text" id="firstname">
```

Form Placeholder



When it fails

Placeholders disappear when typing, losing context. Often have low contrast by default. Are NOT a replacement for an label.



The Solution

Use visible labels. Use placeholders only for "hints" (e.g., "DD.MM.YYYY"). Ensure 4.5:1 contrast for placeholder text if used.

Forms



Learning: Native HTML

Custom Simulation

Time spent on: CSS focus styles, keyboard JS, ARIA attributes, AT testing, bug fixing.

Native Element

Time spent on: Styling the look. Everything else is handled by the browser engine.

02. CSS

Aesthetic Presentation, Functional States

Visible Focus States

```
* { outline: none; }
```

```
:focus { outline: none !important; }
```

Visible Focus States

Keyboard users depend on seeing the "Focus Ring". Never use `outline: none;` without a high-contrast replacement.

Contrast > 3:1 against background

Area change (Border, Ring, Background)

Interactive Elements States

Interaction requires clear visual feedback. The **Focus State** is the most important for keyboard users.

:focus

Mandatory. Must be distinct and high contrast. Never use `outline: none` without a custom replacement.

:hover

Good for mouse users, but not enough on its own. Often styles match `:focus` for consistency.

:active / :visited

Helps orientation. Contrast ratio of 3:1 required for graphical components.

Properly Hiding Content

| Technique | Visual | Screen Reader | Keyboard Focus |
|---------------------------------|---------|---------------|----------------|
| <code>display: none</code> | Hidden | Hidden | No |
| <code>visibility: hidden</code> | Hidden | Hidden | No |
| <code>aria-hidden="true"</code> | Visible | Hidden | Avoid! |

CSS Pitfalls



Absolute Units

Avoid px for font-size. It prevents browser zooming. Use rem or em to respect user preferences.



Color-Only Info

Never convey info only via color. Add icons, patterns, or text (e.g., for error messages).



List Semantics

list-style: none can strip semantics in some browsers. Sometimes requires repair with role="list".

Contrast

Standard text needs **4.5:1** ratio. Large text needs **3:1**.

Don't rely on color alone.

Test with tools.

<https://webaim.org/resources/contrastchecker/>

Contrast Ratio
8.15:1
[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

Contrast Ratio
1.1:1
[permalink](#)

Normal Text
WCAG AA: **Fail**
WCAG AAA: **Fail**
The five boxing wizards jump quickly.

Large Text
WCAG AA: **Fail**
WCAG AAA: **Fail**
The five boxing wizards jump quickly.

Graphical Objects and User Interface Components
WCAG AA: **Fail**
★
Text Input

03. JavaScript

JavaScript is cool!

In general, JavaScript is neither necessary nor problematic for a page's accessibility.

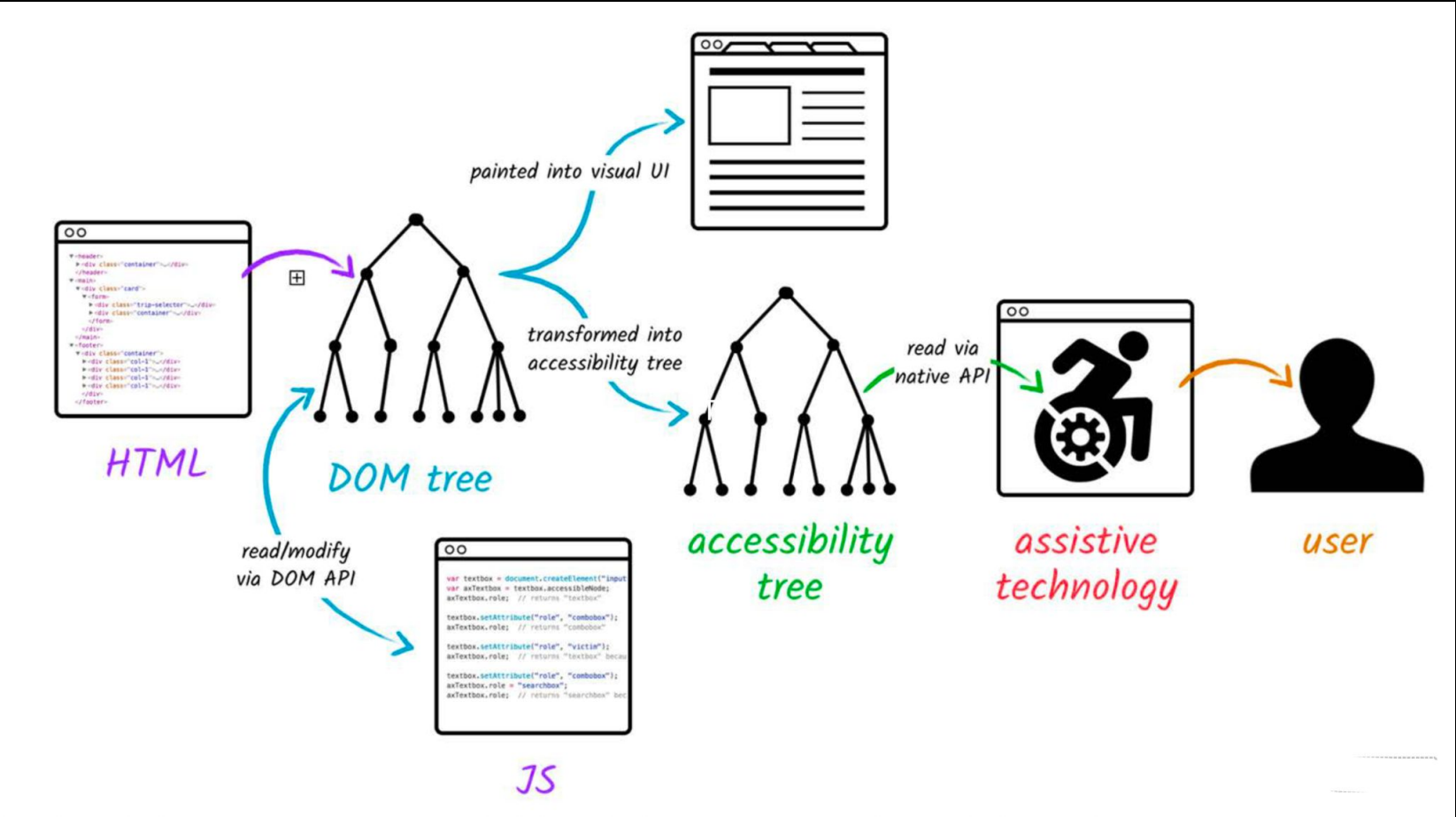
However, JavaScript can significantly improve accessibility in some cases

JavaScript can also ruin a page's accessibility

Example:

Focus Management via JavaScript

DOM & Accessibility Tree



DOM & Accessibility Tree

- More direct access to the accessibility tree, for example to retrieve computed properties (e.g., accessible name)
- Assistive technology events (via DOM events)
- Defining references without ID references

04. ARIA

Accessible Rich Internet Applications

What is ARIA?

ARIA is a semantic extension to HTML (supplemental vocabulary). It bridges the gap for complex widgets that HTML alone cannot describe.

Roles

Defines **what** an element is (e.g., role="slider").

Properties

Describes **characteristics** (e.g., aria-labelledby).

States

Describes **current condition** (e.g., aria-expanded).

ARIA Live Regions

Used for dynamic content changes that don't move focus (e.g., status messages, search result counts).

aria-live="polite"

AT waits until current speech is finished before announcing change.

aria-live="assertive"

Interrupts current speech immediately. Use for critical alerts only.

```
<div role="status" aria-live="polite">3 results found</div>
```

The 5 Rules of ARIA

Rule 1: If you can use native HTML, **do so**. Don't use ARIA.

Rule 2: Do not change native semantics, unless you really have to

Rule 3: All interactive ARIA controls must be keyboard accessible.

Rule 4: Do not use `role="presentation"` or `aria-hidden="true"` on focusable elements.

Rule 5: All interactive elements must have an **accessible name**.

Widget Pattern: Disclosure

Commonly used for "Hamburger Menus" or Accordions.

```
<button aria-expanded="false" aria-controls="menu-list">  
  Menu  
</button>  
<ul id="menu-list" hidden>...</ul>
```

When clicked, JS toggles `aria-expanded` and the `hidden` attribute.

Widget Pattern: Modal Dialog

`</>` `role="dialog"` and `aria-modal="true"`

 **Focus Trap:** Focus must stay inside until closed.

 **ESC Key:** Must close the dialog.

 **Restore:** Focus must return to the trigger button.

Examples

```
<button class="btn" aria-label="Schließen">  
|   <span class="icon-close"></span>  
</button>
```

Examples

```
<nav aria-label="Hauptnavigation">  
<!-- Linkliste -->  
</nav>
```

05. Testing & Tools

Validating Your Work

Testing: The Essentials



Lighthouse / axe

Automated checks in DevTools. Catches ~30-40% of issues (Missing alts, contrast, roles).



WAVE

Browser extension for visual feedback on page structure and errors.







BITV Test

The German standard for public sector sites. Comprehensive 60-step procedure.

Manual Testing Checklist

Automated tools are not enough. You **must** do manual checks.

-  **Keyboard:** Can I reach every interactive element?
-  **Focus:** Can I see where I am?
-  **Screen Reader:** Does the tab order and labels make sense?
-  **Zoom:** Can I zoom to 200% / 400% without horizontal scrolling?

Daily Frontend Best Practices





HTML-First: Start with content structure, then add CSS/JS.

Native Over ARIA: If works, don't build an ARIA accordion.

Alt Text is Code: Treat alt attributes as vital data, not optional metadata.

Audit Early: Test components while building, not after "Go-Live".

Summary & Key Takeaways

-  **BFSG 2025:** Accessibility is no longer "nice to have".
-  **Semantics First:** Use the right tag for the right job.
-  **ARIA:** Use only when native HTML reaches its limits.
-  **Everyone Wins:** Better accessibility = Better SEO + Better UX + Better Legal Safety.

Any Questions?

Thank you for your attention!

Let's build an inclusive TYPO3 ecosystem together.

Kai Grosz

kai@grosz.dev

<https://www.linkedin.com/in/kai-grosz/>

